UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Université Bretagne Sud

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ BRETAGNE SUD

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Kilian FATRAS

Optimal Transport and Deep Learning : Learning from one another

Thèse présentée et soutenue à Rennes, le 25 November 2021
Unité de recherche : IRISA
Thèse N° : 604

Rapporteurs avant soutenance :

Marco Cuturi	Professor, CREST - ENSAE, Institut Polytechnique de Paris.
Christian Wolf	Maître de Conférence, HDR, LIRIS, INSA de Lyon.

Composition du Jury :

Présidente :	Michèle Sebag	Directrice de Recherche CNRS, Université Paris Saclay.
Examinateurs :	Adam M. Oberman	Full Professor, Department of Mathematics and Statistics, McGill University.
	Laetitia Chapel	Maîtresse de Conférence, Université Bretagne Sud.
Dir. de thèse :	Nicolas Courty	Professeur des universités, Université Bretagne Sud.
Co-dir. de thèse :	Rémi Flamary	Maître de Conférence, HDR, CMAP, École Polytechnique.

Remerciements

Ce manuscrit contient tous les travaux de ma thèse qui ont été réalisés de Novembre 2018 à Février 2021. Avant de clôturer ce bien beau chapitre, je tiens à remercier toutes les personnes qui m'ont permis de l'écrire.

Je tiens tout d'abord à remercier mes directeurs de thèse, les professeurs Rémi Flamary et Nicolas Courty, pour les différentes directions de recherches qu'ils m'ont suggérés pendant ces trois années. J'aimerais aussi les remercier de m'avoir permis de travailler étroitement avec l'équipe Panama de l'IRISA et l'INRIA Rennes, cela m'a permis de m'épanouir professionnellement et personnellement pendant ces trois années.

I would like to thank Marco Cuturi and Cristian Wolf for their report and the time they took to read my PhD manuscript. I also want to thank Adam M. Oberman, Michèle Sebag and Laetitia Chapel for being members of my PhD jury.

A Rennes, j'ai eu la chance de travailler dans l'équipe Panama, et j'aimerais remercier chaque membre que j'ai côtoyé. En particulier, merci tout d'abord à Younès, tes conseils et tes compétences en mathématiques ont fait de moi un bien meilleur mathématicien. Je garde de très beaux souvenirs de nos travaux sur le problème du transport optimal avec des minis lots. Merci à Rémi Gribonval, dont la rigueur nous a permis de parfaire un bien joli formalisme. Merci à Jérémy, Rémi C., Axel, Stéphanie, Caglayan et Clément. Et merci à mon autre équipe Obélix.

I also want to thank my other collaborators. Thank you Bharath, Sylvain, Devis, Jean-Christophe, Szymon and Thibault. Many results which are in this manuscript were discovered thanks to your help.

Merci aux *Artistes* que je n'ai pas encore cités: Lucas et Stéphane. Les nombreuses discussions auront animées nos soirées pour le meilleur et pour le pire.

Merci à Rémi L. et Arthur pour la relecture de ce manuscrit. Mon anglais s'est un peu rouillé avec le temps... Je vais y remédier très vite.

Merci à Fabian, ta rencontre a été déterminante afin de me rappeler à quel point j'aime la science. Merci aussi pour ton aide et ta bienveillance quant à ma recherche de PostDoc. Et merci à Michèle, qui m'a fait aimer le goût de la rigueur et qui m'a fait voir pour la toute première fois la beauté des mathématiques. Pour la petite histoire, il s'agissait du théorème de Cayley-Hamilton.

Merci à ma famille. Bien que le prochain chapitre m'éloigne de vous géographiquement, je suis sûr que je reviendrai sur nos terres bretonnes.

Enfin, un grand merci à Floriane. Merci de m'avoir supporté dans les victoires comme dans les moments de doute. Ces travaux portent aussi ton empreinte. J'ai hâte d'écrire le prochain chapitre à tes côtés de l'autre côté de l'océan qui nous a vu grandir.

Résumé (Français)

La communauté d'intelligence artificielle a été capable de créer des algorithmes pour certaines tâches non triviales dont la performance est comparable ou supérieure aux performances humaines. Bien que non triviales, ces tâches peuvent être résolues avec la puissance de calcul des ordinateurs en suivant une liste formelle de règles. Par exemple en 1997, Deep Blue, un super calculateur capable de jouer aux échecs a battu le champion du monde d'échecs: le russe Garry Kasparov. Ce fut la première victoire d'un ordinateur face à un champion du monde d'échecs. Deep Blue était basé sur un algorithme de force brute qui explorait un nombre maximum de coups dans un temps donné et choisissait le meilleur d'entre eux. Cependant, nonobstant ces succès, certaines tâches qui sont naturelles pour les êtres humains, telles que la classification d'objets ou la compréhension de conversations, restent compliquées à résoudre numériquement. Ces problèmes reposent sur des concepts intuitifs et sont compliqués à formaliser. C'est pourquoi une méthode similaire à Deep Blue ne peut pas capturer toutes les complexités cachées. Ces différents concepts nous amènent à considérer une nouvelle approche pour résoudre ces tâches: *l'apprentissage machine*.

Les différences et forces des algorithmes d'apprentissage machine par rapport aux autres algorithmes sont leurs capacités à apprendre sans l'aide d'opérateurs humains. En faisant des erreurs sur des exemples d'entraînement, les algorithmes s'adaptent afin de se rapprocher de la réponse désirée. Ils sont ensuite capable de faire des prédictions sur des situations qu'ils n'ont pas vu lors de leur entraînement. Cela leur donne l'unique capacité d'effectuer des tâches qui sont naturelles pour des êtres humains. Les potentielles applications de l'apprentissage machine sont nombreuses et le domaine a déjà été appliqué aux voitures autonomes, à la détection de cancers ou à la traduction. Récemment, une classe spécifique d'algorithme d'apprentissage machine a poussé les performances encore plus loin. Cette classe d'algorithme est appelée *apprentissage profond*.

Les modèles d'apprentissage profond sont un cas particulier d'apprentissage machine. Les différences résident dans leurs constructions. Ce sont des réseaux de neurones artificiels qui sont composés de nombreuses couches, elles mêmes composées de nombreux neurones. Cela amène ces architectures à posséder un grand nombre de neurones ce qui augmente le nombre de calculs à effectuer afin de les entraîner. Leur entraînement n'a été possible qu'avec l'arrivée de cartes graphiques puissantes ce qui explique leur avènement récent. Cependant, le succès de l'apprentissage profond est aussi dû aux développements récents de domaines mathématiques, qui ont aidé les réseaux de neurones à résoudre les problèmes désirés. Cette thèse porte sur l'étude des différentes interactions de l'apprentissage profond avec l'un de ces domaines appelé *transport optimal*.

Apprentissage profond et transport optimal. Les premiers architectures de réseaux de neurones peuvent être tracées au milieu du vingtième siècle [Rosenblatt 1958] et les premières architectures de réseaux profonds, ainsi que leur entraînement, trente ans plus tard [Rumelhart 1986, LeCun 1989, Lecun 1998]. Bien que ces architectures ont été inventés il y a une trentaine d'années, les chercheurs ont fait face aux limites de calculs des ordinateurs à cause de la taille des réseaux. En 2012, sans se décourager face à ces

limites, les chercheurs ont réussi à avoir de meilleurs résultats avec des modèles d'apprentissage profond qu'avec des algorithmes d'apprentissage machine sur un problème concret de classification. Cette année là, le réseau de neurones AlexNet [Krizhevsky 2012], entraîné sur une tâche de classification avec des gradients stochastiques et des cartes graphiques, a atteint la plus haute performance jamais réalisé sur la compétition ImageNet [Deng 2009]. Ce premier succès retentissant sur un problème de classification, couplé avec l'amélioration des cartes graphiques, a amené la communauté scientifique à reconsidérer et à développer le domaine de l'apprentissage profond.

Depuis ce succès, l'apprentissage profond est devenu la classe d'algorithme la plus performante pour réaliser des décisions basées sur des données. Pour pousser les performances de l'apprentissage profond encore plus loin, de nombreux travaux de recherche ont été conduits afin de trouver de nouvelles architectures de réseaux de neurones. Par exemple en classification, l'architecture ResNet s'est montrée particulièrement performante afin de réduire certains problèmes d'entraînement [He 2016]. En segmentation, les chercheurs ont développé l'architecture U-net [Ronneberger 2015]. Une autre approche qui a permis d'améliorer significativement les performances des modèles d'apprentissage est l'utilisation des minis lots couplée avec un algorithme d'optimisation. En effet, les réseaux de neurones sont entraînés avec des algorithmes d'optimisation de premier ordre comme par exemple la descente de gradient, malgré le fait que ces réseaux ne soient pas convexes. Dans un régime de données massives, utiliser la descente de gradient signifierait le calcul d'un gradient à chaque itération de la descente, ce qui coûte cher à calculer. C'est pourquoi afin d'accélérer le calcul d'une itération, la communauté s'est tournée vers l'optimisation stochastique et l'algorithme de gradient stochastique (SGD). Le prix à payer d'utiliser SGD est une convergence plus lente nécessitant un plus grand nombre d'itérations de l'algorithme. Cependant, son coût global reste inférieur car ses itérations sont moins coûteuses. Plusieurs variantes de SGD sont apparues telles qu'Adam [Kingma 2015] ou RMSprop [Tieleman 2012a]. Il est donc naturel de s'assurer de la convergence de ces algorithmes stochastiques vers une bonne solution lorsqu'une nouvelle fonction de coût est utilisée avec les réseaux de neurones.

Suivant ces développements fondamentaux, l'apprentissage profond est aujourd'hui état de l'art sur bon nombre de domaines à la fois en apprentissage supervisé et non supervisé, tels qu'en adaptation de domaine [Ganin 2015], en segmentation [Ronneberger 2015] et en modèles génératifs [Brock 2019]. Son utilisation s'étend aussi à d'autres domaines comme la physique [Schütt 2017] ou la biologie [Jumper 2021]. L'amélioration rapide de la compréhension de l'apprentissage profond est aussi possible grâce aux multiples ressources libres. Nous souhaiterions mentionner à ce titre les librairies PyTorch [Paszke 2017] et TensorFlow [Abadi 2015].

Cependant, il est important de mentionner que les réseaux de neurones profonds ne sont pas les seuls moteurs derrière les succès de l'apprentissage profond. Par exemple, les problèmes d'adaptation de domaine et les modèles génératifs nécessitent de comparer des ensembles de données entre eux. Sur ces problèmes, les réseaux de neurones transforment les données sur lesquelles ils sont appliqués en nouvelles données. Mathématiquement, cela correspond à créer une mesure image et donc à de nouveaux ensembles de données. Ainsi, résoudre ces problèmes nécessite une notion de distance entre des ensembles de données. Pour définir une telle distance, nous devons tout d'abord correctement représenter les données sur lesquelles nous travaillons.

Données d'entraînement comme mesure empirique. Afin de représenter les données mathématiquement, nous choisissons le cadre des mesures de probabilité. Les mesures de probabilité empiriques prennent la forme d'une somme de Diracs $\alpha_n = \sum_{i=1}^n a_i \delta_{\boldsymbol{x}_i}$, où $\forall i \in [1, n], a_i > 0$ et $\sum_{i=1}^n a_i = 1$. Cette représentation est appelée la représentation Lagrangienne et elle représente l'ensemble des données

comme un seul objet mathématique. Comme nous supposerons que les données sont tirées de manière indépendante et identiquement distribuées, nous choisissons des poids uniformes pour la mesure α_n, *i.e.*, $\forall i \in [1, n], a_i = \frac{1}{n}$. Pour des applications spécifiques, comme dans le Chapitre 6, il est possible de considérer d'autres poids.

Ainsi, la distance mathématiques dont nous avons besoin est une distance entre mesures de probabilité. Il est donc possible de recourir aux divergences de Csiszar [Csiszar 1975]. Dans cette thèse, nous nous concentrons sur le transport optimal pour comparer ces distributions. Le transport optimal peut être vu comme une méthode générique pour créer une distance entre des ensembles de points à partir d'une distance entre deux points. Contrairement aux divergences précédemment mentionnées, il considère des aspects géométriques pour comparer les différentes distributions. C'est donc naturellement que le transport optimal a été utilisé en apprentissage profond. Plus surprenant, l'apprentissage profond a été utilisé afin d'estimer le transport optimal continu.

Transport optimal et apprentissage profond. L'origine du transport optimal remonte au 18ème siècle dans les travaux de Gaspard Monge [Monge 1781]. Bien que le transport optimal a trouvé de nombreuses applications en mathématiques théoriques et appliquées [Santambrogio 2015, Villani 2009], il a aussi rencontré un certain succès en apprentissage machine [Peyré 2019]. Parmi ces succès, nous pouvons noter son utilisation en adaptation de domaine en transportant les données sources vers le domaine cible [Courty 2017a] ou bien en alignant les données plongées dans les réseaux de neurones [Damodaran 2018]. De plus, nous pouvons aussi mentionner son utilisation comme fonction de coût pour des problèmes de classification multi-étiquettes, où le transport a été utilisé entre les prédictions des modèles d'apprentissage et les étiquettes des données [Frogner 2015]. Cependant, l'application la plus célèbre du transport optimal en apprentissage profond est son utilisation avec les réseaux génératifs adversaires [Goodfellow 2014b] où il a été utilisé en tant que fonction dans *Wasserstein GAN* [Arjovsky 2017]. Nous terminons par les différentes applications en biologie [Schiebinger 2019], en astrophysique [Frisch 2002] ou en traitement du signal [Kolouri 2017].

Afin de comprendre ses succès et d'améliorer les performances des méthodes basées sur le transport optimal, les chercheurs ont étudié plusieurs aspects du transport optimal. Le premier aspect est l'estimation statistique du transport car en apprentissage profond, les données utilisées vivent en grandes dimensions. Plusieurs recherches ont été faites pour étudier le taux de convergence en population. Le cas du transport optimal exact a été étudié dans [Dudley 1969, Weed 2019] et sa variante entropique dans [Genevay 2019, Mena 2019]. Concernant le calcul du transport optimal, celui-ci est bien trop élevé pour que le transport soit utilisé sur des jeux de données à large échelle. C'est pourquoi beaucoup de recherches ont été faites afin de réduire ce coût de calcul. Avec une régularisation entropique, le transport peut être efficacement calculé sur des cartes graphiques avec l'algorithme de Sinkhorn [Cuturi 2013]. De plus, la complexité algorithmique de l'algorithme de Sinkhorn a été étudiée dans [Altschuler 2017]. D'autres stratégies basées sur une approximation du transport avec des minis lots ont été développées [Genevay 2018, Damodaran 2018, Salimans 2018] mais un formalisme rigoureux était manquant.

Enfin, des variantes du transport optimal ont trouvé de nombreuses applications. Par exemple, la distance de Gromov-Wasserstein permet de comparer des mesures de probabilités qui vivent dans différents espaces métriques. Elle a été appliquée pour des modèles génératifs [Mémoli 2011, Bunne 2019] ou sur des graphes [Vayer 2019a]. Des variantes avec des contraintes relaxées ont été utilisées afin de rendre le transport robuste aux données aberrantes [Chapel 2020, Balaji 2020].

Apprentissage profond et transport optimal. L'apprentissage profond peut-être utilisé pour approximer la formulation continue du transport optimal. Comme les mesures de probabilité sont inconnues dans beaucoup d'applications, nous nous reposons sur des mesures empiriques des mesures inconnues. Ainsi nous ne pouvons estimer que la version discrète du transport optimal. Les potentielles duaux du transport optimal peuvent être approximer avec un réseau de neurones comme dans [Arjovsky 2017], ou nous pouvons utiliser un réseau de neurones pour approximer une transformation entre les domaines.

L'objectif de cette thèse est d'améliorer l'état des connaissances sur :

- Comment le transport optimal peut-être utilisé en apprentissage profond pour de nouvelles applications ?
- Est-ce que l'approximations du transport optimal par minis lots définit un problème de transport?

Les contributions de cette thèse

Cette thèse a débuté en Novembre 2018 et contient des contributions sur les interactions du transport optimal et de l'apprentissage profond. Les différentes contributions peuvent être séparées en deux parties. Dans la première partie, nous développons l'utilisation du transport optimal en tant que fonction de coût pour l'apprentissage profond.

- Dans notre travail initial [Fatras 2021a], nous étudions le problème des étiquettes corrompues. Cela correspond au problème d'apprentissage supervisé où une proportion des étiquettes est corrompue, ce qui diminue les performances de généralisation des modèles d'apprentissage profond [Zhang 2017]. Pour résoudre ce problème, nous proposons d'utiliser le transport optimal, entre les étiquettes et les prédictions, comme fonction de coût dans l'entraînement virtuel adversaire (VAT) [Miyato 2018a]. VAT promeut une uniformité locale de la prédiction du classifier en pénalisant les changements brutaux de la classification pour de faibles perturbations des données. La pénalisation est "isotrope" entre toutes les classes, ce qui peut ne pas être optimal lorsque deux classes sont similaires. Incorporer le transport optimal dans VAT permet de moduler la force de la régularisation en fonction de la similarité sémantique des classes. Intuitivement, nous voulons des frontières complexes lorsque les classes sont similaires et des frontières lisses lorsque les classes ne le sont pas. Ce comportement est encouragé en choisissant de nouveaux coûts de transport comme discuté dans le Chapitre 5.

- Dans un second travail, nous avons utilisé le transport optimal afin de générer des données qui sont incorrectement classifiées pour un classifier donné [Burnel 2021]. Nous utilisons WGAN, une variante de réseaux adversaires génératifs [Goodfellow 2014b] basée sur le transport optimal, qui est capable de générer des données réalistes [Arjovsky 2017, Gulrajani 2017]. A partir du classifier fixé et des données d'entraînement, nous avons développé plusieurs stratégies pour créer de nouvelles mesures de probabilité pour les données d'entraînement. Ces nouvelles mesures donnent plus d'importance aux exemples incorrectement classifiés par le classifier. Ensuite, nous donnons cette nouvelle distribution au WGAN afin de générer des données incorrectement classifiées.

Les publications associées à ces contributions sont rassemblées dans la liste suivante:

- [Burnel 2021] Jean-Christophe Burnel, Kilian Fatras, Rémi Flamary and Nicolas Courty. *Generating natural adversarial Remote Sensing Images*. In: IEEE Transactions on Geoscience and Remote Sensing (TGRS), 2021.

- [Fatras 2021a] Kilian Fatras, Bharath Damodaran, Sylvain Lobry, Rémi Flamary, Devis Tuia and Nicolas Courty. *Wasserstein Adversarial Regularization on label noise.* In: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2021.

- [Burnel 2020] Jean-Christophe Burnel, Kilian Fatras, Nicolas Courty. *Generating natural adversarial hyperspectral examples with a modified Wasserstein GAN.* In: C&ESAR 2020.

Dans la seconde partie, nous étudions comment une approximation usuelle et performante du transport optimal amène à des questions ouvertes sur ce dernier. Afin de réduire la complexité algorithmique dans de nombreuses applications d'apprentissage profond, le transport est calculé entre des minis lots des données, tirés uniformément aléatoirement. Les réseaux profonds sont ensuite entraînés avec SGD. La stratégie des minis lots a aussi été utilisé avec le transport optimal de manière fructueuse [Genevay 2018, Muzellec 2020, Damodaran 2018]. Cependant, calculer le transport optimal entre des minis lots ne correspond pas à calculer le transport entre les mesures de probabilité entières. C'est pourquoi nous présentons une étude théorique du transport optimal entre minis lots en tant que nouvelle fonction de coût.

- Dans nos travaux [Fatras 2021c, Fatras 2021b], nous proposons d'abord un formalisme rigoureux de l'approximation par des minis lots. Nous construisons un estimateur non biaisé qui respecte les marginales des mesures de probabilités. Nous donnons aussi une solution en forme close pour des données 1D. Elle illustre les effets de l'approximation par minis lots sur le plan de transport. Nous démontrons aussi que les minis lots construisent des connexions non optimales. Nous proposons ensuite une nouvelle fonction de coût, basée sur le transport entre minis lots, qui corrige quelques pertes causées par l'approximation. Nous étudions ensuite théoriquement les propriétés statistiques et d'optimisation du transport optimal, afin de faire un premier pas dans la compréhension des différents succès en apprentissage profond.

- Dans un travail successif, nous étudions aussi quelques limites de l'approximation des minis lots [Fatras 2021b]. Nous montrons empiriquement que les connections non optimales peuvent diminuer les performances des réseaux de neurones sur des applications spécifiques. Ces connexions sont des conséquences du tirage des minis lots ainsi que des contraintes sur les marginales. En effet, au niveau des minis lots, le tirage des données peut créer des données aberrantes qui seront transportées. Afin de remédier à ce problème, nous proposons d'utiliser une variante du transport à contraintes relaxées appelée *transport optimal non balancé* avec les minis lots. Nous montrons ensuite sur différentes expériences que notre méthode surpasse l'approximation classique.

Ces contributions ont été publiées dans plusieurs publications:

- [Fatras 2021b] Kilian Fatras, Thibault Séjourné, Nicolas Courty and Rémi Flamary. *Unbalanced minibatch Optimal Transport; applications to Domain Adaptation.* In: International Conference on Machine Learning (ICML), 2021.

- [Fatras 2021c] Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval and Nicolas Courty. In: arXiv:2101.01792 (Under Review)

- [Fatras 2020b] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval and Nicolas Courty. *Learning with minibatch Wasserstein: asymptotic and gradient properties.* In: the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), 2020.

- [Fatras 2020a] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval and Nicolas Courty. *Divergence Wasserstein par lots*. In: 52èmes Journées de Statistiques de la Société Française de Statistique.

Durant sa thèse, l'auteur a aussi contribué à la librairie libre de transport optimal en python, qui a été reconnue par la communauté à travers une publication. Finalement, il a aussi travaillé avec sur des algorithmes stochastiques pour des régularisations complexes.

- [Flamary 2021] Rémi Flamary, et al. *POT: Python Optimal Transport*. In: Journal of Machine Learning Research (JMLR) - Open Source Software, 2021.
- [Pedregosa 2019] Fabian Pedregosa, Kilian Fatras and Mattia Casotto. *Proximal Splitting Meets Variance Reduction*. In: the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), 2019.

Dans la prochaine section, nous détaillons les différentes parties et chapitres constituant cette thèse.

Organisation du manuscrit.

Cette thèse est divisée en trois parties correspondant aux différentes contributions faites par l'auteur. La première partie fait un état des lieux de la littérature sur le transport optimal numérique ainsi que de son utilisation en apprentissage profond. La seconde partie rassemble les premières contributions du transport optimal comme fonction de coût en apprentissage profond. Après une courte introduction sur les exemples adversaires, nous présentons notre régularisation basée sur des perturbations des données et sur le transport optimal. Nous l'appliquons sur des problèmes de labels corrompus. Nous présentons ensuite notre stratégie de pondérations des poids des mesures de probabilité afin de générer des exemples incorrectement classifiés par un classifier pré-entraîné. Finalement, la troisième partie est dédiée à l'étude de l'approximation du transport optimal par des minis lots. Nous décrivons un formalisme rigoureux ainsi qu'une étude de ses limites. Nous détaillons à présent l'organisation du manuscrit.

Chapitre 2 rassemble des connaissances mathématiques et numériques élémentaires sur le transport optimal aux lecteurs. Il présente les concepts fondamentaux derrière les formulations de Kantorovich ainsi que de certaines variantes utilisées dans ce manuscrit. Parmi toutes les variantes, nous présentons la variante à régularisation entropique ainsi que la variante du transport non balancé et la variante Gromov-Wasserstein. Nous présentons aussi les algorithmes de résolutions du transport optimal régularisé avec l'entropie. Un lecteur familier avec ces concepts peut passer ce chapitre, cependant il définit aussi des notations qui seront utilisées tout au long de cette thèse.

Chapitre 3 décrit les différentes utilisations du transport optimal en tant que fonction de coût en apprentissage profond. Plus spécifiquement, il est divisé en deux parties. La première partie évoque les différents problèmes de classification telles que la classification supervisée, la classification semi-supervisée, la classification avec labels corrompus et l'adaptation de domaine. Nous détaillons ensuite la littérature sur les solutions aux problèmes de classification qui utilisent le transport optimal. La seconde partie est dédiée à l'étude de modèles génératifs et comment le transport optimal a été utilisé avec succès. Nous discutons d'abord des modèles génératifs standards puis comment ces modèles ont été modifiés afin d'utiliser le transport optimal. Un lecteur familier avec les applications standards d'apprentissage profond pourrait ne pas lire cette partie. Cependant, elles rassemblent des concepts et des algorithmes qui seront utilisés dans les différentes expériences présentées dans cette thèse.

Chapitre 4 nous amène à la seconde partie de ce manuscrit. Il présente les différents concepts d'exemples adversaires en apprentissage profond. Nous donnons une définition générale ainsi que les stratégies basiques afin de générer de tels exemples. Nous expliquons ensuite comment ces exemples peuvent pirater des modèles pré-entraîner et les différents concepts derrière ces attaques pirates. Finalement, nous montrons comment ces exemples peuvent être utilisés pour définir une nouvelle stratégie d'apprentissage appelée *entraînement adversaire virtuel* (VAT), pour des problèmes de classification semi-supervisé. Cet algorithme promeut une prédiction locale uniforme autour de chaque donnée d'entraînement. Il le fait en pénalisant de large changement dans la classification.

Chapitre 5 présente notre première contribution. Il étudie le problème de classification avec des labels corrompus. Basé sur l'entraînement adversaire virtuel, nous utilisons le transport optimal dans cette régularisation qui permet de lisser les frontières entre les classes très différentes, comme le fait VAT. Cependant, le transport optimal nous permet aussi d'obtenir des frontières complexes entre les classes proches les unes des autres. Nous évaluons ensuite notre méthode sur des jeux de données standards et concrets.

Chapitre 6 présente une nouvelle distribution de probabilité qui permet de générer des exemples incorrectement classifiés par un classifier pré-entraîné. Après avoir décrit plusieurs stratégies de pondération des poids, nous créons empiriquement une nouvelle distribution basée sur une fonction *softmax* . Nous utilisons ensuite un modèle génératif afin de générer ces données incorrectement classifiées. Afin de montrer la pertinence de cette approche, nous évaluons les exemples générés afin de pirater des classifiers entraînés sur des données satellites.

Chapitre 7 ouvre la troisième et dernière partie de cette thèse. Nous présentons l'approximation du transport optimal, qui est étudié théoriquement et empiriquement dans cette troisième partie. Afin d'étudier cette approximation théorique, nous rappelons quelques résultats avancés de statistiques et d'optimisation du transport optimal. Nous présentons les différentielles généralisées de Clarke qui définissent une notion de régularité pour les fonctions localement Lipschitz [Clarke 1990]. Nous terminons ce chapitre par une courte introduction aux U-statistiques qui apparaissent dans le formalisme des minis lots [Hoeffding 1948, Hoeffding 1963].

Chapitre 8 est dédié à la définition rigoureuse du formalisme qui permet d'exprimer le transport optimal par minis lots. Nous commençons tout d'abord par définir un formalisme dans le cas de mesures empiriques et nous illustrons les conséquences sur le plan de transport. Nous prouvons aussi qu'une solution en forme close existe lorsque les données vivent en 1D. Ensuite, nous définissons un formalisme général pour toutes distributions et nous montrons que le transport optimal par minis lots n'est pas une distance. Cela nous amène à définir une nouvelle fonction qui corrige certaines pertes du transport par minis lots. Nous étudions la positivité de cette fonction de coût ainsi que ces performances statistiques. Après, nous présentons des bornes de déviation entre les estimateurs que nous avons créés et leurs espérances pour différentes hypothèses sur les mesures de probabilité. Nous terminons ce chapitre avec des expériences sur des modèles génératifs et sur du transfert de couleur.

Chapitre 9 termine les contributions de cette thèse. Nous discutons les différentes limites du transport optimal par minis lots à cause du tirage des minis lots ainsi que de la formulation originale du transport optimal. Comme le transport a des contraintes sur les marginales, il est sensible aux données aberrantes.

Or nous montrons que le tirage des minis lots peut créer artificiellement de telles données aberrantes. Afin de réduire ces sensibilités, nous proposons de remplacer le transport optimal entre les minis lots par une variante qui a des marginales relaxées telle que le transport optimal non balancé. Nous démontrons empiriquement que le transport optimal non balancé entre les minis lots a de meilleures performances que le transport optimal original entre les minis lots sur des expériences de flots de gradients et d'adaptation de domaine.

Chapitre 10 conclut cette thèse et discute les potentielles directions pour les futures travaux sur les interactions entre le transport optimal et l'apprentissage profond.

Appendice A rassemble les différentes preuves des résultats qui ont été annoncés dans les chapitres précédents.

Contents

Notations

Linear algebra

$\mathbf{a}, \mathbf{A}$	all vectors in $\mathbb{R}^d$ and matrices are written in bold. The coordinates will be written $A_{i,j}$ for matrices and a_i for vectors without bold.
$\|.\|, \langle , \rangle$	a norm (depends on the context) and an inner product
ℓ_p	denotes the standard $\|.\|_p$ norm
$\mathrm{tr}, \det$	is the trace operator for matrices *i.e.* $\mathrm{tr}(\mathbf{P}) = \sum_{i,j} P_{i,j}$ and the determinant operator
$\|.\|_{\mathcal{F}}$	is the Frobenius norm for matrices *i.e.* $\|\mathbf{P}\|_{\mathcal{F}} = \sqrt{\mathrm{tr}(\mathbf{P}^T\mathbf{P})}$
$\langle ., . \rangle_{\mathcal{F}}$	is the inner product for matrices *i.e.* $\langle \mathbf{P}, \mathbf{Q} \rangle_{\mathcal{F}} = \mathrm{tr}(\mathbf{Q}^T\mathbf{P})$
$\oslash$	element-wise division operator for two vectors, *i.e.* $\mathbf{u} \oslash \mathbf{v} = (\frac{u_i}{v_i})_i$
$\mathcal{M}_m(\mathbb{R})$	set of square (real) matrices of size m

Measure theory

$\mathcal{M}(\mathcal{X})$	the set of Borel finite signed measures on a space $\mathcal{X}$
$\mathcal{M}_+^1(\mathcal{X})$	the set of probability measures on a space $\mathcal{X}$
α, β	probability measures
$\alpha \otimes \beta$	the product measure of two probability measures α, β, *i.e.*, $\alpha \otimes \beta(A \times B) = \alpha(A)\beta(B)$
$\alpha^{\otimes m}$	m-product measure α
$\mathrm{supp}(\alpha)$	the support of α
$\delta_{\boldsymbol{x}}$	the Dirac measure on $\boldsymbol{x}$, *i.e.* $\delta_{\boldsymbol{x}}(\boldsymbol{y}) = 0$ if $\boldsymbol{y} \neq \boldsymbol{x}$ else 1
$\#$	the push forward operator.
$\mathcal{U}(\alpha, \beta)$	the set of couplings of two continuous probability measures α, β
$\mathfrak{L}(\alpha, \beta, c)$	the Kantorovitch cost between two probability measures α, β
$W_p(\alpha, \beta, c)$	the p-Wasserstein distance between two probability measures α, β
$\mathcal{G}W_p(\alpha, \beta, c_{\mathcal{X}}, c_{\mathcal{Y}})$	the p-Gromov-Wasserstein distance between two probability measures α, β
$\mathrm{OT}_{\phi}^{\tau}(\alpha, \beta, c)$	the unbalanced optimal transport cost
h	optimal transport kernel
ε	entropic regularization coefficient
τ	marginal penalty coefficient
C^m	ground cost matrix of size $m \times m$
$\mathrm{nSG}(\rho, \sigma^2)$	space of norm sub-Gaussian random variables
α_n	empirical distribution
n	number of data
$\boldsymbol{X}$	data n-tuple
Σ_n	the set of probability vectors in $\mathbb{R}_+^n$ (or histograms with n bins), *i.e.* $\Sigma_n = \{\alpha \in \mathbb{R}_+^n \mid \sum_{i=1}^n a_i = 1\}$
$\mathbf{a} \in \Sigma_n$	probability vector
$\boldsymbol{u} \in \Sigma_n$	uniform probability vector
Σ	set of all sequences of probability vectors

Functions

$C(\mathcal{X}), C(\mathcal{X}, \mathcal{Y})$	the set of continuous functions form $\mathcal{X}$ to $\mathbb{R}$ (*resp.* from $\mathcal{X}$ to $\mathcal{Y}$)
$C_b(\mathcal{X}), C_b(\mathcal{X}, \mathcal{Y})$	the set of continuous and bounded functions form $\mathcal{X}$ to $\mathbb{R}$ (*resp.* from $\mathcal{X}$ to $\mathcal{Y}$)
$C^p(\mathcal{X})$	the set of functions of class C^p.
$Lip_k(\mathcal{X})$	the set of Lipschitz functions on $\mathcal{X}$ with Lipschitz constant k
$L^p(\alpha)$	the set of p-integrable functions with respect to a measure α, *i.e.* $f \in L^p(\alpha)$ if $\int \lvert f\rvert^p \mathrm{d}\alpha < +\infty$
∇	the gradient operator

Minibatch

m	minibatch size
I	index m-tuple
$[\![n]\!]^m$	set of all index m-tuples
$\mathcal{P}^m$	set of all index m-tuples without replacement
$\mathcal{P}^{m,o}$	set of all ordered index m-tuples without replacement
$\sum_{i\in I} f(i)$	sum over all elements of tuple I
$\Pi_{i\in I} f(i)$	product over all elements of tuple I
w	reweighting function
P	probability law to draw index m-tuples
$\overline{h}^m_{w,P}$	minibatch kernel OT loss
$\widetilde{h}^{m,k}_{w,P}$	incomplete MBOT loss
$\Lambda_{h,w,P}$	debiased minibatch loss
$\widetilde{\Lambda}^k_{h,w,P,C(\boldsymbol{X},\boldsymbol{Y})}$	incomplete debiased MBOT loss
$\overline{\Pi}^h_{w,P}$	MBOT plan
$\widetilde{\Pi}^{h,k}_{w,P}$	incomplete MBOT plan
$\overline{h}^{\mathtt{U}}$	MBOT loss (sampling without replacement)
$\overline{h}^{\mathtt{W}}$	MBOT loss (sampling with replacement)
$\overline{\Pi}^h_{\mathtt{U}}$	OT plan (sampling without replacement)
$\overline{\Pi}^h_{\mathtt{W}}$	OT plan (sampling with replacement)
$\mathtt{Loc}_{\mathtt{A}}$	local mean constraint
$\mathtt{Loc}_{\mathtt{G}}$	local product constraint
D, γ	minibatch local constraints

Acronyms

OT, UOT	are the acronyms for Optimal Transport and Unbalanced Optimal Transport
W,GW	stands respectively for Wasserstein and Gromov-Wasserstein
MBOT, UMBOT	are the acronyms for MiniBatch Optimal Transport and Unbalanced MiniBatch Optimal Transport
ML, DL	are the acronyms for Machine Learning and Deep Learning respectively

Chapter 1

Introduction

Contents

The community of artificial intelligence has been able to design human-level performance algorithms for non trivial tasks. While being non trivial, these tasks can be solved using list of formal rules and the calculus power of computers. For instance in 1997, Deep Blue, a supercalculus computer which played chess, was the first computer to beat a chess world champion, the Russian Garry Kasparov. It was based on a *brute force* algorithm which explored the maximum possible number of moves in a certain time and chose the best one among them. However, despite these successes, some challenging tasks that are natural to human beings, like classifying different objects or understanding conversations, remained hard to solve. These problems rely on intuitive concepts which are hard to formalize and a method like the one used by Deep Blue can not catch all the hidden complexity. We thus need to rely on another approach to perform these tasks, which is called *Machine Learning*.

The difference and strength of machine learning algorithms lie in their ability to learn from data, without requiring human operators. By making mistakes on training samples, the algorithms are able to adapt in order to match the desired outcome. They are then capable of making predictions on situations they have not seen during their training. It gives them the unique ability to achieve repetitive tasks which are natural to human beings. The potential applications of machine learning are wide and the domain has already been applied for autonomous vehicles, cancer detection or language translation. Recently, a specific class of machine learning algorithms pushed the performances of the domain even further by improving almost all applications. This class of algorithms is called *Deep Learning*.

Deep learning models are a particular instance of machine learning algorithms. They are artificial neural networks which are composed of many layers with many neurons. This leads to a very high number of neurons and training these networks is computationally challenging. It has only be possible thanks to a new class of hardware called *Graphics Processing Units* (GPUs). These networks allow a more complex representation of training samples and thus they can make more complex predictions. However, deep learning's recent successes are also due to the development of recent mathematical fields, which helped the neural networks to achieve the desired tasks. This thesis is about studying the different interactions of deep learning with one of these mathematical areas called *Optimal Transport* (OT).

1.1 Deep Learning and Optimal Transport

The rise of Deep Learning. The first neural network architecture can be traced back to the middle of the twentieth century [Rosenblatt 1958] and the first deep neural network architectures, as well as their training procedure, thirty years latter [Rumelhart 1986, LeCun 1989, Lecun 1998]. In spite of their early design, researchers faced computational limits due to the size of the networks. Undeterred by these limits, in 2012, researchers made a first breakthrough on a real-world problem using these architectures. That year, the heavy neural network AlexNet [Krizhevsky 2012], trained on a classification task with stochastic gradient descent (SGD) and GPUs [Robbins 1951a], achieved by a large margin the highest performance ever on the ImageNet competition [Deng 2009]. This first success in classification problems, coupled with the computational improvements resulting from high performing GPUs, led the research community to reconsider and develop the area of *Deep Learning*.

Since this success, deep learning has arisen as the current most competitive method to make data-driven decisions. To improve its performances even further, a lot of work has been done to find relevant architectures depending on the problem. For instance in computer vision and classification, the ResNet architecture empirically eased the training of deep neural networks [He 2016]. In segmentation, researchers developed the U-net architecture [Ronneberger 2015]. Another successful approach that improved deep learning results was the use of minibatch training, coupled with a performing optimizer. Indeed, neural networks are trained using gradient-based algorithms, such as gradient descent, despite the fact that they are not convex. In a big data regime, using gradient descent would mean to compute a full gradient at each iteration. Thus to accelerate the training of neural networks, the community has relied on stochastic optimization [Bottou 2010]. Stochastic optimization only needs to compute a gradient of a minibatch of data. The price to pay for using SGD is a slower convergence. However, the overall performance is faster as one SGD iteration is a lot cheaper than one iteration of gradient descent. Several SGD variants were published to fasten the training of neural networks or improve their results such as Adam [Kingma 2015] or RMSprop [Tieleman 2012a]. It is thus a natural question to investigate the convergence of SGD towards a good solution when a new loss function is used with neural networks.

Following these corner-stone developments, deep learning has reached state-of-the-art results on many supervised and unsupervised problems, such as for instance domain adaptation [Ganin 2016], segmentation [Ronneberger 2015] and generative modelling [Brock 2019]. Its use also expanded to other research domains such as physics [Schütt 2017] and biology [Jumper 2021]. The rapidly growing knowledge of deep learning is also possible thanks to the open-source mindset in the field. We highlight several deep learning Python open-source libraries such as PyTorch [Paszke 2017] or TensorFlow [Abadi 2015], as well as the JMLR open-source journal edition, which encourages the creation of open-source libraries.

Nonetheless, it is important to mention that deep neural networks were not the only reason behind some recent machine learning successes. For instance, domain adaptation and generative modelling are applications which require to compare sets of samples. A deep neural network is a transformation applied on samples that mathematically defines a pushforward operator. When samples are fed to the network, it leads to new sets of "embedded" samples, which correspond to pushforward distributions, that we want to compare. Thus some mathematical notion of distance between sets of samples is needed. In order to define such a distance, we first need to correctly represent sets of training samples as data distributions.

Training data as empirical measure. We consider the framework of probability measures, where we associate an empirical probability measure to the collection of training data $(\boldsymbol{x}_i)_{i=1}^n$. The empirical measure takes the form of a sum of Diracs $\alpha_n = \sum_{i=1}^n a_i \delta_{\boldsymbol{x}_i}$, where $\forall i \in [1, n], a_i > 0$ and $\sum_{i=1}^n a_i = 1$. This

representation is called the *Lagrangian representation* and it represents a set of data as one mathematical object. As we suppose that the data are independent and identically distributed, we choose the weights as uniform, *i.e.*, $\forall i \in [1, n], a_i = \frac{1}{n}$. For specific applications, as presented in Chapter 6, it is possible to reweight this distribution.

Thus the mathematical notion of distance we need is a distance between probability measures. It is possible to rely on divergences like Maximum Mean Discrepancies [Gretton 2012] or Csiszar divergences [Csiszar 1975]. In this thesis, we focus on optimal transport to compare distributions. Optimal transport can be seen as a generic method to lift a distance between two points to a distance between sets of points. Unlike the methods mentioned previously, it considers geometric notions to compare probability measures. It is thus natural that optimal transport was used in deep learning to measure the discrepancy between probability measures. On the other hand, deep learning was also used to estimate the continuous setting of optimal transport.

Optimal Transport for Deep Learning. The origin of optimal transport traces back to the 18th century in the work of Gaspard Monge [Monge 1781]. While optimal transport has found numerous applications in different areas of theoretical and applied mathematics [Santambrogio 2015, Villani 2009], it has also found many successes in machine learning [Peyré 2019]. Among them, we can highlight its use in domain adaptation, where it reached state-of-the-art results, either by transporting source data into the target domain [Courty 2017a] or by aligning embeddings of domains [Damodaran 2018]. Furthermore we can also mention its use as a loss function for multi-label classification [Frogner 2015], where optimal transport was defined between labels and took into account the closeness of classes. Yet, the most famous application of optimal transport in deep learning is probably in Generative Adversarial Networks (GANs), where it was used as a loss function in the so-called *Wasserstein GAN* [Arjovsky 2017]. Following GANs, an optimal transport variant of AutoEncoders has also been designed [Tolstikhin 2018]. We finish our overview by mentioning the different applications in biology [Schiebinger 2019], astrophysics [Frisch 2002] or signal processing [Kolouri 2017].

In order to understand its successes and to improve its performances in deep learning, researchers studied several aspects of optimal transport. The first aspect is the statistical estimation of optimal transport, as in deep learning we mostly deal with high dimensional data. Several research have been done to study its convergence rate in population, known as sample complexity. The exact optimal transport case was studied in [Dudley 1969, Weed 2019] and the entropic-regularized OT case in [Genevay 2019, Mena 2019]. On the other hand, the computational complexity of optimal transport makes it prohibitive to use on large scale datasets. Thus, a lot of research has been done to accelerate the computation of optimal transport. Using an entropic regularization, optimal transport can be efficiently computed on GPUs with the Sinkhorn algorithm [Cuturi 2013]. Furthermore, the algorithmic complexity of the Sinkhorn algorithm was provided in [Altschuler 2017]. Other strategies based on a minibatch approximation were developed in order to reduce the computational complexity [Genevay 2018, Damodaran 2018, Salimans 2018], but a rigorous formalism was lacking.

On another note, variants of optimal transport has found numerous applications. For example, the Gromov-Wasserstein distance allows to compare probability measures which lie in different metric spaces. It has been applied in generative modelling [Mémoli 2011, Bunne 2019] or graphs [Vayer 2019a]. Relaxed marginal variants of optimal transport have been used in many applications to make the transport robust to outliers [Chapel 2020, Balaji 2020].

Deep Learning for Optimal Transport. Deep learning can be used to approximate the continuous formulation of optimal transport. As the probability measures are unknown in many applications, we rely on empirical estimations of these measures. Thus we can only estimate a discrete optimal transport cost. The dual potential of optimal transport can be approximated with a neural network, as done in [Arjovsky 2017], or we can also use a neural network to approximate a mapping between domains [Seguy 2018].

The purpose of this thesis is to bring new knowledge on:

- How can optimal transport be used in deep learning for new applications ?
- Do optimal transport approximations used in deep learning still define a transport problem ?

1.2 Contributions of the thesis

This thesis, started in November 2018, makes some contributions on the use of optimal transport in deep learning. The different contributions can be divided in two distinct parts. In the first part, we develop the use of optimal transport as a loss function for deep learning.

- In the initial work [Fatras 2021a], we study the label noise problem. It corresponds to a supervised learning setting where a proportion of labels are corrupted, which hurts generalization of deep neural networks [Zhang 2017]. To solve this problem, we propose to use optimal transport, between labels, as a loss function in the well known perturbation-based regularization, the *Virtual Adversarial Training* (VAT) [Miyato 2018a]. VAT promotes a uniform local prediction of the classifier by penalizing large local changes in the classification. The penalization is "isotropic" between all classes, which might not be optimal when two classes are similar. Incorporating optimal transport in VAT allows us to modulate the regularization strength with respect to the semantic similarity of classes. Intuitively, we want to have complex classifier boundaries for classes which are similar to each other and smooth boundaries for classes which are not. This behaviour is encouraged with well-chosen new ground costs as discussed in Chapter 5.
- In a second work, we use optimal transport to generate samples which are misclassified for a given classifier [Burnel 2021]. We rely on WGAN, an optimal transport variant of generative adversarial networks [Goodfellow 2014b], which has been used successfully to generate highly realistic data [Arjovsky 2017, Gulrajani 2017]. Given the classifier and training data, we develop several weighting strategies to create new probability measures for training data, where misclassified data receive bigger weights than correctly classified data. Then, we feed the new probability measure to the WGAN in order to generate misclassified data.

The publications associated to these contributions are gathered in the following list:

- [Burnel 2021] Jean-Christophe Burnel, Kilian Fatras, Rémi Flamary and Nicolas Courty. *Generating natural adversarial Remote Sensing Images*. In: IEEE Transactions on Geoscience and Remote Sensing (TGRS), 2021.
- [Fatras 2021a] Kilian Fatras, Bharath Damodaran, Sylvain Lobry, Rémi Flamary, Devis Tuia and Nicolas Courty. *Wasserstein Adversarial Regularization on label noise*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2021.

- [Burnel 2020] Jean-Christophe Burnel, Kilian Fatras, Nicolas Courty. *Generating natural adversarial hyperspectral examples with a modified Wasserstein GAN*. In: C&ESAR 2020.

In the second part, we study how a usual and successful deep learning approximation can lead to open theoretical questions in optimal transport. In order to reduce the computational complexity in many deep learning applications, the loss function is computed between minibatches of data, drawn uniformly at random. The deep neural network is then trained with SGD. The minibatch strategy has also been used with optimal transport and showed impressive results [Genevay 2018, Muzellec 2020, Damodaran 2018]. However, computing optimal transport between minibatches of data does not correspond to computing optimal transport between the full input measures. Thus we present a theoretical study of the optimal transport minibatch approximation.

- In our works [Fatras 2020b, Fatras 2021c], we first provide a rigorous formalism of the minibatch approximation. We build unbiased empirical estimators which respect the marginals of input measures. We also provide a closed-form solution when data lie in 1D, which illustrates the effect of the minibatch approximation on the transport plan, and we empirically show that minibatch optimal transport creates non-optimal connections. We then propose a new loss function, based on minibatch OT, to correct some loss of distance properties, due to the minibatch approximation. We then theoretically study the statistical and optimization properties of minibatch optimal transport, to make a first step in understanding its successes in deep learning.

- In a successive work, we also study some limits of the minibatch approximation [Fatras 2021b]. We empirically show that the non-optimal connections can harm performances of neural networks on specific applications. These connections are consequences of the minibatch sampling and the marginal constraints. Indeed, at the minibatch level, the sampling can create outliers which are transported. To mitigate this effect, we propose to use a relaxed marginal OT variant, called *Unbalanced Optimal Transport* (UOT), at the minibatch level. We then show on different experiments that our method successfully outperforms minibatch OT.

These contributions have led to several publications which are gathered in the list below:

- [Fatras 2021b] Kilian Fatras, Thibault Séjourné, Nicolas Courty and Rémi Flamary. *Unbalanced minibatch Optimal Transport; applications to Domain Adaptation*. In: International Conference on Machine Learning (ICML), 2021.

- [Fatras 2021c] Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval and Nicolas Courty. In: arXiv:2101.01792 (Under Review)

- [Fatras 2020b] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval and Nicolas Courty. *Learning with minibatch Wasserstein: asymptotic and gradient properties*. In: the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), 2020.

- [Fatras 2020a] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval and Nicolas Courty. *Divergence Wasserstein par lots*. In: 52èmes Journées de Statistiques de la Société Française de Statistique.

During his PhD, the author also contributed to the Python Optimal Transport open-source library which was acknowledged by the community through a publication. Finally, he also had a collaboration with Fabian Pedregosa on stochastic optimization algorithms for complex penalties.

- [Flamary 2021] Rémi Flamary, et al. *POT: Python Optimal Transport.* In: Journal of Machine Learning Research (JMLR) - Open Source Software, 2021.
- [Pedregosa 2019] Fabian Pedregosa, Kilian Fatras and Mattia Casotto. *Proximal Splitting Meets Variance Reduction.* In: the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS), 2019.

In the next section, we provide a detailed presentation of the different parts and chapters constituting this thesis.

1.3 Organization of the manuscript

The thesis is divided in three parts corresponding to the different contributions made by the author. The first part reviews the basic concepts of numerical optimal transport and its use in deep learning. The second part discusses the first contributions on optimal transport as a loss function in deep learning. After a short introduction about adversarial examples, we present our perturbation-based regularization, coupled with optimal transport, for label noise problems. We then present our reweighting strategies of empirical distributions in order to generate data which are misclassified by a given classifier. Finally, the third and final part is focused on studying the minibatch approximation of optimal transport. We provide a rigorous formalism of this approximation and then, empirically study some of its limits. We now detail the organization of the manuscript.

Chapter 2 is written to bring basic mathematical and numerical knowledge of optimal transport to the reader. It presents the fundamental concepts behind the Kantorovich formulation as well as different variants used in this manuscript. Among all the different variants, we give a particular focus on entropic-regularized optimal transport, unbalanced optimal transport and the Gromov-Wasserstein distance. We also present basic solvers of specific optimal transport variants. A reader familiar with the concepts of numerical optimal transport may skip this part, although it defines crucial notations that will be used along the thesis.

Chapter 3 reviews the use of optimal transport as a loss function in deep learning. More specifically, it is split in two parts. The first part gathers the different classification problems such as supervised learning, semi-supervised learning, learning with label noise and domain adaptation. We then review the literature on classification with an optimal transport loss. The second part is dedicated to study generative modelling problems with optimal transport. We first discuss the standard generative model methods and then how optimal transport has been used to improve these models. A reader familiar with standard deep learning applications might skip this chapter but it gathers basic concepts and algorithms that will be used in the different experiments presented in this manuscript.

Chapter 4 leads us to the second part of this manuscript. It introduces the concept of adversarial examples in deep learning. We give a general definition and the different basic strategies to generate such examples. We then explain how these adversarial examples can attack a pre-trained classifier and the different concepts associated to these attacks. Finally, we show how these adversarial examples can be used to define a new training algorithm, called *Virtual Adversarial Training* (VAT), for semi-supervised problems. This algorithm promotes a uniform prediction of the classification around each input. It does so by penalizing large changes in the classification.

Chapter 5 presents the first contribution of this manuscript. It studies the classification problem of learning with noisy labels. It is a setting where a fraction of the labels are corrupted. Based on the Virtual Adversarial Training, we use an optimal transport regularization to force the neural network to smoothen its decision boundary between classes that are very different from each other, as VAT does. However, optimal transport allows us to keep complex boundaries between classes close to each other. We then evaluate our method on standard and real-world label noise benchmarks.

Chapter 6 aims at introducing a new reweighted empirical distribution in order to generate data which are misclassified by a pre-trained classifier. After defining several strategies to reweight the different samples, we empirically design a new empirical distribution based on a softmax function. We then use a generative model to generate misclassified data. To show the relevance of this approach, our reweighted distribution is applied on real-world remote sensing data to fool pre-trained classifiers.

Chapter 7 opens the third and final part of this manuscript. We present the minibatch approximation of optimal transport, which is studied theoretically and empirically in this third part. In order to study this approximation theoretically, we review more advanced statistical and optimization results of optimal transport. We introduce the Clarke generalized gradients, which define a notion of regularity for locally Lipschitz functions [Clarke 1990]. We finish with the review of U-statistics [Hoeffding 1948, Hoeffding 1963] which arises in the sampling formalism of minibatches.

Chapter 8 is dedicated to define a rigorous formalism of minibatch optimal transport. We start by defining a formalism for uniform empirical measures and illustrate the consequences on the transport plan. We also highlight a closed-form formula when data lie in 1D. Next, we define a general formalism for any discrete distribution. Afterwards, we review the metric properties of minibatch optimal transport which leads us to define a new loss function to correct some loss of metric properties. We also study the positivness of this loss function. We then present deviation bounds and optimization properties for different hypotheses on probability measures. We finish with extensive experiments in generative modelling and color transfer.

Chapter 9 closes the contributions of the manuscript. We discuss the different limits of minibatch optimal transport due to the minibatch sampling and the OT formulation. As OT has marginal constraints, it is sensitive to outlier data and the minibatch sampling artificially creates such outliers. To mitigate these limits, we propose to replace optimal transport by a variant with relaxed marginals known as the *Unbalanced Optimal Transport*. We then empirically show that unbalanced minibatch optimal transport outperforms minibatch optimal transport on gradient flows and domain adaptation experiments.

Chapter 10 concludes this thesis and reviews the potential directions for future works on the interactions of optimal transport and deep learning.

Appendix A gathers the proof of the different results from the precedent chapters.

Part I

Optimal Transport and Machine Learning

Chapter 2

Introduction to computational Optimal Transport

Contents

In this chapter, we rigorously define the notion of optimal transport. We first detail the probability measure framework which is ubiquitous in machine learning. We then review the optimal transport definition, its metric properties and its computational and memory complexities. Afterwards, we introduce the entropic-regularized optimal transport and review the benefits and consequences of this formulation. The presentation of two optimal transport variants will close this chapter.

2.1 Motivations: dealing with measures

Machine Learning (ML) models aim at learning the best models that achieve a given goal from training data. The training data depend on the problem considered. In computer vision, data are images or videos, while in natural language processing, data are sentences or texts. In this manuscript, we mostly consider the area of computer vision and we focus on images in particular. Specifically, the images represent daily objects, such as cars or planes, human faces or remote sensing images. The tasks we want the models to achieve can be split in two categories. The first category is a classification task, where we want a model

to predict the class of a given image. The second category is image generation, where we want a model to generate images which would look like training data. Machine learning models are trained to achieve these desired tasks by minimizing a certain quantity. These quantities are estimated using probability distributions, which is a framework that allows to represent a set of data as one mathematical object. In this section, we define formally the probability distributions in the discrete and continuous cases. We then show how it appears in the two machine learning problems mentioned above. We finish this section by discussing how probability distributions are estimated in practice when they are unknown.

Definitions of probability distributions. We now give a formal definition of probability measures and we start with the discrete case. We consider n samples from the data space $\mathcal{X}$ as $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$. Let Σ_n denote the probability simplex of order n, *i.e.,* the set of positive vectors which sum to 1, namely

$$\Sigma_n = \{\boldsymbol{a} \in \mathbb{R}^+ : \sum_{i=1}^{n} a_i = 1\}. \tag{2.1}$$

In order to avoid degeneracy issues, we suppose that all $a_i > 0$. A discrete probability measure is defined as follows.

Definition 1. *Given a tuple of weights $\boldsymbol{a} \in \Sigma_n$ and locations $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\} \in \mathcal{X}$, a discrete probability measure reads*

$$\alpha = \sum_{i=1}^{n} a_i \delta_{\boldsymbol{x}_i}, \tag{2.2}$$

Where $(\delta_{\boldsymbol{x}_i})_{i\in[1,n]}$ is a Dirac at the position $\boldsymbol{x}_i$. The set of Diracs $\{\delta_{\boldsymbol{x}_1}, \cdots, \delta_{\boldsymbol{x}_n}\}$ is called the support of the measure α.

The probability measure framework also allows to handle the case where we have an infinite number of samples, that we refer to the continuous case. This occurs in generative modelling where we can generate an infinite number of data. In this setting, the probability measure definition needs to be adapted. Consider the set of Radon measures $\mathcal{M}(\mathcal{X})$ on the data space $\mathcal{X}$ which is equipped with a distance d. We denote the set of probability measures on $\mathcal{X}$ as $\mathcal{M}_+^1(\mathcal{X})$. We say that $\alpha \in \mathcal{M}_+^1(\mathcal{X})$ is a probability measure if it can be integrated against any continuous function f on $\mathcal{X}$ with compact support, formally

$$\int_{\mathcal{X}} f(\boldsymbol{x}) d\alpha(\boldsymbol{x}) \in \mathbb{R} \text{ and } \int_{\mathcal{X}} d\alpha(\boldsymbol{x}) = 1.$$

We now give concrete examples on why the measure framework is adapted for the tasks we consider.

Motivating examples. In generative modelling, the purpose is to generate training-like data. A parametrized model denoted β_θ where θ is the parametrization vector, *i.e.,* $\theta \mapsto \beta_\theta$, is designed to generate images. To make them look like training data, we want to minimize the distance between the generated data and the training data with respect to parameters θ. We achieve this goal by minimizing the distance between the corresponding probability measures of generated data β_θ and training data α. In order to compute the dissimilarities between measures, it is common to rely on a contrast function or divergence $L : \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{X}) \to \mathbb{R}_+$. Thus, the goal is to find the optimal θ^* which minimizes the distance L between the measures β_θ and α, *i.e.* $\theta^* = \text{argmin}_\theta\, L(\alpha, \beta_\theta)$.

The purpose of classification is to predict the class of a given image. In the supervised learning setting, we have access to an image $\boldsymbol{x}_i$ and its label $\boldsymbol{y}_i$, which describes the class of the given image $P(\boldsymbol{y}_i|\boldsymbol{x} = \boldsymbol{x}_i)$. The label is a one-hot vector, where all coordinates are set to 0 except the one which represents the class

of $\boldsymbol{x}$, which is set to 1. It is thus a probability vector, *i.e.,* $\boldsymbol{y}_i \in \Sigma_{n_C}$ where n_C is the number of classes. To classify data, we wish to learn a function $f : \mathcal{X} \mapsto \Sigma_{n_C}$ which describes the relationship between a sample $\boldsymbol{x}$ and its label $\boldsymbol{y}$. The function is learnt by minimizing the errors between its predictions and the correct labels, which are probability distributions with the same support. The error takes the form of a distance between these probability measures. It is typically calculated with the cross-entropy.

Definition 2. *Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be two probability vectors. Then the cross-entropy loss reads*

$$L_{CE}(\mathbf{a}, \mathbf{b}) = -\sum_{i=1}^{n} a_i \log(b_i). \tag{2.3}$$

We give a more detailed description of supervised learning in Section 3.1.1 as well as some of its variants.

Empirical estimation. Ideally, machine learning models would be trained on the true probability distribution of data α. However in practice, the distribution α is unknown and we have access to independent and identically distributed (*i.i.d.*) random samples drawn from α instead. We can thus estimate the measure α using a discrete probability measure, which support is the set of empirical samples, and the weights $\boldsymbol{a} \in \Sigma_n$ are uniform. Formally, we define the empirical measure of α as $\alpha_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{\boldsymbol{x}_i}$. It is known that α_n converges to α as the number of data n grows to infinity and we report to Section 7.2.1 for a longer discussion.

For instance in the supervised learning setting of classification, the data and their labels are sampled from an unknown joint measure $\mathcal{P}_{\mathcal{X}\times\mathtt{Y}}$. So its empirical counter-part is defined as

$$\alpha_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{(\boldsymbol{x}_i, \boldsymbol{y}_i)}. \tag{2.4}$$

where $\delta_{(\boldsymbol{x}_i, \boldsymbol{y}_i)}$ is a Dirac on the joint space of data and labels $\mathcal{X} \times \mathtt{Y}$. Hence approximating the real joint measure $\mathcal{P}_{\mathcal{X}\times\mathtt{Y}}$ is an important problem in order to get the best possible function with minimum prediction errors.

As probability measures are at the heart of many machine learning problems, finding a suitable distance to measure the discrepancy between them is a fundamental problem. Optimal transport has become a key tool to achieve this goal and is the topic of the next sections.

2.2 Kantorovich formulation of Optimal Transport

We start by giving an intuition about optimal transport, which measures the distance between two probability measures by taking into account the data space geometry. Intuitively it minimizes the displacement cost of a measure to another one. In this section, we introduce the formal optimal transport framework between arbitrary measures. We review the primal formulation, which is a linear program, and then we review its corresponding dual formulation. Afterwards, we study the metric properties of optimal transport. Next, we illustrate the particular case when data lie in one dimension, where a closed-form solution of optimal transport is available. Finally, we discuss the computational and memory complexities of solving the OT problem.

2.2.1 Primal and dual Optimal Transport formulations

We start by defining the primal formulation in the case of two continuous measures. Optimal transport measures a distance between two probability measures $(\alpha, \beta) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{Y})$ by considering a ground

cost $c : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}_+$ between the spaces $\mathcal{X}$ and $\mathcal{Y}$. It looks for a probabilistic coupling to move a measure to the other. A probabilistic coupling π is a joint probability measure with marginals α and β, formally:

Definition 3. *Let $\alpha \in \mathcal{M}_+^1(\mathcal{X}), \beta \in \mathcal{M}_+^1(\mathcal{Y})$ be two probability measures. A probabilistic coupling π is a joint probability such that its marginals are equal to α and β. We denote $\pi \in \mathcal{U}(\alpha, \beta)$ with*

$$\mathcal{U}(\alpha, \beta) = \left\{\pi \in \mathcal{M}_+^1(\mathcal{X}, \mathcal{Y}) : \boldsymbol{P}_{\mathcal{X}}\#\pi = \alpha, \boldsymbol{P}_{\mathcal{Y}}\#\pi = \beta\right\},$$

where $\boldsymbol{P}_{\mathcal{X}}\#\pi$ (resp. $\boldsymbol{P}_{\mathcal{Y}}\#\pi$) is the marginalization of π over $\mathcal{X}$ (resp. $\mathcal{Y}$). $\boldsymbol{P}_{\mathcal{X}}$ (resp. $\boldsymbol{P}_{\mathcal{Y}}$) denotes the projection over the space $\mathcal{X}$ (resp. $\mathcal{Y}$), and $\#$ denotes the pushforward operator.

The pushforward operator can be defined as follows.

Definition 4. *The pushforward operator for a continuous map $T : \mathcal{X} \mapsto \mathcal{Y}$ and for any measurable set $B \subset \mathcal{Y}$ is defined as: $\beta(B) = \alpha\big(\{\boldsymbol{x} \in \mathcal{X} : T(\boldsymbol{x}) \in B\}\big)$.*

Note that the notion of pushforward operator is important to formalize the image generation problem. We are now ready to state the Kantorovich formulation of optimal transport between continuous measures.

Definition 5. *Let $\alpha \in \mathcal{M}_+^1(\mathcal{X}), \beta \in \mathcal{M}_+^1(\mathcal{Y})$ be two probability measures and π a probabilistic coupling. The Kantorovich optimal transport formulation between probability distributions α and β is defined as*

$$\mathfrak{L}(\alpha, \beta, c) = \min_{\pi \in \mathcal{U}(\alpha,\beta)} \int_{\mathcal{X}\times\mathcal{Y}} c(\boldsymbol{x}, \boldsymbol{y}) d\pi(\boldsymbol{x}, \boldsymbol{y}). \tag{2.5}$$

It is called the Kantorovich problem [Kantorovich 1942] and we refer in this thesis to this problem as the continuous optimal transport formulation. It can be defined between measures of the same mass, but as our applications are machine learning tasks, measures are mostly probability measures. The ground cost represents the space geometry and influences the optimal connections between measures. Solving some machine learning problems requires to design a specific ground cost as it has been done in domain adaptation [Courty 2017b] or generative modelling [Genevay 2018]. To assure the existence of optimal transport plans, we make the following hypotheses: unless said otherwise, $\mathcal{X}$ and $\mathcal{Y}$ are subsets of $\mathbb{R}^d$, where d is the space dimension. Regarding the ground cost, it can only be a positive and semi-continuous function on $\mathcal{X} \times \mathcal{Y}$ but for the sake of simplicity, we suppose that it is a positive continuous function on the space $\mathcal{X} \times \mathcal{Y}$. We refer the reader to [Santambrogio 2015, Chapter 1] for a detailed review about the existence of optimal couplings for the Kantorovich problem. Note that the use of "min" instead of "inf" is justified in the formulation as the probabilistic coupling $\pi = \alpha \otimes \beta \in \mathcal{U}(\alpha, \beta)$ is always admissible. We can also define the optimal transport problem for discrete measures.

Remark 1. *Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ and $\boldsymbol{Y} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$ be two data n-tuples. Let $\alpha = \sum_{i=1}^n a_i \delta_{\boldsymbol{x}_i}$ and $\beta = \sum_{i=1}^n b_i \delta_{\boldsymbol{y}_i}$ be two* discrete *probability measures with weights* $\mathbf{a}$ *and* $\mathbf{b}$ *respectively. Let $C \in \mathbb{R}_+^{n\times n}$ be a ground cost matrix between the support of α and β,* i.e., *positions $\boldsymbol{X}$ and $\boldsymbol{Y}$, which elements are $C_{i,j} = c(\boldsymbol{x}_i, \boldsymbol{y}_j), \forall (i, j) \in [\![n]\!]^2$. The Kantorovich optimal transport discrete formulation between probability distributions α and β is defined as*

$$\mathfrak{L}(\alpha, \beta, c) = \mathfrak{L}(\mathbf{a}, \mathbf{b}, \boldsymbol{C}) = \min_{\Pi \in \boldsymbol{U}(\mathbf{a},\mathbf{b})} \langle \Pi, \boldsymbol{C} \rangle_F, \tag{2.6}$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenious product and where $\boldsymbol{U}(\mathbf{a}, \mathbf{b}) = \left\{\Pi \in \mathbb{R}_+^{n\times n} : \Pi \mathbf{1}_n = \boldsymbol{a}, \Pi^T \mathbf{1}_n = \boldsymbol{b}\right\}$.

When we do not deal with sum of Diracs but with a given ground cost $\boldsymbol{C}$ and random probability vectors $\mathbf{a}$ and $\mathbf{b}$, we can also define an optimal transport problem $\mathfrak{L}(\mathbf{a}, \mathbf{b}, \boldsymbol{C})$. The continuous and discrete

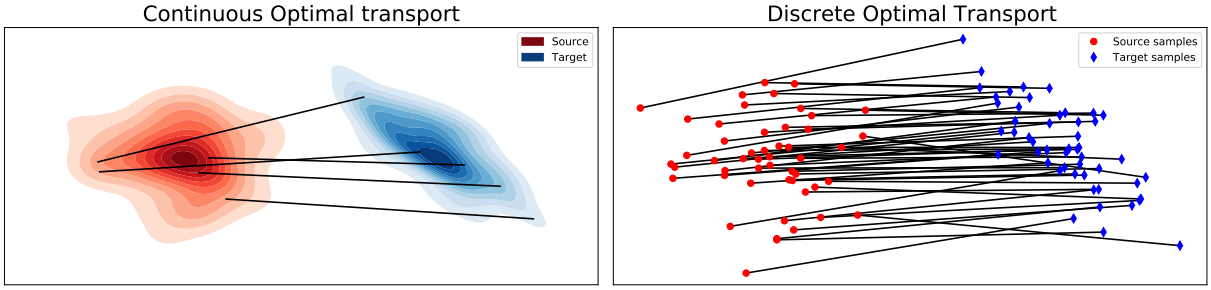


Figure 2.1: Optimal transport illustration between 2D measures. The left image represents the continuous optimal transport setting. The right image represents the discrete optimal transport setting. The black lines represent the optimal connections.

cases are illustrated in Figure 2.1. The last case of interest is the semidiscrete case, where one of the measures is discrete while the other is continuous. We report to the next section for a detailed discussion about this particular case.

When using optimal transport to compare probability measures with one measure being parametrized by a vector θ, *i.e.*, $\theta^* = \text{argmin}_\theta \, \mathfrak{L}(\alpha, \beta_\theta, c)$, the corresponding estimator is usually found in the literature under the name of *Minimum Kantorovich Estimator* [Bassetti 2006, Peyré 2019].

Kantorovich duality The Kantorovich problem (2.5) is the minimization of a linear problem with respect to the transport plan π. As such it is a constrained convex minimization problem. It can be naturally transformed in a dual problem called Kantorovich dual, which is a constrained concave maximization problem. We start this section by defining the *c-transform* which is at the heart of many duality theorems.

Definition 6 (c-transform)**.** *Let $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$, $\mathcal{X}, \mathcal{Y}$ be subsets of $\mathbb{R}^d$ and $f : \mathcal{X} \to \overline{\mathbb{R}}$ be a function. We define its* c-transform *as the function $f^c : \mathcal{Y} \to \overline{\mathbb{R}}$:*

$$f^c(\boldsymbol{y}) = \inf_{x \in \mathcal{X}} c(\boldsymbol{x}, \boldsymbol{y}) - f(\boldsymbol{x}) \tag{2.7}$$

and the $\bar{c}$-transform *of a function $g : \mathcal{Y} \to \overline{\mathbb{R}}$ as the function $g^{\bar{c}} : \mathcal{X} \to \overline{\mathbb{R}}$:*

$$g^{\bar{c}}(\boldsymbol{x}) = \inf_{y \in \mathcal{Y}} c(\boldsymbol{x}, \boldsymbol{y}) - g(\boldsymbol{y}). \tag{2.8}$$

Fonctions that can be written as f^c or $g^{\bar{c}}$ are called respectively c-concave *or* $\bar{c}$-concave *functions.*

One important property of the *c-transform* is that $\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}, f(\boldsymbol{x}) + f^c(\boldsymbol{y}) \leq c(\boldsymbol{x}, \boldsymbol{y})$. The c-transform is a generalization of Fenchel-Legendre conjugate, where the Fenchel-Legendre conjugate of a function G is defined as:

$$\forall \boldsymbol{y} \in \mathbb{R}^d, F(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^d} \langle \boldsymbol{x}, \boldsymbol{y} \rangle - G(\boldsymbol{x}).$$

It is a convex function for any function G. The following fundamental proposition makes a connections between the continuous primal and dual formulations.

Proposition 1 (Dual for arbitrary measures)**.** *The Kantorovich problem admits the dual*

$$\mathfrak{L}(\alpha, \beta, c) = \sup_{(f,g) \in \mathcal{R}(c)} \int_{\mathcal{X}} f(\boldsymbol{x}) d\alpha(\boldsymbol{x}) + \int_{\mathcal{Y}} g(\boldsymbol{y}) d\beta(\boldsymbol{y}), \tag{2.9}$$

where the set of admissible dual potentials is :

$$\mathcal{R}(c) = \{(f, g) \in \mathcal{C}(\mathcal{X}) \times \mathcal{C}(\mathcal{Y}) : \forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}, f(\boldsymbol{x}) + g(\boldsymbol{y}) \leq c(\boldsymbol{x}, \boldsymbol{y})\}.$$

The continuous functions f, g are called the Kantorovich potentials. The following remark highlights the dual formulation in case of discrete measures.

Remark 2 (Dual for discrete measures)**.** *The dual optimal transport problem in the discrete setting is*

$$\mathfrak{L}(\mathbf{a}, \mathbf{b}, C) = \max_{(\boldsymbol{f}, \boldsymbol{g}) \in \mathcal{R}(C)} \langle \boldsymbol{f}, \mathbf{a} \rangle + \langle \boldsymbol{g}, \mathbf{b} \rangle, \tag{2.10}$$

where the set of admissible dual potentials is :

$$\mathcal{R}(C) = \{(\boldsymbol{f}, \boldsymbol{g}) \in \mathbb{R}^n \times \mathbb{R}^n : \forall i, j \in [\![n]\!]^2, \boldsymbol{f}_i + \boldsymbol{g}_j \leq C_{i,j}\}.$$

The equality between discrete formulations holds thanks to the strong duality for linear programs [Bertsimas 1997] and the proof can be found in [Peyré 2019, Section 2.5].

Kantorovich - Rubinstein W_1 formulation Using the *c-transform*, one can make the dual problem dependant on only one dual potential. It is useful in the context of semidiscrete optimal transport, where one of the measure is discrete while the other is continuous. Formally, we have:

$$\begin{aligned}
\mathfrak{L}(\alpha, \beta, c) &= \sup_{(f,g) \in \mathcal{R}(c)} \int_{\mathcal{X}} f(\boldsymbol{x}) d\alpha(\boldsymbol{x}) + \int_{\mathcal{Y}} g(\boldsymbol{y}) d\beta(\boldsymbol{y}), \\
&= \sup_{f \in L^1(\alpha)} \int_{\mathcal{X}} f(\boldsymbol{x}) d\alpha(\boldsymbol{x}) + \int_{\mathcal{Y}} f^c(\boldsymbol{y}) d\beta(\boldsymbol{y}), \\
&= \sup_{g \in L^1(\beta)} \int_{\mathcal{X}} g^{\bar{c}}(\boldsymbol{x}) d\alpha(\boldsymbol{x}) + \int_{\mathcal{Y}} g(\boldsymbol{y}) d\beta(\boldsymbol{y}).
\end{aligned} \tag{2.11}$$

This formulation based on *c-transform* leads to another appealing formulation called the Kantorovich - Rubinstein W_1 dual formulation. When we transport two measures in the metric space $(\mathcal{X}, d)$ with the cost $c = d$, the dual formulation can be rewritten as a supremum over 1-Lipschitz dual potential. A full discussion can be found in [Peyré 2019, Chapter 6.]. We have then:

$$\mathfrak{L}(\alpha, \beta, d) = \sup_{g \in Lip_1(\mathcal{X})} \int_{\mathcal{X}} g(\boldsymbol{x}) \mathrm{d}\alpha(\boldsymbol{x}) - \int_{\mathcal{X}} g(\boldsymbol{y}) \mathrm{d}\beta(\boldsymbol{y}) \tag{2.12}$$

This formulation is at the heart of the Wasserstein GAN [Arjovsky 2017] used for generative modeling, where the ground cost is the usual Euclidean distance. Note that the cost appears in the Lipschitz property. After defining the primal and dual formulations, we state the optimal transport metric properties and its assumptions.

2.2.2 Metric Properties

Optimal transport is a metric between measures when certain properties are satisfied by the ground cost. It can be seen as a canonical method to lift a ground distance between points to a distance between measures. We first recall the metric definition.

Definition 7 (Metric definition)**.** *A metric on a space $\mathcal{Z}$ is a function $f : \mathcal{Z} \times \mathcal{Z} \mapsto \mathbb{R}_+$ which verifies the following axioms $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{Z}$:*

i $d(\boldsymbol{x}, \boldsymbol{y}) = 0 \iff \boldsymbol{x} = \boldsymbol{y}$ *(separability axiom)*

ii $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ *(symmetry)*

iii $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ *(triangle inequality)*

When the metric space is a space of measures, the elements $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ are measures and the axioms remain the same. We now state under what conditions the optimal transport cost is a metric.

Theorem 2.2.1. *We assume $\mathcal{X} = \mathcal{Y}$ and that it is equipped with a distance denoted d. Let $p \in [1, +\infty[$, then for the ground cost $c = d^p$, the optimal transport cost*

$$W_p(\alpha, \beta, d) = \left(\mathfrak{L}(\alpha, \beta, c^p)\right)^{1/p} \tag{2.13}$$

is indeed a metric, called the p-Wasserstein distance*, between probability measures α and β.*

In addition to these metric properties, some statistical properties are discussed in Section 7.2. The metric properties of optimal transport make it a good candidate for comparing probability measures in ML. After defining the formalism and the optimal transport metric properties, we now give a specific case where the optimal transport has a closed-form solution.

2.2.3 1D data closed-form

The optimal transport problem does not have a closed-form solution in general. However when data lie on the real line $\mathbb{R}$, a closed-form solution can be expressed for general probability measures on $\mathbb{R}$. The 1D closed-form is well known in the OT literature and is at the heart of an OT variant called the *sliced Wasserstein distance* [Rabin 2012, Bonnotte 2013], which we detail in the next paragraph. The *closed-form solution* of linear optimal transport is given in the next result and discussed in more details in [Santambrogio 2015, Theorem 2.9] and [Peyré 2019, Remark 2.30].

Theorem 2.2.2 (Closed-form expression on the real-line)**.** *Assume that $\mathcal{X} = \mathbb{R}$, $\alpha, \beta \in \mathcal{M}_+^1(\mathbb{R})$. Let F_α be the cumulative distribution function:*

$$\forall t \in \mathbb{R},\ F_\alpha(t) = \alpha(]-\infty, t]) \tag{2.14}$$

and F_α^{-1} its pseudo inverse, namely:

$$\forall x \in [0,1],\ F_\alpha^{-1}(x) = \inf\{t \in \mathbb{R} \mid F_\alpha(t) \geq x\}. \tag{2.15}$$

If $c(x, y) = h(y - x)$ where h is strictly convex then the Kantorovich problem (2.5) has a unique optimal transport plan given by $\pi^* = (F_\alpha^{-1} \times F_\beta^{-1})\#\mathcal{L}_{[0,1]}$ where $\mathcal{L}_{[0,1]}$ is the Lebesgue measure restricted to $[0, 1]$.

Moreover if α is atomless π^* is supported on the map $T(x) = F_\beta^{-1}(F_\alpha(x))$, i.e. $\pi^* = (id \times T)\#\alpha$. If h is only convex then the optimal transport plan π^* is still optimal but not necessarily unique.

Remark 3. *Suppose the probability measures α and β are discrete uniform measures with the same number of Diracs,* i.e., *$\alpha = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}, \beta = \frac{1}{n}\sum_{i=1}^n \delta_{y_i}$. Then if the measure supports $(x_i)_{i\in[\![n]\!]}$ and $(y_i)_{i\in[\![n]\!]}$ are respectively sorted on $\mathbb{R}$, the identity scaled by a uniform weight is an optimal transport plan. This case is illustrated in the Figure 2.2.*

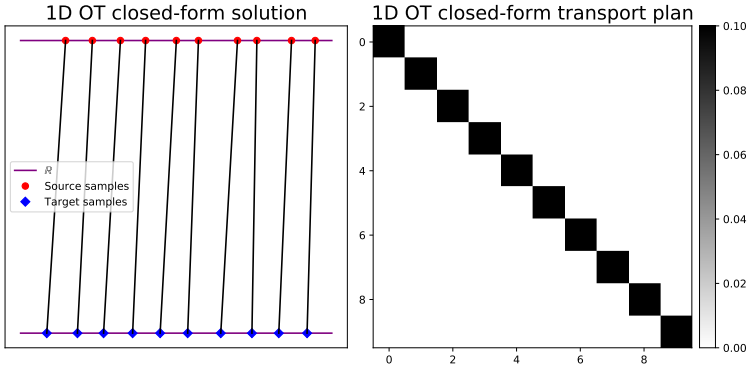


Figure 2.2: Illustration of OT between 1D uniform probability measures with sorted supports. The number of samples is set to 10. The optimal transport plan is the identity scaled by $\frac{1}{n}$.

In the case of non uniform weights and different number of atoms $\alpha = \sum_{i=1}^{n_1} a_i \delta_{x_i}, \beta = \sum_{i=1}^{n_2} b_i \delta_{y_i}$*, the 1D closed-form sends as much mass it can between* x_1 *and* y_1 *and then add the remaining mass to* y_2*. The operation is repeated until there is no mass left.*

As for two discrete uniform probability measures with the same number of atoms and sorted supports, the optimal transport plan is a scaled identity matrix, it can be solved by a sort algorithm in $\mathcal{O}(n \log(n))$ operations.

Sliced Wasserstein distance: taking advantage of the 1D closed-form This appealing property led to the creation of the so-called *sliced Wasserstein distance*. It consists in averaging the Wasserstein distance between projected data in 1D over all the directions in space. Formally it is defined as follows

$$SW_p\left(\mu_f, \mu_g, c\right)^p = \int_{S^{d-1}} W_p\left(\mu_{f_\theta}, \mu_{g_\theta}, c\right)^p d\theta, \tag{2.16}$$

where S^{d-1} is the sphere of dimension d and f_θ, g_θ the projected data in 1D. In practice, it is estimated with a Monte-Carlo method. It was first introduce in [Rabin 2012, Bonnotte 2013] and then used in the context of generative models [Kolouri 2016, Kolouri 2018, Kolouri 2019a, Liutkus 2019]. However in the general case, the computational complexity of OT is much higher than $\mathcal{O}(n \log(n))$ and is discussed in the next section.

2.2.4 Computational and memory complexities

For discrete probability measures, the Kantorovich formulation (2.6) is a linear program and can be solved using standard linear program solvers [Dantzig 1997]. One can then use the network simplex algorithm to solve the Kantorovich dual formulation. This algorithm has a computational complexity of $\mathcal{O}(n^3 \log(n))$ where n is the number of samples in the input measures. We report interested readers to [Peyré 2019, section 3.] for a more detailed discussion on how to solve the OT linear program. In a more simple case, we can make a connection between the Kantorovich formulation and the assignment problem.

Assignment problem We consider the special case of two discrete probability measures with the same number of atoms. For any permutation $\sigma \in \text{Perm}(n)$, we write the corresponding transport plan Π_σ which

is equal to $\left(\Pi_\sigma\right)_{i,j} = \begin{cases} \frac{1}{n} & \text{if } j = \sigma(i) \\ 0 & \text{otherwise} \end{cases}$. In this case, minimizing the assignment problem reads

$$\min_\sigma \langle \Pi_\sigma, C \rangle_F = \min_\sigma \frac{1}{n} \sum_i C_{i,\sigma_i}.$$

While this formulation is greater or equal than the Kantorovich problem, one can show that for uniform weights, the two formulations are equal as proved in [Peyré 2019, Proposition 2.1]. The assignment problem can be solved using the *Auction algorithm* in $\mathcal{O}(n^3)$.

Memory complexity The discrete primal formulation of optimal transport requires to store the ground cost $\boldsymbol{C}$ and the optimal transport plan. Thus its memory complexity is of order $\mathcal{O}(n^2)$. Regarding the dual formulation, it could be computed by storing the data in memory to compute the constraints, as well as the dual potentials and the probability vectors. Thus its memory complexity is of order $\mathcal{O}(nd)$ which is smaller than the primal formulation for big data scenario.

In this section, we defined the Kantorovich problem and reviewed its primal and dual formulations. We then discussed the different cases it defines a metric between probability distributions. Finally, we provided a closed-form solution when data lie in 1D and discussed its computational and memory complexities. In the next section, we regularize the Kantorovich problem with an entropic regularization.

2.3 Entropic regularization: formulation and practical interest

In this section, we discuss the introduction of the entropic-regularized optimal transport. It is defined by adding an entropic regularization penalty on the coupling to the original Kantorovich problem. This regularized loss has several important advantages with respect to original OT. For instance, it can be efficiently computed on GPUs using a 5 line algorithm. Furthermore, the resulting formulation is smooth with respect to input histogram weights and to the ground cost. Also, it can be differentiated using automatic differentiation. We present the entropic-regularized optimal transport definition, its equivalent dual formulation, the available solvers and the consequences on the optimal solutions. We finish by introducing a divergence based on entropic-regularized OT.

2.3.1 Definition and basic properties

The Kantorovich formulation has several interests but has also some downsides. The first problem is the non uniqueness of the transport plan which makes OT non differentiable *w.r.t.* the ground cost c, as discussed in Section 7.2.2. The second is its cubical computational complexity. Using an entropic regularization on the coupling helps to mitigate these two problems. This idea was first used in [Schrodinger 1931] and it was used for OT in [Wilson 1969]. It was then reintroduced recently in the OT literature in [Cuturi 2013] with the rise of GPUs. The entropic-regularized optimal transport problem between continuous probability measures with a ground cost c is defined as

$$\mathfrak{L}^\varepsilon(\alpha, \beta, c) = \min_{\pi \in \mathcal{U}(\alpha,\beta)} \int_{\mathcal{X}\times\mathcal{Y}} c(\boldsymbol{x}, \boldsymbol{y}) d\pi(\boldsymbol{x}, \boldsymbol{y}) + \varepsilon H(\pi|\xi), \tag{2.17}$$

$$\text{with } H(\pi|\xi) = \int_{\mathcal{X}\times\mathcal{Y}} \log\left(\frac{d\pi(\boldsymbol{x}, \boldsymbol{y})}{d\alpha(\boldsymbol{x})d\beta(\boldsymbol{y})}\right) d\pi(\boldsymbol{x}, \boldsymbol{y}) - \int_{\mathcal{X}\times\mathcal{Y}} d\pi(\boldsymbol{x}, \boldsymbol{y}) + \int_{\mathcal{X}\times\mathcal{Y}} d\alpha(\boldsymbol{x})d\beta(\boldsymbol{y}),$$

where $\xi = \alpha \otimes \beta$ and $\varepsilon \geq 0$ is the regularization coefficient. We denote it as the *entropic OT loss*.

Remark 4. *In the discrete measures case the problem becomes:*

$$\mathfrak{L}^{\varepsilon}(\mathbf{a}, \mathbf{b}, \boldsymbol{C}) = \min_{\Pi \in \boldsymbol{U}(\mathbf{a}, \mathbf{b})} \langle \Pi, \boldsymbol{C} \rangle + \varepsilon H(\Pi | \mathbf{a} \otimes \mathbf{b}),$$
$$\text{with } H(\Pi | \mathbf{a} \otimes \mathbf{b}) = \sum_{i,j} \Pi_{i,j} \log(\frac{\Pi_{i,j}}{\mathbf{a}_i \mathbf{b}_j}) - \Pi_{i,j} + \mathbf{a}_i \mathbf{b}_j. \tag{2.18}$$

Adding this penalty leads to several interesting outcomes. First of all, the problem is now strongly convex *w.r.t.* the coupling π and remains linear in the cost c. Indeed for discrete measures, if one computes the Hessian of the Kullback-Leibler divergence *w.r.t.* the transport plan, one gets $\frac{\partial^2 H}{\partial \Pi_{i,j}^2}(\Pi|\xi) = 1/\Pi_{i,j}$, which is a matrix with positive entries. Hence the problem is now strongly convex and as such, has a unique solution. Regarding the dual of the entropic-regularized OT problem we have the following results.

Proposition 2. *The dual of entropic-regularized OT reads*

$$\mathfrak{L}^{\varepsilon}(\alpha, \beta, c) = \sup_{(f,g) \in \mathcal{C}(\mathcal{X}) \times \mathcal{C}(\mathcal{Y})} \int_{\mathcal{X}} f(\boldsymbol{x}) d\alpha(\boldsymbol{x}) + \int_{\mathcal{Y}} g(\boldsymbol{y}) d\beta(\boldsymbol{y})$$
$$- \varepsilon \int_{\mathcal{X} \times \mathcal{Y}} \exp(\frac{-c(\boldsymbol{x}, \boldsymbol{y}) + f(\boldsymbol{x}) + g(\boldsymbol{y})}{\varepsilon}) d\alpha(\boldsymbol{x}) d\beta(\boldsymbol{y}), \tag{2.19}$$

Following the same logic as before, we can express the dual formulation between discrete probability measures.

Remark 5 (Dual for discrete measures)**.**

$$\mathfrak{L}^{\varepsilon}(\mathbf{a}, \mathbf{b}, \boldsymbol{C}) = \max_{(\boldsymbol{f}, \boldsymbol{g}) \in \mathbb{R}^n \times \mathbb{R}^n} \langle \boldsymbol{f}, \mathbf{a} \rangle + \langle \boldsymbol{g}, \mathbf{b} \rangle - \varepsilon \langle \exp(\frac{\boldsymbol{f}}{\varepsilon}), \exp(\frac{-\boldsymbol{C}}{\varepsilon}) \exp(\frac{\boldsymbol{g}}{\varepsilon}) \rangle. \tag{2.20}$$

The proof of this proposition can be found in [Peyré 2019, Proposition 4.4]. The equality between the primal and the dual formulations is proven by the general duality theorem (Karush–Kuhn–Tucker conditions and Lagrangian formulation, see [Bertsekas 1997, Chapter 5]). Furthermore, expressing the Lagrangian of the dual formulation leads to the expression of the optimal transport plan.

Proposition 3. *The optimal transport plan of the discrete entropic OT has the form:*

$$\Pi_{i,j} = e^{\frac{f_i}{\varepsilon}} e^{\frac{-C_{i,j}}{\varepsilon}} e^{\frac{g_j}{\varepsilon}} = u_i K_{i,j} v_j, \tag{2.21}$$

where $K = e^{-\frac{\boldsymbol{C}}{\varepsilon}}$.

See [Peyré 2019, Proposition 4.3] for more details. The form of the optimal solution can be further simplified. It also highlights a simple algorithm to solve the entropic-regularized OT problem, which is the topic of the next subsection.

2.3.2 Solving entropic Optimal Transport

In this section, we detail the possible solvers of entropic-regularized optimal transport. We first detail why the Sinkhorn algorithm can be used in this context and then we focus on stochastic solvers.

Sinkhorn algorithm The optimal transport plan of entropic-regularized OT has the matrix multiplication form $\Pi = \text{diag}(\boldsymbol{u}) \boldsymbol{K} \text{diag}(\boldsymbol{v})$, thanks to Proposition 3. Furthermore, the transport plan needs to verify the marginal constraints, thus it reads $\Pi \mathbf{1_n} = \text{diag}(\boldsymbol{u}) \boldsymbol{K} \text{diag}(\boldsymbol{v}) \mathbf{1}_n = \boldsymbol{u} \odot \boldsymbol{K} \boldsymbol{v} = \mathbf{a}$. The marginal constraint on $\mathbf{b}$ equivalently gives $\Pi^\top \mathbf{1_n} = \boldsymbol{v} \odot \boldsymbol{K}^\top \boldsymbol{u} \mathbf{1}_n = \mathbf{b}$. This problem is known in the

Algorithm 1 Sinkhorn Algorithm

Inputs : histograms $\mathbf{a}$ and $\mathbf{b}$, cost matrix $\boldsymbol{C}$, regularization coefficient ε
$\boldsymbol{K} = \exp(-\boldsymbol{C}/\varepsilon)$
Define at random $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$
while $(\boldsymbol{u}, \boldsymbol{v})$ did not converge **do**
    $\boldsymbol{u} = \mathbf{a} \oslash (\boldsymbol{K}\boldsymbol{v})$
    $\boldsymbol{v} = \mathbf{b} \oslash (\boldsymbol{K}^\top \boldsymbol{u})$
end while
return $\boldsymbol{u}, \boldsymbol{v}$

numerical analysis community as the matrix scaling problem and it can be solved using the Sinkhorn algorithm [Sinkhorn 1967]. The Sinkhorn algorithm is detailed in Algorithm 1 and it can be interpreted as a block coordinate ascent on the dual problem. A practical feature of the Sinkhorn algorithm is that it can be efficiently solved on GPUs as noted in [Cuturi 2013], allowing a fast parallel computation. Regarding its complexities, let $\eta = \frac{4\log(n)}{\varepsilon}$, then the Sinkhorn algorithm produces an optimal solution Π^* such that $\langle \mathbf{C}, \Pi^* \rangle_{\mathcal{F}} \leq \mathfrak{L}^\varepsilon(\mathbf{a}, \mathbf{b}, C) + \eta$ after $O(\|C\|_\infty^3 \log(n)\eta^{-3})$ iterations [Altschuler 2017]. A full discussion on the Sinkhorn algorithm with its analogies, convergence results and complexities can be found in [Peyré 2019, section 4.].

The Sinkhorn algorithm has seen many variants published in the recent years. As numerical stability appear for small regularization value ε, a stabilized version was developed in [Chizat 2018a] which is based on log-sum-exp iterations and is mathematically equivalent to the original version. Several published papers have focused on studying the Sinkhorn algorithm complexity [Altschuler 2017, Pham 2020]. To accelerate the computation of the solution, one can rely on a greedy strategy [Altschuler 2017, Abid 2018], or on reducing the problem by finding negligible components [Alaya 2019]. Other strategies such as Bregman projection operators [Thibault 2017] or gradient descent acceleration [Dvurechensky 2018] are possible. Regarding its use in machine learning, as the entropic formulation is a smooth formulation *w.r.t.* the cost or the input histograms, a Sinkhorn AutoDiff formulation can be developed as done in [Genevay 2018]. To speed up computations, they used a fixed number of Sinkhorn iterations. Finally, the entropic formulation can also be solved using stochastic algorithms.

Stochastic solvers for entropic-regularized OT The entropic OT formulation can be expressed as an expectation over data, allowing the use of stochastic algorithms to solve the OT problem. The idea is to express the entropic-regularized OT formulation using the *c-transform*. Suppose $\varepsilon > 0$, then using the *c-transform*, the dual OT program reads

$$\sup_{f,g \in C(\mathbb{R}^d) \times C(\mathbb{R}^d)} \mathop{\mathbb{E}}_{\mathbf{x}\sim\alpha, \mathbf{y}\sim\beta} [F_\varepsilon(f(\mathbf{x}), g(\mathbf{y}))], \tag{s-D}$$

$$\sup_{g \in C(\mathbb{R}^d)} \mathop{\mathbb{E}}_{\mathbf{x}\sim\alpha} [G_\varepsilon(\mathbf{x}, g)], \tag{s-SD}$$

where $F_\varepsilon(f(\mathbf{x}), g(\mathbf{y})) = f(\mathbf{x}) + g(\mathbf{y}) - \varepsilon e^{\frac{f(\mathbf{x})+g(\mathbf{y})-c(\mathbf{x},\mathbf{y})}{\varepsilon}}$ and $G_\varepsilon(\mathbf{x}, g) = \int_{\mathcal{Y}} g(\mathbf{y}) \mathrm{d}\beta(\mathbf{y}) - \varepsilon \log(\int e^{\frac{1}{\varepsilon}(g(\mathbf{y})-c(\mathbf{x},\mathbf{y}))} \mathrm{d}\beta(\mathbf{y})) - \varepsilon$. Those smoothed problems are called the smooth dual (s-D) and the smooth semi-dual (s-SD) problems. The two problems are unconstrained maximization problems of an expectation and the idea is to use Stochastic gradients descent [Robbins 1951a] (SGD) or reduced variance stochastic (RVS) algorithms [Schmidt 2017a, Defazio 2014, Johnson 2013, Nguyen 2017] to compute their

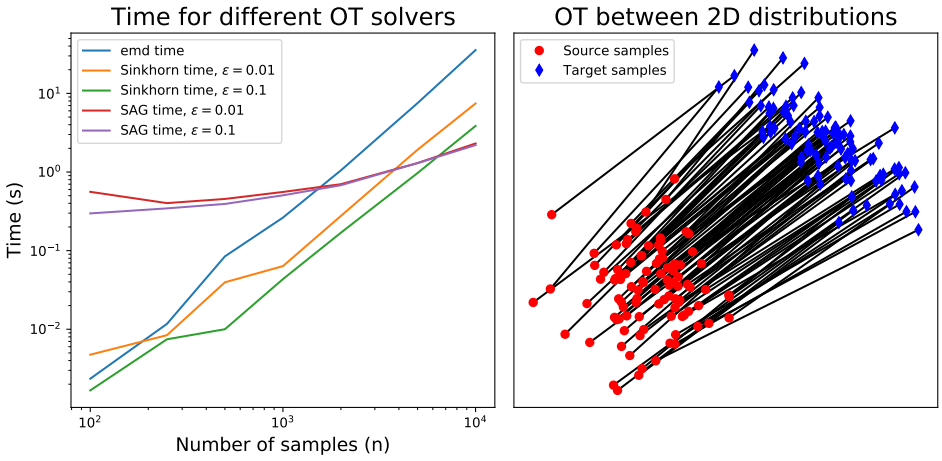


Figure 2.3: Time comparison of optimal transport variants between 2D measures with growing support. The figure shows that the entropic-regularized OT with the Sinkhorn algorithm or the stochastic algorithms are faster to compute than original OT for a large number of samples. We used the solvers from the POT library [Flamary 2021].

solution. These equations were first used in [Genevay 2016] and the authors compared the different stochastic algorithms to compute the solutions. We first discuss the discrete case.

When both measures $\alpha = \sum_{i=1}^{n} a_i \delta_{\mathbf{x}_i}, \beta = \sum_{j=1}^{m} b_j \delta_{\mathbf{y}_j}$ are discrete, the equations become:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta} \in \mathbb{R}^n \times \mathbb{R}^m} \sum_{i=1,j=1}^{n,m} F_\varepsilon(\alpha_i, \beta_j) a_i b_j, \tag{s-Ddis}$$

$$\max_{\boldsymbol{\beta} \in \mathbb{R}^m} \sum_{i=1}^{n} G_\varepsilon(\mathbf{x}_i, \boldsymbol{\beta}) a_i. \tag{s-SDdis}$$

The stochastic gradients computation complexity of (s-SDdis) is $\mathcal{O}(n)$. RVS algorithms can be used to solve the (s-SDdis) problem and have a converge rate of $\mathcal{O}(t^{-1})$ but at the price of a full gradient computation [Johnson 2013] or the storage of past stochastic gradients [Schmidt 2017a, Defazio 2014]. When the extra cost of RSV algorithms can not be handled, one can rely on SGD with a convergence rate of $\mathcal{O}(t^{-\frac{1}{2}})$. A time comparison between the computation of original OT and entropic-regularized OT with the Sinkhorn algorithm or stochastic algorithms can be found in Figure 2.3. It shows that the stochastic formulation is slower than the Sinkhorn algorithm or the network simplex algorithm for a small number of data but faster in the Big Data regime. These poor performances in the small data regime might come from a non efficient Python loop in the stochastic solvers from the POT library [Flamary 2021], which would decrease the time performances. Regarding (s-Ddis), its stochastic gradient computation cost is of order $\mathcal{O}(m^2)$, where m is the minibatch size. It has been solved using directly stochastic gradients between small minibatches in [Seguy 2018]. Due to this small gradient computation cost, it was proven to converge faster than SGD on the (s-SDdis) formulation [Seguy 2018].

Now suppose that one of the measures is continuous, another particular advantage of the smooth semi-dual formulation (s-SDdis) is that it can tackle the semi-discrete OT case, which is the formalism of generative models. Unfortunately RVS algorithms can not be used anymore as one of the measures is continuous and we need to rely on SGD. Thus this approach is not scalable in the Big Data regime.

Finally, the continuous OT case has also been studied in several works. [Genevay 2016] represented the dual variables with kernel expansions, [Seguy 2018] parametrized the dual variables with a neural network.

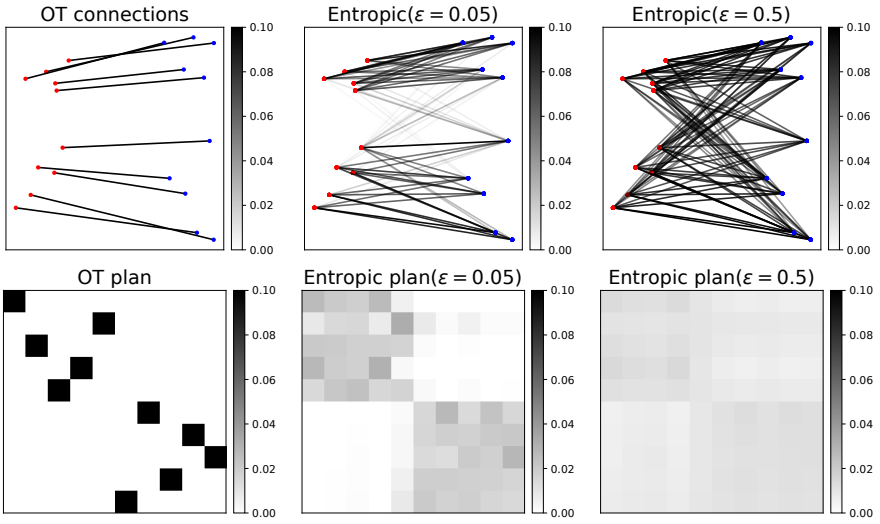


Figure 2.4: Optimal transport variants between 2D measures. For the entropic-regularized OT variants, the ε coefficients are set to 0.05 and 0.5.

Stochastic solvers for exact OT In the context of unregularized OT, the W_1 Kantorovich-Rubinstein formulation could be used as done in [Arjovsky 2017]. The challenge of the W_1 formulation is to ensure that the potential is indeed 1-Lipschitz. The solutions are either based on weight clipping [Arjovsky 2017] or making the gradient norm less or equal to 1 [Gulrajani 2017].

We have shown many desirable complexities of entropic-regularized OT. However, it is not the panacea and we develop its limits in the next section.

2.3.3 Entropic bias: solution far from original Optimal Transport solution

Adding the entropic regularization leads to several advantages, but it also changes the original problem and we discuss the consequences on the transported samples and the metric properties in this section.

The optimal transport plan of the entropic-regularized OT problem is not the plan with minimum transport cost anymore. Thus, using the entropic regularization breaks the separability axiom of distances and $\mathfrak{L}^\varepsilon(\alpha, \alpha, c) \neq 0$. This is defined in the literature as the entropic bias [Feydy 2019]. The greater the regularization coefficient ε is, the more uniform the transport plan gets as illustrated in Figure 2.4. Thus the optimal barycentring mapping of the entropic-regularized OT is no longer the target measure but a shrunk measure, whose support is *smaller* than the one of the target measure. The size of the barycentric mapping support depends on the coefficient ε. We now formally state the limit properties of entropic-regularized OT depending on the regularization parameter ε with the following proposition.

Proposition 4. *When the regularization coefficient ε converges to 0, the entropic-regularized optimal transport problem converges to the original optimal transport problem. Furthermore, when the regularization coefficient converges to ∞, the transport plan converges to the product measure $\mathbf{a} \otimes \mathbf{b}$,* i.e.,

$$\varepsilon \to 0 \implies \mathfrak{L}^\varepsilon(\mathbf{a}, \mathbf{b}, C) \to \mathfrak{L}(\mathbf{a}, \mathbf{b}, C), \tag{2.22}$$

$$\varepsilon \to \infty \implies \Pi^\varepsilon \to \mathbf{a} \otimes \mathbf{b}. \tag{2.23}$$

The proof of the above proposition can be found in [Peyré 2019, Proposition 4.1] when the OT problem is regularized with the entropy function. The case for the Kullback-Leibler regularization follows the same

steps. To mitigate this issue and recover the target solution, a debiased loss has been introduced in the literature and it is the topic of the next section.

Different OT regularization The regularization of optimal transport has several flavors. In this section, we detail some other useful variants but which are out of the scope of this manuscript. In domain adaptation, some well-chosen regularizations of optimal transport were designed to ensure a special behaviour on connections and the transport plan. A $l_p l_1$ norm regularization was proposed in [Courty 2014] in order to concentrate the transport information on elements of the same class. In [Courty 2017a], the authors introduced a group-lasso regularization to promote the mass transfer from source data with the same labels to a given target data. They also introduced a Laplacian regularization which aims at preserving the data structure – approximated by a graph – during transport. Those problems were solved using a general conditional gradient. A squared 2-norm and a group-lasso regularizations of optimal transport were studied in [Blondel 2018], leading to sparse and group-sparse transportation plans contrary to the entropic-regularized optimal transport plan.

2.3.4 Sinkhorn divergence, a debiased entropic Optimal Transport approach

Despite the appealing advantages of the entropic-regularized OT, the associated entropic bias might lead to poor performances in practice [Feydy 2019]. To mitigate this issue, a debiased loss has been introduced, called the Sinkhorn Divergence.

The idea is simple, if the entropic-regularized OT $\mathfrak{L}^\varepsilon(\alpha, \beta, c)$ is not zero for $\varepsilon > 0$ and $\alpha = \beta$, then we just need to remove the non-zero bias for both measures α and β. The Sinkhorn Divergence is defined as follows:

$$S^\varepsilon(\alpha, \beta, c) = \mathfrak{L}^\varepsilon(\alpha, \beta, c) - \frac{1}{2}(\mathfrak{L}^\varepsilon(\alpha, \alpha, c) + \mathfrak{L}^\varepsilon(\beta, \beta, c)). \tag{2.24}$$

The discrete case is straightforward by replacing the entropic-regularized OT terms by their discrete counterparts. It is clear that we recover $S^\varepsilon(\alpha, \alpha, c) = 0$ by construction of the loss. It can still be computed with the same order of computational complexity as the entropic-regularized OT, by using the Sinkhorn algorithm on each term for instance. The Sinkhorn divergence has been proven to interpolate between OT and Maximum Mean Discrepancy (MMD). MMD are integral probability metrics which consist of the integration of a positive kernel k over a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{X}$ [Gretton 2012] and they are defined as

$$\|\alpha - \beta\|_k^2 = \mathbb{E}_{\alpha\otimes\alpha}\left[k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right] + \mathbb{E}_{\beta\otimes\beta}\left[k\left(\boldsymbol{y}, \boldsymbol{y}'\right)\right] - 2\mathbb{E}_{\alpha\otimes\beta}[k(\boldsymbol{x}, \boldsymbol{y})]. \tag{2.25}$$

We have the following proposition regarding Sinkhorn divergence's interpolation behaviour.

Proposition 5. *Let α and β be probability measures and c a ground cost.*

$$\lim_{\varepsilon\to 0} S^\varepsilon(\alpha, \beta, c) = \mathfrak{L}(\alpha, \beta, c) \text{ and } \lim_{\varepsilon\to+\infty} S^\varepsilon(\alpha, \beta, c) = \frac{1}{2}\|\alpha - \beta\|_{-c}^2. \tag{2.26}$$

The proof of this proposition can be found in [Ramdas 2017]. The entropic-regularized OT is a positive cost as the sum of two positive functions. But this might not be the case for the Sinkhorn Divergence as we remove two positive terms. Fortunately, its has been proven in [Feydy 2019, Theorem 1] that the Sinkhorn Divergence is a convex, smooth, positive loss function that metrizes the convergence in law. We state their theorem.

Theorem 2.3.1 (Theorem 1 [Feydy 2019].). *Let $\mathcal{X}$ be a compact metric space with a Lipschitz cost function $c(\boldsymbol{x}, \boldsymbol{y})$ that induces, for $\varepsilon > 0$, a positive universal kernel $k_\varepsilon(\boldsymbol{x}, \boldsymbol{y}) = \exp(-c(\boldsymbol{x}, \boldsymbol{y})/\varepsilon)$. Then, S^ε defines a symmetric positive definite, smooth loss function that is convex in each of its input variables. It also metrizes the convergence in law: for all probability Radon measures α and $\beta \in \mathcal{M}(\mathcal{X})_1^+$.*

$$0 = S^\varepsilon(\alpha, \alpha, c) \leq S^\varepsilon(\alpha, \beta, c), \tag{2.27}$$

$$\alpha = \beta \iff S^\varepsilon(\alpha, \beta, c) = 0, \tag{2.28}$$

$$\alpha_n \rightharpoonup \alpha \iff S^\varepsilon(\alpha_n, \alpha, c) \rightharpoonup 0. \tag{2.29}$$

This theorem ensures that the unique minimizer of $S^\varepsilon(\alpha, \beta, c) = 0$ is $\alpha = \beta$ on contrary to entropic-regularized OT. It is a smoothed function and it can be approximated with an AutoDiff strategy. The Sinkhorn divergence has been used successfully in the context of generative models [Genevay 2018] or gradient flows [Feydy 2019] for instance. An important aspect of the Sinkhorn Divergence is its sample complexity which interpolates between OT and MMD, it is a concept that we detail in Section 7.2.1.

2.4 Optimal Transport extensions

In this section we review several other variants of optimal transport. The first variant is called the Gromov-Wasserstein distance and can compare measures which lie in different metric spaces. We present it as we investigate the theoretical justifications of the Gromov-Wasserstein distance computed between minibatches as presented in Chapter 8. The second variant is the unbalanced optimal transport. UOT allows to compare probability distributions which do not share the same mass by relaxing the marginal constraints on the coupling. This formulation is also robust to data outliers, *i.e.,* it is not affected negatively by these outliers. The Unbalanced OT variant is introduced because it mitigates some downsides of optimal transport computed between minibatches as detailed in Chapter 9. There exist other variants of interests such as the multi-marginal optimal transport [Peyré 2019] or factored coupling OT [Forrow 2019], but we do not present them as they are out of the scope of this manuscript.

2.4.1 Comparing measures lying in different spaces with Gromov-Wasserstein

Classical OT distances are defined when a relevant ground cost between the measures is defined. Unfortunately, the case where the source and the target data live in different metric spaces, such as $\mathbb{R}^2$ and $\mathbb{R}^3$, does not fall into this framework. In this section, we review the Gromov-Wasserstein (GW) distance [Mémoli 2011], which can compare measures lying in different metric spaces. It has been investigated in the past few years and relies on comparing intra-domain distances. The general setting corresponds to computing couplings between metric measure spaces $(\mathcal{X}, c_\mathcal{X}, \alpha)$ and $(\mathcal{Y}, c_\mathcal{Y}, \beta)$, where $(c_\mathcal{X}, c_\mathcal{Y})$ are distances, while α and β are measures on their respective spaces. One defines the Gromov-Wasserstein distance as:

$$\mathcal{GW}_p((\alpha, c_\mathcal{X}), (\beta, c_\mathcal{Y})) = \left(\min_{\pi \in \mathcal{U}(\alpha,\beta)} \int_{\mathcal{X}^2 \times \mathcal{Y}^2} |c_\mathcal{X}(\boldsymbol{x}, \boldsymbol{x}') - c_\mathcal{Y}(\boldsymbol{y}, \boldsymbol{y}')|^p d\pi(\boldsymbol{x}, \boldsymbol{y}) d\pi(\boldsymbol{x}', \boldsymbol{y}') \right)^{1/p}. \tag{2.30}$$

In the discrete case the ground cost takes the form of a tensor $C_{i,j,k,l} = |C_{i,k} - C'_{j,l}|$, where C is the ground cost between the source data and C' the ground cost between the target data. We can interpret the $\mathcal{GW}$ distance as follows: the coupling tends to associate samples that share common relations with

the other samples in their respective metric spaces. The Gromov-Wasserstein distance is not convex in the transport plan, but it is linear in the tensor cost.

However, the Gromov-Wasserstein distance is challenging to compute, as a non convex quadratic program which is NP hard must be solved [Peyré 2016]. To address this issue from another perspective, one can realign the spaces $\mathcal{X}$ and $\mathcal{Y}$ using a global transformation before using the classical Wasserstein distance [Alvarez-Melis 2019]. In this paper, the authors cast the problem as a joint optimization over transport couplings and transformations chosen from a flexible class of invariances. Furthermore, an entropic variant of Gromov-Wasserstein [Peyré 2016] has been proposed to reduce its computational complexity following the same framework defined in Section 2.3. A Sinkhorn-like algorithm can also be adapted to the entropic-regularized Gromov-Wasserstein problem. A Sinkhorn Gromov-Wasserstein divergence has been introduced in [Bunne 2019], but its positivity remains an open question. More recently, a sliced variant has been introduced in [Vayer 2019b] in the case of the specific squared Euclidean ground cost. Finally, a tree variant was proposed to accelerate the computation of Gromov-Wasserstein in [Le 2019]. We now review the metric properties of the Gromov-Wasserstein distance.

Metric properties Gromov-Wasserstein $\mathcal{GW}$ defines a distance between different metric measure spaces under some hypotheses on the measure spaces. Let us first define isometries:

Definition 8 (Isometry)**.** *Let $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$ be two metric spaces. An isometry is a sujective map $\varphi : \mathcal{X} \to \mathcal{Y}$ that preserves the distances:*

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}, d_{\mathcal{Y}}(\varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}')) = d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}'). \tag{2.31}$$

Thanks to the metric properties of $d_{\mathcal{X}}$, an isometry is bijective. Indeed for $\varphi(\boldsymbol{x}) = \varphi(\boldsymbol{x}')$ we have $d_{\mathcal{Y}}(\varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}')) = 0 = d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{x}')$ and hence $\boldsymbol{x} = \boldsymbol{x}'$. By construction, the inverse map φ^{-1} is an isometry as well. The isometry concept can be extended between measure spaces.

Definition 9 (Strong isomorphism)**.** *Let $(\mathbb{R}^d, d_{\mathbb{R}^d}, \alpha)$, $(\mathbb{R}^q, d_{\mathbb{R}^q}, \beta)$ be two measure spaces. We say that $(\mathbb{R}^d, d_{\mathbb{R}^d}, \alpha)$ is* strongly isomorphic *to $(\mathbb{R}^q, d_{\mathbb{R}^q}, \beta)$ if there exists a bijection $\varphi : supp(\alpha) \to supp(\beta)$ such that:*

i φ is an isometry, i.e. $d_{\mathbb{R}^q}(\varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}')) = d_{\mathbb{R}^d}(\boldsymbol{x}, \boldsymbol{x}')$ for $\boldsymbol{x}, \boldsymbol{x}' \in supp(\alpha)^2$.

ii φ pushes α forward to β, i.e. $\varphi \# \alpha = \beta$.

However, the notion of strong isomophism relies on the notion of metric. This, in turn, means that metrics should be used in the GW distance definition. That is why we introduce the concept of weak isomorphism in order to consider other ground costs.

Definition 10 (Weak isomorphism)**.** *Let $(\mathbb{R}^d, c_{\mathbb{R}^d}, \alpha)$, $(\mathbb{R}^q, c_{\mathbb{R}^q}, \beta)$ be two measure spaces. We say that $(\mathbb{R}^d, c_{\mathbb{R}^d}, \alpha)$ is* weakly isomorphic *to $(\mathbb{R}^q, c_{\mathbb{R}^q}, \beta)$ if there exists $(\mathbb{R}^{q'}, c_{\mathbb{R}^{q'}}, \zeta)$, with $supp(\zeta) = \mathbb{R}^{q'}$ and maps $\varphi_0 : \mathbb{R}^{q'} \to \mathbb{R}^d$,$\varphi_1 : \mathbb{R}^{q'} \to \mathbb{R}^q$ such that:*

i $c_{\mathbb{R}^{q'}}(\boldsymbol{z}, \boldsymbol{z}') = c_{\mathbb{R}^d}(\varphi_0(\boldsymbol{x}), \varphi_0(\boldsymbol{x}')) = c_{\mathbb{R}^q}(\varphi_1(\boldsymbol{x}), \varphi_1(\boldsymbol{x}'))$ for $\boldsymbol{z}, \boldsymbol{z}' \in (\mathbb{R}^{q'})^2$.

ii $\varphi_0 \# \zeta = \alpha$ and $\varphi_1 \# \zeta = \beta$.

The isometry notion allows us to compare two different metric spaces. We are now ready to enumerate the Gromov-Wasserstein metric properties.

Theorem 2.4.1 (Metric properties of GW). *Let $(\mathbb{R}^d, d_{\mathbb{R}^d}, \alpha)$, $(\mathbb{R}^q, d_{\mathbb{R}^q}, \beta)$ be two measure spaces. The Gromov-Wasserstein is symmetric, positive and satisfies the triangle inequality. More precisely, let $(\mathbb{R}^d, d_{\mathbb{R}^d}, \alpha)$, $(\mathbb{R}^q, d_{\mathbb{R}^q}, \beta)$, $(\mathbb{R}^{q'}, d_{\mathbb{R}^{q'}}, \zeta)$ be measure spaces, we have:*

$$\mathcal{GW}_p((\alpha, d_{\mathbb{R}^d}), (\beta, d_{\mathbb{R}^q})) \leq \mathcal{GW}_p((\alpha, d_{\mathbb{R}^d}), (\zeta, d_{\mathbb{R}^{q'}})) + \mathcal{GW}_p((\zeta, d_{\mathbb{R}^{q'}}), (\beta, d_{\mathbb{R}^q})). \tag{2.32}$$

The separability axiom is verified under the following condition:

i $\mathcal{GW}_p((\alpha, d_{\mathbb{R}^d}), (\beta, d_{\mathbb{R}^q})) = 0$ *if and only if* $(\mathbb{R}^d, d_{\mathbb{R}^d}, \alpha)$ *and* $(\mathbb{R}^q, d_{\mathbb{R}^q}, \beta)$ *are strongly isomorphic.*

ii $\mathcal{GW}_p((\alpha, d_{\mathbb{R}^d}), (\beta, d_{\mathbb{R}^q})) = 0$ *if and only if* $(\mathbb{R}^d, d_{\mathbb{R}^d}, \alpha)$ *and* $(\mathbb{R}^q, d_{\mathbb{R}^q}, \beta)$ *are weakly isomorphic.*

iii More generally, for any $p \geq 1$, $\mathcal{GW}_p((\alpha, d^p_{\mathbb{R}^d}), (\beta, d^p_{\mathbb{R}^q})) = 0$ *if and only if* $(\mathbb{R}^d, d_{\mathbb{R}^d}, \alpha)$ *and* $(\mathbb{R}^q, d_{\mathbb{R}^q}, \beta)$ *are strongly isomorphic.*

The positivity is clear by the $\mathcal{GW}$ definition. The symmetry and triangle inequality proof can be found in [Chowdhury 2019, Theorem 16]. For (ii), the proof can be found in [Sturm 2012, Lemma 1.10]. For (iii), the proof is available in [Chowdhury 2019, Theorem 18]. We now state the distance theorem for Gromov-Wasserstein.

Theorem 2.4.2 (*GW* is a distance). *$\mathcal{GW}_p$ is a distance on $\left(\mathbb{R}^d, \|.\|_d, \alpha \in P_p(\mathbb{R}^d); d \in N\right)$ quotiented by the strong isomorphisms, where $\|.\|_d$ is a norm on $\mathbb{R}^d$.*

The Gromov-Wasserstein distance allows to compare probability measures which lie in non comparable metric spaces. In the next section, we detail an OT variant which can compare measures with different masses.

2.4.2 Unbalanced Optimal Transport: relaxation of mass constraints

Due the mass conservation constraint, optimal transport can not compare measures with different mass. Unbalanced OT is a generalization of 'classical' OT that relaxes the conservation of mass constraints by allowing the system to either transport or create and destroy mass. In this section, we discuss this loss and its advantages over the original optimal transport formulation.

The unbalanced considered optimal transport formulation is built upon [Liero 2017, Frogner 2015] which replaces the 'hard' marginal constraints of OT by 'soft' penalties using Csiszàr divergences. A dynamical formulation has been investigated in [Chizat 2017, Chizat 2018b] but is out of the scope of this manuscript. There exists other extensions of the static formulations of OT. A famous one is partial OT which consists in transporting a fixed budget of mass [Figalli 2010a] or to move mass in and out of the system at a fixed cost [Figalli 2010b]. Another line of work proposes to optimize over various sets of Lipschitz functions [Hanin 1992, Piccoli 2014, Schmitzer 2017]. One can also replace Csiszàr divergences by integral probability metrics [Nath 2020].

To define this unbalanced variant, consider a convex, positive, lower-semicontinuous function ϕ such that $\phi(1) = 0$. Define $\phi'_\infty = \lim_{x \to +\infty} \phi(\boldsymbol{x})/\boldsymbol{x}$ that we suppose strictly positive. Csiszàr divergences D_ϕ are measures of discrepancy that compare pointwise ratios of mass using a penalty ϕ and are defined as $D_\phi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{y}_i \neq 0} \boldsymbol{y}_i \phi\left(\frac{\boldsymbol{x}_i}{\boldsymbol{y}_i}\right) + \phi'_\infty \sum_{\boldsymbol{y}_i = 0} \boldsymbol{x}_i$. Total Variation and Kullback-Leibler divergences are particular instances of such divergences. Consider two positive measures $\alpha, \beta \in \mathcal{M}_+(\mathcal{X})$. The Unbalanced

Algorithm 2 Sinkhorn Algorithm

Inputs : histograms $\mathbf{a}$ and $\mathbf{b}$, cost matrix $\boldsymbol{C}$, regularization coefficient ε, penalisation coefficient τ
$\boldsymbol{K} = \exp(-\boldsymbol{C}/\varepsilon)$
Define at random $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$
while $(\boldsymbol{u}, \boldsymbol{v})$ did not converge **do**
    $\boldsymbol{u} = (\mathbf{a} \oslash (\boldsymbol{K}\boldsymbol{v}))^{\frac{\tau}{\tau+\varepsilon}}$
    $\boldsymbol{v} = (\mathbf{b} \oslash (\boldsymbol{K}^\top \boldsymbol{u}))^{\frac{\tau}{\tau+\varepsilon}}$
end while
return $\boldsymbol{u}, \boldsymbol{v}$

Optimal Transport (UOT) program between measures and cost c is defined as

$$\mathrm{OT}_\phi^\tau(\alpha, \beta, c) = \min_{\pi \in \mathcal{M}_+(\mathcal{X}^2)} \int c d\pi + \tau(D_\phi(\pi_1 \| \alpha) + D_\phi(\pi_2 \| \beta)), \tag{2.33}$$

where π is the transport plan, π_1 and π_2 the plan's marginals and τ the marginal penalization. Note that the marginals of π are no longer equal to (α, β) in general. Regarding the metric properties we have the following theorem.

Theorem 2.4.3. *Consider the square Euclidean ground cost $c(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2$ and the Csiszàr divergence $D_\phi = \mathtt{KL}$, then the Unbalanced OT cost $\mathrm{OT}_\phi^\tau(\alpha, \beta, c)^{\frac{1}{2}}$ is the Gaussian–Hellinger distance, which is a distance on $M_1^+(\mathbb{R}^d)$.*

Proof. The proof can be found in [Liero 2017]. □

Finally, note that an unbalanced variant of the Gromov-Wasserstein distance was proposed and studied in [Séjourné 2020]. In this paper, the authors defined a distance and an upper-bounding relaxation which is more tractable. Both of them allow the comparison of metric spaces equipped with arbitrary positive measures up to isometries.

Entropic regularization Straightforwardly, it is possible to define an entropic extension of unbalanced optimal transport. It is defined as

$$\mathrm{OT}_\phi^{\tau,\varepsilon}(\alpha, \beta, c) = \min_{\pi \in \mathcal{M}_+(\mathcal{X}^2)} \int c d\pi + \varepsilon \mathtt{KL}(\pi | \alpha \otimes \beta) + \tau(D_\phi(\pi_1 \| \alpha) + D_\phi(\pi_2 \| \beta)). \tag{2.34}$$

The formulation considered is computable via a generalized Sinkhorn algorithm [Chizat 2018a, Séjourné 2019] which is proven to converge and detailed in Algorithm 2. A recent analysis of the Sinkhorn algorithm for $D_\phi = \mathtt{KL}$ proved a convergence rate of $\tilde{O}(n^2/\varepsilon)$ [Pham 2020]. For these reasons, we rely in this manuscript on the unbalanced optimal transport with Kullback-Leibler divergence as our function ϕ and the entropic regularization. We illustrate the effect on the transport plans in Figure 2.5. We can see that when the parameter τ is high we indeed recover entropic-regularized OT, but when this parameter is low, the intensity of the connections declines.

Recently, an Unbalanced Sinkhorn Divergence was proposed in [Séjourné 2019]. It was proven to be a generalization of the original Sinkhorn Divergence. Furthermore it keeps the same properties, *i.e.,* when $e^{-\boldsymbol{C}/\varepsilon}$ is a positive definite kernel, it is a convex, symmetric, positive definite loss function which metrizes the convergence in law [Séjourné 2019]. The Unbalanced Sinkhorn Divergence reads

$$S_\phi^{\tau,\varepsilon}(\alpha, \beta, c) = \mathrm{OT}_\phi^{\tau,\varepsilon}(\alpha, \beta, c) - \frac{1}{2}\mathrm{OT}_\phi^{\tau,\varepsilon}(\alpha, \alpha, c) - \frac{1}{2}\mathrm{OT}_\phi^{\tau,\varepsilon}(\beta, \beta, c) + \frac{\varepsilon}{2}(m_\alpha - m_\beta)^2, \tag{2.35}$$

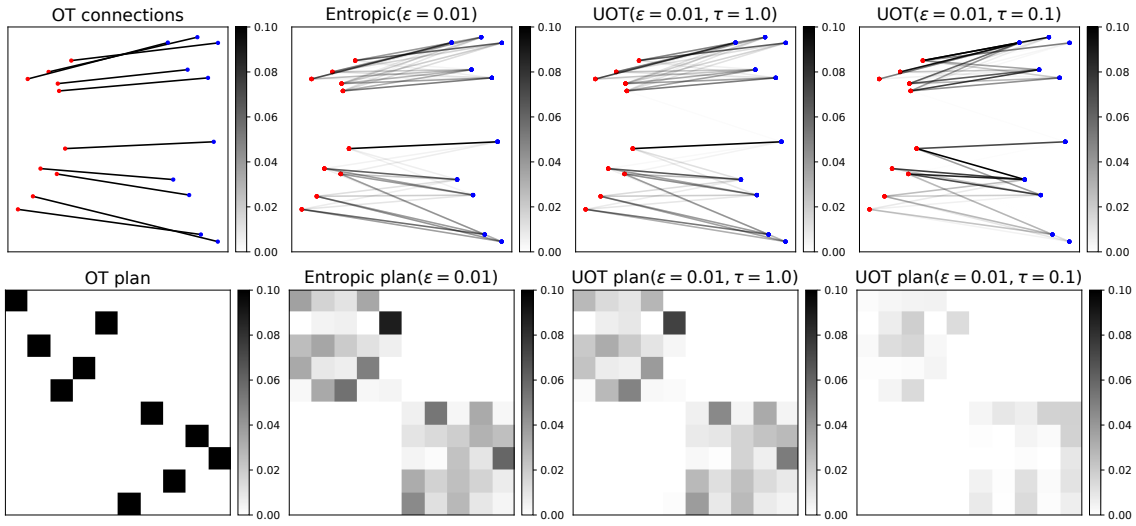


Figure 2.5: Optimal transport illustration between 2D measures. From left to right, we have an illustration of original OT, entropic-regularized OT and entropic-regularized unbalanced OT for two parameters τ.

where $m_\alpha = \int d\alpha$ and $m_\beta = \int d\beta$. The main difference is the introduction of the mass terms which are necessary to ensure positiveness and to recover a divergence. Those terms are equal and cancel each other in the case of balanced OT. Thus the Unbalanced Sinkhorn Divergence allows to mitigate between accelerated computations and conservation of key theoretical guarantees of the unbalanced formulation.

Discussion A downside of the unbalanced formulation is that for small values τ, the unbalanced OT cost is very small because too much of the mass is destroyed. This leads to a transport plan with a very small norm, and thus to gradients with a small norm which might increase the training time of neural network in domain adaptation [Fatras 2021b] for instance. This will be discussed in Chapter 9.

2.5 Conclusion

We introduced the concept of optimal transport in this chapter. We reviewed its primal definition, which corresponds to a linear program in the discrete case, and the corresponding dual formulation. Afterwards, we reviewed the 1D optimal transport closed-form solution. We then discussed the computational and memory complexities of OT. Next, we presented the entropic-regularized optimal transport variant, its impact on the OT problem and some of its solvers. To recover the separability axiom from the entropic regularization, we introduced the Sinkhorn Divergence which merges the best of entropic-regularized OT and exact OT. Finally, we discussed two other optimal transport variants of interests: the Gromov-Wasserstein distance which compares probability measures in different metric spaces and the unbalanced optimal transport variant which relaxes the marginal constraints and can compare measures with different mass. In the next section, we present more formally the supervised learning setting as well as the adversarial formulation and the connections with optimal transport.

Chapter 3

Optimal Transport in Deep Learning

Contents

3.1 Classification using neural networks

In this section, we start by presenting how optimal transport can be used in a context of classification with neural networks. We present formally the supervised learning problem for multi-class and multi-label classification. Then we review how OT can be used in a supervised learning setting. We then present the domain adaptation problem, which consists of transferring knowledge from a source dataset to a target dataset, and how optimal transport is a well suited tool to achieve this transfer. After discussing classification problems, we review basic methods to generate highly realistic data. We start with autoencoders and follow with generative adversarial networks. We also discuss how optimal transport was used to solve these problems.

3.1.1 Supervised learning and multi-label classification

Supervised learning

We start with the problem of supervised learning. In supervised learning, we want to find a function $f \in \mathcal{F}$ that describes the relationship between a random feature vector $\boldsymbol{x}$ and a random target vector $\boldsymbol{y}$, which follow the joint measure $\mathcal{P}(\mathcal{X}, \mathtt{Y})$. To this end, we first define a loss function L that penalizes the differences between the prediction $f(\boldsymbol{x})$ and the target $\boldsymbol{y}$, for instance an image $\boldsymbol{x}$ and the given class encoded in a label $\boldsymbol{y}$. Then, we define the average of the loss function L over the data measure $\mathcal{P}$, also known as the *expected risk*:

$$R(f) = \int L(f(\boldsymbol{x}), \boldsymbol{y}) d\mathcal{P}(\boldsymbol{x}, \boldsymbol{y}). \tag{3.1}$$

Unfortunately the joint measure $\mathcal{P}(\mathcal{X}, \mathtt{Y})$ is unknown and we have instead access to empirical samples drawn from this joint measure. Formally, let the *i.i.d.* data and their label be denoted as

$(\boldsymbol{x}_1, \boldsymbol{y}_1), \cdots, (\boldsymbol{x}_n, \boldsymbol{y}_n)$. We thus approximate the expected risk with the empirical measure, called empirical risk. We then minimize this empirical risk known as *empirical risk minimization* (ERM) [Vapnik 1998]. It is defined as

$$\text{argmin}_\theta \frac{1}{n} \sum_{i=1}^{n} L(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i), \tag{3.2}$$

where f_θ is the classifier parametrized by a vector θ such as a deep neural network. We consider a multi-class problem, *i.e.,* $\boldsymbol{y} = \boldsymbol{e}_c = [0, \cdots, 0, 1, 0, \cdots, 0]$, where 1 is at the c-th position. $\boldsymbol{y}$ is a "one-hot" vector. In general, the last function of a neural network is a softmax activation function whose output is a probability vector. The hypothesis of multi-class supervised learning is that one image has a single element to classify. However, real-world images have several classes. For instance autonomous vehicles need to deal with a rich environment composed of cars, traffic signals, pedestrians and many other important elements. To correctly classify all these elements, the supervised learning paradigm needs to be adapted. It is called multi-label classification and it is the topic of the next section.

Multi-label classification and Optimal Transport

In this section we review the multi-label classification problem and then, how we can use optimal transport losses to tackle this problem.

Multi-label classification In multi-label classification, the label is not a "one-hot" vector anymore and thus not a probability vector. In this problem, we can encode several information in the label and the formalism is the following. Each element of the label corresponds to a class which potentially appears in the image. Thus an image with a dog, a human and a frisbee would have three "1" for each element in the label. So the neural network outputs now a vector where its elements are all probabilities for the different classes to be in the image. In order to get a probability classification for each output element, the last activation function of a neural network is usually a sigmoid function [Kang 2006]. The function f is still learnt through empirical risk minimization and the contrast loss is usually a Kullback-Leibler divergence [Long 2015a].

However the usual loss L for classification penalizes the error in an *isotropic* manner, *i.e.,* they give the same importance to all errors between all classes independently. This behaviour does not reflect real life impact of errors, for instance confusing a cat and a dog is a bigger mistake than confusing two breeds of dogs. In the next paragraph, we discuss how optimal transport can be used in ERM to take into account the class similarities.

Multi-label Optimal Transport loss Let $f_\theta : \mathcal{X} \mapsto (\mathcal{P})^{n_C}$ be a map, where $\mathcal{P}$ denotes a probability and where n_C is the number of classes. In multi-label classification, the function $f_\theta(\boldsymbol{x})$ and the label $\boldsymbol{y}$ are measures of mass $\|f_\theta(\boldsymbol{x})\|_1$ and $\|\boldsymbol{y}\|_1$ respectively. Thus we can use unbalanced optimal transport cost as our function L to measure the dissimilarities between the label and the model output as the mass $\|f_\theta(\boldsymbol{x})\|_1$ and $\|\boldsymbol{y}\|_1$ are generally different. The strength of optimal transport losses is that it can take into account class similarities contrary to other losses. The similarities are encoded in the ground cost $\boldsymbol{C}$. Formally, the UOT cost can be computed as $h(f_\theta(\boldsymbol{x}), \boldsymbol{y}, \boldsymbol{C})$, where h is the entropic-regularized unbalanced optimal transport cost [Frogner 2015] as introduced in Section 2.4.2. Note that it is a smooth function *w.r.t.* the histograms, thus it has well defined gradients to train the model f_θ with stochastic gradient descent [Bottou 2010]. The difficulty is to choose an informative ground cost. In [Frogner 2015], authors chose to rely on an Euclidean distance between the Word2Vec class representation [Mikolov 2013]. This

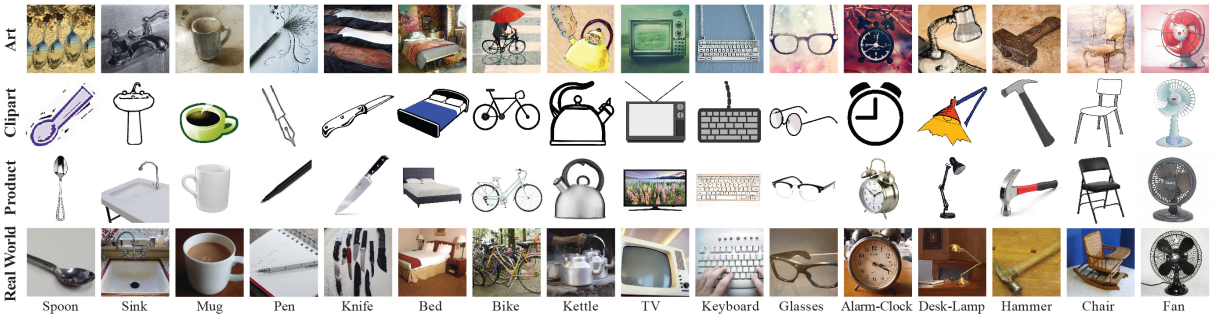


Figure 3.1: Illustration of the Office Home dataset.

choice allows to measure the similarities between the classes. Using the UOT loss in ERM improved the performances of the trained models on standard real-world datasets. Straightforwardly, the unbalanced optimal transport cost could be used in a multi-class learning problem as detailed in the previous section. In the next section, we discuss a problem where we want to classify unlabelled data using labeled data of two different domains.

3.1.2 Domain adaptation

In this section, we introduce the problem of Domain Adaptation (DA). It consists of transferring knowledge from a source dataset to a target dataset by using the labeled source data to classify the unlabeled target data [Pan 2010]. We follow the settings of unsupervised DA where we have a labeled source domain dataset $\mathcal{D}_s = \left\{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\right\}_{i=1}^n, \boldsymbol{x}_i \in \mathbb{R}^d$ and an unlabelled target domain dataset $\mathcal{D}_t = \left\{(\boldsymbol{x}_j^t)\right\}_{j=1}^n, \boldsymbol{x}_j \in \mathbb{R}^d$. Other variants exist such as semi-supervised domain adaptation, where a few labels are available in the target domain and supervised domain adaptation where all labels in the target domain are available. There are also extensions of domain adaptation called multi-source domain adaptation or multi-target domain adaptation where we have respectively access to several source or target datasets. We give an illustration of the Office-Home dataset [Venkateswara 2017] with different domains, see Figure 6.12.

In vanilla domain adaptation setting, both domains share the same label space $\mathcal{Y}_s = \mathcal{Y}_t$. Other variants can be considered such as partial domain adaptation, where extra classes in the source domain are present but not in the target domain $\mathcal{Y}_t \subset \mathcal{Y}_s$ and open set domain adaptation, where extra classes are in the target domain but not in the source domain $\mathcal{Y}_s \subset \mathcal{Y}_t$. Those two variants need dedicated solutions that we discuss latter. Usually, at least one of the two following assumptions is made in most domain adaptation methods:

- Class imbalanced: Label distributions are different in the two domains ($\mathcal{P}_s(\boldsymbol{y}) \neq \mathcal{P}_t(\boldsymbol{y})$), but the conditional distributions of the samples with respect to the labels are the same ($\mathcal{P}_s(\boldsymbol{x}^s|\boldsymbol{y}) = \mathcal{P}_t(\boldsymbol{x}^t|\boldsymbol{y})$). Open set and partial DA settings are special cases of class imbalanced;

- Covariate shift: Conditional distributions of the labels with respect to the data are equal $\mathcal{P}_s(\boldsymbol{y}|\boldsymbol{x}^s) = \mathcal{P}_t(\boldsymbol{y}|\boldsymbol{x}^t)$. However, data distributions in the two domains are supposed to be different $\mathcal{P}_s(\boldsymbol{x}^s) \neq \mathcal{P}_t(\boldsymbol{x}^t)$.

Based on the covariate shift hypothesis, [Sugiyama 2008] used an importance re-weighting strategy to learn a classifier, however it requires a large enough overlapping of distributions. Kernel alignment has also been proposed [Zhang 2013]. Other works learn a feature map $\mathcal{T}$ such that the new representations

of input data are matching, $\mathcal{P}_s(\mathcal{T}(\boldsymbol{x})) = \mathcal{P}_t(\mathcal{T}(\boldsymbol{x}))$ [Gong 2016]. Other approaches consider different class of maps such as projection [Baktashmotlagh 2013, Gong 2012, Fernando 2013, Long 2014, Gong 2016] and affine transform [Zhang 2013]. We can also use different divergences between the probability measures of source and target data [Si 2010, Long 2014, Gong 2016], but unfortunately these divergences require that the measures share the same support.

Another line of works is the alignment strategy in feature spaces and it can be summarized as follows; the neural network is used as a feature extractor and we align the extracted data features when data share the same class. The alignment of data spaces can be done with weight sharing [Sun 2016], reconstruction [Aljundi 2016], backpropagation [Ganin 2015] or by adding Maximum Mean Discrepancy and association-based losses between source and target layers [Long 2015b, Haeusser 2017]. The alignment can also be achieved by an adversarial formulation which pushes CNN to be able to discriminate whether a sample comes from the source or the target domain [Luo 2017, Tzeng 2015, Ganin 2016, Long 2018, Chen 2020]. Finally, the most recent works extend this adversarial logic to the use of generative models [Liu 2017, Sankaranarayanan 2018, Murez 2018], they use a class-conditioning or cycle consistency term to learn the discriminative embedding, such that embedded similar images in both domains are projected close by in the latent space.

In parallel of the empirical methods, several works were dedicated to make an analysis of the alignment strategy. They developed a bound on the target domain error rate by a source error rate plus an estimable term reflecting a certain distance between domains, called the alignment term, plus a non estimable term that is assumed to be small when adaptation is possible [Zhang 2019, Germain 2013]. [Ben-David 2007] considered the bounds for 0-1 loss in binary classification setting by introducing a divergence term that takes into account the complexity of the hypothesis space, and to a case where the hypothesis case is a RKHS [Cortes 2014]. In the next section, we discuss how optimal transport has been successful at solving the domain adaptation problem.

Optimal Transport in Domain Adaptation

Optimal transport has been successfully used on the domain adaptation problem. Earlier work [Courty 2014, Courty 2017a] used optimal transport to find a coupling between the source and the target domains, then they transported the source data and their labels into the target domain with a barycentric mapping from the coupling. Finally they learnt a classifier on the transported samples in the target domain. This strategy is illustrated in Figure 3.2. Authors studied a great variety of regularized optimal transport to create optimal connections between samples. As we discussed in Section 2.3.3, a $l_p l_1$ norm regularization was proposed in [Courty 2014] in order to concentrate the transport information on elements of the same class. In [Courty 2017a], a group-lasso regularization was used to promote the mass transfer from source data with the same labels to a given target data. They also introduce a Laplacian regularization which aims at preserving the data structure – approximated by a graph – during transport. On the theoretical side, [Redko 2017] justified that the Wasserstein metric can be used as a divergence measure between distributions to obtain generalization guarantees.

We now detail more recent approaches. JDOT is a method where optimal transport was used to find a coupling on the joint data and label distributions [Courty 2017b]. The ground cost incorporated a term on the data features and on the labels. Thus for two losses d and L, two data $\boldsymbol{x}_i, \boldsymbol{y}_i$ and $\boldsymbol{x}_j$ and a classifier f_θ, the cost is of the form $c(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{x}_j, f_\theta(\boldsymbol{x}_j)) = \eta d(\boldsymbol{x}_i, \boldsymbol{x}_j) + L(\boldsymbol{y}_i, f_\theta(\boldsymbol{x}_j))$. Intuitively, matching close source and target samples with similar labels has a smaller cost than with different labels. η is a positive parameter which balances the metric in the feature space and the label loss. When η is set to

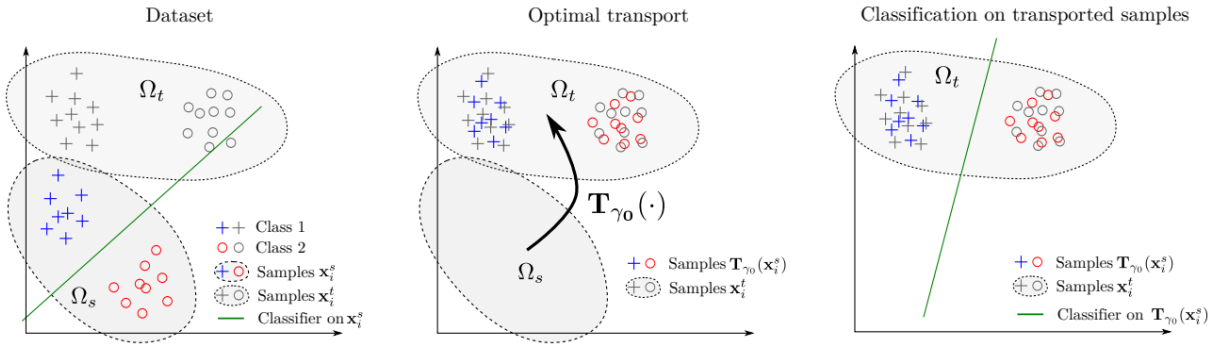


Figure 3.2: Illustration of [Courty 2017a] method. (left) dataset for training, i.e. source domain and target domain. Note that a classifier estimated on the training examples clearly does not fit the target data. (middle) a data dependent transportation map is estimated and used to transport the training samples onto the target domain. Note that this transformation is usually not linear. (right) the transported labeled samples are used for estimating a classifier in the target domain. Courtesy of [Courty 2017a].

$+\infty$, we recover the initial setting from [Courty 2017a]. Finally, authors [Courty 2017b] also provided a generalization bound to justify their method theoretically.

Based on this approach, DEEPJDOT [Damodaran 2018] used a neural network g_ψ as feature extractor, and used it instead of the data in the ground cost. Hence, the ground cost was $c(g_\psi(\boldsymbol{x}_i), \boldsymbol{y}_i, g_\psi(\boldsymbol{x}_j), f_\theta(g_\psi(\boldsymbol{x}_j))) = \eta d(g_\psi(\boldsymbol{x}_i), g_\psi(\boldsymbol{x}_j)) + L(\boldsymbol{y}_i, f_\theta(g_\psi(\boldsymbol{x}_j)))$. The feature extractor g_ψ and the classifier f_θ are then trained together to match the source and the target measures. The full DEEPJDOT loss and architecture is illustrated in Figure 3.3. It is important to note that the coupling was computed between minibatches of source and target data to keep a small computational complexity. This approximation is used in several problems and is the topic of Part III. This strategy reached state-of-the-art results. Other methods based on DEEPJDOT, which focused on the coupling, were published [Xu 2020a, Fatras 2021b] and other methods were designed to handle the case of open set DA [Xu 2020b] and partial DA [Fatras 2021b].

Several other alignment strategies based on optimal transport were developed [Mengxue 2020]. Authors built an attention-aware transport distance, which can be viewed as the prediction feedback of the iteratively learned classifier, to measure the domain discrepancy. Another work learnt the transport cost for the specific task of domain adaptation [Kerdoncuff 2020]. Finally, optimal transport can be coupled with adversarial learning to learn a domain invariant feature representation. [Shen 2018] uses a neural network, denoted by the domain critic, to estimate empirical Wasserstein distance between the source and target samples and optimizes the feature extractor network to minimize the estimated Wasserstein distance in an adversarial manner. And more recently, [Dhouib 2020b] introduced an OT-based adversarial approach to encourage large margin separation.

In this section, we have reviewed the domain adaptation problem as well as the different designed methods to solve it. We also discussed the specific case where optimal transport has been used to solve this problem. In the next section, we do not consider a classification problem but a data generation problem.

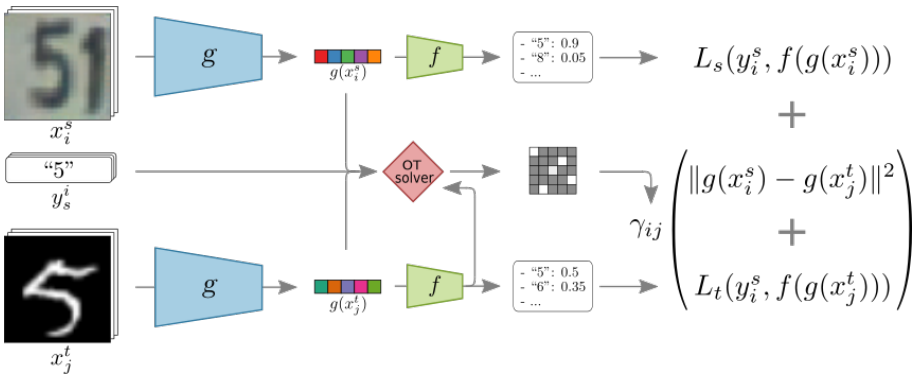


Figure 3.3: Overview of the DEEPJDOT method. While the structure of the feature extractor g and the classifier f are shared by both domains, they are represented twice to distinguish between the two domains. Both the latent representations and labels are used to compute per batch a coupling matrix Π that is used in the global loss function. Courtesy of [Damodaran 2018].

3.2 Generating high realistic data with Deep Learning

In this section, we review two main methods to generate highly realistic data. The first one is the autoencoder method. It aims to transform inputs into latent vectors and then, transform these same latent vectors as their corresponding inputs, with the least possible amount of distortion. We will review the basic idea and how optimal transport has been useful to solve it. Then we will review the Generative Adversarial Networks. Following an adversarial training, we will show how deep neural networks can generate data in an unsupervised way and how optimal transport has also been coupled with Generative Adversarial Networks.

3.2.1 AutoEncoder: Encoding data as a latent space

AutoEncoder [Rumelhart 1986] has become a standard tool to generate data because it has seen many improvements recently. It is composed of two distinct parts. An encoder whose the purpose is to encode an input into a latent vector which lies in a latent space. The second element is a decoder which aims to decode the latent vector into the original input. The encoder and the decoder are trained in order to minimize the reconstruction loss between the output and the original input. Then the latent space and the decoder are used to generate new data. Unfortunately, when the encoder is trained in this manner, it does not give a useful representation and sampling from the latent space becomes hard [Bengio 2013].

To overcome these difficulties, Variational Autoencoders (VAEs) [Kingma 2014] minimizes a variational upper bound on the Kullback-Leibler divergence. The upper bound is a reconstruction loss between the output and the input coupled with a regularization. Formally, we denote the measure associated to real and training data as $\mathcal{P}_r \in \mathcal{M}_1^+(\mathcal{X})$ (a positive Radon measure). We denote a latent variable model $\mathcal{P}_G$, specified by the prior distribution $\mathcal{P}_{\boldsymbol{z}}$ on the latent space $\mathcal{Z}$, and the decoder $\mathcal{P}_G(X|Z)$, which models the conditional likelihood. For two random variables X, Z, we denote the encoder as $Q(Z|X)$ which approximates the posterior distribution of the decoder. The regularization is $\mathbb{E}_{\mathcal{P}_r}\big(D_{\mathrm{KL}}(Q(Z,X), \mathcal{P}_{\boldsymbol{z}})\big)$ and it captures how distinct the image by the encoder of each training set is from the prior.

The total loss of VAEs is then:

$$D_{\mathrm{VAE}}(\mathcal{P}_r, \mathcal{P}_G) = \inf_{Q(Z,X)\in\mathcal{Q}} \mathbb{E}_{\mathcal{P}_r}\Big(- \mathbb{E}_{Q(Z,X)} \log(p_G(X|Z)) + D_{\mathrm{KL}}(Q(Z,X), \mathcal{P}_{\boldsymbol{z}})\Big), \tag{3.3}$$

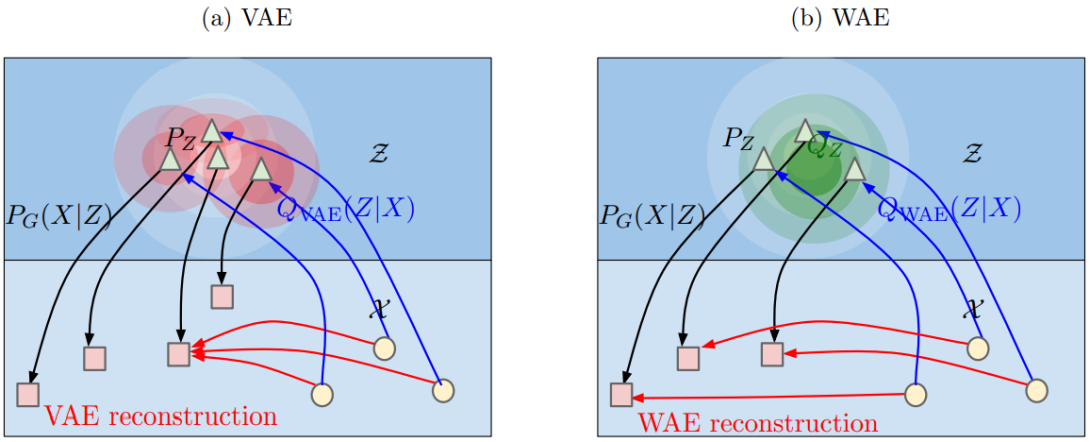


Figure 3.4: Both VAE and WAE minimize two terms: the reconstruction cost and the regularizer penalizing discrepancy between $\mathcal{P}_{\boldsymbol{z}}$ and distribution induced by the encoder Q. VAE forces $Q(Z|X = \boldsymbol{x})$ to match $\mathcal{P}_{\boldsymbol{z}}$ for all the different input examples $\boldsymbol{x}$ drawn from $\mathcal{P}_r$. This is illustrated on picture (a), where every single red ball is forced to match $\mathcal{P}_{\boldsymbol{z}}$ depicted as the white shape. Red balls start intersecting, which leads to problems with reconstruction. In contrast, WAE forces the continuous mixture $Q_Z := \int Q(Z|X)d\mathcal{P}_r$ to match $\mathcal{P}_{\boldsymbol{z}}$, as depicted with the green ball in picture (b). As a result latent codes of different examples get a chance to stay far away from each other, promoting a better reconstruction. Courtesy of [Tolstikhin 2018].

where $\mathcal{Q}$ is the set of, possibly restricted, conditional probability distributions. The decoder $\mathcal{P}_G(X|Z)$ is parametrized by a neural network and can have any form as long as its density $p_G(\boldsymbol{x}|\boldsymbol{z})$ can be computed and differentiated with respect to G's parameters. A standard choice is a Gaussian distribution $\mathcal{P}_G(X|Z) = \mathcal{N}((X; G(Z)), \sigma^2 \cdot I)$. If $\mathcal{Q}$ is the set of all conditional probability distributions $Q(Z, X)$, the VAE loss coincides with the negative marginal log-likelihood $D_{\text{VAE}}(\mathcal{P}_r, \mathcal{P}_G) = -\mathbb{E}_{\mathcal{P}_r} \log(\mathcal{P}_G(X))$. However to make the Kullback-Leibler divergence (3.3) tractable in closed-form, [Kingma 2014] uses a standard normal distribution $\mathcal{P}_{\boldsymbol{z}}$ and restricts $\mathcal{Q}$ to Gaussian distributions. That is why VAE is *minimizing an upper bound* on the KL-divergence $D_{\text{KL}}(\mathcal{P}_r, \mathcal{P}_G)$. To reduce the gap between the Kullback-Leibler divergence and VAE's loss, one possibility is to consider a larger set of encoder function $\mathcal{Q}$ as done in [Mescheder 2017, Makhzani 2016]. The encoder is learnt using Generative Adversarial Networks and their adversarial learning formulation. We detail this class of generative models in the next section.

Optimal Transport meets AutoEncoders

As the purpose of autoencoders is to reduce the discrepancy between the measures $\mathcal{P}_r$ and $\mathcal{P}_G$, [Tolstikhin 2018] proposed to rely on optimal transport. Following the above VAE framework, [Tolstikhin 2018] focused on deterministic decoders $G(X|Z)$, *i.e.*, $\mathcal{P}_G = G_\sharp \mathcal{P}_{\boldsymbol{z}}$. In this case, the Wasserstein distance can be rewritten as

$$W(\mathcal{P}_G, \mathcal{P}_{\boldsymbol{z}}, c) = \inf_{Q(Z|X): Q_{\boldsymbol{z}} = \mathcal{P}_{\boldsymbol{z}}} \mathbb{E}_{X \sim \mathcal{P}_r} \mathbb{E}_{Z \sim Q(Z|X)} \big[c(X, G(Z))\big], \tag{3.4}$$

where $Q_{\boldsymbol{z}} = Q(Z|X)_\sharp \mathcal{P}_r = \mathbb{E}_{X \sim \mathcal{P}_r} Q(Z|X)$. The full theorem and its proof can be found in [Tolstikhin 2018, Theorem 1]. This theorem allows us to optimize over random encoders $Q(Z|X)$ instead of optimizing over all couplings between the input measures. We relax the problem by removing the constraint on $Q_{\boldsymbol{z}}$ and adding a penalty instead. Thus the full WAE loss is

Figure 3.5: VAE (left column), WAE-MMD (middle column), and WAE-GAN (right column) trained on CelebA dataset. In "test reconstructions" odd rows correspond to the real test points. Courtesy of [Tolstikhin 2018].

$$D_{\mathrm{WAE}}(\mathcal{P}_G, \mathcal{P}_{\boldsymbol{z}}, c) = \inf_{Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{X\sim\mathcal{P}_r}\mathbb{E}_{Z\sim Q(Z|X)}\big[c(X, G(Z))\big] + \eta D_{\mathrm{Z}}(Q_{\boldsymbol{z}}, P_{\boldsymbol{z}}), \tag{3.5}$$

where $\mathcal{Q}$ is any nonparametric set of probabilistic encoder and η the penalty constant. Similarly to VAEs, the encoder and the decoder are parametrized by a neural network. [Tolstikhin 2018] proposed two different regularizations D_{Z}. The first one is called WAE-GAN where the divergence is equal to the Jensen-Shannon divergence $D_{\mathrm{Z}} = D_{\mathrm{JS}}$. The divergence is estimated using a discriminator and adversarial training as done in Generative Adversarial Networks. The other considered penalty is a MMD loss (see Section 2.3) and it is called WAE-MMD. MMD can be estimated with U-statistics as detailed in Section 7.3 and thus can be optimized with SGD. A strength of WAE on VAE is that the encoder is a deterministic mapping. We illustrate the difference between WAE and VAE in Figure 3.4 and we gathered some samples of the three methods in Figure 3.5. Several other autoencoder variants of WAE were published in [Patrini 2019, Kolouri 2019b].

In this section, we described the autoencoder as a generative model. We formulate the Variational AutoEncoder and then the Wasserstein AutoEncoder which takes an optimal transport loss to measure the distance between measures. In the next section, we describe another class of generative models called

Generative Adversarial Networks.

3.2.2 Generate realistic data with Generative Adversarial Networks

In this section, we describe a popular method to generate data, the *Generative Adversarial Networks* (GANs). After describing the initial formulation, we review a more recent formulation based on optimal transport. We close this section by discussing specific applications of GANs.

A two player game formulation

Generative Adversarial Networks [Goodfellow 2014a] have become a popular unsupervised method for data generation. Their objective is to learn a model that can generate data which are similar to a given training dataset. It has initially been designed as a two player game between two DNNs, a generator and a discriminator. The discriminator tries to predict if an image is real or generated and the generator tries to fool the discriminator with its generated images. More formerly, learning a GAN corresponds to minimizing a divergence between the measures of generated data and training data. The choice of this distance is fundamental and has led to several variants of GANs [Arjovsky 2017, Gulrajani 2017, Miyato 2018b, Nowozin 2016, Li 2017a, Genevay 2018]. Formally, let the measure associated to real and training data be denoted $\mathcal{P}_r \in \mathcal{M}_1^+(\mathcal{X})$ and the generated data distribution as $\mathcal{P}_G \in \mathcal{M}_1^+(\mathcal{X})$. G is a deterministic generator, classically expressed as a neural network, that takes as input a random vector $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}_q, \mathbf{I}_q)$ to generate a sample $G(\boldsymbol{z}) \sim \mathcal{P}_G$, where q is the dimension of this random variable. We generally assume that $q \ll d$, with d the dimensionality of the training data. Then, we want to reduce the distance with the real data measure $\mathcal{P}_r$. $\mathcal{P}_z$ is usually a Gaussian or a uniform distribution. GANs improve the generated data quality by minimizing the Jensen-Shannon divergence between the measures $\mathcal{P}_G$ and $\mathcal{P}_r$:

$$\min_{\psi} \max_{\phi} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_r}[\log \mathcal{D}_\phi(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim\mathcal{P}_z}[\log(1 - \mathcal{D}_\phi(G_\psi(\boldsymbol{z})))]. \tag{3.6}$$

However as the true measures $\mathcal{P}_G$ and $\mathcal{P}_r$ are unknown, we only consider their empirical measures from the available true data and the generated ones. Several variants were considered to control the label/class of generated data [Mirza 2014, Odena 2017]. Figure 3.6 gathers some highly realistic GAN-generated examples. The GAN formulation takes the form of a min-max optimization problem. It has thus bring a lot of attention to this underlooked optimization problem and several work were published to justify convergence or empirical methods to fasten convergence [Gidel 2019a, Gidel 2019b].

Wasserstein GAN: Optimal Transport meets GANs

The Wasserstein GAN [Arjovsky 2017] relies on a different divergence than the Jensen-Shannon divergence. It uses the Wasserstein distance to measure the discrepancy between the generated measure and the data measure. Unfortunately, the Wasserstein distance is intractable when one of the measure is continuous as it is the case for the generated measure $\mathcal{P}_{G_{\psi,\sharp}}$. Hence, [Arjovsky 2017] proposed to approximate the Kantorovich-Rubinstein duality defined in Section 2.2.1. The difficulty of this formulation is to satisfy the Lipschitz property of the map f. Thus we approximate f with a neural network, namely

$$W_1(\mathcal{P}_r, \mathcal{P}_{G_{\psi,\sharp}}) = \min_{\psi} \max_{\phi\in\Phi} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_r}[\mathcal{D}_\phi(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim\mathcal{P}_z}[\mathcal{D}_\phi(G_\psi(\boldsymbol{z}))], \tag{3.7}$$

where $\mathcal{D}_\phi$ is the dual potential and is within the set of 1-Lipschitz functions parameterized by weights ϕ, denoted Φ. Analogous to GANs, we still call $\mathcal{D}$ the "discriminator" although it is actually a real-valued

Figure 3.6: Samples generated by BigGAN model at 512×512 resolution. Courtesy of [Brock 2019].

function. In order to respect the 1-Lipschitz constraint, [Arjovsky 2017] used a weight clipping trick on the discriminator's weights during the optimization procedure. However this practice leads to unstability during optimization and poor minima. Another approach was proposed in [Gulrajani 2017], involving a gradient penalty that enforces the norm of the gradient to be equal to one, thus promoting $\mathcal{D}$ 1-Lipschitz. Formally it is defined as

$$\min_{\psi}\max_{\phi}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_r}[\mathcal{D}_\phi(\boldsymbol{x})]-\mathbb{E}_{\boldsymbol{z}\sim\mathcal{P}_z}[\mathcal{D}_\phi(G_\psi(\boldsymbol{z}))]+\lambda\mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathcal{P}_{\hat{x}}}[(\max\{0,\|\nabla\mathcal{D}_\phi(\hat{\boldsymbol{x}})\|_2-1\})^2], \tag{3.8}$$

where $\mathcal{P}_{\hat{x}}$ is the measure of samples along the straight lines between a pair of points from $\mathcal{P}_r$ and $\mathcal{P}_G$ and λ the regularization parameter. In practice, the true measure $\mathcal{P}_r$ is unknown and only an empirical counterpart $\widehat{\mathcal{P}}_r$ with n *i.i.d.* samples from $\mathcal{P}_r$ is available.

Other variants of GANs based on optimal transport have been published in the literature [Genevay 2018, Salimans 2018,Fatras 2021c]. These methods relied on primal OT by using other ground cost [Salimans 2018] or by using extracted features learned in an adversarial manner [Genevay 2018, Fatras 2021c]. We discuss in more details these methods in Section 7.1.

Other applications of GANs

In this section, we discuss the different applications of GANs. When GANs were introduced they were first applied to images. Since then, large scale GAN training in order to get high resolution and diverse images was achieved in Big GAN [Brock 2019] as illustrated in Figure 3.6. GANs also allow to tackle the super resolution task and it was studied in [Ledig 2017] and illustrated in Figure 3.7. In order to control the attributes of the generated images, a new generator architecture was proposed in Style GAN [Karras 2019]. Other work on image inpainting use GANs to reconstruct images [Liu 2018] or even modify them with some given labels [Jo 2019]. GAU-GAN [Park 2019] proposed to synthesize an image solely based on some given labels.

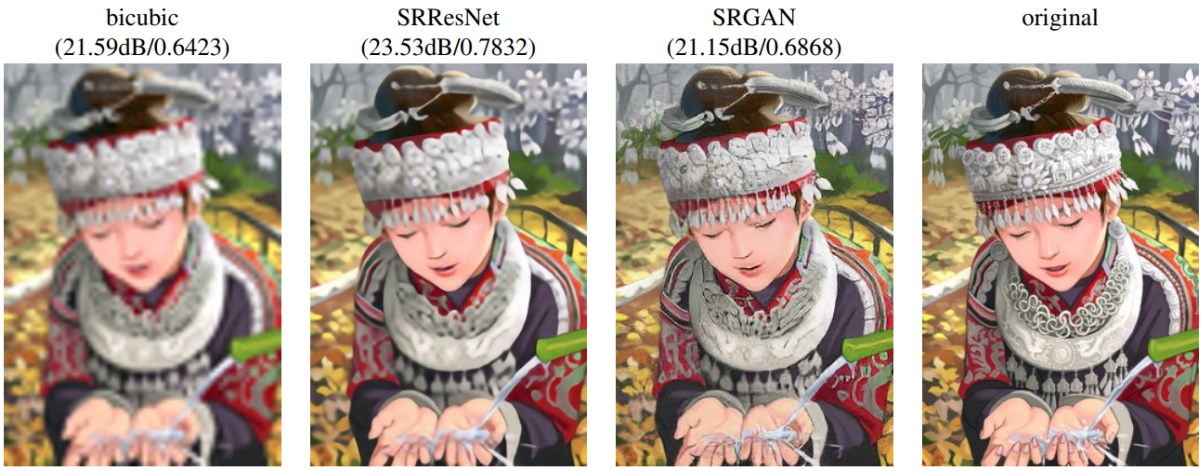


Figure 3.7: Super resolution samples examples from different technique. From left to right : bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Courtesy of [Ledig 2017]

Recently; a line of work focused on image-to-image translation problem with GANs [Isola 2017] or on video-to-video [Wang 2018b]. The goal was to learn a mapping function from a source input to a corresponding realistic target output that precisely depicts the content of the source input. The input can also be matched to several outputs as done in [Choi 2018b]. When the corresponding output of the input is not available, CycleGAN can be used to achieve the input-to-output translation problem in an unsupervised manner [Zhu 2017, Bansal 2018]. A similar approach was built for motion transfer in order to create videos of people dancing known as *Every Body Dance* [Chan 2019]. This strategy was also applied to domain adaptation [Murez 2018].

3.3 Conclusion

In this chapter, we have introduced the supervised learning setting for multi-class and multi-label problems. We have presented how optimal transport can be used to tackle the supervised learning problem, with a ground cost which encodes the class similarities. We then review the domain adaptation problem, it consists of classifying unlabeled data in a target domain using labeled data from a source domain. Optimal transport is used to align the domains of source and target data. In order to align the domains, recent methods used neural networks as feature extractors.

Then we reviewed two standard methods to generate highly realistic data. We first introduce the autoencoders which tries to reconstruct a given image from a latent space with the minimum of distortion. It takes the form of minimizing a divergence between two distributions, a setting particularly well suited for optimal transport, which was studied in Wasserstein autoencoder. Finally, we presented generative adversarial neural networks. Following an adversarial training between a generator and a discriminator, we showed that it also minimizes a divergence between measures. Hence, an optimal transport based variant, called Wasserstein GAN, was used in place of the original divergence. We reviewed the different advantages of this method and its implementation.

In the next part, we introduce the notion of adversarial examples and how it can be used to attack and to train a neural network for classification tasks. We then introduce the first two contributions of this

manuscript; a method based on virtual adversarial training to learn in the presence of noisy labels; and a method to generate data which are misclassified by a pre-trained classifier.

Part II

Optimal Transport meets Adversarial Examples

Chapter 4

Adversarial examples in Deep Learning

Contents

In this chapter we review adversarial examples in deep learning. We first give a definition and general algorithms to generate such examples and detail how we can attack a classifier with such examples. We then describe the semi-supervised learning problem and how we can design a new training strategy based on adversarial examples for semi-supervised learning.

4.1 Adversarial examples: attacking a classifier

4.1.1 Adversarial examples: an unexpected weakness

In this section, we present the concept of adversarial attacks. An adversarial example is an example which was designed in order to fool a pre-trained classifier. The notion of adversarial examples for neural networks was first discussed in [Szegedy 2013]. They observed that perturbed images with small modifications can lead to a large change in the classification and to incorrect class predictions. These examples pose security issues as one could for instance fool an autonomous vehicle by using stickers or paint on traffic signals in order to make the car fail to interpret the signals. Adversarial examples thus create new lines of research to make the neural networks robust to such attacks or to use this concept for training purposes.

Attack and defense lines of research have studied the adversarial robustness of neural networks, i.e. the sensitivity of neural networks against these adversarial attacks. The attack line of research aims at creating adversarial examples for a pre-trained classifier. It also studies whether the adversarial examples can be transferred to other unseen pre-trained classifiers. To attack a classifier, we can have different hypotheses regarding the knowledge of the pre-trained classifier [Goodfellow 2015, Song 2018]. In order to make the neural network more robust to adversarial examples, efficient methods include specific training procedures with better adversarial resistance [Zhang 2018a, Papernot 2016], modified

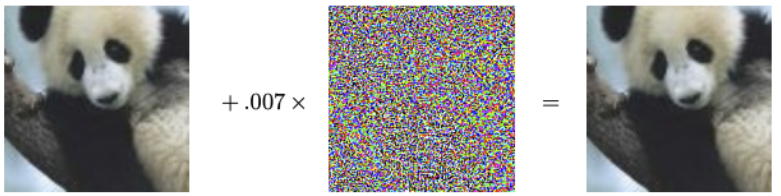


Figure 4.1: Example of adversarial example for a GoogLeNet [Szegedy 2014] trained on ImageNet. $J(\theta, \boldsymbol{x}, \boldsymbol{y})$ is the loss function to train the neural network. Courtesy of [Goodfellow 2015].

training data [Goodfellow 2015, Szegedy 2013] or the input samples before feeding them to classifiers [Hendrycks 2017, Samangouei 2018].

4.1.2 Generating adversarial examples

Generating adversarial examples with the classifier's gradients. Turning to the existing literature on adversarial examples, there exists two main strategies to generate such examples. The first strategy is to add a small perturbation to images that are correctly classified, *i.e.*, $\boldsymbol{x}^{adv} = \boldsymbol{x} + \boldsymbol{r}^a, \|\boldsymbol{r}^a\| \leq \rho$. The small perturbation, which can be computed from the classifier gradients, fools the classifier into making the wrong prediction. Intuitively, the gradient leads toward the direction with the most variation in the prediction. This is the main idea of the Fast Gradient Sign Method (FGSM) algorithm [Goodfellow 2015] and it is illustrated in Figure 4.1. Formally, let θ be the parameters of our neural network f_θ, the true class probability $q(\boldsymbol{x})$ and a divergence D. As $q(\boldsymbol{x})$ is unknown, we use the available label $\boldsymbol{y}$ instead. We look for the perturbation that leads to the largest change in the classification

$$\boldsymbol{r}^a = \underset{\boldsymbol{r}, \|\boldsymbol{r}\| \leq \rho}{\text{argmax}}\ D(\boldsymbol{y}, f_\theta(\boldsymbol{x} + \boldsymbol{r})). \tag{4.1}$$

Furthermore, estimating $\boldsymbol{r}^a$ is untractable but we can approximate it with a linear approximation of D with respect to $\boldsymbol{r}$ in (4.1). Thus the adversarial example is generated as $\boldsymbol{x}^a = \boldsymbol{x} + \rho\,\text{sign}(\nabla_{\boldsymbol{x}} L(f_\theta(\boldsymbol{x}), \boldsymbol{y}))$ and the gradient is computed using backpropagation. One problem of this approach is that one needs to know the full classifier architecture to compute the gradients. Another problem is that even a small perturbation can harm the image quality and lead to images that are not similar to the original data anymore. Such images can easily be detected as adversarial images [Feinman 2017, Pang 2018].

Generating adversarial examples with a GAN. The second paradigm does not require access to the classifier. It looks for adversarial examples only by having access to the classifier's prediction $f_\theta(\boldsymbol{x})$. Most of the techniques using the second strategy try to find adversarial examples using GANs [Goodfellow 2014a]. For instance, to generate adversarial examples without requiring the classifier's gradients, [Zhao 2018] uses a GAN and an invertor. The generator is a function which maps a latent space to the true data space. On the other hand, the invertor is an inverse function which maps the true data space to the latent space and they use it to look for adversarial examples. They add small perturbations to the latent vector $\boldsymbol{z}^*$ which represents a data $\boldsymbol{x}^*$ until they find an adversarial example. This method allows them to have very realistic adversarial examples without using the classifier's gradients. In [Song 2018] the authors use an

AC-GAN [Odena 2017] to model the data distribution with the assumption that an ideal model could generate all the sets of legitimate data. With such a model they search in the latent space for all the adversarial examples. Their search is done by minimizing the confidence score of a classifier while having the auxiliary classifier still predicting the correct class. So the adversarial examples look like true images and are adversarial for the attacked classifier.

4.1.3 Concepts of adversarial attacks

We gather now the different concepts related to adversarial attacks and some definitions. We rely on notations used in [Goodfellow 2015, Zhao 2018, Song 2018, Brown 2018]. We define by $\boldsymbol{y}^{pred} = f_\theta(\boldsymbol{x})$ the operation of getting the vector in the simplex of prediction (for every classes) $\boldsymbol{y}^{pred}$ of the datum $\boldsymbol{x}$ with respect to f_θ. Furthermore for a classifier f_θ, the probability to belong to the correct class y of a sample $\boldsymbol{x}$ is denoted as $f_\theta^y(\boldsymbol{x})$. Without loss of generality, we identify here $\mathcal{X}$ to a set of normalized images with d pixels $\mathcal{I} \triangleq \{[0,1]^d\}$. Depending on the application, we note that $\mathcal{X}$ can be different (*e.g.* spectral data). $\mathcal{N}_\mathcal{I} \subseteq \mathcal{I}$ stands for the subset of all natural images. By *natural* images, we mean an image which is likely to be drawn from the empirical distribution of training images $\mathcal{P}_r$, therefore related to a given training set of images. Assume that we dispose of an oracle o, such that $o(\boldsymbol{x})$ is the ground truth of the considered classification or prediction problem. In an attack scenario, we are given a classifier f_θ that one wants to fool. This can be understood in several ways. Before, we precise two different cases: A **white box attack** is an attack where an attacker has complete access to the classifier's structure and parameters θ (*e.g.* when we have access to the neural network gradients). Oppositely, a **black box attack** is an attack where the only knowledge is the outputs $\{\boldsymbol{y}^{pred}\}$ for some inputs $\{\boldsymbol{x}\}$. Note that we suppose in the following that we have access to a fixed set of pairs $\{\boldsymbol{x}, \boldsymbol{y}^{pred}\}$ but we cannot query the classifier on new samples (since we could compute approximated gradients similarly to FGSM [Goodfellow 2015]). We now give a detailed description about misclassified example categories.

As described in the above section, the idea of **perturbation-based example** is to add a small perturbation to an input in order to create an adversarial example. Most of the time; the perturbation is limited by a factor ρ. The perturbed example is an adversarial examples if $f_\theta(\boldsymbol{x}) \neq f_\theta(\boldsymbol{x}^{adv})$. Formally, it can be written as $\{\boldsymbol{x}^{adv} \in \mathcal{I} \mid \exists \boldsymbol{x}^{test} \in \mathcal{I}, \|\boldsymbol{x}^{adv} - \boldsymbol{x}^{test}\| < \rho \wedge o(\boldsymbol{x}^{adv}) = o(\boldsymbol{x}^{test}) = f_\theta(\boldsymbol{x}^{test}) \neq f_\theta(\boldsymbol{x}^{adv})\}$. **Natural examples** are misclassified data-like images. It can be misclassified training or testing data for instance. Some perturbation-based examples are also natural examples. Formally, it can be written as $\{x_{adv} \in \mathcal{N}_\mathcal{I} \mid o(\boldsymbol{x}^{adv}) \neq f_\theta(\boldsymbol{x}^{adv})\}$. The last category is **unrestricted examples**. In this context there are both a pre-trained classifier and an oracle. Unrestricted examples are samples where the oracle and the classifier predict different result. Formally, it can be written as $\{\boldsymbol{x}^{adv} \in \mathcal{I} \mid o(\boldsymbol{x}^{adv}) \neq f_\theta(\boldsymbol{x}^{adv})\}$.

We are now ready to describe different attacks against the pre-trained classifier, *i.e.,* the different strategies to fool the classifier. **Untargeted attack** stands for an attack from any adversarial example. **Targeted attack** acknowledges an attack with an adversarial example from a source label, which is classified as a target label, $\boldsymbol{y}^{target} = f_\theta(\boldsymbol{x}^{adv}) \neq \boldsymbol{y}^{source} = o(\boldsymbol{x}^{adv})$.

In the next section, we study how one can use adversarial examples to define a new training procedure for neural networks.

4.2 Virtual Adversarial Training for semi-supervised learning

In this section, we describe the semi-supervised learning problem, a variant of supervised learning, and we detail a state-of-the-art algorithm to solve it based on adversarial examples and semi-supervised learning

hypotheses.

4.2.1 Semi-supervised learning

Semi-supervised learning is a classification task where some data have labels $(\boldsymbol{x}_1, \boldsymbol{y}_1), \cdots, (\boldsymbol{x}_l, \boldsymbol{y}_l)$ while the remaining data are unlabelled $\boldsymbol{x}_{l+1}, \cdots, \boldsymbol{x}_n$ such that the number of labelled data is usually small, *i.e.,* $l \ll n$. This is a challenging problem because neural networks are known to need a large amount of data to achieve good generalization [Zhang 2017]. Hence learning neural network classifiers in semi-supervised learning is a difficult problem. To overcome the lack of labelled data, several works relied on using a relationship graph between the labeled and the unlabeled data to spread the label information [Belkin 2004]. A more recent approach uses input perturbations with deep neural networks [Zhang 2018a, Berthelot 2019, Miyato 2018a]. The method we describe in the next section, called *Virtual Adversarial Training* (VAT) [Miyato 2018a], is one of these methods. They rely on the semi-supervised learning assumptions [Chapelle 2006, Section 1.2],

1. Data which are close to each other are more likely to share the same label.
2. Data tend to form clusters, and data in the same cluster are more likely to share a label.

VAT is built on the semi-supervised hypotheses to classify unlabeled data using labeled data. It promotes a local uniformity of the classifier in a ball around each input, thus data close to each other should share the same label as well as data inside a cluster. It is the topic of the next section.

4.2.2 Virtual Adversarial Training: the power of approximations

In this section, we introduce the *Virtual Adversarial Training*. We start by presenting the adversarial training framework and then how we use it to define a regularization for semi-supervised learning.

Adversarial Training. To make the model f_θ more robust to adversarial examples, one can use the generated adversarial examples from the FGSM method to define a new loss function called *adversarial training* [Goodfellow 2015]. For a sample $\boldsymbol{x}$ with corresponding label $\boldsymbol{y}$, the adversarial training framework aims at minimizing a measure of discrepancy between the true class probability, $q(\boldsymbol{x})$, and the prediction of a model f_θ for a slightly perturbed version of $\boldsymbol{x}$, $f_\theta(\boldsymbol{x} + \boldsymbol{r}^a)$. We discussed the computation of $\boldsymbol{r}^a$ in Section 4.1. In other words, adversarial training tries to minimize the change in classification in a ball around each input. It takes the form of the following regularization,

$$\begin{aligned} L_{\text{AT}}(\boldsymbol{x}, \theta) &= D(q(\boldsymbol{x}), f_\theta(\boldsymbol{x} + \boldsymbol{r}^a)), \\ \text{where } \boldsymbol{r}^a &= \underset{\boldsymbol{r}, \|\boldsymbol{r}\| \leq \rho}{\operatorname{argmax}} D(q(\boldsymbol{x}), f_\theta(\boldsymbol{x} + \boldsymbol{r})). \end{aligned} \tag{4.2}$$

In Eq. (4.2), $D(p_1, p_2)$ measures the discrepancy between the probability vectors p_1 and p_2 thanks to the divergence D. For instance, D can be the Kullback-Leibler divergence. ρ represents the radius of the local ball-shaped neighbourhood, where we seek the adversarial sample.

Approximating the true measure. The true class probability $q(\boldsymbol{x})$ is unknown and intractable in practice. In semi-supervised learning, most of samples do not have a label $\boldsymbol{y}$ so we can not rely on it. That is why, [Miyato 2018a] proposed to use a 'virtual' label, i.e., the current label estimate from the

classifier $f_\theta(\boldsymbol{x})$. This approximation makes sense as at the end of the training, the current estimate should be close to the true label when the number of data is large. Another strength of relying on $f_\theta(\boldsymbol{x})$ is that the quantity is defined even for unlabelled data. Since in VAT $f_\theta(\boldsymbol{x})$ is taken as a label, it becomes a constant that we refer to as $\hat{f}_\theta(\boldsymbol{x})$, leading to the following regularization term,

$$\begin{aligned} L_{\text{VAT}}(\boldsymbol{x}, f_\theta) &= D(\hat{f}_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x} + \boldsymbol{r}^a)), \\ \text{where } \boldsymbol{r}^a &= \underset{\boldsymbol{r}, \|\boldsymbol{r}\| \leq \rho}{\text{argmax}}\, D(\hat{f}_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x} + \boldsymbol{r})). \end{aligned} \tag{4.3}$$

Using $f_\theta(\boldsymbol{x})$ instead of $q(\boldsymbol{x})$ promotes a local uniformity in the predictions around each input. Thus, L_{VAT} can be seen as a measure (negative) of the local smoothness, or also as an estimation of the local Lispschitz constant in the ρ neighborhood of $\boldsymbol{x}$ with respect to the metric D, hence a measure of complexity of the function. The full regularization term of VAT is the expectation of L_{VAT} over all samples,

$$\mathcal{R}_{\text{VAT}}(\boldsymbol{X}, f_\theta) = \frac{1}{N} \sum_{i=1}^{N} L_{\text{VAT}}(\boldsymbol{x}_i, f_\theta). \tag{4.4}$$

By penalizing large local changes in classification, the classifier should have a smooth local prediction, thus giving the same label to data close to each other. This local spread of information should also spread among the whole cluster, thanks to the semi-supervised hypothesis.

Adversarial samples computation. VAT requires an efficient computation of adversarial samples. One could use the gradient with respect to the input but because of differentiability, it vanishes in $\boldsymbol{r} = 0$. When we approximate D in $\boldsymbol{r} = 0$ through the second order Taylor expansion, we have

$$D(\hat{f}_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x} + \boldsymbol{r})) \underset{r=0}{\sim} \frac{1}{2} \boldsymbol{r}^t \boldsymbol{H}_{\boldsymbol{r}} \boldsymbol{r}. \tag{4.5}$$

However, computing the Hessian $\boldsymbol{H}_{\boldsymbol{r}}$ with respect to the input is costly. Instead we use an iterative method to estimate the dominant Hessian's eigenvector that represents the direction in which the classification function will change the most. We use the power iteration method [Golub 2000, Chapter 6]. The algorithm starts from a random normalized vector $\boldsymbol{d}$ and computes the gradient w.r.t $\xi \boldsymbol{d}$ as follows:

$$\boldsymbol{d} \longleftarrow (\nabla_{\boldsymbol{d}} D(\hat{f}_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x} + \xi \boldsymbol{d})) - \nabla_{\boldsymbol{d}} D(\hat{f}_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x})))/\xi,$$

where ξ be a small real number. Since the second term is zero, we update the value of $\boldsymbol{d}$ with the calculated gradient $\nabla_{\boldsymbol{d}} D(\hat{f}_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x} + \xi \boldsymbol{d}))/\xi$. This operation is repeated $t_{\max}$ times, but the literature suggests that only one iteration is sufficient to achieve state-of-the-art results [Miyato 2018a]. In practice, the value of ξ is set to 10^{-6}.

Once the adversarial direction $\boldsymbol{d}$ is defined, one can obtain the adversarial example with $\boldsymbol{r}^a = \rho \boldsymbol{d} / \|\boldsymbol{d}\|_2$, by projecting onto the sphere of radius ρ. The adversarial samples follow the direction where the classification function has the biggest variation w.r.t D.

4.3 Conclusion

We have presented a weakness of deep neural networks when they are trained in a classical manner for classification problems: the adversarial examples. Adversarial examples are malicious data, which look like training data, but are misclassified for a given pre-trained classifier. Thus they can be used to attack the pre-trained classifier. We discussed the standard strategies to generate such examples, by input

modification or by using GANs, and we reviewed the different concepts for attacking classifiers. We then showed how the adversarial examples can be used to design a new training algorithm for neural networks called adversarial training. Following this training strategy, virtual adversarial training was designed to train a model in semi-supervised learning, a setting where only a few data have a label. For a given input, VAT uses the current neural network prediction as a label and penalizes large changes in the prediction for a perturbed version of the input.

In the next chapters, we introduce two contributions. The first contribution is an extension of the VAT algorithm which uses an optimal transport loss in a learning with noisy labels scenario. We review the label noise problems and the different lines of research to solve it. We then introduce our second contribution and finally we evaluate it on different benchmarks. The second chapter is dedicated to the generation of natural examples, training images which are misclassified, using GANs. We introduce a simple and efficient reweighting strategy in order to give more importance to misclassified samples during the training of GANs.

Chapter 5

Wasserstein Adversarial Regularization for learning with noisy labels

Contents

In this chapter, we present the problem of learning with noisy labels problem and different lines of research to solve this problem. We then present how adversarial training can be used in the context of noisy labels. Next, we present our main contribution: using a Wasserstein loss in the virtual adversarial training to promote local smoothness between very different classes and complex boundaries between classes similar to each other. We also present the selection of an insightful ground cost. Afterwards, we extensively evaluate our proposed method on different classical learning with noisy labels benchmarks. These contributions have been published in IEEE Transactions on Pattern Analysis and Machine Intelligence [Fatras 2021a].

5.1 Learning with noisy labels

In this section, we first present the label noise problem, we also discuss the type of noise we consider for labels and the related work.

5.1.1 Definition and setting

The learning with noisy labels setting shares a lot of similarities with supervised learning. Like the supervised learning setting, we suppose that for each data $\boldsymbol{x}$, we have access to its corresponding label $\boldsymbol{y}$

which encodes its class. The notable difference is that a portion of the label $\boldsymbol{y}$ are corrupted and encodes the wrong class of $\boldsymbol{x}$. For instance, an image of a bird whose label is a plane. Such labels are called *noisy labels*. It is a common problem in practice since annotating large datasets is a challenging and costly task, which is practically impossible to do perfectly for every task at hand. It is then most likely that datasets will contain incorrectly labeled data, which induces noise in those datasets and can hamper learning. Furthermore, the probability of facing this problem increases when the dataset contains several fine grained classes that are difficult to distinguish [Schroff 2011, Krause 2016, Dubey 2018]. These noisy labels are problematic because as pointed out in [Zhang 2017], deep convolutional neural networks have huge memorization abilities and can learn very complex functions. That is why training a neural network with noisy labels can lead to poor generalization [Arpit 2017, Wang 2018a, Choi 2018a]. We now discuss the choice of noise setting we consider.

The label noise problem arises whenever some elements $\boldsymbol{y}$ do not match the real class of $\boldsymbol{x}$. Several scenarii exist: in the *symmetric* label noise, labels can be flipped uniformly across all classes, whereas in the *asymmetric* label noise, labels $\boldsymbol{y}$ in the training set can be flipped with higher probability toward specific classes. We note that the first scenario, while thoroughly studied in the literature, is highly improbable in real situations: for example, it is more likely that an annotator mislabels two breeds of dogs than a dog and a car. Hence, noise in labels provided by human annotators is not symmetric since annotators make mistakes depending on class similarities [Misra 2016]. In the next section, we describe the different lines of research to mitigate the influence of these noisy labels.

5.1.2 Related work on label noise

The label noise problem has been considered mainly in three ways in recent literature.

First are *data cleaning methods*: [Brodley 1999] uses a set of learning algorithms to create classifiers that serve as noise filters for the training data. [Vahdat 2017, Xiao 2015, Li 2017b] learn relations between noisy and clean labels before estimating new labels for training. In [Lee 2018], few human who verified labels were necessary to detect noisy labels and adapt learning. Authors in [Natarajan 2013] use a simple weighted surrogate loss to mitigate the influence of noisy labels. In [Jiang 2018, Ren 2018], the methods rely either on a curriculum or on meta-gradient updates to re-weight training sets and downweight samples with noisy labels.

Second are *Transition probability-based methods*: [Liu 2014, Menon 2015, Sukhbaatar 2014, Patrini 2017, Hendrycks 2018] estimate a probability for each label to be flipped to another class and use these estimations to build a noise transition matrix. In [Sukhbaatar 2014], the authors add an extra linear layer to the softmax in order to learn the noise transition matrix itself, while [Hendrycks 2018] uses a small set of clean validation data to estimate it. [Patrini 2017] proposes a forward/backward loss correction method, which exploits the noise transition matrix to correct the loss function itself.

Third are *regularization-based methods:* In [Reed 2015, Ma 2018], the authors use a mixture between the noisy labels and network predictions. In [Tanaka 2018a, Kun 2019a], the regularization is achieved by alternatively optimizing the network parameters and estimating the true labels while the authors of [Han 2018, Yu 2019, Hongxin 2020] propose peer networks feedbacking each other about predictions for the noisy labels. [Song 2019] proposes to replace noisy labels in the mini-batch by the consistent network predictions during training, while [Chen 2019] proposes noisy cross-validation to identify samples that have correct labels. In [Wang 2019, Zhang 2018b, Ghosh 2017], robust loss functions are proposed to overcome limitations of cross entropy loss function. Our proposed method follows this strategy where the regularization promotes local smoothness to fight the label noise influence and it is the topic of the next

section.

5.2 Wasserstein Adversarial Regularization for label noise

In this section, we present how we can use the Virtual Adversarial Training algorithm to tackle the label noise problem but discuss some limits. We then introduce an optimal transport based extension of VAT which corrects these limits.

5.2.1 Adversarial regularization: smoothing local predictions

To prevent a classifier to overfit on noisy labels, we would like to regularize its decision function in areas where the local uniformity of training labels is broken, *i.e., when some labels are under represented in specific data area.* To achieve such desired local uniformity, robust optimization can be used. This amounts to enforce that predicted labels are uniform in a local neighborhood $\mathcal{U}_i$ of data point $\boldsymbol{x}_i$. This changes the total loss function in the following way:

$$\arg\min_{\theta} \sum_{i=1}^{N} \max_{\boldsymbol{x}_i^u \in \mathcal{U}_i} L_{\text{CE}}(f_\theta(\boldsymbol{x}_i^u), \boldsymbol{y}_i) \tag{5.1}$$

Because the robust optimization problem [Caramanis 2011] in (5.1) is hard to solve exactly, adversarial training [Goodfellow 2015, Shaham 2015] was proposed as a possible surrogate. Instead of solving the max inner problem, it suggests to replace it by enforcing uniformity in the direction of maximum perturbation, called the adversarial direction as presented in Section 4.1. Mainly used for robustness against adversarial examples, adversarial training has been successfully used in semi-supervised learning with the *Virtual Adversarial Training* algorithm presented in Section 4.2.2. Thus to solve the label noise problem, we propose to use the VAT regularization. The optimized loss is then

$$L_{\text{tot}}(\boldsymbol{x}_i, \boldsymbol{y}_i, f_\theta) = L_{\text{CE}}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) + \eta R_{\text{VAT}}(\boldsymbol{x}_i, f_\theta), \tag{5.2}$$

Where like in VAT, $f_\theta(\boldsymbol{x}_i)$ is the neural network prediction of the input and $f_\theta(\boldsymbol{x}_i + \boldsymbol{r}^a)$ is the neural network prediction of the adversarial input. A sound choice for D can be the Kullback-Leibler (KL) divergence. R_{VAT} promotes local uniformity in the predictions without using the potentially noisy label $\boldsymbol{y}_i$: therefore, it reduces the influence of noisy labels, since it is computed from the prediction $f_\theta(\boldsymbol{x}_i)$ that can be correct when the true label is not. Our regularization shares strong similarities with Virtual Adversarial Training (VAT) [Miyato 2018a], at the notable exception that we do not consider a semi-supervised learning problem and that we regularize on the labeled training positions $\boldsymbol{x}_i$, where VAT is applied on unlabeled samples. Thus we call this framework AR which stands for *Adversarial Regularization.* It can be shown that this regularization acts as a label smoothing technique:

Proposition 6. *Let D be the Kullback-Leibler divergence. Let $\gamma = \frac{\eta}{\eta+1} \in [0, 1[$. Let $\boldsymbol{y}^a = f_\theta(\boldsymbol{x} + \boldsymbol{r}^a)$ be the predicted adversarial label. Let H be the entropy. The regularized learning problem $L_{tot}(\boldsymbol{x}, \boldsymbol{y}, f_\theta)$ in (5.2) is equivalent to :*

$$L_{\text{tot}}(\boldsymbol{x}, \boldsymbol{y}, f_\theta) \equiv L_{\text{CE}}(f_\theta(\boldsymbol{x}), (1-\gamma)\boldsymbol{y} + \gamma \boldsymbol{y}^a) - \gamma H(\boldsymbol{y}^a).$$

Proof. The total learning loss for one sample $(\boldsymbol{x}, \boldsymbol{y})$ is:

$$L_{\text{tot}}(\boldsymbol{x}, \boldsymbol{y}, f_\theta) = L_{\text{CE}}(f_\theta(\boldsymbol{x}), \boldsymbol{y}) + \eta R_{\texttt{AR}}(\boldsymbol{x}, f_\theta). \tag{5.3}$$

We write the $R_{\texttt{AR}}$ regularization as:

$$\begin{aligned} R_{\texttt{AR}}(\boldsymbol{x}, f_\theta) &= D_{\mathrm{KL}}(f_\theta(\boldsymbol{x}+\boldsymbol{r}^a), f_\theta(\boldsymbol{x})), \\ \text{where } \boldsymbol{r}^a &= \underset{\boldsymbol{r}, \|\boldsymbol{r}\| \leq \rho}{\operatorname{argmax}} D_{\mathrm{KL}}(f_\theta(\boldsymbol{x}+\boldsymbol{r}), f_\theta(\boldsymbol{x})). \end{aligned} \tag{5.4}$$

We have that:

$$\begin{aligned} D_{\mathrm{KL}}(f_\theta(\boldsymbol{x}+\boldsymbol{r}^a), f_\theta(\boldsymbol{x})) &= \sum_c f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)^{(c)} \log \frac{f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)^{(c)}}{f_\theta(\boldsymbol{x})^{(c)}} \\ &= \sum_c f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)^{(c)} \log f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)^{(c)} - \sum_c f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)^{(c)} \log f_\theta(\boldsymbol{x})^{(c)} \\ &= -\sum_c f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)^{(c)} \log f_\theta(\boldsymbol{x})^{(c)} - H(f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)). \end{aligned} \tag{5.5}$$

where H is the entropy function. Consequently, the total loss can be rewritten as:

$$L_{\mathrm{tot}}(\boldsymbol{x}, \boldsymbol{y}, f_\theta) = -\sum_c (\boldsymbol{y}^{(c)} + \eta f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)^{(c)}) \log f_\theta(\boldsymbol{x})^{(c)} - \eta H(f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)) \tag{5.6}$$

Here $\eta \in \mathbb{R}^+$. Let $\eta = \frac{\varepsilon}{1-\varepsilon}$ with $\varepsilon \in [0, 1[$. We have the following equivalence:

$$\begin{aligned} (1-\varepsilon) L_{\mathrm{tot}}(\boldsymbol{x}, \boldsymbol{y}, f_\theta) &= -\sum_c ((1-\varepsilon)\boldsymbol{y}^{(c)} + \varepsilon f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)^{(c)}) \log f_\theta(\boldsymbol{x})^{(c)} - \varepsilon H(f_\theta(\boldsymbol{x}+\boldsymbol{r}^a)) \\ &= L_{\mathrm{CE}}(f_\theta(\boldsymbol{x}), \underbrace{(1-\varepsilon)\boldsymbol{y} + \varepsilon f_\theta(\boldsymbol{x}+\boldsymbol{r}^a))}_{\text{Interpolated label}} - \varepsilon \underbrace{H(f_\theta(\boldsymbol{x}+\boldsymbol{r}^a))}_{\text{adversarial label entropy}} \end{aligned} \tag{5.7}$$

□

This leads to the following interpretation: instead of learning over the exact label or over a mix between the exact label and the network prediction, we learn over an interpolation between the data $\boldsymbol{y}_i$ and the adversarial label $\boldsymbol{y}_i^a$, while maximizing the entropy of the adversarial label (i.e. blurring the boundaries of the classifier). Note that label smoothing techniques are known to create cluster of classes in the penultimate layer of the neural network [Müller 2019], thus leading to appealing behaviour in the presence of noisy labels. Related developments can be found in adversarial label smoothing (ALS) [Shafahi 2018, Goibert 2019], which aims at providing robustness against adversarial attacks.

Yet, one of the major limit of this approach is that the regularization is conducted with the same magnitude between all classes without considering potential class similarities. As a consequence, a strong regularization can remove the label noise, but also hinder the ability of the classifier to separate similar classes where a complex boundary is needed. In the next section we present how we overcome this issue. We propose to replace D by a geometry-aware divergence taking into account the specific relationships between the classes for the classifier to not overfit on noisy labels when the classes are significantly different and to have complex boundaries when they are similar. Taking these relations into account avoid overfitting on noisy labels between dissimilar class, while allowing complex boundaries between classes with high similarities.

5.2.2 Wasserstein adversarial regularization to consider class similarities

To make the divergence aware of specific relationships between classes, we replace the "isotropic" divergence D with an optimal transport cost computed in the labels space. We name our proposed method Wasserstein

Adversarial Regularization. In a related way, [Frogner 2015] used the Wasserstein distance as a loss in a learning system between the output of the model for multi-label learning as detailed in Section 3.1.1. In this setting, OT can be computed because in multi-class problem, the classifier output and the label are probability measures. Note that the computational complexity of exact OT scales with respect to the number of classes instead of the number of data as main OT applications, *i.e.,* the complexity is $\mathcal{O}(n_C^3 \log(n_C))$. In classical learning with noisy labels benchmarks, the number of classes can go up to 1000 classes. The interest of optimal transport is to take into consideration the geometry of the label space. We define the proposed regularization term R_{WAR} as follows:

$$\begin{aligned} &R_{\text{WAR}}(\boldsymbol{x}_i) = \mathfrak{L}^{\varepsilon}(f_\theta(\boldsymbol{x}_i + \boldsymbol{r}_i^a), f_\theta(\boldsymbol{x}_i), \boldsymbol{C}) \\ &\text{with } \boldsymbol{r}_i^a = \underset{\boldsymbol{r}_i, \|\boldsymbol{r}_i\| \leq \rho}{\operatorname{argmax}} \mathfrak{L}^{\varepsilon}(f_\theta(\boldsymbol{x}_i + \boldsymbol{r}_i^a), f_\theta(\boldsymbol{x}_i), \boldsymbol{C}). \end{aligned} \tag{5.8}$$

In practice, we will use the solution of the sharp entropic variant of the optimal transport problem [Cuturi 2013, Luise 2018]:

$$\begin{aligned} &\mathfrak{L}^{\varepsilon}_{\text{sharp}}(f_\theta(\boldsymbol{x}_i + \boldsymbol{r}_i^a), f_\theta(\boldsymbol{x}_i), \boldsymbol{C}) = \langle \Pi^*_\varepsilon, \boldsymbol{C} \rangle \\ &\text{with } \Pi^*_\varepsilon = \underset{\Pi \in U(f_\theta(\boldsymbol{x}_i + \boldsymbol{r}_i^a), f_\theta(\boldsymbol{x}_i))}{\operatorname{argmin}} \langle \Pi, \boldsymbol{C} \rangle - \varepsilon H(\Pi) \end{aligned}$$

Using this regularized version has several advantages similar to the entropic-regularized OT that we recall: *i)* it lowers the computational complexity to near-quadratic [Altschuler 2017], *ii)* when $\varepsilon > 0$, it turns the problem into a strongly convex one, which makes the OT loss differentiable with respect to the input measures [Peyré 2019, proposition 4.6], *iii)* it allows to vectorize the computation of all Wasserstein distances in a batch, which is particularly appealing for training deep neural nets and finally *iv)* it is a better approximation than the entropic-regularized OT [Luise 2018]. Based on [Genevay 2018], we use the *AutoDiff* framework, which approximates the derivative of this regularization with a fixed number of iterations of the Sinkhorn algorithm. We recall that using a large ε results in smoothing the loss and speeds up the Sinkhorn algorithm, but drives the solution away from the true OT solution. As such, we recommend to use a small ε to stay as close as possible to the true OT loss. Note that for all the experiments in the main task, we have used same regularization parameter ($\varepsilon = 0.05$) with 20 Sinkhorn iterations, to illustrate that, while the regularization parameter is important, it is not too sensitive across datasets. To summarize, for each data in a batch, we compute the OT between the input data network prediction and the adversarial input data network prediction. Hence the OT complexity is of order $\mathcal{O}(m \times n_s \times n_C^2)$, where m is the batch size, n_s the computation budget and n_C the number of classes. We compare the computation time of our method in the experimental section. The full model update using WAR can be found in Algorithm 3.

Choice of ground cost. The ground cost $\boldsymbol{C}$ reflects the geometry of the label space. It bridges the gap between AR and WAR. An uninformative 0-1 ground cost, *i.e.* 0 over the diagonal and 1 everywhere else, would give the total variation (TV) loss (Remark 2.26 in [Peyré 2019]), which could also be used as D in the AR framework. Below, we refer to this special case as $\texttt{WAR}_{0-1}$. To define a $\boldsymbol{C}$ matrix encoding class relationships, multiple choices are possible. Relying on expert knowledge, one could set it manually, but this becomes unpractical when a large number of classes is present, and this knowledge is not always available, or even prone to errors. We propose two variants to estimate the ground cost:

- In absence of prior information about the nature of the source of labelling errors, we first propose to rely on semantic distances based on word embeddings such as *word2vec* [Mikolov 2013]. Similarities

Algorithm 3 Wasserstein Adversarial Training (WAT) model update

Inputs: $\boldsymbol{x}, \boldsymbol{y}, \rho, \xi, \eta, \varepsilon, \theta_t, \gamma, k_{\max}$, ground cost $\boldsymbol{C}$
Select sample $(\boldsymbol{x}, \boldsymbol{y})$
Random $\boldsymbol{d} \in \mathbb{R}^d$
for k=1, $\cdots$, $k_{\max}$ **do**
    $\boldsymbol{r} = \xi * \boldsymbol{d}/\|\boldsymbol{d}\|_2$
    $\boldsymbol{d} \longleftarrow \nabla_r \mathfrak{L}^{\varepsilon}(\hat{f}_{\theta_t}(\boldsymbol{x}), f_{\theta_t}(\boldsymbol{x}+\boldsymbol{r}), \boldsymbol{C})/\xi$
end for
$\boldsymbol{r}^a = \rho \boldsymbol{d}/\|\boldsymbol{d}\|_2$
$L_{\text{WAR}}(\boldsymbol{x}_i, f_{\theta_t}) = \mathfrak{L}^{\varepsilon}(\hat{f}_{\theta_t}(\boldsymbol{x}), f_{\theta_t}(\boldsymbol{x}+\boldsymbol{r}^a), \boldsymbol{C})$
return $\theta_{t+1} = \theta_t - \gamma \nabla_{\theta_t} \big(L_{\text{CE}}(f_{\theta_t}(\boldsymbol{x}_i), \boldsymbol{y}_i) + \eta L_{\text{WAR}}(\boldsymbol{x}_i, f_{\theta_t}) \big)$

between classes are then defined via Euclidean distances in the embedding space, as proposed in [Frogner 2015]. Finally, as our method requires large values of the cost between similar classes, we apply the function $e^{-\delta}$ (where δ is the Euclidean distance between the two class names) element-wise and set the diagonal of $\boldsymbol{C}$ to 0. We denote it $\texttt{WAR}_{\text{w2v}}$;

- Our second ground cost computation relies on the distance between mean of embedded data. We use a pre-trained neural network (details given in experimental section) to embed data and calculate the distance between mean of each class. The rationale is that two close classes, in terms of their mean distances, should be harder to distinguish, and as such the method should allow for a complex boundary to discriminate them. We denote it $\texttt{WAR}_{\text{embed}}$.

While both options are only approximations of the difficulty to discriminate two classes, we show in the experimental section that they nonetheless provide better results than the uninformative 0-1 cost and random standard normal ground cost matrices. Note that we considered pre-calculated and static ground costs. Updating dynamically the ground cost might be an interesting research direction as future work. Other application dependent options could be designed with respect to the problem at hand. We finally note that estimating the ground cost $\boldsymbol{C}$ could be very interesting but, despite some recent progress in metric learning for OT [Cuturi 2014, Huang 2016, Li 2019], adapting them for WAR is a difficult task, and it would be prone to overfitting due to the presence of label noise in the data.

Function smoothness and ground metric. Now we discuss how the proposed regularization term regularizes the model f_θ with a smoothness controlled by the ground metric $\boldsymbol{C}$. It is not possible to extend the label smoothing in Proposition 6 to WAR because the OT cost does not admit a close form solution, but we can still show how R_{WAR} promotes label smoothness. To this end, we look at the regularization term $\mathfrak{L}^{\varepsilon}_{\text{sharp}}(\hat{f}_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x}+\boldsymbol{r}), \boldsymbol{C})$ for a given sample $\boldsymbol{x}$ and a pre-computed $\boldsymbol{r}$. We can prove the following proposition:

Proposition 7. *Minimizing R_{WAR} with a symmetric cost $\boldsymbol{C}$ such that $C_{i,i} = 0, \forall i$ is equivalent to minimizing an upper bound of a weighted total variation (TV) norm between $f_\theta(\boldsymbol{x})$ and $f_\theta(\boldsymbol{x}+\boldsymbol{r})$.*

$$\underline{c} TV(f_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x}+\boldsymbol{r})) \leq \sum_k \underline{c}_k |f_\theta(\boldsymbol{x})_k - f_\theta(\boldsymbol{x}+\boldsymbol{r})_k| \leq \mathfrak{L}^{\varepsilon}_{\text{sharp}}(f_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x}+\boldsymbol{r}), \boldsymbol{C}), \tag{5.9}$$

where $\underline{c}_k = \min_{i, i \neq k} c_{k,i}$ is the minimal off-diagonal cost for row k of $\boldsymbol{C}$ and $\underline{c} = \min_k \underline{c}_k$ is a global minimum out of the diagonal.

Proof. We aim to prove the following relations for two probability measures α and β that

$$\underline{c}TV(\alpha,\beta) \leq \sum_i \underline{c}_i |a_i - b_i| \leq \mathfrak{L}^{\varepsilon}_{\text{sharp}}(\alpha,\beta,c).$$

It is well known and obvious from [Cuturi 2013, Luise 2018] that the optimal OT matrix of regularized OT, $\Pi^{\star}_{\varepsilon}$, leads to a larger OT loss than the exact OT solution $\Pi^{\star}$. This means that

$$W_{\boldsymbol{C}}(\alpha,\beta) = \langle \Pi^{\star}, \boldsymbol{C}\rangle \leq \langle \Pi^{\star}_{\varepsilon}, \boldsymbol{C}\rangle = \mathfrak{L}^{\varepsilon}_{\text{sharp}}(\alpha,\beta,c) \tag{5.10}$$

and the relation is strict when $\varepsilon > 0$.
Now if we suppose that the cost matrix is symmetric and $C_{i,i} = 0$ and $C_{i,j} > 0$ when $i \neq j$ then solving OT means that the maximum amount of mass on the diagonal of $\Pi^{\star}$ since it leads to a 0 cost. Under constraints $U(\alpha,\beta)$ this maximum amount is equal to $\Pi^{\star}_{i,i} = \min(a_i, b_i), \forall i$. This implies that for a given row i in $\Pi^{\star}$ the amount of mass not on the diagonal row i is $\sum_{j\neq i} \Pi^{\star}_{i,j} = \max(a_i - b_i, 0)$ because of the left marginal constraint in U. Note that a similar result can be expressed with the column j such that the mass not on the diagonal of column j is $\sum_{i\neq j} \Pi^{\star}_{i,j} = \max(b_j - a_j, 0)$. This obviously means that for a given column/row index k we have $\sum_{i\neq k} \Pi^{\star}_{i,k} + \Pi^{\star}_{k,i} = |a_k - b_k|$.

Let's write $A_k = \sum_{i\geq k, j\geq k} T^{\star}_{i,j} C_{i,j}$. We have that $W_{\boldsymbol{C}}(\alpha,\beta) = A_1$. Now we remark that

$$A_k - A_{k+1} \geq \underline{c}_k |a_k - b_k|.$$

We can write that

$$A_1 = A_1 - \sum_{k=2} A_k + \sum_{k=2} A_k.$$

Since $A_N = 0$ because $C_{N,N} = 0$, it turns out that $A_1 = \sum_{k=1}(A_k - A_{k+1})$. Lower bounding every elements of the sum by the previous minoration gives that:

$$A_1 = \sum_{k=1}(A_k - A_{k+1}) \geq \sum_{k=1} \underline{c}_k |a_k - b_k| \geq \underline{c}TV(\alpha,\beta), \tag{5.11}$$

which gives the desired results. □

By minimizing the proposed R_{WAR} regularization with $\boldsymbol{r}$ belonging in a small ball around $\boldsymbol{x}$, we actually minimize a local approximation of the Lipschitz constant of f_θ. This has the effect of smoothing-out the model around $\boldsymbol{x}$ and makes it more robust to label noise. One can see the effect of the cost matrix in the center term of (5.9), where the values in the ground metric correspond to a weighting of a total variation, hence controlling the effect of the regularization and the adversarial direction $\boldsymbol{r}$ during adversarial computation. Interestingly, the Wasserstein distance can be bounded both below ($\underline{c}$) and above ($\bar{c}$) by Total Variation and weighted total variation similarly to the equation above. Finally, in practice we minimize the expectation of the OT loss, which means that we will penalize areas of high density similarly to a regularization with the Sobolev norm (i.e. penalizing the expected norm of the model gradient [Mroueh 2018]), while keeping a finer control of the class relations, since we use the ground loss $\boldsymbol{C}$ that promotes anisotropy. After demonstrating and discussing the theoretical advantages of our method, we illustrate it on a toy experiment.

Illustration of the effect of R_{WAR} We illustrate AR and WAR in a simple toy 3-classes classification problem using the Scikit-learn library [Pedregosa 2011a] with noise in Figure 5.1.

Each column of the figure corresponds to a divergence function D. The top row illustrates the values on the simplex, while the bottom row shows the classification predictions when using D as adversarial

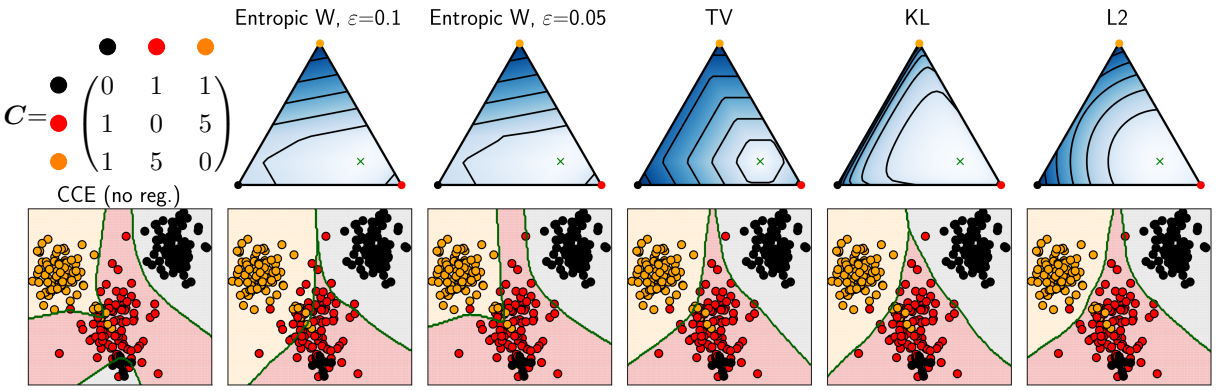


Figure 5.1: Illustration of the regularization geometry for different losses in the adversarial training. (Top) Regularization values on the simplex of class probabilities. Each corner stands for a class. All losses are computed with respect to a prediction represented as the green x. Colors are as follows: white is zero while darker is bigger. In the case of WAR, the ground cost $\boldsymbol{C}$ is given on the left. (Down) Classification boundaries when using these losses for regularization. The unregularized classifier (CCE) is given on the left.

regularization. From left to right, we compare the effect of training with the cross entropy alone (CCE, no regularization), R_{WAR} with $\varepsilon = 0.1$ and $\varepsilon = 0.05$, as well as R_{AR} with TV (that is the same as R_{WAR} with $0-1$ loss), KL (as used in VAT [Miyato 2018a]) and L2 divergences as D. For the classification problem, we generated two close classes (in orange and red), as well as a third (in black), which is far from the others. Then, we introduced noisy labels (of the black class) in the region of the red class.

On this toy example, CCE overfits the noisy black labels, yet is able to distinguish the red and orange classes. The R_{AR} regularizers, being class agnostic, correct for the noisy black labels in the bottom part, but smooth the complex decision function between the orange and red classes. On the contrary, R_{WAR} uses a different cost per pair of classes, illustrated in the top left panel of Figure 5.1: the smallest cost is set between the red and black classes, which has the effect of promoting adversarial examples in that direction. This cost / smoothing relation is due to the fact that our problem is a minimization of the OT loss: in other words, the higher the cost between the classes, the less the binary decision boundary will be smoothed. Finally, the effect of the global ε parameter can also be appreciated in the classification results: while using $\varepsilon = 0.05$, the smoothing of the loss is decreased and the final decision boundary between the mixed classes keeps all its complexity.

5.3 Numerical experiments on learning with noisy labels and openset noise

We evaluate the proposed approach WAR on both image classification and semantic segmentation tasks. We first showcase the performance of WAR on a series of image classification benchmarks (Section 5.3.1), and then consider two real-world cases: first, the classification of clothing images from online shopping websites (Section 5.3.2) and then the semantic segmentation of land use in sub-decimeter resolution aerial images (Section 5.3.3). We also evaluated our method in the context of openset noise which consists to have image whom labels are not encoded, for instance non digit images in the MNIST dataset (Section 5.3.4). Following the community evaluation strategies, we used WAR on four different architectures depending on

benchmarks. On each benchmark, WAR outperformed state-of-the-art competitors showing the robustness of our methods on the used architectures. The code can be found here*.

5.3.1 Image classification on simulated benchmark datasets

Table 5.1: Test accuracy (%) of different models on Fashion-MNIST (F-M), Cifar-10, and Cifar-100 datasets with varying noise rates (0% – 40%). The mean accuracies and standard deviations averaged over the last 10 epochs of three runs are reported, and the best results are highlighted in **bold**.

Data /	noise	CCE	Backward [Patrini 2017]	Forward [Patrini 2017]	Unhinge [Rooyen 2015]	Bootsoft [Reed 2015]	CoTeaching [Han 2018]	CoTeaching+ [Yu 2019]	D2L [Ma 2018]	SL [Wang 2019]	Pencil [Kun 2019b]	JoCoR [Wei 2020]	$\texttt{WAR}_{\text{w2v}}$ Ours
F-M	0%	94.69±0.11	94.86±0.04	94.81±0.04	**95.12 ± 0.03**	94.79±0.02	94.28±0.04	93.62±0.01	94.47±0.02	94.18±0.04	93.19 ± 0.23	94.60 ± 0.04	94.70±0.02
	20%	89.02±0.47	88.84±0.10	91.03±0.12	90.04± 0.08	88.17±0.11	91.24±0.06	92.26±0.02	89.12±0.15	93.36±0.03	92.50 ± 0.09	91.01 ± 0.08	**93.37±0.08**
	40%	78.85±0.56	81.74±0.08	82.85 ±0.2	78.32 ±0.15	73.84±0.28	86.83±0.10	86.15±0.03	78.98±0.25	86.83±0.07	90.17 ± 0.03	84.86 ± 0.20	**90.41±0.02**
Cifar-10	0%	91.76±0.04	91.63 ± 0.04	91.59 ± 0.03	**92.27 ± 0.04**	91.67 ± 0.03	90.12 ±0.04	88.47±0.14	91.29±0.02	90.48±0.05	87.90 ± 0.39	91.94 ± 0.02	91.88±0.31
	20%	85.26±0.09	84.67 ± 0.1	85.70 ± 0.08	87.09 ± 0.05	85.35 ± 0.8	86.19 ±0.07	82.97±0.25	86.64 ±0.12	87.51±0.06	87.09 ± 0.25	87.93 ± 0.09	**89.12±0.48**
	40%	76.23±0.15	73.49 ± 0.14	75.10 ± 0.15	77.94 ± 0.1	74.32 ± 0.2	80.87±0.09	72.65±0.10	73.12 ±0.43	76.87±0.12	84.48 ± 1.04	82.07 ± 0.11	**84.76±0.25**
Cifar-100	0%	68.60±0.09	69.53±0.07	70.12±0.07	70.54±0.06	69.81±0.04	65.42±0.06	58.93±0.14	**70.93 ±0.02**	68.27±0.06	63.32 ± 0.24	69.29 ± 0.26	68.16±0.18
	20%	58.81±0.10	59.23±0.08	59.54±0.05	61.06±0.06	58.97±0.08	56.55±0.08	44.88±0.14	60.90±0.03	58.41±0.08	59.05 ± 0.34	60.43 ± 0.07	**62.72±0.16**
	40%	42.45±0.12	43.02±0.09	42.17±0.1	42.87±0.07	41.73±0.08	42.73±0.08	29.94±0.34	42.61±0.04	40.97±0.12	45.70 ± 0.67	42.26 ± 0.45	**58.86±0.21**
Avg,. rank		7.7	6.9	6.2	4.2	8.1	7.1	10.1	6.7	7.0	6.2	5.0	**2.5**
noise only		8.8	8.2	7.1	5.6	10.0	5.8	9.5	7.3	6.0	3.5	5.0	**1.0**

Datasets and noisy labels simulations We consider three image classification benchmark datasets: Fashion-MNIST (F-M) [Xiao 2017], and CIFAR-10 / CIFAR-100 [Krizhevsky 2009]. Fashion-MNIST consists of 60′000 gray scale images of size 28×28 with 10 classes. CIFAR-10 and CIFAR-100 consist of 50′000 color images of size 32×32 covering 10 and 100 classes, respectively. Each dataset also contains 10′000 test images with balanced classes.

Since we want to evaluate robustness to noisy labels, we simulated label noise in the training data only. For all datasets, we introduced 0%, 20% and 40% of noise in the labels. We considered only asymmetric noise, a class-conditional label noise where each label y_i in the training set is flipped into y_j with probability $P_{i,j}$. As described above, asymmetric noise is more common in real-world scenarios than symmetric noise, where the labels are flipped uniformly over all the classes. For CIFAR-10 and CIFAR-100, we follow the asymmetric noise simulation setting by [Patrini 2017], where class labels are swapped only among similar classes with probability p (i.e. the noise level). For Fashion-MNIST, we visually inspected the similarity between classes on a t-SNE plot of the activations of the model trained on clean data; we then swapped labels between overlapping classes ($\rightarrow$: one-directional swap, $\leftrightarrow$ mutual swap): DRESS $\rightarrow$ T-SHIRT/TOP, COAT $\leftrightarrow$ SHIRT, SANDAL $\rightarrow$ SNEAKER, SHIRT $\rightarrow$ PULLOVER, ANKLE BOOT $\rightarrow$ SNEAKER.

Baselines We compared the proposed WAR with an informative $\boldsymbol{C}$ matrix based on the `word2vec` embedding ($\texttt{WAR}_{\text{w2v}}$) against several state-of-the-art methods: `Unhinged` [Rooyen 2015], `Bootstrapping` [Reed 2015], `Forward` and `Backward` loss correction [Patrini 2017], `Dimensionality driven learning (D2L)` [Ma 2018], `Co-Teaching` [Han 2018], `Co-Teaching+` [Yu 2019], `Symmetric cross entropy (SL)` [Wang 2019], `Pencil` [Kun 2019b] and finally `JoCoR` [Wei 2020]. We then compare $\texttt{WAR}_{\text{w2v}}$ with $\texttt{WAR}_{\text{embed}}$, `AR` and $\texttt{WAR}_{0-1}$. Finally, as a baseline for all the considered methods we also included a `categorical cross entropy (CCE)` loss function. All the methods shared the same architecture and training procedures, as detailed below.

*https://github.com/bbdamodaran/WAR

Table 5.2: Comparison of variants of WAR with AR with varying noise rates (0% – 40%). The mean accuracies and standard deviations averaged over the last 10 epochs of three runs are reported, and the best results are highlighted in **bold**.

Methods	Fashion-MNIST			CIFAR-10			CIFAR-100		
	0%	20%	40%	0%	20%	40%	0%	20%	40%
AR	**94.81±0.09**	93.10±0.14	89.74±0.10	91.49±0.07	88.91±0.09	81.98±0.25	67.83±0.10	**65.44±0.11**	55.75±0.14
WAR_{0-1}	94.60±0.03	90.99±0.07	86.03±0.20	90.94±0.12	86.12±0.21	74.15±0.34	65.78±0.15	60.56±0.14	51.00±0.31
WAR_{w2v}	94.70±0.02	**93.37±0.08**	**90.41±0.02**	91.88±0.31	89.12±0.48	84.76±0.25	**68.16±0.18**	62.72±0.16	**58.86±0.21**
$\text{WAR}_{\text{embed}}$	94.63±0.09	93.25±0.15	90.20±0.36	**91.88±0.12**	**89.93±0.02**	**85.08±0.32**	66.58±0.21	62.82±0.76	55.79±0.25

Model Similarly to other works [Han 2018, Miyato 2018a], we used a 9 layer CNN for the three image classification benchmark datasets: Fashion-MNIST, CIFAR10, and CIFAR100 as shown in Table 5.3. Between each layer we use a batch norm layer, a drop-out layer and a leaky-relu activation function with slope of 0.01. We use the Adam optimizer for all our networks with an initial learning rate of 0.001 with coefficient $(\beta_1, \beta_2) = (0.9, 0.999)$ and with mini-batch size of 256. The learning rate is divided by 10 after epochs 20 and 40 for Fashion-MNIST (60 epochs in total), after epochs 40 and 80 for CIFAR-10 (120 epochs in total), and after epochs 80 and 120 for CIFAR-100 (150 epochs in total). While training WAR, we set $\eta = 0$ for 15 epochs for faster convergence, as we observed that the network does not overfit on noisy labels at early stages of training. The input images are scaled between [-1, 1] for Fashion-MNIST, and mean subtracted for the CIFAR10, and CIFAR100 datasets before feeding into the network. The proposed method WAR, AR, and cross entropy loss functions are implemented in PyTorch, and for the JoCoR[†], Pencil[‡], Co-teaching[§], Co-teaching+[¶], SL[‖] method we used the PyTorch code provided by the authors. For the rest of the state-of-the-art methods (dimensionality driven learning[**], forward, backward loss correction and robust loss functions [††]: unhinged and boot strapping) the experiments are conducted using the Keras code provided by respective authors. We used similar layer initialization for all the methods in Pytorch and Keras.

For WAR_{w2v}, we set the hyper-parameters η= 10, ε= 0.05, and ρ= 0.005 for all the datasets. The hyper-parameters of the baselines are set according to their original papers. The noise transition matrix for the Forward and Backward method is estimated from the model trained with cross entropy [Patrini 2017].

Results Classification accuracies are reported in Table 5.1. Results show that WAR_{w2v} consistently outperforms the competitors by large margins, across noise levels and datasets. In particular, WAR_{w2v} achieved improvements of 2-3% points on fashion-MNIST/CIFAR-10, and 15% on CIFAR-100 at the highest noise level. This demonstrates that the inclusion of class geometric information during training mitigates the effect of over-fitting to noisy labels. On the most noisy datasets, the most competitive method with WAR_{w2v} is Pencil, however Pencil tends to decrease the performance when the dataset has a smaller percentage of noisy labels which is not the case of our method. Besides WAR_{w2v} and Pencil, JoCoR, Unhinged, Co-Teaching, and SL also performed well. The Forward and Backward method performed slightly better than CCE, which is most likely due to the burden in accurately estimating the noise transition matrix. It is noted that Co-Teaching uses true noise estimate, and the accuracy might drop if the noise

[†]https://github.com/hongxin001/JoCoR
[‡]https://github.com/yikun2019/PENCIL
[§]https://github.com/bhanML/Co-teaching
[¶]https://github.com/xingruiyu/coteaching_plus
[‖]https://github.com/YisenWang/symmetric_cross_entropy_for_noisy_labels
[**]https://github.com/xingjunm/dimensionality-driven-learning
[††]https://github.com/giorgiop/loss-correction

Fashion-MNIST	CIFAR-10	CIFAR-100
28×28×1	32×32×3	32×32 ×3
3×3 conv, 128 LReLU		
3×3 conv, 128 LReLU		
3×3 conv, 128 LReLU		
2×2 max-pool, stride 2		
dropout, p=0.25		
3×3 conv, 256 LReLU		
3×3 conv, 256 LReLU		
3×3 conv, 256 LReLU		
2×2 max-pool, stride 2		
dropout, p=0.25		
3×3 conv, 512 LReLU		
3×3 conv, 256 LReLU		
3×3 conv, 128 LReLU		
avg-pool		
dense 128 → 10	dense 128 → 10	dense 128 → 100

Table 5.3: CNN models used in our learning with noisy label experiments on Fashion-MNIST, CIFAR-10 and CIFAR-100.

ratio is estimated directly from the noisy data. Furthermore, performance of `Co-Teaching+`[‡‡] is surprising lower than the one of `Co-Teaching` on two datasets, in contrast to the observations in [Yu 2019]. From our experiments, we observed that `Co-Teaching+` underperforms when the noise is class-dependent and the model considered has a wide capacity.

Importance of encoding class relationships to better assess the significance of including class relationships and to study informative cost matrices, we compare `WAR` in three settings: 1) $\texttt{WAR}_{0-1}$: a version of `WAR` with an uninformative 0-1 ground cost (the cost matrix is a matrix of ones, except for the diagonal); 2) $\texttt{WAR}_{\texttt{w2v}}$: the version of WAR with the ground cost defined by *word2vec* embedding (as in all other experiments in this chapter); and 3) $\texttt{WAR}_{\texttt{embed}}$: a version of WAR with a ground cost defined by similarity across classes obtained by off the shelf CNNs.

$\texttt{WAR}_{\texttt{embed}}$ is computed as follows: After embedding our data with a pre-trained ResNet18 [He 2016], we compute each class centroid (which is an imperfect statistic, as belonging to one class is determined from potentially noisy labels), then we compute the distance between them. Finally, we apply the same exponential trick as for $\texttt{WAR}_{\texttt{w2v}}$. The intuition behind $\texttt{WAR}_{\texttt{embed}}$ is the following: when classes are close (similar) in a relevant embedding, the noisy labels will lead to closer class centroids. Here the pretrained neural network can be seen as generic feature extractor used to assess class similarity only, similarly to [Damodaran 2018, Damodaran 2020]. Note that for Fashion-MNIST, we computed the centroids directly in the original data space. We also compared the performance of our ground costs against 20 standard normal cost matrices drawn randomly: in all cases, our proposed ground costs were ranked at the first and second positions, showing the relevance to have a good crafted ground cost. Finally, we also compare these results against `AR` using KL divergence. We used η=5 and $\rho = 0.005$ (similarly to `WAR` approaches), and followed the `WAR` training procedure. Table 5.2 reports the performance of $\texttt{WAR}_{\text{w2v}}$, $\texttt{WAR}_{\texttt{embed}}$, $\texttt{WAR}_{0-1}$ and `AR`,

[‡‡] We used the code provided by the authors: https://github.com/xingruiyu/coteaching_plus

η	AR	WAR_{0-1}	WAR_{w2v}	$\text{WAR}_{\text{embed}}$
0.5	77.49±0.18	76.80±0.22	77.05±0.36	76.96 ± 0.15
1	77.25±0.25	76.35±0.21	76.90±0.37	77.00 ± 0.31
5	81.37±0.21	74.76±0.15	80.16±0.36	83.13 ± 0.12
10	76.84±0.84	74.14±0.16	84.76±0.25	85.08 ± 0.32
20	57.36±0.10	75.58±0.18	86.73±0.20	80.36 ± 0.22

Table 5.4: Test accuracy (in %) of adversarial regularization methods: AR, WAR_{0-1}, WAR_{w2v} and $\text{WAR}_{\text{embed}}$ with different η values on CIFAR-10 dataset with 40% noise level. The average accuracies and standard deviations over last 10 epochs are reported for one run.

and shows that WAR_{w2v} is consistently better than AR and WAR_{0-1} (except in one case), and outperformed AR significantly by a 2-3% margin at the highest noise level. When looking at the two encodings of the $\boldsymbol{C}$ matrix, $\text{WAR}_{\text{embed}}$ and WAR_{w2v} results are on par most of the time with the *word2vec* encoding showing better performances on Fashion-MNIST and CIFAR-100 and the CNN embedding being slightly better on CIFAR-10. These results show that both embeddings are senseful and provide good priors against the label noise. We found the *word2vec* embedding a good choice in our experiments, especially because they can be applied widely according to the semantics of the classes, while $\text{WAR}_{\text{embed}}$ needs access to a meaningful pre-trained model, which can be complicated depending on the problem (e.g. for semantic segmentation or other less mainstream tasks than image classification). Regarding computation time, all experiments were performed on a single GPU GTX TITAN. For one epoch on the CIFAR10 dataset, $\text{WAR}_{\text{embed}}$ took 187 seconds, AR took 165 seconds and the usual cross-entropy loss took 57 seconds. We recall that in WAR the quadratic complexity of the Wasserstein distance [Altschuler 2017] only applies to the number of classes and stays of linear complexity *w.r.t.* the number of samples, hence the effect of WAR to the computational burden is relatively small. Thus the replacement of the KL divergence by the Wasserstein distance is of the same order of complexity. To demonstrate this point, we trained WAR and AR on the imagenet32 dataset with clean labels and the 9 layer CNN network. This dataset has 1000 classes and more than 1.2M 32×32 images. On average per epoch, the AR method took 1409 seconds, while WAR method took 1712 seconds, so it is only 20% more computationally expensive.

Sensitivity analysis of η: We conducted an experimental study to analysis the sensitivity of the trade-off parameter (η) between the cross entropy and the adversarial regularization term on CIFAR-10 dataset with 40% noise level. The experimental results with different values of η are shown in Table 5.4 and the result reveals that as η increases, AR, $\text{WAR}_{\text{embed}}$ and WAR_{w2v} are robust to the label noise. However for the higher η, AR and $\text{WAR}_{\text{embed}}$ do over-smoothing and decreases the classification accuracy. On the other hand, WAR_{w2v} increases the accuracy as η increases. This behaviour shows the capability of our proposed method WAR_{w2v} to preserve the discrimination capability between similar classes. It is noted that WAR_{w2v} points the gradient direction towards the low cost classes, as a result it does not over smooths between the conflicting classes, thus maintaining the discrimination ability. Furthermore, WAR_{0-1} with uninformative ground cost did not provide better results, and it is mostly similar with different values of η. This observation reinstates need of having meaningful ground cost to capture the relationship between the classes in the dataset, and to guide gradient direction with respect to the ground cost.

In this section, we evaluated our method on standard benchmarks where we simulated the label noise ourselves. We also reviewed the proof of concept on the importance to choose a good ground cost and made a sensitivity analysis. In the next section, we evaluate our method on a real-world dataset which contains label noise.

Table 5.5: Test accuracy of different models on Clothing1M dataset with ResNet-50. (*) refers to results reproduced by us. (†) means WAR_{w2v} result at the epoch showing the best validation accuracy assessed on the clean validation set.

	Unsupervised								Using y_{val}			
Method	CCE [Wang 2019]	CCE (*)	bootsoft [Rooyen 2015]	D2L [Ma 2018]	GCE [Zhang 2018b]	SL [Wang 2019]	JoCoR (*)	WAR_{w2v} Unsup.	Forward [Patrini 2017]	JOF [Tanaka 2018b]	Pencil (*)	WAR_{w2v} (†)
Acc.	68.80	68.65	68.94	69.47	69.75	71.02	69.78	**71.61**	69.84	72.16	69.66	**72.20**

5.3.2 Image classification on real-world noisy label benchmark datasets

Dataset In this section, we demonstrate the robustness of $\texttt{WAR}_{\texttt{w2v}}$ on a large scale real-world noisy label dataset, Clothing1M [Xiao 2015]. The Clothing1M dataset contains 1 million images of clothing obtained from online shopping websites and has 14 classes. The labels have been obtained from text surrounding the images and are thus extremely noisy. The overall accuracy of the labels has been estimated to ≈61.54%. The dataset also contains additional manually refined clean data for training (50k samples), validation (14k) and testing (10k). We did not use the clean training and validation data in this work, as $\texttt{WAR}_{\texttt{w2v}}$ assumes clean labels are unavailable (which is generally a more realistic assumption). Nonetheless, we report the results obtained using the model from the epoch with the lowest validation error, calculated on the clean validation set (last column of Table 5.5) to show how close to it our unsupervised strategy can get and to compare with recent methods (e.g. [Tanaka 2018b]). Clean test data was only used to evaluate the performance of the different approaches when learning with label noise.

Experimental setup and Results Similar to [Patrini 2017, Wang 2019], we used ResNet-50 pre-trained on ImageNet for a fair comparison between methods. For $\texttt{WAR}_{\texttt{w2v}}$, the hyperparameters are similar to those of the previous experiment, except $\rho = 0.5$. Regarding the training procedure for the Clothing1M dataset we give the following details. Data pre-processing includes resizing the image to 256 x 256, center cropping a 224 x 224 patch from the resized image, and performing mean subtraction. We used a batch size of 64 and learning rate of 0.002 to update the network with Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9, and weight decay of 0.001. The learning rate is divided by 10 after 5 epochs (10 epochs in total). We set $\rho = 0.5$ and divided by 10 (after 5 epochs) when the training loss is not decreasing. For [Kun 2019b], the first stage was for three epochs with step size of $2e^{-3}$, the second for 6 epochs with step size $2e^{-4}$, and the last stage was for 6 epochs with step size $2e^{-5}$. For JoCoR [Wei 2020], we got better results by setting the learning rates as follows: we use the step size $2e^{-3}$ for the three first epochs, then we divided it by 10 for 6 epochs and finally used $2e^{-5}$ for the remaining 6 epochs.

For fairness of comparison, we only selected the methods which are similar to ours (robust loss functions), and which have the similar training methods (optimizer, learning rate, batchsize, epochs) and architectures. We compared $\texttt{WAR}_{\texttt{w2v}}$ against the competitors from [Patrini 2017, Wang 2019], and also reproduced the `CCE` accuracy by our own experiments. Results are reported in Table 5.5 and our method achieved the highest performance compared to all the baselines. Note that Forward uses a mix of noisy and clean labels to estimate the noise transition matrix, while JOF and $\text{WAR}_{\text{w2v}}(\dagger)$ use clean validation labels only to report the test accuracy with respect to the best validation score. This shows the competitive advantage of our method.

In this section we evaluated `WAR` on a real-world clothing images where it achieved state-of-the-art results. In the next section, we applied our method on semantic segmentation with real-world remote sensing data.

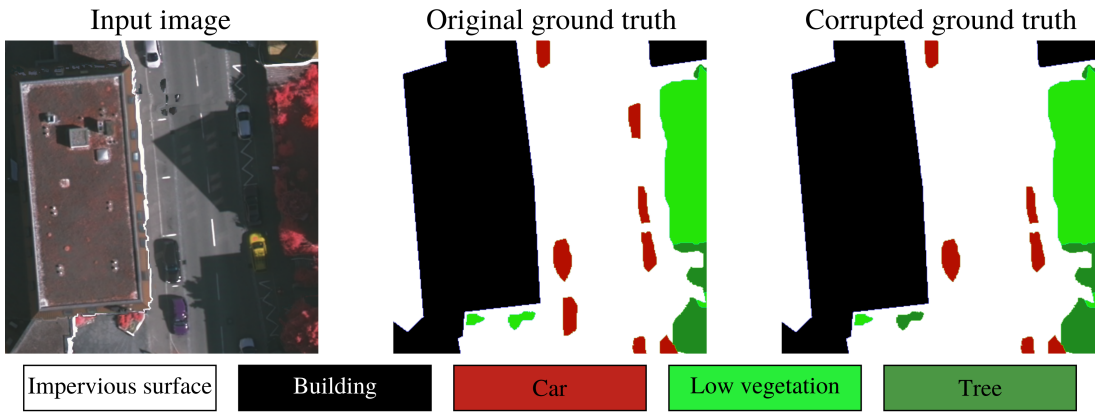


Figure 5.2: Comparison of the original and the corrupted ground truths for the semantic segmentation experiment.

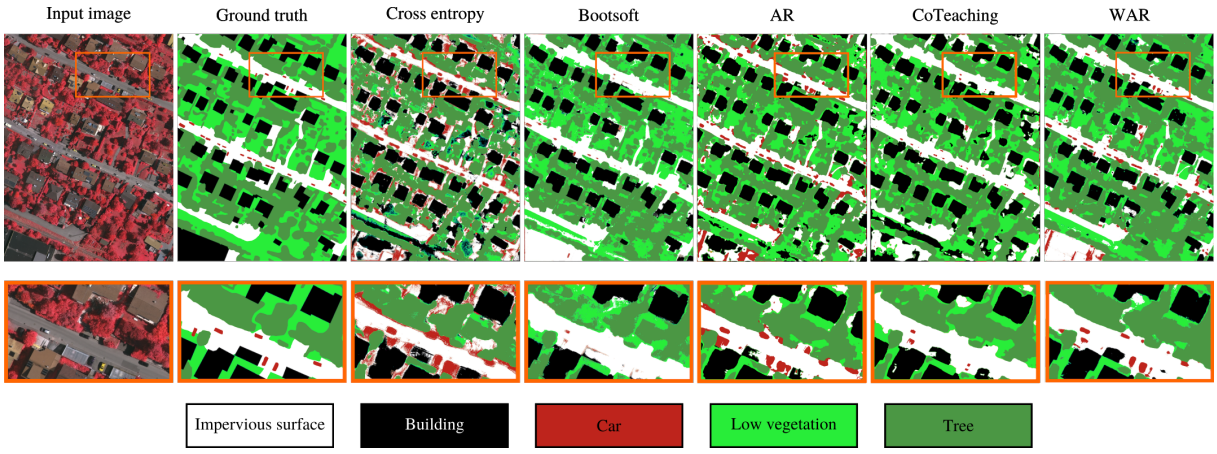


Figure 5.3: Semantic segmentation maps obtained on the test set of the ISPRS Vaihingen dataset (tile #12 of the original data). The top row shows the full image, and the second row shows a close-up of the area delineated in orange.

5.3.3 Semantic segmentation of aerial images

Datasets and noisy labels simulations In this experiment we consider the task of assigning every pixel of an aerial image to an urban land use category. We use a widely used remote sensing benchmark, the ISPRS Vaihingen semantic labeling dataset[§§]. The data consist of 33 tiles (of varying sizes, for a total of $168'287'871$ pixels) acquired by an aircraft at the ground resolution of 9cm. The images are true orthophotos with three spectral channels (near infrared, red, green). A digital surface model (DSM) and a normalized digital surface model (nDSM) are also available, making the input space 5-dimensional. Among the 33 tiles, we used the initial data split (11 tiles for training, 5 for validation and 17 for testing). As ground truth, six land cover classes (impervious surfaces, building, low vegetation, tree, car, background/clutter) are densely annotated.

We simulated label noise by swapping labels at the object level rather than flipping single pixels. An object is the connected component of pixels sharing the same label. We also focused on plausible labeling errors: for instance, a car could be mislabeled to an impervious surface, but not to a building or a tree.

[§§]http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html

Table 5.6: Per class F1 scores, average F1 score and overall accuracy (%) on the test set of Vaihingen. The best results (on the noisy dataset) are highlighted in **bold**.

Class	CCE	CCE	Bootsoft	CoTeaching	AR	$\texttt{WAR}_{\texttt{w2v}}$
Training set	Clean	Noisy				
Buildings	90.29	75.06	88.34	75.1	81.6	**89.04**
Cars	58.91	14.21	10.98	**26.6**	21.6	25.78
Imperv. surf.	85.76	62.20	**82.66**	76.9	70.9	79.01
Low vegetation	76.32	25.92	61.40	57.4	57.8	**71.56**
Trees	84.72	70.89	80.49	77.5	78.9	**82.92**
Average F1	79.20	49.65	64.77	63.6	62.2	**69.66**
Total Accuracy	83.89	63.95	**78.87**	74.6	77.1	78.43

Following this methodology, a third of the connected components had the label flipped. An example of the corrupted label data is shown in Figure 5.2.

Model We used a U-Net architecture [Ronneberger 2015], modified to take the 5 channels input data as inputs. The network was trained for 300 epochs (with 90°, 180° or 270° rotations and vertical or horizontal flips as data augmentation) using the Adam optimizer with an initial learning rate of 10^{-4} and coefficients $(\beta_1, \beta_2) = (0.9, 0.999)$. After 10 epochs, the learning rate is set to 10^{-5}. Furthermore, we predict on the full image using overlapping patches (200 pixels overlap) averaged according to a Gaussian kernel centered in the middle of the patch ($\sigma = 1$). This architecture has been shown to perform well on the semantic segmentation task in general, and on this dataset in particular [Xu 2018]. Using this methodology, we obtain an overall accuracy on the clean data of 83.89%, which is close to the state-of-the-art for this dataset.

Results We compare $\texttt{WAR}_{\texttt{w2v}}$ with standard `CCE`, `Bootsoft`, `Co-Teaching` and `AR`. The results, computed on the full test ground truth (including boundaries) and averaged over 2 runs, are reported in Table 5.6. Note that the classes are unbalanced and, for most of them, the F1-score is improved using $\texttt{WAR}_{\texttt{w2v}}$, except for the dominant class (impervious surfaces). This leads to a much higher average F1-score using $\texttt{WAR}_{\texttt{w2v}}$ (compared to its competitors), while the overall accuracy is only slightly decreased compared to `Bootsoft`. This behavior can be seen in the maps shown in Figure 5.3. We can see in the close-ups that `Bootsoft` performs poorly in detecting the cars, which are often confused with generic impervious surfaces.

We applied our method for semantic segmentation on remote sensing data. In the next section, we make a final evaluation on image classification with open set noise.

5.3.4 Image classification with open set noisy labels

As a final benchmark, following [Wang 2018c, Yu 2019] we evaluate our proposed method on realistic open set noisy labels scenarios. The open set noisy datasets are created by replacing some training images with out of domain images, while keeping the labels and number of images per class unchanged. In our experiments, we simulated open set noisy datasets following [Wang 2018c, Yu 2019] for CIFAR 10 dataset by replacing images with others coming either from the SVHN or ImageNet32 (images of size 32×32) [Oord 2016] datasets.

Table 5.7: Test accuracy on CIFAR10 dataset with 40% openset samples from SVHN and ImageNet32.

Dataset	CCE	Forward	Iterative	Co-teaching	Co-teaching+	$\texttt{WAR}_{\text{w2v}}$
CIFAR-10+ 40% SVHN	67.45	56.70	77.73	80.95	77.53	**82.03**
CIFAR-10+ 40% ImageNet32	66.69	66.77	79.38	80.34	75.47	**80.61**

CIFAR-10+SVHN	CIFAR-10+ImageNet32
32×32×3	32×32 ×3
3×3 conv, 64 ReLU	
3×3 conv, 64 ReLU	
2×2 max-pool, stride 2	
3×3 conv, 128 ReLU	
3×3 conv, 128 ReLU	
2×2 max-pool, stride 2	
3×3 conv, 196 ReLU	
3×3 conv, 16 ReLU	
2×2 max-pool, stride 2	
dense 256 → 10	

Table 5.8: CNN models used in our open set noisy label experiments on CIFAR-10 with openset noise from SVHN and ImageNet32.

Experimental set-up and Results. For the openset noise experiments, we used a 6 layers CNN with one fully connected layer following [Wang 2018c, Yu 2019] (see Table 5.8). Between each convolutional layer we used a batch normalization layer and a momentum of 0.1. The images are scaled in the range [-1, 1] and the networks are trained by SGD with learning rate 0.01, weight decay 10^{-4} and momentum 0.9. The learning rate is divided by 10 after 40 and 80 epochs (100 in total). Table 5.7 reports the classification accuracy on the CIFAR-10 dataset with 40% of openset noise. The experimental setting is similar to [Wang 2018c] in order to ensure a fair comparison with the state-of-the-art methods. The scores for the Iterative and Forward method are from [Wang 2018c], while for other competing methods we report our replicated scores. Our method outperformed the competitors on both openset noisy datasets. Our method is significantly better than all competitors for openset samples from SVHN, while for ImageNet32 openset samples we are slightly better than Co-teaching, and significantly better than remaining methods. Note that our method does not need to reject openset samples on contrary to competitors.

5.4 Conclusion

In this chapter, we introduced a regularization based on optimal transport in order to learn in a noisy label setting. Noisy labels often occur in a context of supervised learning where labels of given images do not encode their classes correctly. These noisy labels impact negatively the learning of neural networks as they have huge memorization abilities [Zhang 2017]. We show that using the VAT regularization is effective to counter-balance the label noise influence on the neural networks. However when two classes are similar, we want to modulate the regularization to keep complex boundaries. We achieved this by using optimal transport as our divergence in the regularization and it is called *Wasserstein Adversarial Training*. To encode the class similarities, we used the distance between the word2vec class representations

or the distance between embedded data of different classes. We have evaluated our methods on standard benchmarks, two real-world datasets and in an open set setting.

Chapter 6

Generating natural adversarial Remote Sensing Images

Contents

In this chapter, we present how to adapt a Wasserstein GAN in order to generate natural examples of a given pre-trained classifier. The considered setup is the following: our intent is to ***generate untargeted natural adversarial examples in a black box scenario*, which is detailed in Section 4.1.3.** We first review how GANs were used to generate remote sensing images, in what purposes and how they can be used to generate adversarial examples. We then present our main contribution: a reweighted discrete measure fed to a Wasserstein GAN to generate natural examples. The reweighted measure gives bigger weights to misclassified samples by the given classifier. We present several reweighting strategies and discuss their pros and cons. We finish this chapter by presenting the performance of our reweighted measure to generate adversarial examples on real-world remote sensing data. These contributions have been published to IEEE Transactions on Geoscience and Remote Sensing [Burnel 2021].

6.1 GANs for remote sensing images

Generative modelling for remote sensing data has been investigated in several references. GANs were used to generate such data in [Audebert 2018]. They used a conditional GAN architecture, where they take as inputs a hyperspectral class and a random noise, to control the class of generated data. They showed that the generated spectra quality is genuine-looking and physically plausible. Furthermore, they experimentally validated that the generated samples can be used for data augmentation strategy to improve classification accuracy. The generation of super resolved remote sensing data was investigated

in [Jiang 2019]. To this end; they relied on two main subnetworks combined with an adversarial learning strategy. GANs have also been used to learn unsupervised representations as done in [Lin 2017]. They used the generator to generate data-like images and the discriminator was used as a feature extractor for classification purposes. [Duan 2018] also used GANs for learning representation by incorporating a non-local layer into their architecture. GANs have also been used in remote sensing for domain adaptation on hetereogenous data [Voreiter 2020], which might be seen as transfer learning with different type of data.

In the next section we introduce our main contribution. We propose to reweight the empirical measure in order to generate adversarial data. We then discuss the different reweighting strategies and their pros and cons.

6.2 Adversarial Reweighting WGAN

We recall the notations. As we have only access to n samples $\boldsymbol{x} \in \mathcal{X}$ of an unknown measure α, we mostly deal with underlying probability measures. The measure associated to real and training data is denoted $\mathcal{P}_r \in \mathcal{M}_1^+(\mathcal{X})$ and the generated data distribution as $\mathcal{P}_G \in \mathcal{M}_1^+(\mathcal{X})$. G is a deterministic generator, classically expressed as a neural network, that takes as input a random vector $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}_q, \mathbf{I}_q)$, where q is the dimension of this random variable. We generally assume that $q \ll d$, with d the dimensionality of the training data.

6.2.1 Adversarial Reweighting WGAN

To generate data similar to the training data, we use an empirical distribution with uniform weight $\widehat{\mathbb{P}}_r$ in the classical GAN training, i.e $\widehat{\mathbb{P}}_r = 1/n \sum_i^n \delta_{\boldsymbol{x}_i}$. It is customary for empirical distributions to suppose that the samples are drawn *i.i.d.,* from the underlying distribution. One solution to change the GAN goal is to change the empirical distribution $\widehat{\mathbb{P}}_r$. Instead of encouraging the generator to generate data close to the true data, our method, called ARWGAN, encourages the generator to generate realistic data that make the classifier's prediction fail. A natural solution to reach this goal is to give a bigger weight in the distribution to true misclassified data than to those that are correctly classified. The resulting weighted distribution, denoted $\widehat{\mathbb{P}}_r^a$, must have the following form: $\widehat{\mathbb{P}}_r^a = \sum_{i=1}^n p(\boldsymbol{x}_i)\delta_{\boldsymbol{x}_i}$, where: $\sum_{i=1}^n p(\boldsymbol{x}_i) = 1$. The GAN is trained to minimize the Wasserstein distance between the generator distribution $\widehat{\mathbb{P}}_G$ and the reweighted true distribution $\widehat{\mathbb{P}}_r^a$. Finally, our new loss function is of the form:

$$\min_{\psi} \max_{\phi} \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_r^a}[\mathcal{D}_\phi(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim \mathbb{P}_z}[\mathcal{D}_\phi(G_\psi(\boldsymbol{z}))] + \lambda \mathbb{E}_{\hat{\boldsymbol{x}} \sim \widehat{\mathbb{P}}_{\hat{x}}}[(\max\{0, \|\nabla \mathcal{D}_\phi(\hat{\boldsymbol{x}})\|_2 - 1\})^2]. \tag{6.1}$$

Intuitively, the generator is a map between the latent space and the adversarial data area of the true distribution. To the best of our knowledge, it is the first time the distribution is modified for generative purpose. Now we describe and review the impact of different reweighting methods.

6.2.2 Several flavors of reweighting

Several possibilities exist to modify the empirical distribution of the true data. We give a list of the different weighting strategies for $p(\boldsymbol{x})$. We start by the considering only misclassified data.

Hard weighting

An intuitive way to generate adversarial examples is to only consider misclassified data, where a constant weight is given to misclassified data and 0 for correctly classified data. Let N_m be the number of misclassified data, then the misclassified data have normalized weights equal to $1/N_m$. The downside of this weighting strategy is that few data are left for training in the context of a very accurate classifier. This is problematic as GANs are data hungry which means that this method is not efficient to generate adversarial examples. To summary we have $p_{\text{hard}}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \text{ is correctly classified by } f_\theta \\ \frac{1}{N_m} & \text{otherwise} \end{cases}$.

Soft weighting

Another weighting approach is to use the prediction score from the classifier. Such approach could take into account examples which are correctly classified but with a low confidence. We refer to those samples as *soft adversarial examples*. In order to consider the soft adversarial examples, we can use a softmax function. It is defined as:

$$S(f_\theta^y(\boldsymbol{x}), w) \triangleq \frac{\exp(w * [f_\theta^y(\boldsymbol{x}) - f_\theta^y(\boldsymbol{x})_{\max}])}{\sum_{i=0}^{K} \exp(w * [f_\theta^y(\boldsymbol{x}_i) - f_\theta^y(\boldsymbol{x})_{\max}])}, \tag{6.2}$$

where $f_\theta^y(\boldsymbol{x})$ is the prediction to belong to the correct class, $f_\theta^y(\boldsymbol{x})_{\max}$ is the maximal probability among the batch of data to belong to the correct class and w is a temperature coefficient that controls the entropy of the resulting distribution. However as the objective is to generate adversarial data, we consider $1 - f_\theta^y(\boldsymbol{x})$ rather than $f_\theta^y(\boldsymbol{x})$. The soft weighting strategy can be expressed as $p_{\text{soft}}(\boldsymbol{x}) \triangleq S(1 - f_\theta^y(\boldsymbol{x}), w)$. Figure 1.(c) is an example of weighting using the softmax function with a temperature $w = 5$. Note that in the case of $w = 0$, we recover the original WGAN and when w tends to infinity, we recover the hard weighting strategy.

Weight clipping. In order to make the softmax strategy more robust to potential outliers in the dataset, we propose to apply a clipping function to all prediction vectors $f_\theta(\boldsymbol{x})$. We denote the threshold κ and clip the prediction vector $f_\theta(\boldsymbol{x})$ to this maximum value κ. The clipping function is applied to the prediction vectors before the softmax strategy. In our experiments, we selected a threshold of 75%. Putting all together, the weighted distribution is :

$$p_{\text{clip}}(\boldsymbol{x}) \triangleq S(\min(\kappa, (1 - f_\theta^y(\boldsymbol{x})), w). \tag{6.3}$$

Example of weights. To demonstrate the relevance of the different reweighting methods, we use a toy dataset to illustrate the softmax approach. See Figure 1. (a) to (f), where we represent data from a certain class with points cloud, the classifier's decision boundary with a black line, so that points under the line are correctly classified and points over the line are misclassified. When we get closer and beyond the classification line, the point clouds become bigger with a bigger weight. However, this might be problematic when the training data is tainted by outliers. For instance, in the presence of few misclassified data with high inaccurate predictions and a majority of misclassified data with medium inaccurate prediction, the former get a disproportionately large weight as shown in Figure 1.(e). The effect of the threshold can be seen in Figure 1.(f). where the outlier now has a similar weight to the other misclassified samples.

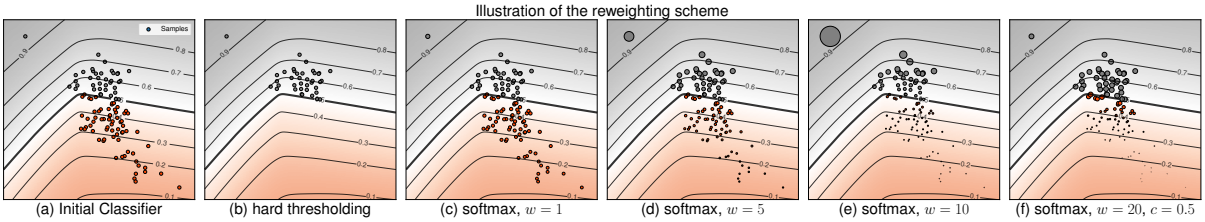


Figure 6.1: Illustration of different reweighting strategies for adversarial data generation. (a) is the standard uniform weight between data. (b) is a hard weighting where we only consider misclassified data. (c), (d) and (e) are softmax weighting strategy for different temperatures. (f) is the combination of softmax and clipping strategies.

Minibatch weighting

We numerically found that computing the reweighting directly on the full distributions gives numerical instabilities when batches with small weighted data are selected. A possible solution to this problem is to compute the reweighting strategy on batches from the distributions. In practice, we draw samples uniformly at random from the training images and make a reweighted distribution on this batch. We denote this distribution as $\mathbb{P}_r^{a\otimes m}$. The loss function becomes an expectation over mini-batches:

$$\min_{\psi}\max_{\phi}\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r^{a\otimes m}}[\mathcal{D}_\phi(\boldsymbol{x})]-\mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_z}[\mathcal{D}_\phi(G_\psi(\boldsymbol{z}))]+\lambda\mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{x}}}[(\max\{0,\|\nabla\mathcal{D}_\phi(\hat{\boldsymbol{x}})\|_2-1\})^2], \tag{6.4}$$

where m is the minibatch size. As we have detailed our different reweighting strategies, we evaluate them on different scenarios in the next section.

6.3 Numerical experiments on generative modelling

In this section we describe the different experiments that were designed to evaluate ARWGAN on simulated and real data. Section 6.3.1 shows a simple example where ARWGAN is trained to fool a classifier on synthetic 2D data. The first experiment on real-life data, provided in Section 6.3.2, aims at fooling a classifier trained on hyperspectral images. In Section 6.3.2, we study the transferability of our adversarial generators across different classifiers. In Section 6.3.3 we focus on segmentation, and train ARWGAN models to modify realistically an image in order to fool the segmentation of a given model. This is done by modifying IRRGB images through a patch in order to make them adversarial. Finally in Section 6.3.4, we measure the capacity of ARWGAN to generate data which can fool the state-of-the-art detector YOLOV3. The code can be found here*.

For all experiments we use a RMSProp optimizer [Tieleman 2012b], with a learning rate of 0.00005 for the generator and 0.0001 for the critic. Experiments were done on a single RTX 2080 Ti GPU.

6.3.1 Adversarial generator for 2D data classification

The first experiment illustrates ARWGAN against a pre-trained classifier on the well known two moons toy dataset [Pedregosa 2011b]. We consider 4000 training samples. On this toy task, we want to generate data which belong to class 1 but are classified as belonging to class 2, i.e., the white point clouds in Figure 6.2. The classifier used is a pre-trained dense 2-layer with a classification accuracy of 92%. The classifier boundary is represented as a black line. The generator and discriminator share the same dense 3 hidden

*https://github.com/PythonOT/ARWGAN

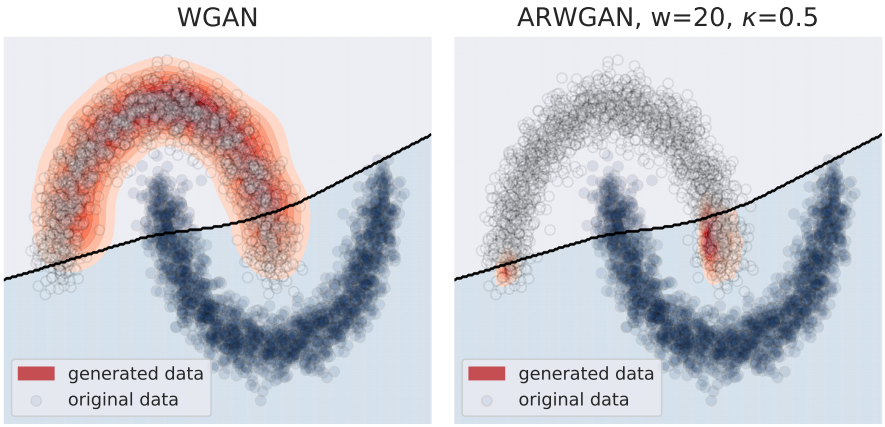


Figure 6.2: Adversarial data generation with WGAN (left) and ARWGAN (right) on two moons dataset. The black line is the classifier boundary. The kernel density estimation of generated data is in red.

layers architecture of size 128, 128 and 64 respectively. The input noise dimension $\boldsymbol{z} \sim \widehat{\mathbb{P}}_z$ is 10. The batch size is 256 and the networks is trained for 600 epochs. We then train both a WGAN and an ARWGAN to see in which zone they generate class 1 data. To visualize the generation area, 1500 samples are generated and a kernel density estimation of these data is performed. For ARWGAN, the temperature w is set to 20 and the clipping value κ is 0.5. We see that WGAN generates data all over the class 1. On the other hand, ARWGAN generates a majority of misclassified data and some correctly classified but close to the classifier boundaries. This illustrates the effectiveness of our method to focus on generating adversarial data for a pre-trained classifier. We now evaluate its performance on real-world remote sensing images.

6.3.2 Adversarial generator for hyperspectral data classification

In this subsection, we investigate the effectiveness of ARWGAN to fool a classifier trained on real-world hyperspectral data. Unlike traditional RGB images, hyperspectral imagery divides the color spectrum in multiple contiguous bands that can outreach the visible spectrum. In those images a pixel is represented by a spectrum, which can be used among others to identify the materials observed in this pixel. We consider two different hyperspectral images with two different classifiers. We first train a pixel classifier that takes a spectrum as input and outputs a material probability vector. We aim to fool the classifiers by generating spectra which are misclassified. In order to evaluate the pertinence of ARWGAN's generated data, we first check if the adversarial data are genuine-looking like the targeted class and then the percentage of correctly classified adversarial data. We evaluate in the first paragraph ARWGAN generated data against a Support vector Machine classifier trained on the Pavia University dataset. In a second paragraph, we consider a convolutional neural network on the Houston dataset. And in a third paragraph, we evaluate the transfer of adversarial examples on unseen classifiers.

Support Vector Machines on Pavia University

We first test our method against a classical SVM [Boser 1992, Hearst 1998] classifier trained on the Pavia University hyperspectral dataset. The Pavia dataset has 9 classes, composed of hyperspectral images with 103 bands of size 610×340px, (see Figure 6.3). The SVM has an accuracy score of 84% and a $\kappa = 0.79$, see the confusion matrix for more details (Figure 6.4). Performances are competitive as we consider a

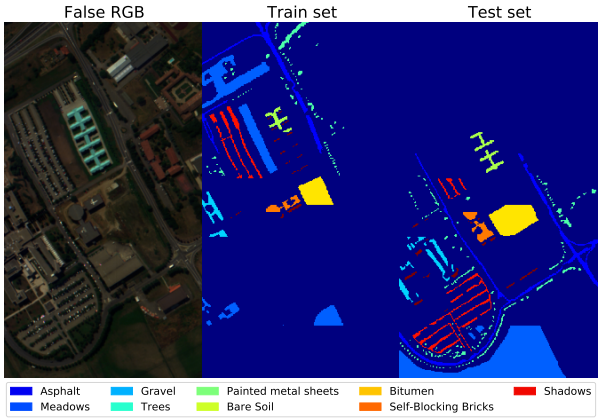


Figure 6.3: [Best viewed in color] Pavia University false RGB (left) and ground truth used for training (center) and testing (right).

spatially disjoint splitting with 1/3 of data for training, as shown in Figure 6.3.

Experimental setting. For this experiment, we focused on generating meadows spectra. We choose 1-dimensional convolution layers for both WGAN's generator and critic as they keep coherence between close spectral bands. The architecture used is detailed in Figure 6.5. We use the whole dataset to train ARWGAN (which does not require labels), with a batch size of 64, λ of 10 and w of 20. Regarding the classifier, its accuracy performance is 84% for the meadows spectra class.

Results. We first check if the generator is able to generate adversarial data that are close to the training data. We consider 1000 adversarial generated spectra. A majority is classified as Bare Soil (80%) and the remaining mostly as Trees (17%). With Figure 6.6 we present a comparison between Meadows spectra, our adversarial Meadows spectra used to reconstruct an image close to the original and the two most predicted classes: Bare Soil and Trees. Our generated spectra are very close to the targeted spectra and the two other spectra appear to be very different. This means that we successfully fool the classifier while having visually valid generated data. We also conduct a comparison with the method from [Song 2018] using the same network architecture and training procedure for comparison purpose. We reconstructed Pavia image from 1000 generated adversarial spectra for both methods in Figure 6.7. We see that while our generated adversarial spectra look like the targeted data it is not the case for generated data from [Song 2018] showing the success of our strategy.

We then check the accuracy of the pre-trained SVM classifier on the generated data. Results are gathered in Table 6.1. We can observe that 35% of ARWGAN's spectra are correctly classified while it goes up to 79% for WGAN's spectra, which confirms that ARWGAN generates a larger proportion of misclassified data.

We have conducted a sensitivity analysis of our temperature parameter w without using any ceiling and gathered the results in Table 6.2. As expected our method tends to generate more adversarial data for bigger temperature. Indeed, for a temperature $w = 1$, ARWGAN generates a few more adversarial example than WGAN, but when w is equal to 5, the generated adversarial example proportion doubles. For further understanding we show the result of using Hard weighting for the Pavia class painted metal sheets, and compare it with a soft weighting strategy. If we look at adversarial proportion for this class then the Hard weighting is able to generate 81% of adversarial example, while Soft weighting have a smaller rate with 18.6% and a WGANs have a rate of only 0.6%. However we illustrate in Figure 6.8 the limit of the Hard weighting strategy, that is overfitting with just a few modes of generated examples very

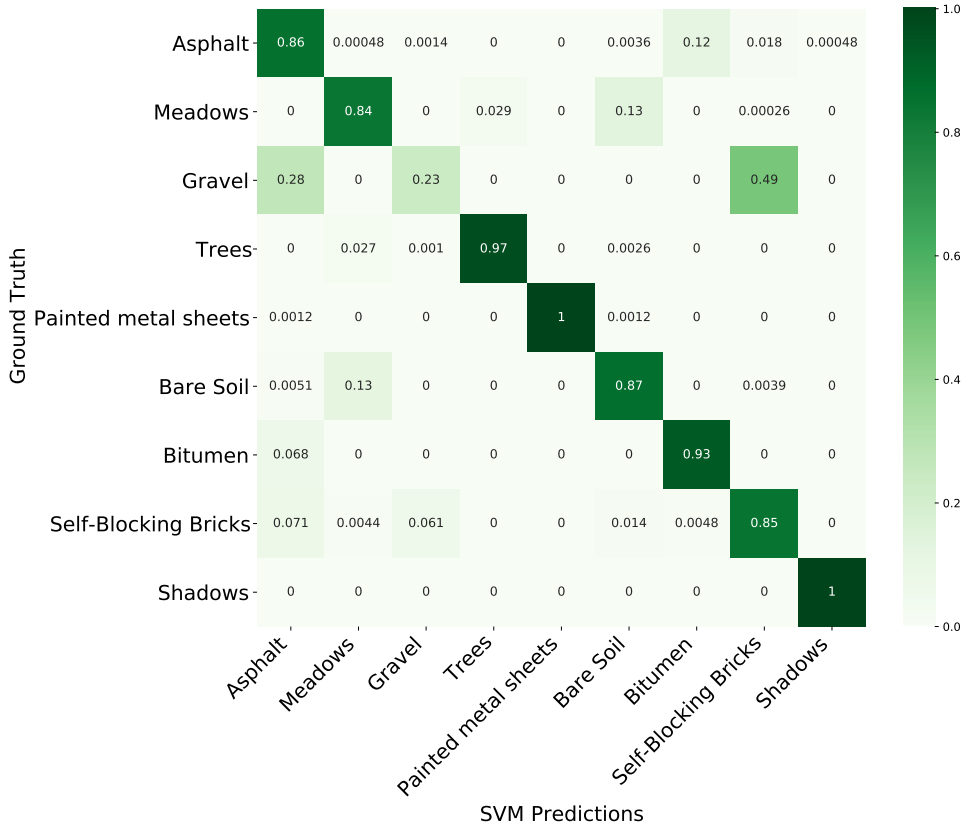


Figure 6.4: SVM Classifier confusion matrix on Pavia dataset

Generator: 226,353 parameters	Critic 23,457 parameters
INPUTS: 128×1	INPUTS: 103×1
Dense → 1664, ReLU	Conv1D k=3 stride=2, filters=16 ReLU
Reshape → 26×64	Conv1D k=3 stride=2, filters=32 ReLU
Conv1D$^\intercal$ k=3 stride=2, filters=32 ReLU	Conv1D k=3 stride=2, filters=64 ReLU
Conv1D$^\intercal$ k=3 stride=2, filters=16 ReLU	Flatten
Conv1D$^\intercal$ k=5 stride=1, 103×1 TanH	Dense → 1

Figure 6.5: Generator (Top) and critic (Bottom) 3 convolutional layer architectures to generate adversarial hyperspectral Pavia data.

similar to those in the dataset, while soft weighting is able to generate more diverse examples.

Deep neural networks on DFC2018

We now test ARWGAN on the DFC2018 dataset [grs] [Xu 2019], which was acquired over the University of Houston campus and its neighbourhoods. The hyperspectral data cover a 380-1050 nm spectral range with 48 bands at a 1-m Ground Sampling Distance. Here classes do not only cover urban classes (buildings, cars, railways, etc.) but it also covers vegetation classes (healthy or stressed grass, artificial turf, etc.). An overview of the dataset is given with Figure 6.9, where we only take 3 of the 48 bands for visualization purpose, and the ground truth labels are shown in the same figure. The considered classifier is based

Table 6.1: Classification accuracy comparison between sprectra from Full Pavia image, classical GAN reconstruction, and ARWGAN reconstruction (10 runs), lower is better

	Accuracy	Std
Real meadows	0.89	/
GAN meadows	0.79	0.004
ARWGAN meadows	**0.35**	0.01

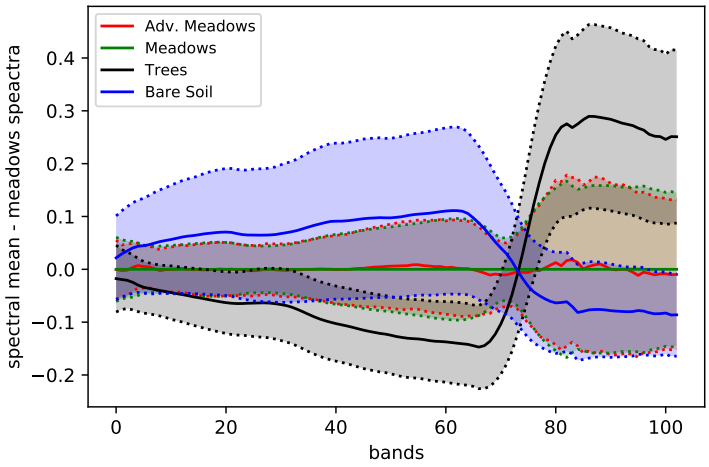


Figure 6.6: [Best viewed in color] Comparison between several class spectra means against meadows spectra from Pavia dataset. All means are reported centered around the mean spectrum of meadows for better visualization. The spectra means are denoted in plain line and the standard deviations are in dotted lines

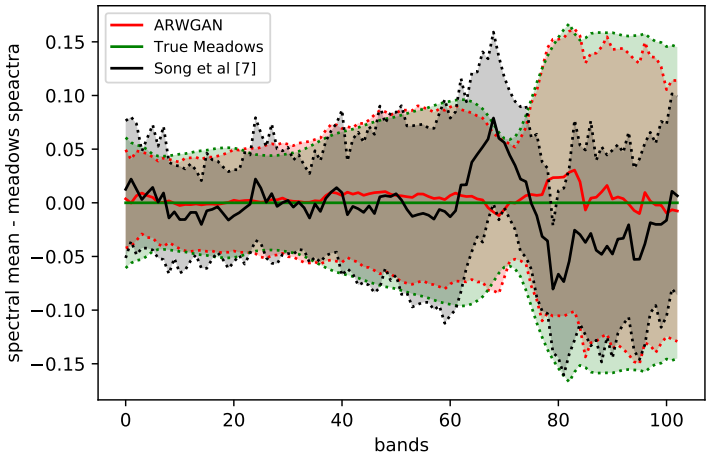


Figure 6.7: [Best viewed in color] Comparison with state-of-the-art [Song 2018] class spectra means against meadows spectra. All means are reported centered around the mean spectrum of meadows for better visualization. The spectra means are denoted in plain line and the standard deviations are in dotted lines

on a simple 1D CNN [Hu 2015], is composed of 3 CNN layers and 1 FC layer for a total of 148,373 parameters. His overall accuracy is 52% with $\kappa = 0.43$ on disjoint train/test, which is consistent with the

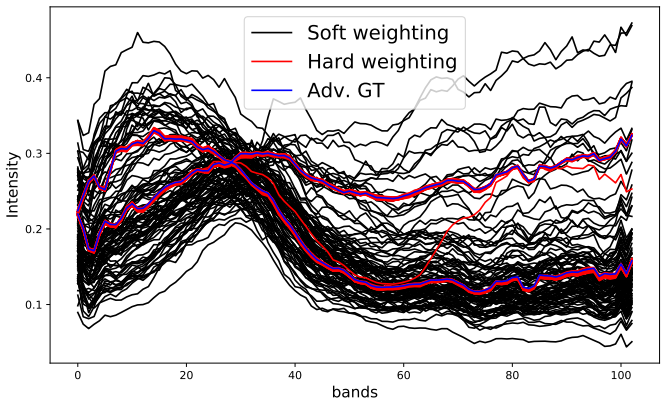


Figure 6.8: Comparison between hard and soft weighting spectra for painted metal sheets class on Pavia dataset. We plot 100 spectra for both weighting strategies.

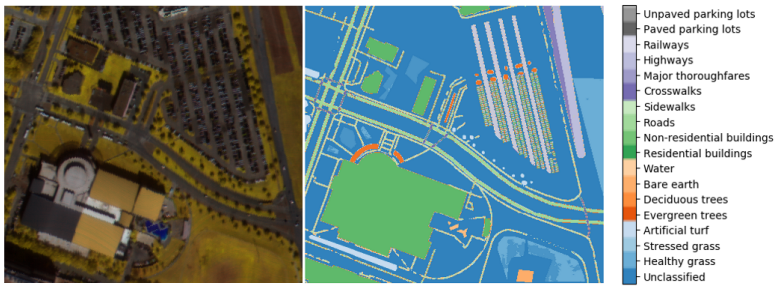


Figure 6.9: Data from the DFC2018 dataset. Example of false RGB image (left) and the ground truth (right).

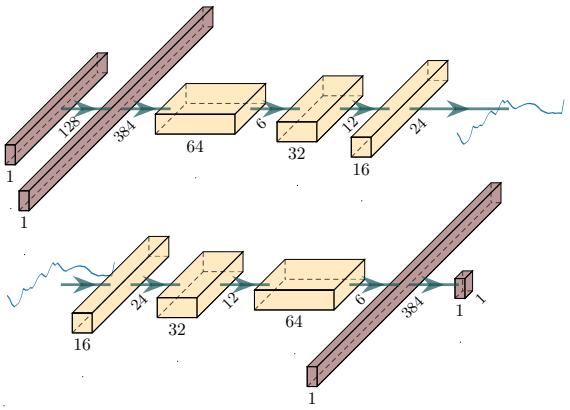
Table 6.2: Sensitivity analysis of the temperature parameter w for generated meadows from Pavia dataset

w	**0**	**1**	**5**	**10**	**20**	**35**	**50**	**100**
adv. proportion	25.5%	29.2%	50.4%	55.3%	57.7%	55.0%	61.6%	60.4%

result observed in recent reviews [Audebert 2019].

Experimental setting. For this experiment, we investigate different adversarial class spectra: adversarial healthy grass spectra, adversarial car spectra and adversarial cross-walk spectra. For the WGAN architecture, we choose 1-dimensional convolution layers for both the generator and the critic as done in the previous experiments. The architecture is detailed in Figure 6.10. We use the whole dataset with a batch size of 64, λ is set to 20 and in this experiment w is set to 20. Regarding the classifier performance, its accuracy on the whole dataset is 82% for the healthy grass spectra, 39% on the Cars class, and only 5% on the Crosswalks class. This allows us to discuss the quality of adversarial generated data according to the classifier performance on a specific class with different accuracies.

Results. We create an adversarial image by generating adversarial spectra for all the pixels of the class we attack. A visualization of the adversarial image for healthy grass can be viewed in Figure 6.11 in false colors, this is the image we refer to in this section. The pre-trained classifier has an overall accuracy of 82% on real healthy grass spectra. On adversarial generated spectra, its performance decreases by 50% as shown in the top row of Table 6.3.



Generator: 110,769 parameters	Critic: 22,561 parameters
INPUTS: 128×1	INPUTS: 48×1
Dense 128 → 384, ReLU	Conv1D k=3 stride=2, 24×16 ReLU
Reshape 384 → 6×64	Conv1D k=3 stride=2, 12×32 ReLU
Conv1Dᵀ k=3 stride=2, 12×32 ReLU	Conv1D k=3 stride=2, 6×64 ReLU
Conv1Dᵀ k=3 stride=2, 24×16 ReLU	Reshape 6×64 → 384
Conv1Dᵀ k=5 stride=2, 48×1 TanH	Dense 384 → 1

Figure 6.10: Generator (Top) and critic (Bottom) 3 convolutional layer architectures to generate adversarial hyperspectral DFC2018 data.

We now check the adversarial quality of generated spectra by comparing their means and standard deviations to the original data. Figure 6.12 gathers all spectra statistics for adversarial healthy grass generation. We compare the target class against the generated adversarial examples from ARWGAN, the generated adversarial examples from [Song 2018] and the two most predicted classes by the classifier (evergreen trees and stressed grass classes). We see that the ARWGAN adversarial spectra statistics match the targeted class better than the predicted class and state-of-the-art adversarial data generation. This confirms the competitiveness of our method to generate data similar to the training data.

We now investigate the performance of ARWGAN to generate adversarial cross-walk and cars spectra. The spectra statistics evaluation can be found in Figure 6.13. We compare the target class, ARWGAN generated data, the state-of-the-art adversarial data and the most predicted class by the classifier. We can see that ARWGAN generated data stays very similar to the target class. Regarding the classifier accuracy, we see in Table 6.3 that the classification accuracy of ARWGAN generated spectra is smaller than for real data, it decreases by 20% for the car class and 4% for the cross-walk class. We also see that there is a correlation between the classifier performance and the number of adversarial examples per batch. For high performing classifier, the number of misclassified data is really low and the GANs have few misclassified examples to be trained on.

Qualitative comparison with state-of-the-art. We now compare more deeply ARWGAN with [Song 2018][†]. We adapted their method to use the same architectures considered here. We used untargeted attacks, which means that we want our spectra to be misclassified but we do not want it to be classified as a specific label. The results are visible with Figure 6.12,6.13. Regarding the Healthy grass class, the adversarial spectrum of [Song 2018] exhibits a lot of noise and does not fit the target class as well as ARWGAN generated spectra. Similarly for the Crosswalks class, their algorithm was not able to produce

[†]following their online implementation : https://github.com/ermongroup/generative_adversary

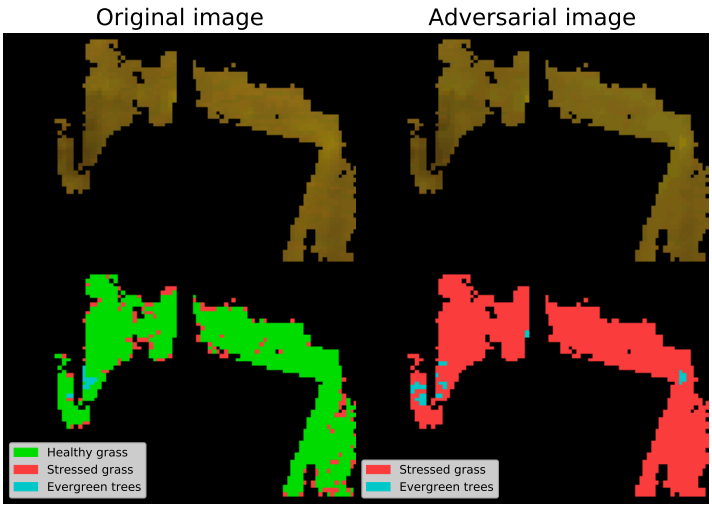


Figure 6.11: [Best viewed in color] Healthy grass visualization. (Top) False RBG image built from original spectra (left) and from adversarial spectra (right). (Bottom) Classifier prediction on the original spectra (left) and adversarial spectra (right).

Table 6.3: Classification accuracy for the pre-trained classifier over ARWGAN generated spectra and real DFC2018 data (10 runs).

	Mean	Std	Real data
Healthy Grass	0.34	0.06	0.82
Car	0.19	0.05	0.39
Crosswalks	0.01	0.009	0.05

spectra that fit the real spectra unlike ARWGAN. The method from [Song 2018], while being able to handle all classes, have greater difficulties that ARGWAN on underrepresented classes even if the classifier has a poor accuracy on these classes.

Cross-model transferability: generalizing adversarial examples to unseen classifier

In this part we consider the question of transferring generated adversarial examples for a pre-trained classifier to attack an unseen classifier. In other words, we want to measure the ability of generated adversarial examples to fool a classifier that was not used during training.

Experimental setting. For this experiment we used two different neural networks, whose details are shown in Table 6.4, and an SVM classifier. While the two neural networks are built upon the same blocks (Dense and Convolutional layers), we can see that they do not share the same performances on the Healthy Grass class, meaning that we can expect that they learned different features. Classifier A has an accuracy of 80% on the Healthy grass class while the accuracy of classifier B is only of 60% on this class. Note that classifier B has a better overall accuracy. Classifier C is a SVM with a 52% overall accuracy and a 75% accuracy on Healthy Grass class.

Results. For this experiment we look at the ratio of adversarial examples generated given classifier A, B or C. All the results are visible in Table 6.5. ARWGAN generates around 90% of adversarial examples against classifiers A, B and C when they are seen during training. From the results, we can see that there

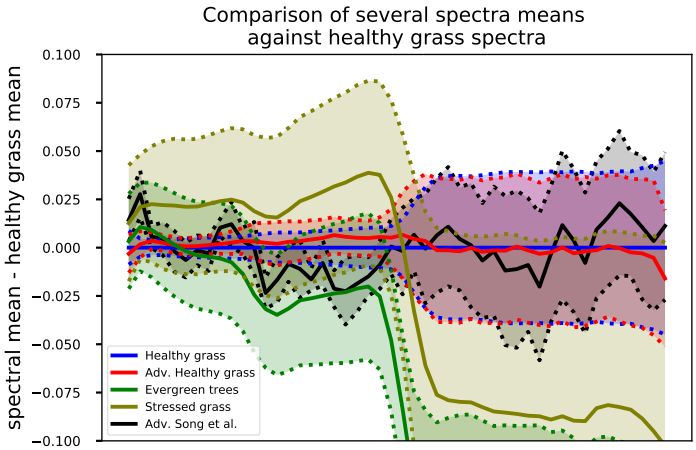


Figure 6.12: [Best viewed in color] Comparison between several class spectra means against healthy grass spectra from DFC2018 dataset. All means are reported centered around the mean spectrum of healthy grass for better visualization. The spectra means are denoted in plain line and the standard deviations are in dotted lines.

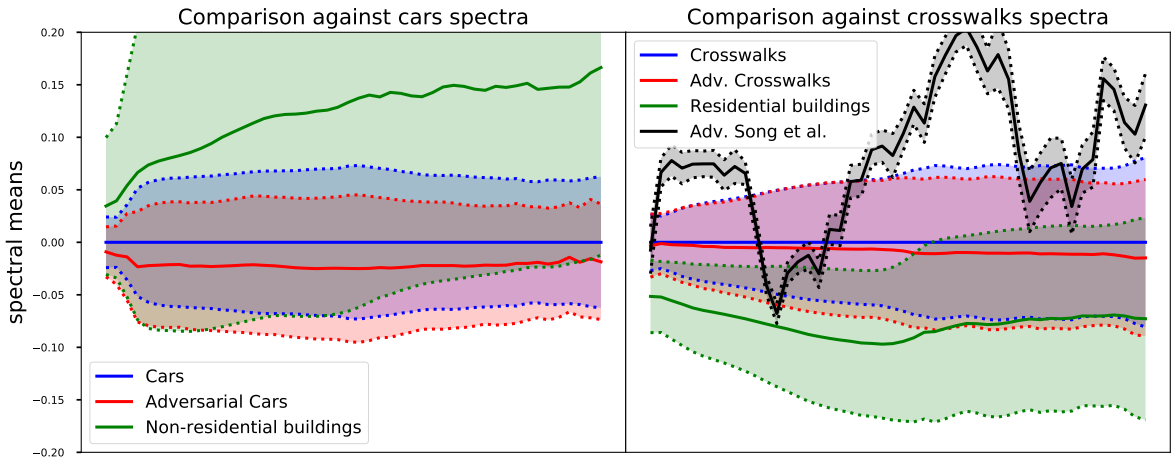


Figure 6.13: [Best viewed in color] Comparison between several class spectra means against car and cross-walk spectra. All means are reported centered around the mean spectrum of car or cross-walk for better visualization. The means are in plain and the standard deviations are in dotted lines.

is a strong transfer of adversarial examples across classifiers. In the case of generators trained against Classifier A, we generate 77.8% of adversarial examples for all classifiers. This shows that our method can be used to attack unseen classifiers.

In this section, we showed that ARWGAN can successfully generate hyperspectral data which genuinely look like training data and that are adversarial. In the next section, we demonstrate that ARWGAN can modify training images in order to make them adversarial.

6.3.3 Adversarial segmentation with modified mask images

GANs are able to generate images but they can also be used to modify existing images. In this segmentation experiment, we now focus on this last ability to modify realistically training data in order to fool the

Table 6.4: Details on the three different classifiers used to test the generalisation of our generator, while having similar overall accuracy we can see by looking at healthy grass accuracy that they do not learned the same features.

	Num. Param.	*Overall Accuracy*	*Healthy Grass Accuracy*	*Convs Layers*	*FCs Layers*
Classifier A	728,532	60%	80%	4	3
Classifier B	186,132	62%	68%	2	2
Classifier C	7,168,080	52%	75%	/	/

Table 6.5: Results for cross model transferability. Each column corresponds to a different generator trained against a classifier and each row corresponds to a test on a classifier or a combination of them. Results are the percentage of adversarial examples for 1000 generated samples (over 10 runs).

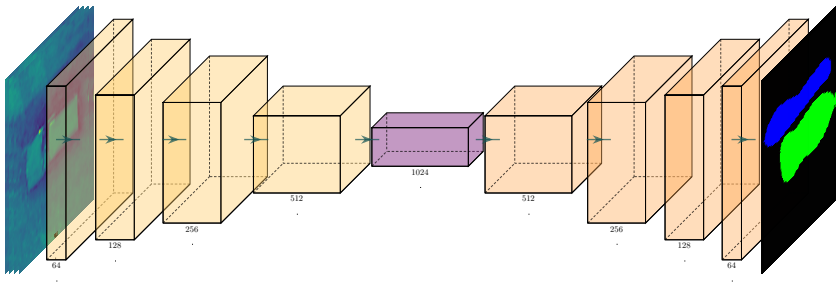
Test \ Train	Classifier A	Classifier B	Classifier C
Classifier A	89.68%, ±0.85%	78.78%, ±1.28%	73.03%, ±1.10%
Classifier B	93.68%, ±0.68%	89.32%, ±0.44%	83.39%, ±1.18%
Classifier C	84.05%, ±0.74%	81.50%, ±1.30%	90.70%, ±0.68%
A & B	88.27%, ±0.79%	77.59%, ±1.43%	71.73%, ±1.12%
A & C	81.58%, ±0.94%	76.26%, ±1.05%	79.23%, ±1.34%
B & C	78.36%, ±0.99%	68.15%, ±1.47%	69.57%, ±1.38%
all	77.80%, ±0.97%	67.52%, ±1.54%	68.71%, ±1.41%

segmentation of a pre-trained model. In order to modify an image we rely on a mask denoted $\boldsymbol{M}$, in the shape of a square at the center of an image. For an image to be both adversarial and realistic, the generator makes slight modifications inside the mask. Our images have pixels which are composed of Red-Green-Blue-InfraRed channels. To evaluate the performance of ARWGAN on this modification task, we compare the predicted segmentation of the original and the modified data and we want the original data to have a better segmentation. Furthermore we check that the modified images are still realistic by using human perceptual evaluations.

Dataset. We consider two datasets. First the Potsdam dataset [pot] which is a dataset composed of 38 6000×6000 pixels patches over the city of Potsdam, Germany. The patches are true orthophotos with four channels red, green, blue, infrared and have a GSD of 5-cm, making it possible to see clearly objects such as cars. We split the 38 images as follows: 28 images are used as training set and the remaining 10 images as test set. In total, we have 9138 cars in the dataset and from the patches, small images of cars of size 128 by 128 were extracted using center of mass of individual cars on patches. Regarding the neural network, we use a segmentation network (SegNet) [Badrinarayanan 2017] as a pre-trained model. Its architecture is provided in Figure 6.14 and its performance on real data is available as a confusion matrix in Figure 6.15.

The second dataset we considered is the Vaihigen dataset [vai]. The main differences with the Potsdam datasets are that images are RGB and have a GSD of 9-cm. We used 23 images as training set and the remaining 10 images as test set.

Experimentation. The car segmentation is done within a 64 by 64 mask in the center of the images. For the generator we used a U-Net [Ronneberger 2015] that takes as inputs both the image and the mask in addition to a latent vector $\boldsymbol{z}$, then it outputs a single new image as shown in Figure 6.16. The generator works as follows: inputs are first compressed in latent information through convolutional and pool layers, then the resulting latent vector is decoded with transposed convolutional layers to give a single new image modified only inside the mask. Our modification task can be viewed as image inpainting and following



INPUTS: 128×128×4 total parameters : 12,173,542
Conv2D k=3, 128×128×64, ReLU, MaxPool
Conv2D k=3, 64×64×128 ReLU, MaxPool
Conv2D k=3, 32×32×256 ReLU, MaxPool
Conv2D k=3, 16×16×512 ReLU, MaxPool
Conv2D k=3, 8×8×1024 ReLU, upsample
Conv2D k=3, 16×16×512 ReLU,
Conv2D k=3, 16×16×512 ReLU, upsample
Conv2D k=3, 32×32×256 ReLU,
Conv2D k=3, 32×32×256 ReLU, upsample
Conv2D k=3, 64×64×128 ReLU,
Conv2D k=3, 64×64×128 ReLU, upsample
Conv2D k=3, 128×128×64 ReLU,
Conv2D k=3, 128×128×32, ReLU
Conv2D k=3, 128×128×6, Softmax

Figure 6.14: Segmentation network for detection on Potsdam dataset

this analogy, we designed a network close to [Jo 2019] which showed impressive results using dilated convolutions [Yu 2016]. In this experiments, the modified loss function is:

$$\begin{aligned}
&\mathcal{L}_2(\mathbb{P}_r^{a\otimes m}, \mathbb{P}_z, \boldsymbol{M}, \psi, \phi, \theta) = \\
&\quad \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbb{P}_r^{a\otimes m}}\big(\mathcal{D}_\phi(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_z}[\mathcal{D}_\phi(G_\psi(\boldsymbol{x}, \boldsymbol{M}, \boldsymbol{z}))]\big) + \lambda\mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{x}}}[(\max\{0, \|\nabla\mathcal{D}_\phi(\hat{\boldsymbol{x}})\|_2 - 1\})^2] \\
&\quad + \eta_1 \sum_{i=0}^{m}((1-\boldsymbol{M}) * (G(\boldsymbol{x}, \boldsymbol{M}, \boldsymbol{z})_i - \boldsymbol{x}_i))^2 - \eta_2 \sum_{i=0}^{m}(\boldsymbol{y} * \boldsymbol{M}) \log[(1 - f_\theta(\boldsymbol{x})) * \boldsymbol{M}].
\end{aligned} \tag{6.5}$$

We add a regularization term to the generator's loss in (6.5), to ensure that only the mask $\boldsymbol{M}$ is modified by penalizing the squared difference between unmasked original image and unmasked modified image and we also add a term to favor adversarial segmentation. The latter modification of the loss function, penalizes the GAN when it modified other pixels than the car pixels. Both of these terms are weighted by constants η_1 and η_2, but taking $\eta_1 = \eta_2 = 1$ in the experiments gave meaningful results.

We now give more details about the training procedure. The critic D_ϕ is composed of 4 blocks of 3 convolutions each. Each block starts with a stride 2 convolution and each convolution has a kernel of size 3. Finally, we use a dense layer to get a scalar from the critic. We used a batch size of 32 and λ is set to 50.

Results. Some examples from the test set are gathered in Figure 6.17. We see in the first row two different cases where ARWGAN performed well. In the first case few modifications of the image led to huge difference in segmentation, the differences for the RES column are in red if the car is not segmented

Figure 6.15: Segmentation network confusion matrix on Potsdam dataset for the considered U-net.

	Original image	Modified image	Image w/o cars
Classifier Potsdam	0.762	0.481	/
Classifier Vaihingen	0.758	0.453	/
Human Potsdam	0.918	0.835	0.945
Mean confidence	2.524	2.240	2.229

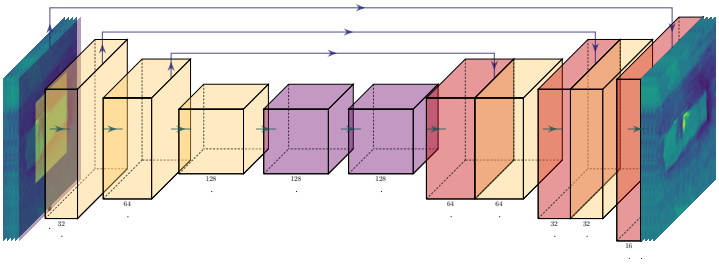
Table 6.6: Results of perceptual evaluation, we can observe that out method lower the accuracy on both dataset while retaining a good accuracy from human perception.

anymore or in green if the car is better segmented.

In the second case we needed much more modification to erase the car from the segmentation. The second row shows two failure cases: a heavy modification of the input which results in a poor image quality and a modification which leads to a better segmentation, thus failing as an adversary. Note that these failure cases do not happen often. This can be seen in the perceptual evaluation, see Table 6.18, that we designed to investigate if the generated results are convincing and adversarial. We did not add a comparison with any method as, to the best of our knowledge, there is no existing method providing region based generation of natural adversarial examples.

We illustrate in Figure 6.18 our method on Vaihingen dataset. While our method can give nice results, we have noticed an unstability during training, suggestng that it might be important to validate η_2 in some applications with long training.

Perceptual evaluation. To assess the quality of our generated patches we conduct a perceptual evaluation where we compare the ability of both trained classifier and humans to classify adversarial samples. The classifier performance is evaluated using the test set, for both original and mask modified images. And as we can see with the first row of Table 6.6, the classifier looses 28 points of accuracy when it is evaluated on data from ARWGAN. However those results are only meaningful if those are real natural adversarial example as we defined it in the first section, meaning that the generated patches are indeed



INPUTS: 128×128×6 total parameters : 7,057,318
GatedConv k=5, s=2, 64×64×32, LReLU
GatedConv k=3, s=2, 32×32×64, lrn, LReLU
GatedConv k=3, s=2, 16×16×128, lrn, LReLU
GatedConvDilat k=3, r=2, 16×16×128, lrn, LReLU
GatedConvDilat k=3, r=4, 16×16×128, lrn, LReLU
GatedDeconv k=3, s=2, 32×32×64, lrn, LReLU
GatedConv k=3, 32×32×64, lrn, LReLU
GatedDeconv k=3, s=2, 64×64×32, lrn, LReLU
GatedConv k=3, 64×64×32, lrn, LReLU
GatedDeconv k=3, s=2, 128×128×4, lrn, LReLU
GatedConv k=5, stride=1, 128×128×4, TanH

Figure 6.16: Mask modification generator architecture on Potsdam and Vaihingen datasets

cars. To this end; we conducted a perceptual evaluation similar to [Song 2018], with the assumption that if a human can detect a car then its segmentation is trivial. We conducted the perceptual evaluation by first taking randomly 50 images where we had cars and 50 images where there were no cars, and then we applied our method to the 50 images with cars, letting us with 150 images in total. We have developed a simple test where 12 images are presented to humans. Among these images, 5 are ground truth cars, 5 were modified using our method and 2 do not have cars in them. In this test you must indicate whether you see a car or not and associate this decision with a confidence level. This test was carried out by 74 persons and the results are gathered in the second and third rows of Table 6.6. Interestingly humans loose 8 points of accuracy, however their confidence remains of the same order: Moderately confident. When we compare the results, we see a 36% drop of performance for the classifier against 9% for Humans, meaning that our method affected far more the classifier than the humans and that our method produces convincing adversarial examples. After demonstrating how to use ARWGAN to modify a training example to make it adversarial, we evaluate it against a state-of-the-art detector.

6.3.4 Adversarial car images for YOLOV3 detector

The last experiment consists in evaluating ARWGAN's performance for generating cars and fooling state-of-the-art detectors. Instead of semantic segmentation where each pixel has a label, the classifier output is bounding boxes surrounding an object. In this experiment, we solely focus on the cars class object, thus the classification task can be summarized as the detection of a car in images. We evaluate the percentage of misclassified data in the generated ARWGAN and WGAN data.

Dataset. We use the same Potsdam dataset [pot] than in previous experiment. We transformed the Potsdam segmentation images to different images with bounding boxes around cars. To train the classifier we used the same train set as the patch modification experiment. However to train the GAN, we used

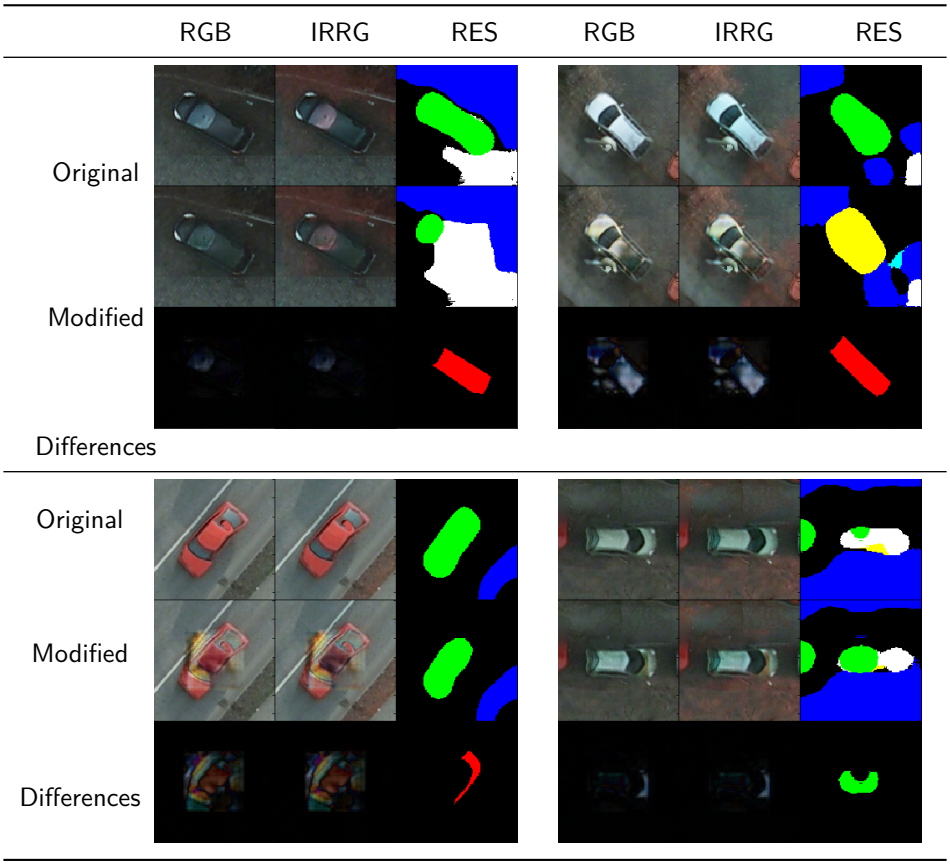


Figure 6.17: Results for patches modification, first row show examples where our method worked, second row show failure cases, and the third row is showing the difference between the two other rows, for the labels we only showed the difference with respect to ground truth of the car class.

both training and testing sets as done in our first experiment, which allows to increase the number of training data.

Experimentation. The selected detector is a YoloV3 [Redmon 2018] that achieves an overall (Train and Test) accuracy score of 99.5% with an objectness threshold of 0.75. This task seems very easy for our classifier as we do not consider the Intersection Over Union. The generator architecture we consider contains three convolutional layers and one dense layer (details in Figure 6.19). We set λ to 50. We consider two different methods, we first train a WGAN and we evaluate its ability to generate adversarial data. Then we used the WGAN generator as initialization for our method ARWGAN, and evaluate the number of generated adversarial data. The critic in this experiment is the same as the previous experiment.

Results. state-of-the-art classifiers have high accuracy and only have a few natural adversarial examples making it hard to train our method. Nevertheless, using a pre-trained WGAN generator as initialization for our generator leads to a high number of generated adversarial data with good image quality. Example of natural adversarial car images can be found in Figure 6.20. We see that our generated adversarial images have a better quality than natural adversarial images from the ground truth, called adv. GT, or WGAN generated images. As seen in Table 6.7 the considered WGAN generates only 2.6% adversarial data while our methods improve its score by more than 5 points (more than 3 times more examples are adversarial), showing that even in extreme scenarios, our method is still able to generate

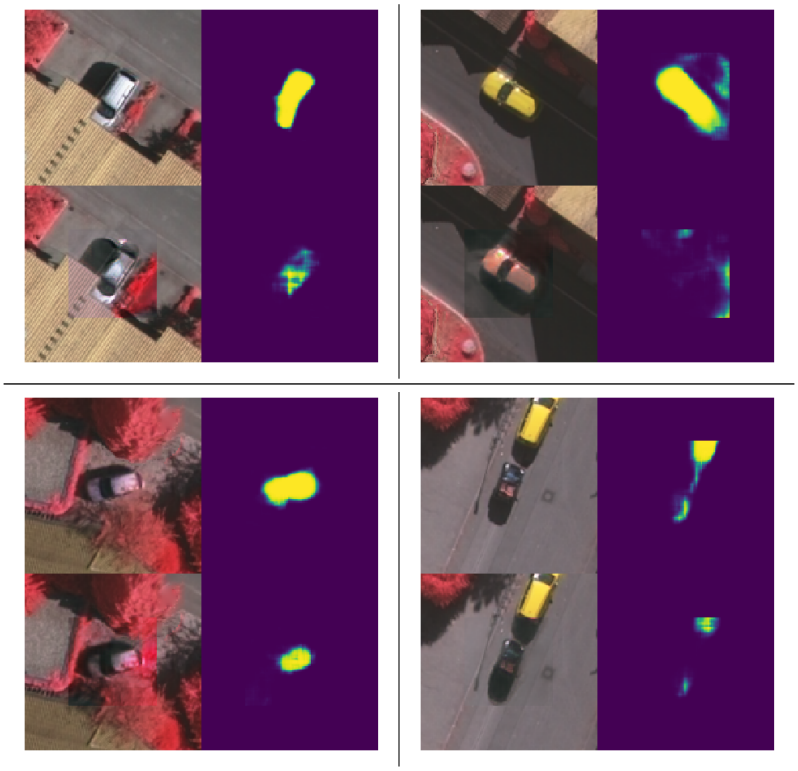
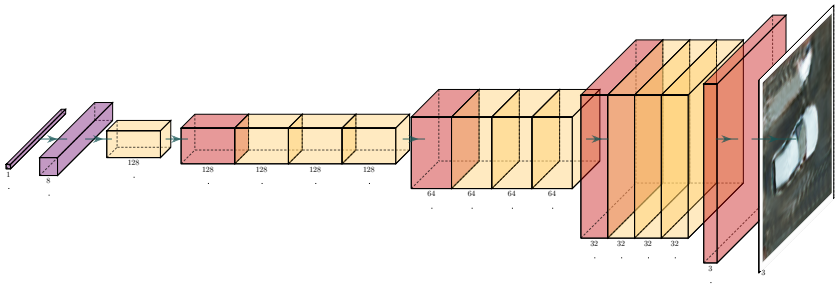


Figure 6.18: Results on Vaihingen dataset. For each example, original (top) and adversarial (bottom) images with car heatmaps. Our method reduces the classifier's probability.



INPUTS: 128×1	number of parameters : 17,048,928
Dense 128 → 8192, ReLU, reshape 8×8×128	GatedConv k=3, s=1, 32×32×64, ReLU
GatedConv k=3, s=1, 8×8×128, ReLU	GatedConv k=3, s=1, 32×32×64, ReLU
GatedDeconv k=3, s=2, 16×16×128, ReLU	GatedDeconv k=3, s=2, 64×64×32, ReLU
GatedConv k=3, s=1, 16×16×128, ReLU	GatedConv k=3, s=1, 64×64×32, ReLU
GatedConv k=3, s=1, 16×16×128, ReLU	GatedConv k=3, s=1, 64×64×32, ReLU
GatedConv k=3, s=1, 16×16×128, ReLU	GatedConv k=3, s=1, 64×64×32, ReLU
GatedDeconv k=3, s=2, 32×32×64, ReLU	GatedDeconv k=3, s=2, 128×128×3, ReLU
GatedConv k=3, s=1, 32×32×64, ReLU	GatedConv k=5, s=1, 128×128×3, ReLU

Figure 6.19: Car generator architecture trained on Potsdam dataset [pot]

adversarial data. Moreover Figure 6.20 shows that despite having few good looking natural adversarial examples in the ground truth, our method manages to generate better looking images.

	Adv. generation rate	std
WGAN	2.6%	±0.94%
Our method	7.8%	±2.5%

Table 6.7: Adversarial generation rate for our pre-trained classifier over generated data from different methods.

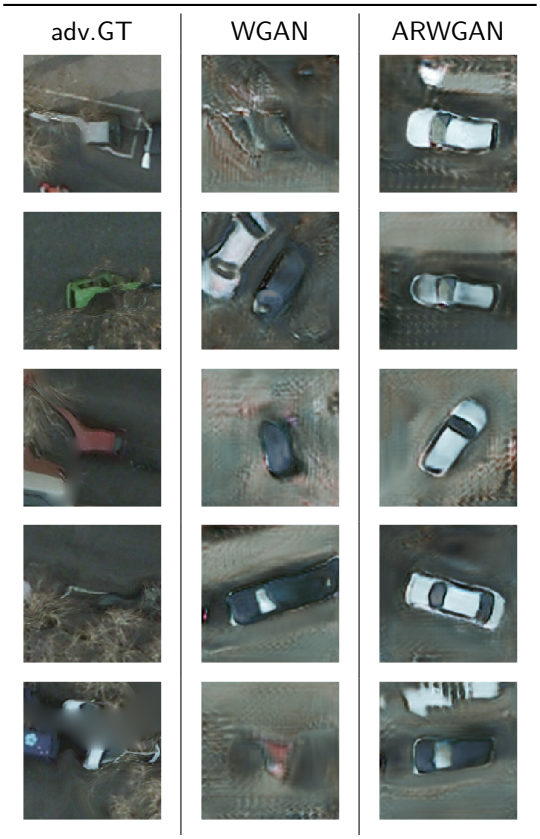


Figure 6.20: Adversarial car images for the YoloV3 detector. Left column are GT adversarial examples, middle column are WGAN adversarial examples and right column are ARWGAN adversarial examples

As expected, the method of [Song 2018] failed to generate adversarial examples on this example, as the associated auxiliary classifier was always surpassed by YOLOV3 in terms of car detection. The auxiliary classifier has to detect if there is a car in the generated examples, which in practice led YoloV3 to also detect the presence of a car.

6.4 Conclusion

In this chapter, we introduced a novel method to generate adversarial data for a given pre-trained classifier. The purpose is to reweight the initial uniform measure and to give larger weights to misclassified data. Thus the WGAN is trained with the reweighted empirical measure and is encouraged to generate misclassified samples. We proposed several reweighting strategies. After describing the limits of a hard weighting strategy, we proposed a soft strategy based on a softmax function. We then extensively evaluated our method on remote sensing data. We started by generating hyperspectral data and demonstrated the realistic quality of the generated samples. They were both misclassified and statistically close to the

targeted class. Furthermore, they were not similar to data of the predicted class. We also evaluated the transfer of adversarial examples between seen and unseen classifiers and we show that a good proportion of ARWGAN adversarial examples can attack unseen classifiers. We then proposed to modify a given training remote sensing image in order to make it adversarial with a WGAN. Using a centered mask on the center of the image, ARWGAN modified what was in the mask in order to make the segmentation fail. Finally, we trained our method to fool a state-of-the-art detector on cars.

Unfortunately, this method can not be straightforwardly adapted to other OT based GANs. The reason is that other OT based GANs rely on the computation of optimal transport between minibatches of uniform empirical measures. This approximation is the topic of the final part of this thesis.

Part III

Theory and applications of Minibatch Optimal Transport

Chapter 7

Optimal Transport statistical properties and subsampling

Contents

When optimal transport is used as a loss function to train a neural network, it has to be computed at each iteration of the optimization procedure. In a big data scenario, the number of samples n is very large. Thus, the cubical computational complexity in n of exact optimal transport makes it prohibitive in practice. To use OT in practice, we can possibly rely on some OT variants. The entropic-regularized OT reduces the cubical complexity to quadratic, which is still expensive. Other strategies were developed to reach complexities close to a linear complexity. Multi-scale OT strategy is one of them [Gerber 2017]. The multiscale decomposition yields a sequence of optimal transport problems, that are solved in a top-to-bottom fashion from the coarsest to the finest scale. It is approximately linear in time and memory in the number of nodes. Another line of research used ground costs computed between well chosen positive features, it reduces the cost of a Sinkhorn iteration to a linear computation [Scetbon 2020]. We can also build low rank transport plan as done in [Forrow 2019, Scetbon 2021].

In this chapter, we discuss another practical solution for large-scale OT. We introduce the concept of optimal transport between minibatches. We detail how it is computed and the empirical benefits of this strategy. Despite strong computational gains, a formal study of the minibatch OT formulation is lacking. In this manuscript, based on the literature regarding the statistical properties of OT and the sampling strategy, we aim at providing a theoretical study of minibatch OT. That is why, in Section 7.2, we discuss the optimal transport statistical and smoothness properties. And then, we finish this chapter by presenting the concept of U-statistics, which is at the heart of our study on minibatch optimal transport.

7.1 Large scale Optimal Transport through minibatches computation

In this section, we review how the use of minibatch with optimal transport appeared.

Empirical interests The Kantorovich-Rubinstein duality turns the original marginal equality constraints into inequality constraints, which can be hard to optimize. These inequalities are generally formulated as specific relations between the two dual potentials, known as the c-transform. In the notable case of Euclidean distance as a ground cost, however, these constraints translate more simply in constraint over the Lipschitzness of the potential. Nonetheless, practical approximations are still difficult to obtain and are still a source of active research [Arjovsky 2017, Gulrajani 2017].

Thus, we might want to compute the primal or dual OT with the given ground cost. Unfortunately, we showed in Section 2.3 that the computational complexity of optimal transport is of order $\mathcal{O}(n^3 \log(n))$ where n is the number of samples and that the memory complexity is $\mathcal{O}(n^2)$. The latter is due to the storage of the ground cost and the coupling, which are square matrices of size n. Thus in the big data scenario, practitioners relied on a common technique in deep learning, a minibatch computation of optimal transport. The idea is simple, instead of estimating the optimal transport cost between the full empirical measures, we only estimate the cost between a subset of them. It takes the form of drawing m samples at random in both the source and the target measures, associating a uniform weight to each of them, and then computing the OT cost between these samples only. This strategy gives a computational complexity of $\mathcal{O}(m^3 \log(m))$ if one uses the original OT cost and a memory cost of $\mathcal{O}(m^2)$ with $m \ll n$.

The minibatch strategy has been successful on several machine learning problems. It has been popular in generative modelling to design variants of the Wasserstein GANs. For instance, [Genevay 2018] developed a GAN based on the AutoDiff Sinkhorn Divergence. After taking two minibatches of training data and generated data, the authors used a feature extractor D_ϕ before applying the squared Euclidean distance as ground cost, *i.e.,* for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $c(\boldsymbol{x}, \boldsymbol{y}) = \|D_\phi(\boldsymbol{x}) - D_\phi(\boldsymbol{y})\|^2$. Then they optimize the feature extractor and the generator as a min max problem on minibatches. In a concurrent work, [Salimans 2018] developed a MMD loss based on a minibatch computation of the entropic-regularized OT. On another note, generative models based on the *sliced Wasserstein distance* [Kolouri 2016, Kolouri 2018, Kolouri 2019a, Liutkus 2019] fall into the minibatch framework where they compute the SWD between minibatches of input distributions.

Another popular example is domain adaptation. We recall that a fundamental technique to transfer the knowledge between domains is to use a domain alignment loss. Based on a feature extractor, the idea is to align the extracted data features of the two domains when data share the same class. The alignment strategy can be achieved by optimal transport on the joint domains of data and labels as done in [Damodaran 2018, Xu 2020a, Fatras 2021b]. The ground cost incorporates a term on the data features and on the labels. However, computing the optimal transport cost between the full distributions at each iteration of the optimization procedure is too costly. Thus practitioners successfully relied on an optimal transport minibatch computation. We nonetheless prove in Section 9 that the minibatch procedure creates negative transfer and we propose a strategy based on unbalanced optimal transport to alleviate this problem.

Minibatch OT has also been used in missing data imputation [Muzellec 2020] or to compute an online OT variant [Mensch 2020]. This approximation was successful in practice even if only subsets of input measures were considered. We now present key statistical elements of OT that we use to justify the empirical success of the minibatch strategy.

7.2 Optimal Transport statistical and smoothness properties

In this section, we discuss the statistical and smoothness properties of optimal transport which are important in the context of machine learning. We start by discussing the approximation of optimal transport between continuous measures by their empirical counter-parts. We then discuss the empirical estimator bias *w.r.t.* the continuous optimal transport. Finally, we discuss the smoothness of optimal transport *w.r.t.* the ground cost, which is at the heart of modern optimization strategy.

7.2.1 Empirical estimation of Optimal Transport

Approximation with empirical measures: the curse of dimensionality

In machine learning, the probability measures $\alpha, \beta \in \mathcal{M}_+^1(\mathcal{X})$ are unknown and we only have access to the empirical counter-parts α_n, β_n. That is why in order to estimate the optimal transport cost $\mathfrak{L}(\alpha, \beta)$, we compute directly the OT cost between the empirical measures $\mathfrak{L}(\alpha_n, \beta_n)$. This approximation is valid because our empirical estimators metricize the weakly convergence, *i.e.,* $\mathfrak{L}(\alpha_n, \beta_n) \underset{n\to+\infty}{\rightharpoonup} \mathfrak{L}(\alpha, \beta)$. Indeed the empirical distribution weakly converges towards the true distribution as the number of samples grows, $\alpha_n \underset{n\to+\infty}{\rightharpoonup} \alpha$ (resp. $\beta_n \underset{n\to+\infty}{\rightharpoonup} \beta$) and the original optimal transport cost, the Sinkhorn divergence and the unbalanced variants metrize the convergence in law [Peyré 2019, Feydy 2019, Séjourné 2019]. While the Gromov-Wasserstein distance does not metricize the weak convergence, it still verifies the property $\alpha_n \underset{n\to+\infty}{\rightharpoonup} \alpha \implies \mathcal{GW}(\alpha_n, \beta_n) \underset{n\to+\infty}{\rightharpoonup} \mathcal{GW}(\alpha, \beta)$, this is straightforward using [Mémoli 2011, Theorem 5.1 (c)] and Wasserstein distance's weak convergence. As the convergence is ensured between the two terms, we want to answer the important question of the convergence rate. This rate is often called the sample complexity. A quick convergence rate means that the quantity $\mathfrak{L}(\alpha, \beta)$ is easy to estimate empirically. We now give the sample complexity of optimal transport and the entropic-regularized OT and compare them to the sample complexity of MMD [Gretton 2012].

Theorem 7.2.1 ([Dudley 1969]). *For $\mathcal{X} = \mathbb{R}^d$ and measures supported on bounded domains, for $d > 2$ and $1 < p < +\infty$ we have*

$$\mathbb{E}(|W_p(\alpha_n, \beta_n) - W_p(\alpha, \beta)|) = \mathcal{O}(n^{-1/d}). \tag{7.1}$$

The expectation is taken with respect to the samples $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{1\leq i\leq n}$. This rate is tight in $\mathbb{R}^d$ if one of the measures has a Lebesgue density. The rate can be refined if the measures are supported on low dimension subdomains of dimension s and the rate becomes $\mathcal{O}(n^{-1/s})$ as shown in [Weed 2019]. Unfortunately, this sample complexity is really slow for high dimensional data and it is known as the *curse of dimension*. The higher the ambient dimension is, the slower the convergence of empirical estimators as we need an exponential number of samples to estimate the optimal transport cost. In comparison, MMD has a sample complexity independent from the dimension and of order $\mathcal{O}(n^{-1/2})$. This makes MMD loss easier to estimate with empirical measures.

As the Sinkhorn Divergence interpolates between optimal transport and MMD, it is a natural question to determine its sample complexity. It is provided in the following theorem:

Theorem 7.2.2 (Theorem 3, [Genevay 2019]). *Consider two measures α and β on $\mathcal{X}$ and $\mathcal{Y}$, two*

subset domains of $\mathbb{R}^d$*, with a* $\mathcal{C}^\infty$*, L-lipschitz cost c. One has:*

$$\mathbb{E}(|\mathfrak{L}^\varepsilon(\alpha_n, \beta_n) - \mathfrak{L}^\varepsilon(\alpha, \beta)|) = \mathcal{O}\left(\frac{e^{\frac{\kappa}{\varepsilon}}}{\sqrt{n}}\left(1 + \frac{1}{\varepsilon^{\lfloor d/2 \rfloor}}\right)\right), \tag{7.2}$$

where $\kappa = 2L|\mathcal{X}| + \|c\|_\infty$ *and the constants only depend on* $|\mathcal{X}|, |\mathcal{Y}|, d$ *and* $\|c^{(k)}\|_\infty$ *for* $k = 0, \cdots, [d/2]$*.* $c^{(k)}$ *denotes the k-th differential and* $|\mathcal{X}|$ *denotes the diameter of the space* $\mathcal{X}$*.*

The entropic-regularized OT's sample complexity interpolates between the optimal transport and the MMD samples complexities. Thus it can be more robust to the curse of dimensionality depending on the regularization coefficient value. Straightforwardly, the Sinkhorn divergence has a similar sample complexity as the sum of three entropic-regularized OT terms. This highlights the practical interest of using the Sinkhorn divergence when data lie in high dimension instead of non-regularized optimal transport. The Sinkhorn divergence sample complexity has been improved with weaker assumptions in [Mena 2019] such as unbounded measures. They also introduced a central limit theorem of entropic-regularized OT which can be applied to estimate the entropy of a random variable corrupted by a Gaussian noise. After reviewing the sample complexity of empirical estimators, we finish this section by stating concentration bounds of original OT and entropic-regularized OT estimators around their mean.

Concentration around the mean of Optimal Transport

One of the main questions in statistical applications is to understand the concentration of a given estimator around the unknown parameter of interest. In our case, we want to measure the deviation and the converge speed of our estimators to their expectation as the number of samples grows. A first result come from [Weed 2019, Proposition 20]:

Proposition 8. *Consider a compact space* $\mathcal{X}$*, for all* $n \geq 0$ *and* $1 \leq p < +\infty$*, we have with probability* $1 - \delta$*:*

$$|W_p^p(\alpha, \alpha_n) - \mathbb{E}W_p^p(\alpha, \alpha_n)| \leq \sqrt{\frac{1}{n}\log(\frac{2}{\delta})} \tag{7.3}$$

Note that this concentration result is only for the p-Wasserstein distance to the power p. For the entropic-regularized OT, we have:

Proposition 9. *Consider a compact space* $\mathcal{X}$ *a compact subspace of* $\mathbb{R}^d$*, a* $\mathcal{C}^\infty$ *and L-Lipschitz cost, for all* $n \geq 1$ *and* $1 \leq p < +\infty$*, we have with probability* $1 - \delta$*:*

$$|\mathfrak{L}^\varepsilon(\alpha, \alpha_n) - \mathbb{E}\mathfrak{L}^\varepsilon(\alpha, \alpha_n)| \leq \frac{\kappa_1}{\sqrt{n}} + \kappa_2\sqrt{\frac{2\log(1/\delta)}{n}}. \tag{7.4}$$

With κ_1, κ_2 *constants detailed in [Genevay 2019].*

For concentration bounds, the convergence speeds are similar between the p-Wasserstein distance to the power p and the entropic-regularized OT. These results are interesting because the concentration bounds do not depend on the ambient dimension d contrary to the sample complexities. This means that the estimator converges to its expectation at a rate of $\mathcal{O}(n^{-\frac{1}{2}})$, where n is the number of samples. In Section 8.3, we introduce concentration bounds for any optimal transport costs computed between minibatches. After reviewing the sample complexity and the concentration properties of the optimal transport estimator and its entropic-regularized counter-part, we now review its bias with respect to the loss $\mathfrak{L}(\alpha, \beta)$.

Biased empirical estimator

We saw that optimal transport suffers from the curse of dimension which penalizes its statistical properties. Unfortunately another weakness occurs when dealing with empirical measures. The optimal transport cost between empirical measures is a biased estimator of optimal transport between continuous measures, *i.e.,* $\mathbb{E}_{\alpha_n,\beta_n}(\mathfrak{L}(\alpha_n,\beta_n,C)) \neq \mathfrak{L}(\alpha,\beta,c)$. This surprising result was first discussed in [Bellemare 2017] and authors proposed a specific integral probability metric [Gretton 2012] to solve this problem. It was studied in more details in [Bińkowski 2018], where authors showed that this bias also exists for all integral probability metrics. The Sinkhorn Divergence and the entropic-regularized OT also suffer from such a bias. In this manuscript, we show that the minibatch procedure does not suffer from a biased estimator but at the price of losing the metric properties, see Section 8.2.1 for more details.

In machine learning, we deal with empirical measures and we have discussed the statistical properties of the optimal transport in this context. In the next section, we discuss another important problem of interest in machine learning, the optimization properties of optimal transport in a data fitting problem.

7.2.2 Minimizing Optimal Transport with respect to a parameter θ

In this section, we review the optimal transport optimization properties when the Wasserstein distance is used to fit a model to empirical data. Given discrete samples $(\boldsymbol{x}_i)_{i=1}^n \in \mathbb{R}^d$ from an unknown distribution α, we want to fit a parametric measure $\theta \mapsto \beta_\theta \in \mathcal{M}_1^+(\mathbb{R}^d)$ to α using an optimal transport cost and the empirical samples. Let Υ be a contrast function, we thus look for the solution of

$$\hat{\theta} = \arg\min_{\theta\in\theta} \quad \Upsilon(\alpha_n, \beta_\theta), \tag{7.5}$$

where α_n is the empirical distribution of sample $(\boldsymbol{x}_i)_{i=1}^n$. When the contrast function is chosen to be the Wasserstein distance, the above optimization problem is known as Minimal Wasserstein estimation [Bernton 2019]. Learning many generative models can also be framed as solving (7.5) [Genevay 2018]. To find the optimal $\theta^\star$ which minimizes the Kantorovich estimator, it is standard to rely on a gradient descent procedure [Genevay 2018].

To ensure that minimizing the above equation leads to the correct minimum of $\Upsilon(\alpha,\beta_\theta)$, we need for the estimator $\Upsilon(\alpha_n,\beta_\theta)$ to be an unbiased estimator of $\Upsilon(\alpha,\beta_\theta)$ with respect to the draw of the data and that we can permute the expectation on the data and the gradient with respect to θ, *i.e.,* $\mathbb{E}_{\alpha_n,\beta_n}\nabla_\theta OT(\alpha,\beta_\theta) = \nabla_\theta \mathbb{E}_{\alpha_n,\beta_n} OT(\alpha,\beta_\theta)$ [Bottou 2018, Robbins 1951b]. We saw in the previous section that the first condition is not met as the Wasserstein estimator is a biased estimator of the Wasserstein distance between continuous distributions. However, minimizing a biased estimator can still lead to good performances in practice. We now study whether the second condition is met by optimal transport or not.

Regularity with respect to the ground cost C

To prove that the exchange between the expectation and the gradient is possible, a common argument is to rely on the differentiation lemma.

Lemma 7.2.1 (Differentiation lemma)**.** *Let V be a nontrivial open set in $\mathbb{R}^p$ and let $\mathcal{P}$ be a probability distribution on $\mathbb{R}^d \times \mathbb{R}^d$. Define a map $\mathcal{F} : \mathbb{R}^d \times \mathbb{R}^d \times V \to \mathbb{R}$ with the following properties:*

- *For any $\theta \in V, \mathbb{E}_{\mathcal{P}}[|\mathcal{F}(\boldsymbol{x},\boldsymbol{y},\theta)|] < \infty$.*
- *For $\mathcal{P}$-almost all $(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}^d \times \mathbb{R}^d$, the map $V \to \mathbb{R}$, $\theta \to \mathcal{F}(\boldsymbol{x},\boldsymbol{y},\theta)$ is differentiable.*

- *There exists a $\mathcal{P}$-integrable function $\varphi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that $|\partial_\theta \mathcal{F}(\boldsymbol{x}, \boldsymbol{y}, \theta)| \leq \varphi(\boldsymbol{x}, \boldsymbol{y})$ for all $\theta \in V$.*

Then, for any $\theta \in V$, $E_{\mathcal{P}}[|\partial_\theta \mathcal{F}(\boldsymbol{x}, \boldsymbol{y}, \theta)|] < \infty$ and the function $\theta \to E_{\mathcal{P}}[\mathcal{F}(\boldsymbol{x}, \boldsymbol{y}, \theta)]$ is differentiable with differential:

$$E_{\mathcal{P}} \partial_\theta [\mathcal{F}(\boldsymbol{x}, \boldsymbol{y}, \theta)] = \partial_\theta E_{\mathcal{P}}[\mathcal{F}(\boldsymbol{x}, \boldsymbol{y}, \theta)]. \tag{7.6}$$

The proof can be found in [Klenke 2008, Theorem 6.28]. The optimal transport loss uses the parameters θ in the ground cost, formally we have $\theta \mapsto \boldsymbol{y}_\theta \mapsto C(\boldsymbol{x}, \boldsymbol{y}_\theta) \mapsto \mathfrak{L}(\alpha_n, \beta_n, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$. Thus we need to assure the differentiability of OT with respect to the the ground cost C as the probability vectors $\mathbf{a}$ and $\mathbf{b}$ remain constant. To study this regularity, we use the Danskin theorem [Bertsekas 1973, proposition B.25] which gives the regularity of a max problem. It is stated as follow:

Proposition 10 (Danskin Theorem)**.** *Let $\boldsymbol{Z} \subset \mathbb{R}^d$ be a compact set and let $\phi : \mathbb{R}^d \times \boldsymbol{Z} \mapsto \mathbb{R}$ be continuous such that $\phi(., \boldsymbol{z}) : \mathbb{R}^d \mapsto \mathbb{R}$ is convex for each $\boldsymbol{z} \in \boldsymbol{Z}$. Then*

1. *The function*

$$f(\boldsymbol{x}) = \max_{\boldsymbol{z} \in \boldsymbol{Z}} \phi(\boldsymbol{x}, \boldsymbol{z}) \tag{7.7}$$

is convex and has directional derivatives given by $f'(\boldsymbol{x}, \boldsymbol{y}) = \max_{\boldsymbol{z} \in \boldsymbol{Z}(\boldsymbol{x})} \phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y})$ where $\phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y})$ is the directional derivatives of $\phi(., \boldsymbol{z})$ at $\boldsymbol{x}$ in direction of $\boldsymbol{y}$ and $\boldsymbol{Z}(\boldsymbol{x})$ is the set of maximum points in (7.7), i.e.,

$$\boldsymbol{Z}(\boldsymbol{x}) = \{\bar{\boldsymbol{z}} \in \boldsymbol{Z} | \phi(\boldsymbol{x}, \bar{\boldsymbol{z}}) = \max_{\boldsymbol{z} \in \boldsymbol{Z}} \phi(\boldsymbol{x}, \boldsymbol{z})\}. \tag{7.8}$$

In particular, if $\boldsymbol{Z}(\boldsymbol{x})$ is a singleton and $\phi(., \bar{\boldsymbol{z}})$ is differentiable at $\boldsymbol{x}$, then f is differentiable at $\boldsymbol{x}$ and $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, \bar{\boldsymbol{z}})$.

2. *If $\phi(., \boldsymbol{z})$ is differentiable for all $\boldsymbol{z} \in \boldsymbol{Z}$, and $\nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, .)$ is continuous on $\boldsymbol{Z}$ for each $\boldsymbol{x}$, then the subdifferential of $f(\boldsymbol{x})$ is given by:*

$$\partial f(\boldsymbol{x}) = \text{conv}\{\nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}, \boldsymbol{z}) | \boldsymbol{z} \in \boldsymbol{Z}(\boldsymbol{x})\}, \forall \boldsymbol{x} \in \mathbb{R}^d, \tag{7.9}$$

where conv *indicates the convex hull operation.*

If we apply the Danskin Theorem to discrete optimal transport, the function ϕ corresponds to optimal transport, $\boldsymbol{x}$ corresponds to the ground cost C and $\boldsymbol{z}$ corresponds to the transport plan Π. The set of transport plan $\boldsymbol{U}(\alpha, \beta)$ is indeed a compact set of measures [Santambrogio 2015, Theorems 1.4, 1.5, 1.7]. Regarding the continuity of optimal transport, the Frobenius product $\langle C, \Pi \rangle$ is obviously continuous and it is linear in C and Π. As $-\phi$ is a linear function of C, it is a convex function of C. Unfortunately, original optimal transport does not always have a unique solution as illustrated in Figure 7.1, thus we only have access to subgradients. Regarding the entropic-regularized OT, the Kullback-Leibler divergence is also a continuous function as the product and sum of continuous functions (we recall that their probability vectors $\mathbf{a}$ and $\mathbf{b}$ have full support, *i.e.,* $\forall i, a_i > 0$). As the entropic-regularized OT is strongly convex, it has a unique solution. Thus it is differentiable with respect to the ground cost $\boldsymbol{C}$ contrary to original OT. So the differentiation lemma works only for entropic-regularized OT. To justify the convergence of a SGD strategy with original OT, we rely on a concept of generalized gradients developed in the next section. It allows us to exchange generalized gradients and expectations.

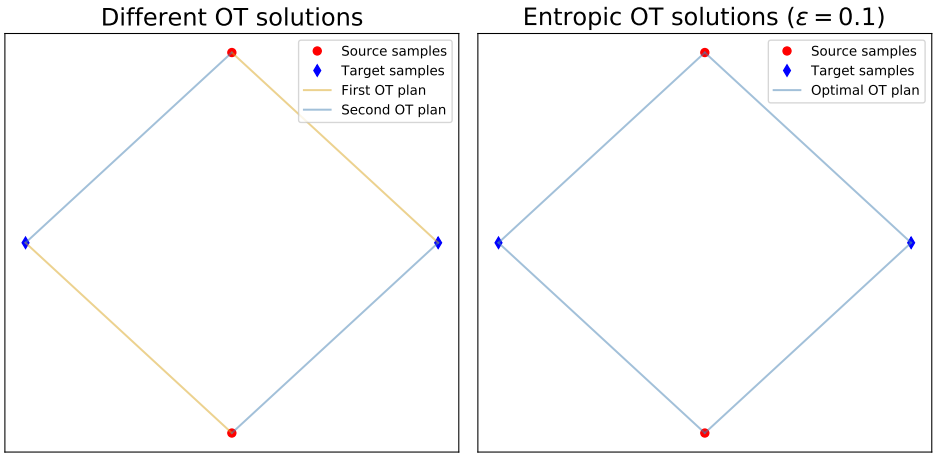


Figure 7.1: Optimal transport plan illustration between 2D measures. The left image represents the original optimal transport plans. The right image represents the unique entropic-regularized optimal transport plan.

Clarke regularity

As the original optimal transport is not differentiable, we can not apply standard first order optimization methods for smooth functions. That is why instead we introduce the Clarke regularity (see [Clarke 1990] for a full survey). A fundamental problem of nonsmooth analysis is to define a generalization of the notion of derivative, that preserves some fundamental properties of the classical notion of derivative and allows for a construction of calculus for nondifferentiable functions. Subdifferentials of convex functions are a well known example of such a generalization. While extremely useful and important in convex analysis and optimization, their obvious drawback is that they are defined only for convex functions. Clarke generalized derivatives (also called Clarke subdifferentials) are well defined for all locally Lipschitz functions. We remind the reader that a function $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is an open set in $\mathbb{R}^d$, is called locally Lipschitz at $\boldsymbol{x} \in \mathcal{X}$ if there exists an open neighbourhood U of $\boldsymbol{x}$, such that f is Lipschitz on U. Such a function is called globally Lipschitz, if it is locally Lipschitz at all points of its domain. We first introduce Clarke generalized directional derivatives:

Definition 11. *Let $f : \mathcal{X} \to \mathbb{R}$ be a locally Lipschitz function defined on an open set $\mathcal{X} \subset \mathbb{R}^d$. Clarke generalized directional derivative of f at $\boldsymbol{x}$ in direction $\boldsymbol{v}$ is denoted by $f^\circ(\boldsymbol{x}; \boldsymbol{v})$ and defined by:*

$$f^\circ(\boldsymbol{x}; \boldsymbol{v}) = \limsup_{h \downarrow 0, \boldsymbol{y} \to \boldsymbol{x}} \frac{f(\boldsymbol{y} + h\boldsymbol{v}) - f(\boldsymbol{y})}{h}.$$

For a given $\boldsymbol{x}$, the function $f^\circ(\boldsymbol{x}; \boldsymbol{v})$ is finite, positively homogeneous, and a subadditive function of $\boldsymbol{v}$. We refer to Proposition 2.1.1 [Clarke 1990] for the proof of those properties. Positive homogenity and subadditivity imply, that $f^\circ(\boldsymbol{x}; \boldsymbol{v})$ is a convex function of $\boldsymbol{v}$. Clarke's subdifferential at $\boldsymbol{x}$ is defined as the subdifferential of $f^\circ(\boldsymbol{x}; \cdot)$ at the origin.

Definition 12. *Let $f : \mathcal{X} \to \mathbb{R}$ be a locally Lipschitz function. The Clarke generalized gradient, also called Clarke subdifferential is a set valued map defined by:*

$$\partial f(\boldsymbol{x}) := \{\boldsymbol{\xi} : f^\circ(\boldsymbol{x}; \boldsymbol{v}) \geq \langle \boldsymbol{\xi}, \boldsymbol{v} \rangle \text{ for all } \boldsymbol{v} \in \mathbb{R}^d\}.$$

We use the symbol ∂f for the Clarke subdifferential, that is commonly used for subdifferentials of convex functions. This is no accident and will not be a cause of confusion, as when a convex function is

locally Lipschitz at $\boldsymbol{x}$, for example when $\boldsymbol{x}$ is in the interior of its domain, then the two notions coincide (see Proposition 2.2.7 [Clarke 1990]). In fact, Clarke generalized gradients share a number of important properties with subdifferentials of convex functions. Namely for any locally Lipsthiz function f defined on an open set X, $\partial f(\boldsymbol{x})$ is a closed, compact set for all $\boldsymbol{x} \in \mathcal{X}$ (Proposition 2.1.2(a) [Clarke 1990]) and the multifunction ∂f has a closed graph (Proposition 2.1.5(b) [Clarke 1990]).

For such defined generalization of derivative, we have a well defined calculus, including Mean Value Theorem (Theorem 2.3.7 [Clarke 1990]), chain rule (Theorem 2.3.9, Theorem 2.3.10 [Clarke 1990]) as well as formulas for generalized derivatives of sums, products and quotients. A problematic aspect of this calculus is that it usually only gives one-sided inclusions in the general case, similarly to how for two convex functions f, g, we always have the following inclusion:

$$\partial(f+g) \subseteq \partial f + \partial g,$$

while in general the reverse inclusion doesn't always hold. A classical condition on the locally Lipschitz function f under which those relations often become exact is regularity.

Definition 13. *(Definition 2.3.4, [Clarke 1990]) A locally Lipschitz function f is said to be regular at x provided that the following two conditions are satisfied:*

1. *For all $\boldsymbol{v}$, the usual one-sided directional derivative $f'(\boldsymbol{x}; \boldsymbol{v})$ exists,*
2. *For all $\boldsymbol{v}$ we have:*

$$f'(\boldsymbol{x}; \boldsymbol{v}) = f^{\circ}(\boldsymbol{x}; \boldsymbol{v}).$$

We will also say that a function f is minus regular, if $-f$ is regular. An intuitive geometric interpretation of regularity is that a function is regular if it doesn't have "upwards dashes" in its graph. For example $\boldsymbol{x} \mapsto |\boldsymbol{x}|$ is regular, since the dash is downwards, but $\boldsymbol{x} \mapsto -|\boldsymbol{x}|$ is not. Nevertheless, the function $\boldsymbol{x} \mapsto -|\boldsymbol{x}|$ behaves equally well as the absolute value function from the perspective of calculus of generalized derivatives since it is minus regular, see Remark 2.3.5 [Clarke 1990]. Finally, we present the most important result which justifies the use of Clarke gradients: the generalization of the differentiation lemma to Clarke regular functions.

Theorem 7.2.3 (Generalized gradients of integral functionals)**.** *Let $(T, \mathcal{T}, \mu)$ be a positive measurable space. Suppose that U is an open subset of a* separable *Banach space Θ, and that we are given a family of functions $f_t : U \mapsto \mathbb{R}$ satisfying the following conditions:*

- *For each $\theta \in U$, the map $t \mapsto f_t(\theta)$ is measurable.*
- *For some $k(\cdot) \in L^1(T, \mathbb{R})$, for all θ_1 and θ_2 in U, and t in T, one has $|f_t(\theta_1) - f_t(\theta_2)| \leq k(t)\|\theta_1 - \theta_2\|$.*

Write $f(\theta) = \int_T f_t(\theta)\mu(dt)$ and suppose f is defined at some point θ in U. Then f is defined and Lipschitz in U. Then we have

$$\partial f(\theta) = \partial \int_T f_t(\theta)\mu(dt) \subset \int_T \partial f_t(\theta)\mu(dt). \tag{7.10}$$

If in addition each $f_t(\cdot)$ is regular at θ, then f is regular at θ and the right hand side expression in (7.10) is an equality.

For the proofs of chain rule and generalized gradient of sum (or expectation) becoming exact for regular functions, we refer once again to [Clarke 1990]. We show that optimal transport is Clarke regular with respect to the ground cost $\boldsymbol{C}$ in Section 8.3.2.

In the next section, we finally introduce a key element of the rigorous study of minibatch OT.

7.3 U-statistics: generalized mean

In this section, we introduce the concept of U-statistics and related quantities. A U-statistic is an average of a kernel h over m-tuples taken from an n-tuple $\boldsymbol{X}$. Thus they are a good formalism to study minibatch OT that allow us to derive statistical and optimization properties. We start by defining what U-statistics are. We then review their use in machine learning and we finish by discussing their concentration bounds.

7.3.1 U-statistics definition and application in Machine Learning

In this section, we present U-statistics which are estimators of specific quantities. The theory of U-statistics can be drawn back to [Hoeffding 1948, Hoeffding 1963]. We start by giving a formal definition.

Definition 14 (Complete U-statistics)**.** *Consider two sequences of* i.i.d *random variables* $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\} \sim \alpha^{\otimes n}$ *and* $\boldsymbol{Y} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\} \sim \beta^{\otimes n}$ *($n \geq 2$) taking values in a measurable space $(T, \mathcal{T})$. Assume $h : \mathcal{X}^{\otimes m} \times \mathcal{Y}^{\otimes m} \mapsto \mathbb{R}$ is a $\mathcal{X}^{\otimes m} \times \mathcal{Y}^{\otimes m}$ measurable function, with $2 \leq m \leq n$, and also that it is a permutation symmetric map, meaning that $h(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_m) = h(\boldsymbol{x}_{\sigma_1}, \cdots, \boldsymbol{x}_{\sigma_m}, \boldsymbol{y}_{\sigma_1}, \cdots, \boldsymbol{y}_{\sigma_m})$ for any m-tuples of data and permutation σ. A two samples U-statistics of order m is defined as:*

$$U_{n,m} = \left(\frac{(n-m)!}{n!}\right)^2 \sum_{I,J \in \mathcal{P}^m} h(\boldsymbol{X}(I), \boldsymbol{Y}(J)), \tag{7.11}$$

where $\mathcal{P}^m$ is the set of m-tuples without replacement and $\boldsymbol{X}(I)$ (resp $\boldsymbol{Y}(J)$) represents the data whose indices correspond to the indices in I (resp. J).

It can be interpreted as a generalized mean over m-tuples of data. We have presented above the two samples U-statistics which results will be applied to optimal transport, but one sample and more general U-statistics can be defined. Regarding the expectation of this quantity, $U_{n,m}$ is clearly an unbiased estimator of $\mathbb{E}_{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m,\boldsymbol{y}_1,\cdots,\boldsymbol{y}_m} h(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_m)$ and one of its properties is that it has the minimum variance among all unbiased estimators. A full review of U-statistics properties can be found in [J Lee 2019]. The combinatorial number of terms makes this quantity hard to use in practice. In order to remedy to this problem, we present a sub-sample quantity called *incomplete U-statistics*. Incomplete U-statistics are defined as following.

Definition 15 (Incomplete U-statistics)**.** *Consider the notations and hypotheses from Definition 14. Pick two integers $k > 0$ and $0 < m \leq n$. Then, we define the incomplete estimator:*

$$\widetilde{U}^k_{n,m}(\mathbf{a}, \mathbf{b}) := \frac{1}{k} \sum_{(I,J) \in \mathbb{D}_k} h(\boldsymbol{X}(I), \boldsymbol{Y}(J)), \tag{7.12}$$

where $\mathbb{D}_k$ is a set of k pairs of m-tuples uniformly drawn independently.

They allow a smaller computational cost than their corresponding complete estimator. We finish the U-statistics definition by discussing a related quantity of interest called *V-statistics*. V-statistics are related to U-statistics with the notable difference that they average the kernel h over m-tuples of indices

with possibly repeated indices, *i.e.*, $I \in [\![n]\!]^m$. So they are not necessarily unbiased estimators due to the repetition of samples. They are defined as

$$V_{n,m} = \left(\frac{1}{n^m}\right)^2 \sum_{I,J \in [\![n]\!]^m} h(\boldsymbol{X}(I), \boldsymbol{Y}(J)). \tag{7.13}$$

Applications in Machine Learning

U-statistics have been used in various machine learning problems as a performance criterion. We present two examples where U-statistics are present: clustering and metric learning. More detailed examples can be found in [Clémençon 2016]. We then finish this section by an application of U-statistics for comparing probability distributions.

Clustering. Consider a distance $d : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$. Clustering is an unsupervised learning task that consists in partitioning a set of data points $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ in a feature space $\mathcal{X}$ into a finite collection of subgroups depending on their closeness with respect to d, in other words data points in the same subgroup should be closer to each other than to those lying in other subgroups. We thus want to evaluate the quality of a partition $\mathbb{P}$ of $\mathcal{X}$ with respect to the clustering of an *i.i.d.* empirical data $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ drawn from α. This quality can be assessed through the within cluster point scatter, which takes the form of a one sample U-statistic of order 2 [Clémençon 2008]:

$$L_{\text{clustering}}(\mathbb{P}) = \frac{2}{n(n-1)} \sum_{i<j} d(\boldsymbol{x}_i, \boldsymbol{x}_j). \sum_{\mathcal{C} \in \mathbb{P}} \mathbb{I}\{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}\}, \tag{7.14}$$

The U-statistic kernel is $h(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(\boldsymbol{x}_i, \boldsymbol{x}_j). \sum_{\mathcal{C} \in \mathbb{P}} \mathbb{I}\{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}\}$.

Metric Learning. Choosing the appropriate metric for a given application is a fundamental problem. For instance in domain adaptation as we detailed above, a joint metric on the extracted feature and the label spaces gave state-of-the-art results [Courty 2017b]. In the context of supervised learning, the goal of metric learning is to find a metric under which data with the same label are close to each other and those with different labels are far away [Bellet 2015, Kulis 2013b]. Consider an *i.i.d.* empirical data $(\boldsymbol{x}_1, \boldsymbol{y}_1), \cdots, (\boldsymbol{x}_n, \boldsymbol{y}_n) \in \mathcal{X}_1 = \mathcal{X} \times \{1, \cdots, j\}$ the empirical pairwise classification performance of a distance $d : \mathcal{X}_1 \times \mathcal{X}_1 \mapsto \mathbb{R}_+$ can be evaluated by:

$$L_{\text{metric}}(d) = \frac{6}{n(n-1)(n-2)} \sum_{i<j<k} \mathbb{I}\{d(\boldsymbol{x}_i, \boldsymbol{x}_j) < d(\boldsymbol{x}_i, \boldsymbol{x}_k), \boldsymbol{y}_i = \boldsymbol{y}_j \neq \boldsymbol{y}_k\}, \tag{7.15}$$

which is a one sample U-statistic of degree 3 with a kernel $h((\boldsymbol{x}, \boldsymbol{y}), (\boldsymbol{x}', \boldsymbol{y}'), (\boldsymbol{x}^\star, \boldsymbol{y}^\star)) = \mathbb{I}\{d(\boldsymbol{x}, \boldsymbol{x}') < d(\boldsymbol{x}, \boldsymbol{x}^\star), \boldsymbol{y} = \boldsymbol{y}' \neq \boldsymbol{y}^\star\}$.

U-statistics as divergence between probability measures. We recall that a MMD loss with a positive kernel k is defined as:

$$\|\alpha - \beta\|_k^2 = \mathbb{E}_{\alpha \otimes \alpha}\left[k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right] + \mathbb{E}_{\beta \otimes \beta}\left[k\left(\boldsymbol{y}, \boldsymbol{y}'\right)\right] - 2\mathbb{E}_{\alpha \otimes \beta}[k(\boldsymbol{x}, \boldsymbol{y})]. \tag{7.16}$$

These above expectations can be estimated using empirical samples $\boldsymbol{X} \sim \alpha^{\otimes n}, \boldsymbol{Y} \sim \beta^{\otimes n'}$ and with the estimator

$$U^2_{\text{MMD}}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{n(n-1)} \sum_{i=1, i \neq j}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{n'(n'-1)} \sum_{i=1, i \neq j}^{n'} k(\boldsymbol{y}_i, \boldsymbol{y}_j) - \frac{2}{nn'} \sum_{i=1}^{n} \sum_{j=1}^{n'} k(\boldsymbol{x}_i, \boldsymbol{y}_j), \tag{7.17}$$

which is composed of two one sample U-statistics of degree 2 and one sample average. When we have the same number of empirical data n, we can estimate it with the estimator

$$U^2_{\mathrm{MMD}}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{n(n-1)} \sum_{i \neq j}^{n} h(\boldsymbol{z}_i, \boldsymbol{z}_j), \tag{7.18}$$

which is a one sample U-statistic and where the U-statistic kernel is equal to $h(\boldsymbol{z}_i, \boldsymbol{z}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) + k(\boldsymbol{y}_i, \boldsymbol{y}_j) + k(\boldsymbol{x}_i, \boldsymbol{y}_j) + k(\boldsymbol{x}_j, \boldsymbol{y}_i)$. We can also estimate the MMD loss using V-statistics, which leads to a biased loss:

$$V^2_{\mathrm{MMD}}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{n(n-1)} \sum_{i,j=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{n'(n'-1)} \sum_{i,j=1}^{n'} k(\boldsymbol{y}_i, \boldsymbol{y}_j) - \frac{2}{nn'} \sum_{i=1}^{n} \sum_{j=1}^{n'} k(\boldsymbol{x}_i, \boldsymbol{y}_j), \tag{7.19}$$

Note that in the V-statistics, the index i can be equal to the index j. Intuitively we expect the empirical test statistic, whether biased or unbiased, to be small if $\alpha = \beta$, and large if the distributions are far apart. U-statistics have appealing concentration bounds and thus, their use is an important element for the statistical study of MMD losses. We present the main U-statistics concentration results in the next section.

7.3.2 U-statistics concentration bounds

In the case of U-statistics, we want to aim at estimating the concentration of $U_{n,m}$ around $\mathbb{E}U_{n,m}$. It is the goal of this section to give the main standard result. We start by recalling the famous Hoeffding lemma:

Lemma 7.3.1 (Hoeffding's Lemma)**.** *Let the real random variable $x \in [a, b]$ and denote $\mathbb{E}x = \mu$. Then for all $s \in \mathbb{R}$:*

$$\mathbb{E}\left[e^{s(x-\mu)}\right] \leq e^{s^2(b-a)^2/8}. \tag{7.20}$$

The proof can be found in [Hoeffding 1963]. This quantity gives us an upper bound on the moment-generating function of any bounded random variable minus its mean value. The goal is now to apply this lemma to U-statistics to get an Hoeffding inequality type bound and by doing so, we get the following result:

Lemma 7.3.2. *Let $\delta \in [0, 1]$ and $m \geq 1$ be fixed. Suppose that the kernel h is bounded,* i.e., *$0 \leq \|h\|_\infty \leq M_h$. We have a concentration bound between $U_{n,m}$ and $\mathbb{E}U_{n,m}$ depending on the number of empirical* i.i.d. *data $\boldsymbol{X} \sim \alpha^{\otimes n}$ and $\boldsymbol{Y} \sim \beta^{\otimes n}$, with probability $1 - \delta$:*

$$|U_{n,m} - \mathbb{E}U_{n,m}| \leq M_h \sqrt{\frac{\log(2/\delta)}{2\lfloor n/m \rfloor}} \tag{7.21}$$

Proof. We fix $r = \lfloor n/m \rfloor$. Let $0 \leq k \leq r-1$, we define the set $I^k := \{km + 1, \cdots, km + m\}$. Then we define the function ζ as :

$$\zeta(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) = \frac{1}{r} \sum_{k=0}^{r-1} \zeta_k(\boldsymbol{X}(I^k), \boldsymbol{Y}(I^k)) = \frac{1}{r} \sum_{k=0}^{r-1} h(\boldsymbol{X}(I^k), \boldsymbol{Y}(I^k)). \tag{7.22}$$

We can see that $\zeta(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n)$ is a sum of independent random variables. In the summation below, σ_x or σ_y denotes a generic permutation of $\{1, \cdots, n\}$. We compute :

$$r \sum_{\sigma_x, \sigma_y} \zeta(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}) = r(n-m)!^2 \sum_{I,J \in \mathcal{I}} h(\boldsymbol{X}(I), \boldsymbol{Y}(J)) \tag{7.23}$$

$$= r(n-m)!^2 \Big(\frac{n!}{(n-m)!}\Big)^2 U_{n,m} \tag{7.24}$$

Thus we have $U_{n,m} = \left(\frac{1}{n!}\right)^2 \sum_{\sigma_x, \sigma_y} \zeta(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)})$, which is a sum of independent random variables. Thus, we now highlight the quantity $U_{n,m} - \mathbb{E}U_{n,m}$ for the different sums we have. As our data are *i.i.d.*, we get:

$$U_{n,m} - \mathbb{E}U_{n,m} = \frac{1}{n!^2} \sum_{\sigma_x, \sigma_y} \zeta(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}) - \mathbb{E}\zeta(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)})$$

$$= \frac{1}{n!^2} \sum_{\sigma_x, \sigma_y} \frac{1}{r} \sum_{k=0}^{r-1} \big(\zeta_k(\boldsymbol{X}(I^k_{\sigma_x}), \boldsymbol{Y}(I^k_{\sigma_y})) - \mathbb{E}\zeta_k(\boldsymbol{X}(I^k_{\sigma_x}), \boldsymbol{Y}(I^k_{\sigma_y}))\big).$$

Where $I^k_{\sigma_x}$ (resp. $I^k_{\sigma_y}$) represents the I^k indices after applying the permutation σ_x (resp. σ_y). In what follows, we abbreviate the summation of ζ over the permutations σ_x, σ_y as $\sum_\sigma \zeta_\sigma$. Thus, for $t > 0$, we have:

$$P(U_{n,m} - \mathbb{E}U_{n,m} \geq t) \leq e^{-\lambda t} \mathbb{E}(e^{\lambda(U_{n,m} - \mathbb{E}U_{n,m})})$$

$$= e^{-\lambda t} \mathbb{E}\left[e^{\lambda \frac{1}{n!^2} \sum_\sigma \zeta_\sigma - \mathbb{E}\zeta_\sigma}\right] \leq e^{-\lambda t} \frac{1}{n!^2} \sum_\sigma \mathbb{E}\left[e^{\lambda \zeta_\sigma - \mathbb{E}\zeta_\sigma}\right] \leq e^{-\lambda t} \max_\sigma \mathbb{E}\left[e^{\lambda \zeta_\sigma - \mathbb{E}\zeta_\sigma}\right],$$

where in the first and second inequalities, we used Markov's and Jensen's inequalities respectively. Furthermore, for any ζ_{σ_0}, we have from Lemma 7.3.1 along with the bounded kernel assumptions

$$\mathbb{E}\big[e^{\lambda \zeta_{\sigma_0} - E\zeta_{\sigma_0}}\big] = \prod_{k=0}^{r-1} \mathbb{E}\big[e^{\frac{\lambda}{r}\zeta_k - E\zeta_k}\big] \leq \prod_{k=0}^{r-1} \mathbb{E}\big[e^{\frac{\lambda^2}{8r^2} M_h^2}\big] = \exp\Big(\frac{\lambda^2}{8r} M_h^2\Big) \tag{7.25}$$

Thus $P(U_{n,m} - \mathbb{E}U_{n,m} \geq t) \leq \exp\Big(-\lambda t + \frac{\lambda^2}{8r} M_h^2\Big)$. Optimizing over $\lambda \in \mathbb{R}_+$ gives the desired result. □

The proof for the one sample U-statistics is the same. The exact same result is true for the V-statistics and the proof can be found in [Hoeffding 1963, section 5.C]. Thus we can see that for a fixed m and δ, the difference between the complete estimator and the expected quantity decreases at a rate of $\mathcal{O}(\sqrt{\frac{m}{n}})$. This concentration result is at the heart of several of our contributions that we detail in the next chapter.

7.4 Conclusion

In this chapter, we presented the empirical interest of computing the optimal transport problem between minibatches. We discussed the practical advantages as well as the different domains where it was used. Then we presented several optimal transport properties that are at the foundation of our theoretical analysis of minibatch OT. We reviewed the optimal transport sample complexities, the concentration bounds, the estimator bias and the regularity with respect to the ground cost c. Finally, we presented the U-statistics and we gave their concentration bounds.

Several works were published to justify the minibatch paradigm. [Bernton 2019] studied statistical inference, performed by the minimization over the parameter space of the Wasserstein distance between model distributions and the empirical distribution of the data. They showed that for generative models, the minimizers of the minibatch loss converge to the true minimizer when the minibatch size and the number of data increase. [Sommerfeld 2019] considered another approach. They show that approximating OT with minibatch OT is a good approximation loss and they exhibit a deviation bound between the two quantities. In this manuscript, we propose a study of the use of minibatch OT as a loss function and we gather all our contributions in the two next Chapters 8 and 9. We propose a rigorous formalism for all empirical measures, which assures that all the mass is transported. Based on the formalism, we give a closed-form in the case where samples lie in 1D and we illustrate the non optimal connections created by the minibatch approximation on the resulting transport plan. We then study the loss properties of minibatch OT and we show that minibatch OT is not a distance. We introduce a new loss function to recover some distance properties and study its positiveness. We evaluated our different theoretical contributions on different generative models and color transfer experiments. Finally, the different illustrations highlighted that the non optimal connections could hurt the generalization of deep neural networks. We finally propose to use a robust OT variant at the minibatch level to mitigate these connections and investigate the performances on domain adaptation experiments.

Chapter 8

Minibatch Optimal Transport

Contents

In this chapter, our goal is to study the minibatch approximation of optimal transport as a loss function. We introduce a rigorous formalism and provide a theoretical study of minibatch optimal transport. In the first section, we formalize the problem, develop empirical estimators of minibatch OT for uniform probability measures and give a closed-form solution in the 1D case. Following the uniform case, we extend empirical estimators to non uniform probability vectors. We then introduce a debiased minibatch OT cost inspired by the Sinkhorn Divergence and review its positivity. We then study the concentration bounds of minibatch OT and its optimization properties. Finally, we evaluate our debiased loss functions and minibatch OT on several experiments. These contributions have been published in the 23rd International Conference on Artificial Intelligence and Statistics [Fatras 2020b] and a longer version is currently submitted to SIAM Journal on Mathematics of Data Science [Fatras 2021c].

Notations Before starting, we recall some notations that will be used in this chapter. Suppose we have access to n data $\boldsymbol{x} \in \mathbb{R}^d$. We first assign a fixed index to each data and then get a n-tuple of data $\boldsymbol{X}$, i.e., $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$. This assignation allows us to draw minibatches of data, and moreover, permutations of assigned indices would not change any result.

As each element inside the data m-tuple has an index, it is then possible to characterize a m-tuple of data with a corresponding m-tuple of indices. A generic element of indices $I = (i_1, \ldots, i_m) \in [\![n]\!]^m$ is called an index m-tuple. We recall that we denote by $\mathcal{P}^m$ the set of m-tuples without replacement. For an index m-tuple $I = (i_1, \ldots, i_m)$, $\boldsymbol{X}(I) = (\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_m})$ is the corresponding data m-tuple, and vice-versa, any data m-tuple can be written $\boldsymbol{X}(I)$ for some index m-tuple I.

We finish by defining extra notations. The characteristic function of the set A, which is equal to 1 if $i \in A$ and 0 otherwise, is denoted $\mathbf{1}_A(i)$. With a slight abuse of notation we write $i \in I$ if the index i appears in the m-tuple I. We also write $\mathbf{1}_I(i)$ for tuples of indices. Regarding the sum over the elements of $I = (i_1, \cdots, i_m)$, we denote it as $\sum_{i \in I} f(i) = \sum_{k=1}^m f(i_k)$ and similarly for the product over the elements $\Pi_{i \in I} f(i) = \Pi_{k=1}^m f(i_k)$.

8.1 Expectation of Optimal Transport over minibatches

Throughout the minimization of a data fitting problem, the optimal transport cost can be computed between minibatches of input empirical measures. This takes the form of 2 m-tuples of samples drawn uniformly at random from the source and the target measures, to associate them a uniform weight $1/m$ and to compute the optimal transport between these sub-measures. It is likely that the selected data change over each iteration and this leads to minimizing the expectation of optimal transport between minibatches of input measures. Formally, consider the minibatches $(\boldsymbol{X}, \boldsymbol{Y})$ drawn from $\alpha^{\otimes m} \otimes \beta^{\otimes m}$, we minimize

$$E_h^m = \mathop{\mathbb{E}}_{(\boldsymbol{X},\boldsymbol{Y}) \sim \alpha^{\otimes m} \otimes \beta^{\otimes m}} [h(\boldsymbol{u}_m, \boldsymbol{u}_m, C^m(\boldsymbol{X}, \boldsymbol{Y}))], \tag{8.1}$$

where h is an OT cost and $\boldsymbol{u}_m$ is a uniform vector of size m. The expectation over minibatches is fundamentally different with respect to the original OT problem between the full input measures. In this section, we start by building a rigorous formalism in the case of empirical measures, we also prove a closed-form solution for 1D data of the minibatch OT problem and give 1D and 2D illustrations. Following the formalism in the uniform case, we extend it to general discrete measures.

8.1.1 Empirical estimators in the uniform case

As we have only access to the empirical measures α_n and β_n, we need to define relevant estimators of E_h^m. Relevant estimators would correctly estimate E_h^m but also transport the entire input measures u_m. We start with defining a ground cost and for simplicity we consider the Euclidean distance, but our formalism can be adapted to any ground cost. We consider:

$$C^{m,p} : (\boldsymbol{X}, \boldsymbol{Y}) \in (\mathbb{R}^{m \times d})^2 \mapsto C^{m,p}(\boldsymbol{X}, \boldsymbol{Y}) = \left(\|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^p \right)_{1 \leq i,j \leq m} \in \mathcal{M}_m(\mathbb{R}), \tag{8.2}$$

The power p is useful when we consider the p-Wasserstein distance. The expectation of the OT cost over minibatches E_h^m can be estimated using U-statistics as presented in Section 7.3. Thus we define the following estimator.

Definition 16 (Minibatch OT)**.** *Let $C = C^{n,p}(\boldsymbol{X}, \boldsymbol{Y})$ be a square matrix of size n. Given an OT loss $h \in \{\mathfrak{L}, W_p, \mathfrak{L}^\varepsilon, S^\varepsilon\}$ and an integer $m \leq n$, we define the following quantity:*

$$\overline{h}_C^m(\boldsymbol{X}, \boldsymbol{Y}) := \frac{(n-m)!^2}{n!^2} \sum_{I,J \in \mathcal{P}^m} h(\boldsymbol{u}_m, \boldsymbol{u}_m, C_{I,J}) \tag{8.3}$$

where for I, J two m-tuples, $C_{(I,J)}$ is the matrix extracted from C by keeping the rows and columns corresponding to I and J respectively.

A similar estimator can be built for Gromov-Wasserstein and we report the reader to its general form in Section 8.1.3. We omit C when clear from context. While it is easier to get an upper bound of minibatch OT and statistical results with the ground cost $C^{n,p}(\boldsymbol{X}, \boldsymbol{Y})$, which is a square matrix of size n, in practice we only need to compute $C_{(I,J)}$ as it is equal to $C^{m,p}(\boldsymbol{X}(I), \boldsymbol{Y}(J))$ and it reduces the memory cost. These quantities represent an average of OT costs h over minibatches of size m. It is clear that these estimators are two sample U-statistics of order $2m$. When one of the measure is continuous while the other is discrete, we have a semi-discrete variant of minibatch OT:

$$\overline{h}_C^m(\boldsymbol{X}, \beta) := \frac{(n-m)!}{n!} \sum_{I \in \mathcal{P}^m} \mathop{\mathbb{E}}_{\boldsymbol{Y} \sim \beta^{\otimes m}} h(\boldsymbol{u}_m, \boldsymbol{u}_m, C(\boldsymbol{X}(I), \boldsymbol{Y})) \tag{8.4}$$

It can be seen as a one sample U-statistic of order m.

Remark 6. *We recall that all OT costs are symmetric. This means that for any permutation I' of a given tuple I, we have $h(\boldsymbol{u}_m, \boldsymbol{u}_m, C_{I,J}) = h(\boldsymbol{u}_m, \boldsymbol{u}_m, C_{I',J})$. The symmetry property allows us to rewrite the minibatch OT definition. Let $\mathcal{P}^{o,m}$ be the set of ordered m-tuples without replacement, we can rewrite (8.3) as:*

$$\overline{h}_C^m(\boldsymbol{X}, \boldsymbol{Y}) := \binom{n}{m}^{-2} \sum_{I,J \in \mathcal{P}^{o,m}} h(\boldsymbol{u}_m, \boldsymbol{u}_m, C_{I,J}) \tag{8.5}$$

In practical settings, since $\overline{h}_C^m(\boldsymbol{X}, \boldsymbol{Y})$ is an expectation over the combinatorial number of all possible pairs of m-tuples I, J, it is often estimated by drawing only k such pairs of m-tuples uniformly, called subsample quantity.

Definition 17 (Minibatch subsampling)**.** *Consider the notations from Definition 16. Pick two integers $k > 0$ and $0 < m \leq n$. Then, we define the incomplete estimator:*

$$\widetilde{h}_C^{m,k}(\boldsymbol{X}, \boldsymbol{Y}) := \frac{1}{k} \sum_{(I,J) \in \mathbb{D}_k} h(\boldsymbol{u}_m, \boldsymbol{u}_m, C_{(I,J)}) \tag{8.6}$$

where $\mathbb{D}_k$ is a set of k pairs of m-tuples drawn uniformly at random.

The subsample quantity for exact optimal transport has then a computational complexity of $\mathcal{O}(km^3 \log(m))$. The value of k depends on the application and is discussed in more details in the experimental section. We finish this paragraph with the following remark:

Remark 7. *In (8.3) and (8.6), the dependence in the measure supports is implicit through the Euclidean ground cost C,* i.e., *$(\boldsymbol{X}, \boldsymbol{Y}) \mapsto C(\boldsymbol{X}, \boldsymbol{Y}) \mapsto h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}))$.*

The transport plan is interesting to see the connections between samples, that is why we build similar estimators.

Transport plan estimators Classical OT losses such as $\mathfrak{L}$ or its entropic variant $\mathfrak{L}^\varepsilon$ are directly associated with a transport plan between measures. We now propose to similarly define a transport plan associated to the proposed minibatch OT losses when it is possible. The main idea is that for each pair of samples $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$, one can average the connections provided by all possible "transport plan between minibatches", with the following definition.

Definition 18 (minibatch transport plan)**.** *We denote by* $\mathrm{Opt}(h, C, \boldsymbol{u}, \boldsymbol{u})$ *the set of all optimal transport plans for* h *an OT cost, cost matrix* C *and a marginal* $\boldsymbol{u}$*. Let* $\mathbf{u}_m \in \mathbb{R}^m$ *be a discrete uniform probability vector. For each pair of index* m*-tuples* $I = (i_1, \dots, i_m)$ *and* $J = (j_1, \dots, j_m)$ *from* $[\![1, n]\!]^m$*, consider* $C' := C_{I,J}$ *the* $m \times m$ *matrix with entries* $C'_{k\ell} = C_{i_k, j_\ell}$ *and denote by* $\Pi^m_{I,J}$ *an arbitrary element of* $\mathrm{Opt}(h, C, \boldsymbol{u}_m, \boldsymbol{u}_m)$*. It can be lifted to an* $n \times n$ *matrix where all entries are zero except those indexed in* $I \times J$*:*

$$\Pi_{I,J} = Q_I^\top \Pi^m_{I,J} Q_J \tag{8.7}$$

where Q_I *and* Q_J *are* $m \times n$ *matrices defined entrywise as*

$$(Q_I)_{ki} = \delta_{i_k, i}, 1 \le k \le m, 1 \le i \le n \tag{8.8}$$

$$(Q_J)_{\ell j} = \delta_{j_\ell, j}, 1 \le \ell \le m, 1 \le j \le n. \tag{8.9}$$

Each row of these matrices is a Dirac vector, hence they satisfy $Q_I \mathbf{1}_n = \mathbf{1}_m$ *and* $Q_J \mathbf{1}_n = \mathbf{1}_m$*.*

We also define the *averaged minibatch transport matrix* which takes into account all possible minibatch couples.

Definition 19 (Averaged minibatch transport matrix)**.** *Consider as in Definition 16 an OT kernel* h *and* $1 \le m \le n$*. Given data* n*-tuples* $\boldsymbol{X}, \boldsymbol{Y}$*, consider for each pair of* m*-tuples* I*,* J*, the uniform vector of size* $m, \boldsymbol{u}_m$*, and let* $\Pi_{I,J}$ *be defined as in Definition 18. The averaged minibatch transport matrix is*

$$\overline{\Pi}^{h,m}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{(n-m)!^2}{n!^2} \sum_{I,J \in \mathcal{P}^m} \Pi_{I,J} \tag{8.10}$$

Similarly to incomplete minibatch OT estimator, we can define incomplete minibatch transport plan estimator. Let two integers $k \ge 1$ *and* $1 \le m \le n$*:*

$$\widetilde{\Pi}^{h,m,k}(\boldsymbol{X}, \boldsymbol{Y}) := \frac{1}{k} \sum_{(I,J) \in \mathbb{D}_k} \Pi_{I,J}, \tag{8.11}$$

where $\Pi_{I,J}$ *is the lifted* $n \times n$ *OT plan between minibatches.*

When there are no possible confusion, we omit the inputs $\boldsymbol{X}, \boldsymbol{Y}$. The average in the above definition can be expressed as a finite weighted sum of $\Pi_{I,J}$. It is therefore well defined for an arbitrary choice of optimal transport plans $\Pi_{I,J}$, and we do not need to concern ourselves with the measurability of selection of optimal transport plans. The same will be true whenever an average of optimal transport plans will be taken in the rest of the manuscript, since all results concerning such averages will be nonasymptotic. We will therefore avoid further mentioning this issue, for the sake of brevity. Note also that the Sinkhorn divergence involves three terms, hence three transport plans, which explains why we do not attempt to define an associated averaged minibatch transport matrix.

This construction is consistent with the construction of minibatch OT. Indeed, for an exact OT loss $\mathfrak{L}$ or a p-Wasserstein distance to the power p, W_p^p, it is straight forward to check that $\overline{h}^m_C(\boldsymbol{X}, \boldsymbol{Y}) =$

$\langle \overline{\Pi}^{h,m}(\boldsymbol{X}, \boldsymbol{Y}), C \rangle$. Thus as $\overline{\Pi}^{h,m}(\boldsymbol{X}, \boldsymbol{Y})$ is an average of connections, we want to known if all source and target mass are entirely transported, *i.e.,* we want to check that $\overline{\Pi}^{h,m}(\boldsymbol{X}, \boldsymbol{Y})$ is an admissible transport plan between uniform probability vectors $\boldsymbol{u}_n$. We have the following proposition

Proposition 11. *The transportation plan $\overline{\Pi}^{h,m}(\boldsymbol{X}, \boldsymbol{Y})$ is an admissible transportation plan between the full input measures $\boldsymbol{u}_n, \boldsymbol{u}_n$. Furthermore, for an exact OT loss kernel h or p-Wasserstein distance to the power p we have : $\overline{h}^m_C(\boldsymbol{X}, \boldsymbol{Y}) \geq h(\boldsymbol{u}_n, \boldsymbol{u}_n, C)$.*

The proof can be found in Appendix A.1.2. The fact that $\overline{\Pi}^{h,m}(\boldsymbol{X}, \boldsymbol{Y})$ is an admissible transportation plan means that even though it is not optimal, we still transport data similarly to OT. Finally, note that because incomplete U-statistics are not U-statistics in general, the incomplete minibatch transport plan estimator does not define a transport plan between the full measures in general, i.e., their marginals are not equal to uniform probability vectors.

8.1.2 Illustration examples on 1D ans 2D data

In this section, we illustrate the minibatch OT framework on two simple cases. The first case is when data lie in 1D. Thanks to the 1D OT close-formed, detailed in Section 2.2.3, we can express a closed-form for minibatch OT and we illustrate the minibatch OT plan for several values of minibatches. We then illustrate the transport plan for 2D toy data.

Closed-form solution for 1D data. We recall that the solution of the 1D OT problem can be computed with a sorting algorithm which gives an appealing $\mathcal{O}(n \log(n))$ complexity compare to the initial $\mathcal{O}(n^3 \log(n))$ of a network simplex solver.

We assume (without loss of generality) that the points are ordered on the $\mathbb{R}$ line. In such a case, we can compute the 1D Wasserstein 1 distance with cost $c(\boldsymbol{x}, \boldsymbol{y}) = |\boldsymbol{x} - \boldsymbol{y}|$ as: $W(\boldsymbol{u}, \boldsymbol{u}) = \frac{1}{n} \sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{y}_i|$ and the OT matrix is simply an identity matrix scaled by $\frac{1}{n}$. Based on the symmetry definition of minibatch OT (8.5) and after a short combinatorial calculus (given in appendix), the 1D minibatch transport plan coefficient between sorted samples $(\overline{\Pi}^{h,m})_{j,k}$ can be computed as:

$$(\overline{\Pi}^{h,m})_{j,k} = \frac{1}{m} \binom{n}{m}^{-2} \sum_{i=i_{\min}}^{i_{\max}} \binom{j-1}{i-1} \binom{k-1}{i-1} \binom{n-j}{m-i} \binom{n-k}{m-i}$$

where $i_{\min} = \max(0, m - n + j, m - n + k)$ and $i_{\max} = \min(j, k)$. $i_{\min}$ and $i_{\max}$ represent the sorting constraints.

We show on the first row of Figure 8.1 the minibatch OT plans $\overline{\Pi}^{h,m}$ with $n = 20$ samples for different values of the minibatch size m. On the second row of the figure a plot of the measures in several rows of $\overline{\Pi}^{h,m}$, to illustrate the number of connections. We give the OT plans for entropic and quadratic regularized OT between full measures for comparison purpose. It is clear from the figure that the OT matrix densifies when m decreases, which is a similar effect to entropic regularization. Regarding the quadratic regularization, the spread of mass is more localized and preserves the sparsity as discussed in [Blondel 2018].

While the entropic regularization spreads the mass in a similar manner for all samples, minibatch OT concentrates the mass at the extremities. Note that the minibatch OT plan solution is for ordered samples and does not depend on the position of the samples once ordered, as opposed to the regularized OT methods. This will be better illustrated in the next example.

Finally, a closed-form is also available in the case of drawing with replacement. We provide it in appendix. Unfortunately, its computational complexity makes it hard to use in practice.

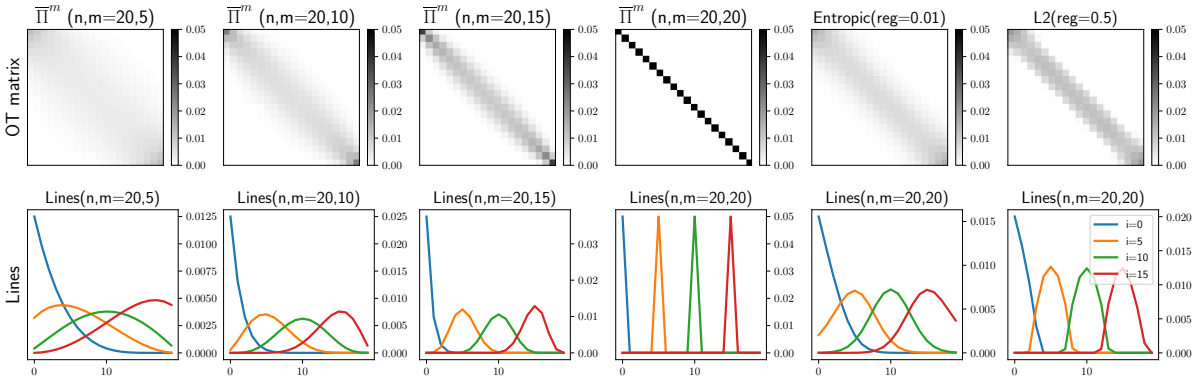


Figure 8.1: Several OT plans between measures with $n = 20$ samples in 1D. The first row shows the minibatch OT plans $\overline{\Pi}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$ for different values of m, the second row provides the shape of the measures on the rows of $\overline{\Pi}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$. h is the exact OT. The two last columns correspond to classical entropic and quadratic regularized OT.

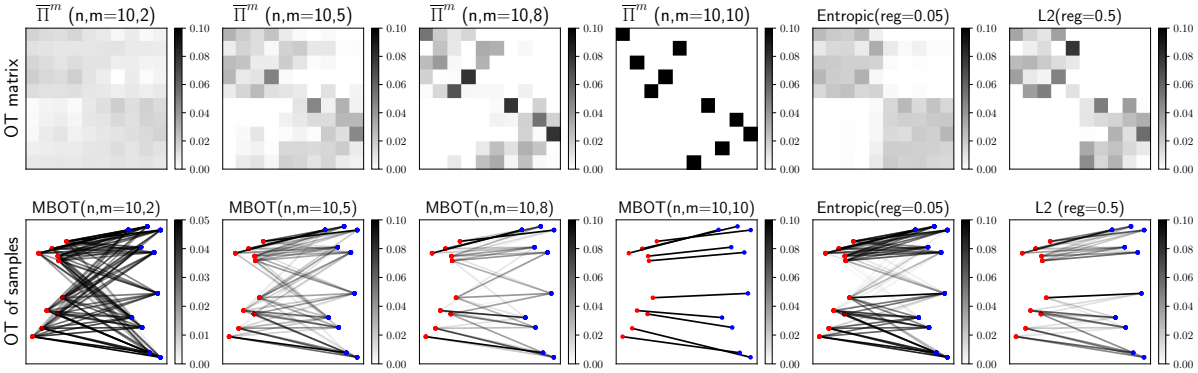


Figure 8.2: Several OT plans between 2D measures with $n = 10$ samples. The first row shows the minibatch OT plans $\overline{\Pi}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$, where h is the exact optimal transport cost, for different values of m. The second row provide a 2D visualization of where the mass is transported between the 2D positions of the sample.

Minibatch Wasserstein in 2D We illustrate several OT plans between two empirical measures of 10 2D-samples each in Figure 8.2. We consider the MBOT transport plan for several batch sizes, the entropic and quadratic regularized OT between full measures. We use two 2D empirical measures (point cloud) where the samples have a cluster structure and the samples are sorted *w.r.t.* their cluster. We first discuss the sampling without replacement. We can see from the OT plans in the first row of the figure that the cluster structure is more or less recovered with the regularization effect of the minibatches (and also regularized OT). On the second row one can see the effect of the geometry of the samples on the spread of mass. Similarly to 1D, for Minibatch OT, samples on the border of the simplex cannot spread as much mass as those in the center and have darker rows. This effect is less visible on regularized OT.

The defined formalism supposed we had uniform probability vectors for the full measure and we considered the probability vectors as uniform at the minibatch level. However as justified by Proposition 11 the above formalism would output a transport plan with uniform marginals. Thus in the case of non uniform probability vectors such as in ARWGAN (see Chapter 6), the formalism would not give the

correct marginals and a GAN using minibatch OT would generate data similar to all training data. We also supposed that we had a sampling of data without replacement to form a minibatch, which does not correspond to the GAN sampling. In the next section, we extend our formalism to non uniform weights and to other sampling.

8.1.3 Empirical estimators in the general case

In this section, we generalize the minibatch OT formalism that we defined in Section 8.1.1. We aim at defining a formalism which can consider different samplings than sampling without replacement and other probability vectors than the uniform.

We recall that discrete probability measures can either be represented as a sum of Diracs $\sum_i a_i \delta_{\boldsymbol{x}_i}$ or with a probability vector and the support of measures $(\mathbf{a}, \boldsymbol{X})$. We chose the latter as it is easier to define formal mathematical objects when we consider discrete probability measures and for consistency with [Peyré 2019]. Indeed sum of Diracs are equal for different indices assignations, *i.e.*, $\sum_{k=1}^n a_{i_k} \delta_{x_{i_k}} = \sum_{k=1}^n a_k \delta_{x_k}$, the result of selecting an element with a given index from the minibatch would depend on the order of Diracs. One can define a discrete probability measure from a probability vector $\mathbf{a} \in \Sigma_n$ and locations $\boldsymbol{X}$ in a canonical way by $\alpha_n = \sum_{i=1}^n a_i \delta_{x_i}$ (see remark 2.1 [Peyré 2019]), and we will often implicitly use this assignment throughout the rest of the chapter (see remark 2.11 [Peyré 2019]).

To define minibatch OT losses for general probability vectors $\mathbf{a}, \mathbf{b}$, a first ingredient is a *reweighting function* w that takes as inputs a discrete probability $\mathbf{a} \in \Sigma_n$ and an m-tuple of indices $I = (i_1, \dots, i_m)$ and outputs a discrete probability vector $\mathbf{a}_I = w(\mathbf{a}, I) \in \Sigma_m$. A second ingredient is a *parametric family of distributions* $\{P_{\mathbf{a}} : \mathbf{a} \in \Sigma_n\}$ such that for each $\mathbf{a} \in \Sigma_n$, $P_{\mathbf{a}}$ is a probability distribution over m-tuples I of indices. The law on probability tuples assures that we have a weighted average of OT kernels and its combination with a suited reweighting function assures all samples are transported. Those ingredients are needed to get unbiased estimator of minibatch OT. Formally, we need:

Definition 20 (Reweighting and probability functions)**.** *A reweighting function is a map w of the form :*

$$w : (\mathbf{a}, I) \in \Sigma_n \times [\![n]\!]^m \mapsto \Sigma_m \tag{8.12}$$

A probability function is a map P of the form :

$$P : \mathbf{a} \in \Sigma_n \mapsto P_{\mathbf{a}} \in \mathcal{P}\big([\![n]\!]^m\big), \tag{8.13}$$

where $\mathcal{P}\big([\![n]\!]^m\big)$ is the set of probability distributions over the set of m-tuples I of indices $[\![n]\!]^m$.

We are now ready to give a formal definition of minibatch OT losses. The idea is to compute the expectation of the OT kernels over minibatches I, furthermore we need the reweighting functions to assure that the OT kernels has probability vectors as inputs.

Definition 21 (Minibatch Wasserstein)**.** *Let $C = C^{n,p}(\boldsymbol{X}, \boldsymbol{Y})$ be a matrix of size $n \times n$. Given a kernel $h \in \{\mathfrak{L}, W_p, W_p^p, \mathfrak{L}^\varepsilon, S^\varepsilon\}$, two reweighting functions w_1, w_2 and two probability functions P^1, P^2 as in (8.12) and (8.13) respectively, we define the minibatch OT loss $\overline{h}^m_{w_1,w_2,P^1,P^2}$ for any $\mathbf{a}, \mathbf{b} \in \Sigma_n$ by :*

$$\overline{h}^m_{w_1,w_2,P^1,P^2,C}(\mathbf{a}, \mathbf{b}) := \mathbb{E}_{I \sim P^1_{\mathbf{a}}} \mathbb{E}_{J \sim P^2_{\mathbf{b}}} h\Big(w_1(\mathbf{a}, I), w_2(\mathbf{b}, J), C_{(I,J)}\Big), \tag{8.14}$$

where for I, J two m-tuples, $C_{(I,J)}$ is the matrix extracted from C by keeping the rows and columns corresponding to I and J respectively. Regarding the incomplete estimator, we have:

$$\widetilde{h}^{m,k}_{w,P}(\mathbf{a}, \mathbf{b}) := \frac{1}{k} \sum_{(I,J) \in \mathbb{D}_k} h\Big(w(\mathbf{a}, I), w(\mathbf{b}, J), C_{(I,J)}\Big) \tag{8.15}$$

where $\mathbb{D}_k$ is a set of k pairs of m-tuples drawn independently from the joint distribution $P_{\mathbf{a}} \otimes P_{\mathbf{b}}$.

Similarly we have for Gromov-Wasserstein:

Definition 22 (Minibatch Gromov-Wasserstein)**.** *Let w_1, w_2, P^1, P^2 be as in Definition 21. We define for two ground costs $C^1 = C^{n,p}(\boldsymbol{X}, \boldsymbol{X})$ and $C^2 = C^{n,p}(\boldsymbol{Y}, \boldsymbol{Y})$. The minibatch Gromov-Wasserstein is defined as:*

$$\overline{\mathcal{GW}}^m_{w_1,w_2,P^1,P^2,C^1,C^2}(\mathbf{a}, \mathbf{b}) := \mathbb{E}_{I\sim P^1_{\mathbf{a}}}\mathbb{E}_{J\sim P^2_{\mathbf{b}}}\mathcal{GW}\Big(w_1(\mathbf{a}, I), w_2(\mathbf{b}, J), C^1_{I,I}, C^2_{J,J}\Big), \tag{8.16}$$

where $C^1_{(I,I)}$ (resp. $C^2_{(J,J)}$) is the matrix extracted from C^1 (resp. C^2) by keeping the rows and columns corresponding to I and I (resp. J and J).

Like in the uniform probability vectors case, the dependence of minibatch OT in the ground cost C will often be omitted when there is no possible confusion. When the reweighting functions and the probability laws on tuples are the same (equal to w and P respectively), we use the following shorthand notations for the above losses : $\overline{h}^m_{w,P}$. With a slight abuse of notation, we also use the notation $\overline{h}^m_{w,P}$ for the $\mathcal{GW}$ loss. Regarding the incomplete estimator, the notable difference is that now the minibatches are drawn according to $P_{\mathbf{a}} \otimes P_{\mathbf{b}}$.

The loss $\overline{h}(\mathbf{a}, \mathbf{b})$ corresponds to an averaged optimal transport distance between sub-probability distributions of input probability distributions $\mathbf{a}$ and $\mathbf{b}$. The minibatch OT losses define weighted U-statistics and V-statistics [J Lee 2019] where the weights depend on the input probability vectors $\mathbf{a}, \mathbf{b}$ and on the laws over index m-tuple $P_{\mathbf{a}}, P_{\mathbf{b}}$. This connection turns out to be central to get quantitative statistical results. Concrete versions of these minibatch OT losses are obtained by specifying its ingredients h, w, and P. We now give a few examples of some *reweighting functions* and *families of distributions*.

Example 1 (Uniform reweighting function)**.** *The* uniform reweighting function $w^{\mathtt{U}}$ *is independent of the input discrete probability* $\mathbf{a}$*. It is defined coordinatewise for any m-tuple $I = (i_1, \ldots, i_m)$ by $w^{\mathtt{U}}_k(\mathbf{a}, I) = \frac{1}{m}$, $1 \leq k \leq m$ and yields to a uniform probability vector in Σ_m.*

Example 2 (Normalized reweighting function)**.** *The* normalized reweighting function $w^{\mathtt{W}}$ *normalizes the restriction of the input discrete probability* $\mathbf{a}$ *to the support of I, to ensure it remains a discrete probability. It is defined coordinatewise for any m-tuple $I = (i_1, \ldots, i_m)$ by $w^{\mathtt{W}}_k(\mathbf{a}, I) = \frac{a_{i_k}}{\sum_{p=1}^m a_{i_p}}$, $1 \leq k \leq m$, which is again a probability vector even if entries in I are repeated.*

For instance, consider four $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$ data with weights $\mathbf{a} = (\frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{2})$, if one picks the batch $I = (2, 4)$, the reweighting functions give $w^{\mathtt{W}}(\mathbf{a}, I) = [\frac{1}{5}, \frac{4}{5}]$ and $w^{\mathtt{U}}(\mathbf{a}, I) = [\frac{1}{2}, \frac{1}{2}]$. In the case of a uniform discrete probability $\boldsymbol{u} \in \Sigma_n$, the two reweighting functions are identical.

Regarding the parametric law on indices, which gives the probability to pick a given batch of samples, we focus on two constructions depending whether sampling is done with or without replacement. Indeed in practice, most of work use a sampling without replacement, and it is easy to design this case with our formalism. We first consider sampling with replacement.

Example 3 (Drawing indices with replacement)**.** *Drawing $i_\ell \in [\![n]\!]$, $1 \leq \ell \leq m$ i.i.d. (with replacement) from the discrete probability distribution $\mathbf{a} \in \Sigma_n$ yields the law on indices*

$$P^{\mathtt{U}}_{\mathbf{a}}(I) = \Pi_{i\in I} a_i. \tag{8.17}$$

Now we give an example of drawing without replacement. The idea is to give a zero probability to pick a batch with repeated indices.

Example 4 (Drawing indices "without replacement")**.** *Given a discrete probability distribution* $\mathbf{a} \in \Sigma_n$*, it is also possible to draw distinct indices* $i_\ell \in [\![n]\!]$*,* $1 \leq \ell \leq m$*, by defining* $P^{\mathtt{W}}_{\mathbf{a}}(I) = 0$ *if the* m*-tuple* I has *repeated indices, otherwise*

$$P^{\mathtt{W}}_{\mathbf{a}}(I) = \frac{1}{m}\frac{(n-m)!}{(n-1)!}\sum_{i\in I} a_i. \tag{8.18}$$

With a uniform discrete probability, $a_i = \frac{1}{n}$*,* $1 \leq i \leq n$*, this law corresponds to drawing the* m*-tuples without repeated indices* I *uniformly at random among all possible* m*-tuples without repeated indices, i.e., drawing the* m *indices* i_p *without replacement. By abuse of language, we will sometimes refer to this law as a draw "without replacement" even for non uniform* $\mathbf{a}$*.*

Note that the minibatch OT between uniform probability vectors in Definition 16 can be recovered with the sampling $P^{\mathtt{W}}_{\mathbf{a}}$ and the reweighting functions $w^{\mathtt{W}}, w^{\mathtt{U}}$.

Minibatch transport plan. Regarding the transport plan, we can also generalize the formulation (8.10).

Definition 23 (Averaged minibatch transport matrix)**.** *Consider as in Definition 21 an OT kernel* h*, two reweighting functions* w_1, w_2 *and a family of probability distributions* $\{P_{\mathbf{a}} : \mathbf{a} \in \Sigma_n\}$ *over index* m*-tuples from* $[\![1, n]\!]$*, where* $1 \leq m \leq n$*. Given discrete probabilities* $\mathbf{a}, \mathbf{b} \in \Sigma_n$ *and data tuples* $\boldsymbol{X}, \boldsymbol{Y}$*, consider for each pair of* m*-tuples* I*,* J *the discrete probabilities* $\mathbf{a}_I = w_1(\mathbf{a}, I) \in \Sigma_m$*,* $\mathbf{b}_J = w_2(\mathbf{b}, J) \in \Sigma_m$*. Let* $\Pi_{I,J}$ *be defined as in Definition 18 but* $\Pi^m_{I,J}$ *is an optimal transport plan with marginals* $\mathbf{a}_I, \mathbf{b}_J$*. The averaged minibatch transport matrix is*

$$\overline{\Pi}^{h,m}_{w_1,w_2,P_{\mathbf{a}},P_{\mathbf{b}}}(\mathbf{a}, \mathbf{b}) = \mathbb{E}_{I\sim P_{\mathbf{a}}, J\sim P_{\mathbf{b}}}\Pi_{I,J} \tag{8.19}$$

where $\Pi_{I,J}$ *is the lifted* $n \times n$ *OT plan between minibatches. For brevity this is simply denoted* $\overline{\Pi}^{h,m}_{w,P}$ *when* $w = w_1 = w_2$ *and* $P = P_{\mathbf{a}} = P_{\mathbf{b}}$ *are clear from context. We can also generalize the incomplete transport plan as folllows:*

$$\widetilde{\Pi}^{h,m,k}_{w,P}(\mathbf{a}, \mathbf{b}) := \frac{1}{k}\sum_{(I,J)\in\mathbb{D}_k} \Pi_{I,J}. \tag{8.20}$$

While the $n \times n$ matrix defined in (8.19) is candidate to be transport plan between $\mathbf{a}$ and $\mathbf{b}$, we need to check if it is indeed admissible, i.e., if it has the right marginals. This is why it is a priori only called an averaged minibatch transport *matrix*.

Proposition 12. *If the reweighting function* w *and the parametric distribution on* m*-tuples* $P_{\mathbf{c}}$ *satisfy the following admissibility condition*

$$\mathbb{E}_{I\sim P_{\mathbf{c}}} Q_I^\top w(\mathbf{c}, I) = \mathbf{c}, \qquad \forall \mathbf{c} \in \Sigma_n \tag{8.21}$$

Then with the notations of Definition 19, the averaged minibatch transport matrix $\overline{\Pi}^{h,m}_{w,P}$ *is an admissible transport plan between the discrete probabilities* $\mathbf{a}, \mathbf{b} \in \Sigma_n$ *in the sense that* $\overline{\Pi}^{h,m}_{w,P}\mathbf{1}_n = \mathbf{a}$ *and* $\mathbf{1}_n^\top \overline{\Pi}^{h,m}_{w,P} = \mathbf{b}^\top$*. Considering the Wasserstein kernel* $h = W_p^p$*, the minibatch loss defined in (16), as the associated coupling* $\overline{\Pi}^{h,m}_{w,P}$ *is not the optimal coupling of the full OT problem, it satisfies*

$$\overline{h}^m_{w,P}(\mathbf{a}, \mathbf{b}) = \langle \overline{\Pi}^{h,m}_{w,P}, C\rangle_F \geq h(\mathbf{a}, \mathbf{b}). \tag{8.22}$$

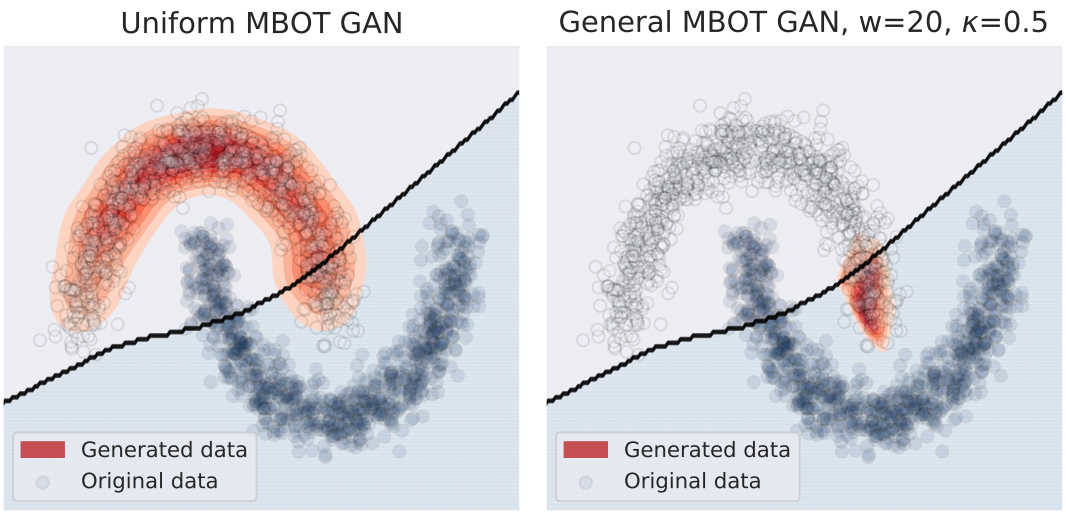


Figure 8.3: Difference between generated 2D data from different MBOTGAN. We used the uniform formalism (left) and the general formalism with replacement $\overline{h}^m_{\mathtt{U}}$ (right). We use the minibatch OT loss as a loss function for GANs. The black line represents the classifier boundary decision. The hyperparameters for the reweighted distributions are set to $\kappa = 0.5$ and $w = 20$. We trained the generator for 30000 iterations.

Under assumption (8.21) one can safely call $\overline{\Pi}^{h,m}_{w,P}(\mathbf{a}, \mathbf{b})$ an averaged minibatch transport *plan*. Our main examples of reweighting functions and parametric probability distributions indeed satisfy the admissibility condition (8.21).

Lemma 8.1.1 (Admissibility)**.** *The uniform reweighting function $w^{\mathtt{U}}$ and the parametric law "with replacement" $P^{\mathtt{U}}$ satisfy the admissibility condition* (8.21)*. The admissibility condition also holds for the parametric law without replacement $P^{\mathtt{W}}$ with the normalized reweighting function $w^{\mathtt{W}}$.*
In contrast for $w^{\mathtt{U}}, P^{\mathtt{W}}$ when $\mathbf{a}$ is not uniform, the resulting OT matrix is not a transportation plan.

Empirical justification. We introduce some notations for the aforementioned Examples 1, 2, 3, 4. When referring to these 4 examples, we define two estimators. $\overline{h}^m_{\mathtt{W}}$ (respectively $\overline{\Pi}^{h,m}_{\mathtt{W}}$) with law $P^{\mathtt{W}}_{\mathbf{a}}$ and reweighting function $w^{\mathtt{W}}$ stands for the minibatch Wasserstein loss (respectively minibatch OT plan) over the m-tuples without repetitions. And $\overline{h}^m_{\mathtt{U}}$ (respectively $\overline{\Pi}^{h,m}_{\mathtt{U}}$) with law $P^{\mathtt{U}}_{\mathbf{a}}$ and reweighting function $w^{\mathtt{U}}$ stands for the minibatch Wasserstein loss (respectively minibatch OT plan) over the m-tuples I.

We now justify empirically that our general formalism can deal with non uniform probability vectors. To this end, we consider generating misclassified data for a pre-trained classifier using the reweighted distribution from ARWGAN (see Chapter 6). To generate data, we use a GAN with minibatch OT loss as loss function. The ground cost is the squared Euclidean cost and we use either the uniform or the general formalism $\overline{h}^m_{\mathtt{U}}$. We follow the experimental setting from Section 6.3.1 and the results can be found in Figure 8.3. It shows that our uniform formalism generates data following the empirical data while our general formalism generates data which are misclassified by the pre-trained classifier.

Drawing data with or without replacement

Our general flexible formalism allows us to define several minibatch strategies by playing with the probability law on tuples. The laws can be also different between the source and the target distributions.

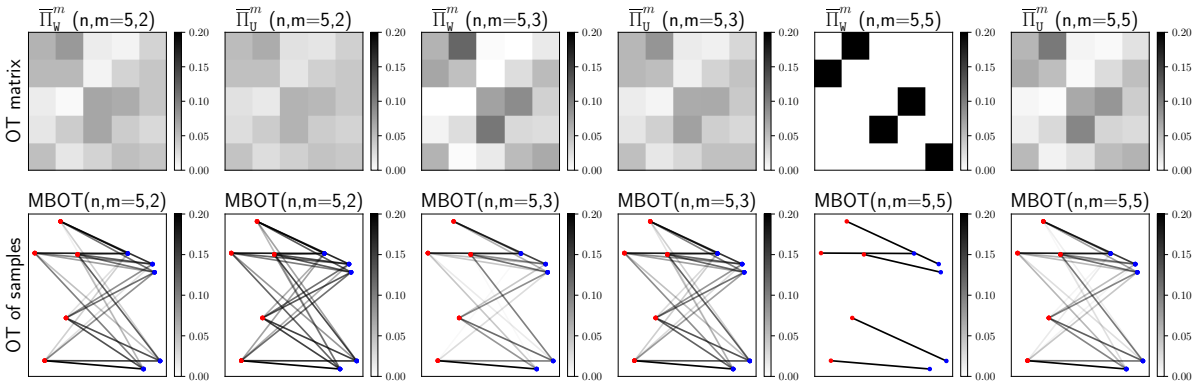


Figure 8.4: Difference between transport plan estimators with 2D distributions and $n = 5$ samples. Each column gives the OT plan $\overline{\Pi}_{\mathtt{W}}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$ or $\overline{\Pi}_{\mathtt{U}}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$ (top) and the shape of the distributions on the rows of the OT plans (bottom). We consider the exact optimal transport cost as h.

In particular, as given in examples, the cases of drawing with or without replacement. An estimator based on sampling without replacement is the most common practice when we have access to n samples. While this drawing has been investigated for minibatch OT losses, the case with replacement, which appears in the GANs formalism, remains an open question that we aim at answering. The minibatch OT losses represent a weighted sum of Wasserstein distance over batches of size m. In the case of sampling without replacement, they are generalized unbiased U-statistics while with a sampling with replacement, we get generalized biased V-statistics [J Lee 2019, Gretton 2012]. Precisely, they are two sample U-statistics or V-statistics of order $2m$ (see [J Lee 2019]) and h is a U-statistic kernel. Interestingly, similar biased and unbiased estimators have been designed to estimate MMD [Gretton 2012].

Finally an important parameter is the value of the minibatch size m. In the case of sampling without replacement, we remark that the minibatch procedure allows us to interpolate between OT, when $m = n$ and averaged pairwise distance, when $m = 1$. This property is not shared by the sampling with replacement. Indeed when $m = n$, it does not correspond to original OT due to the repetition of data. It only converges to the true OT when $n \to \infty$. This effect is illustrated in Figure 8.4. We consider the same setting as in Figure 8.2 but with 5 empirical data. On each column we show the transport plan and the shape of connection between samples. We can see that the estimator $\overline{\Pi}_{\mathtt{U}}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$ has always a denser plan, i.e. a bigger number of connections, than the estimator $\overline{\Pi}_{\mathtt{W}}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$. In particular, when $m = n = 5$, we get the optimal transport plan with $\overline{\Pi}_{\mathtt{W}}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$ while we do not recover it with $\overline{\Pi}_{\mathtt{U}}^{h,m}(\boldsymbol{u}, \boldsymbol{u})$ due to the fact that samples can be repeated. Now that we have rigorously defined how we can build minibatch Wasserstein losses between empirical measures, we study its loss properties.

8.2 Debiased minibatch Optimal Transport: a Sinkhorn divergence approach

In this section, we introduce a new OT cost based on minibatch OT. It follows the form of the Sinkhorn Divergence. We start by motivating this loss function by reviewing the distance properties of minibatch OT. After, we introduce our loss function to recover some loss of minibatch OT and finally we review its positivity.

8.2.1 Metric properties: a fundamental difference

The optimal transport cost can define a distance between probability distributions with some specific ground cost. It is of interest to see if these properties remain true for minibatch OT. Answering this question is the purpose of this section and we start with the following proposition.

Proposition 13 (Estimator properties)**.** *The minibatch OT losses enjoy the following properties:*

- *The losses are symmetric*
- *The losses are not distances*

Proof. We give the proof that minibatch OT losses are not distances. Consider a uniform probability vector and random 3-data tuple $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ with distinct vectors. As $\overline{h}^m_{w,P}$ is a weighted sum of positive terms, it is equal to 0 if and only if each of its term is 0. But consider the minibatch term $I_1 = (i_1, i_2)$ and $I_2 = (i_1, i_3)$, then obviously $h(w(\boldsymbol{u}, I_1), w(\boldsymbol{u}, I_2), C(\boldsymbol{X}(I_1), \boldsymbol{X}(I_2))) \neq 0$ as $\boldsymbol{x}_2 \neq \boldsymbol{x}_3$, finishing the proof. □

The symmetry of the losses is inherited from the optimal transport problem which is itself symmetric. The loss of the separability distance axiom means that for data fitting problems, the final solution will not match the target distribution. The axiom is recovered for minibatches without replacement when $m = n$ as we recover the original OT formulation.

We defined minibatch OT losses and reviewed their basic properties. In what follows, we propose an elegant formulation which fixes this loss.

8.2.2 Debiasing minibatch Wasserstein losses

As we have shown before, the minibatch OT losses are not distances, for general probability vectors and data n-tuple $\boldsymbol{X}$, $\overline{h}^m_{w,P}(\mathbf{b}, \mathbf{b}) > 0$. This leads to an undesirable situation when one uses it for learning purposes as the final solution is not the target distribution but a shrunk version of it. Hence, we would like to debias the losses to get $\overline{h}^m_{w,P}(\mathbf{b}, \mathbf{b}) = 0$. We debias the minibatch OT losses by following the same idea as the Sinkhorn divergence, we remove half of each self term $\overline{h}^m_{w,P}(\mathbf{a}, \mathbf{a})$ and $\overline{h}^m_{w,P}(\mathbf{b}, \mathbf{b})$.

Definition 24 (Debiased Minibatch Wasserstein estimators)**.** *Let $C = C^{n,p}$. Consider $1 \leq m \leq n$ be an integer and h be the OT cost $\mathfrak{L}$, the entropic loss $\mathfrak{L}^\varepsilon$, the Sinkhorn divergence S^ε, or the Gromov-Wasserstein distance $\mathcal{GW}$ for some ground cost $c(\boldsymbol{x}, \boldsymbol{y})$, we define the following quantities:*

$$\Lambda^{h,m}_{w,P,C(\boldsymbol{X},\boldsymbol{Y})}(\mathbf{a}, \mathbf{b}) := \overline{h}^m_{w,P,C(\boldsymbol{X},\boldsymbol{Y})}(\mathbf{a}, \mathbf{b}) - \frac{1}{2}\big(\overline{h}^m_{w,P,C(\boldsymbol{X},\boldsymbol{X})}(\mathbf{a}, \mathbf{a}) + \overline{h}^m_{w,P,C(\boldsymbol{Y},\boldsymbol{Y})}(\mathbf{b}, \mathbf{b})\big), \tag{8.23}$$

That we note when it is clear of context $\Lambda^{h,m}_{w,P}$ and its incomplete counter part:

$$\widetilde{\Lambda}^{h,m,k}_{w,P,C(\boldsymbol{X},\boldsymbol{Y})}(\mathbf{a}, \mathbf{b}) := \widetilde{h}^{m,k}_{w,P,C(\boldsymbol{X},\boldsymbol{Y})}(\mathbf{a}, \mathbf{b}) - \frac{1}{2}\big(\widetilde{h}^{m,k}_{w,P,C(\boldsymbol{X},\boldsymbol{X})}(\mathbf{a}, \mathbf{a}) + \widetilde{h}^{m,k}_{w,P,C(\boldsymbol{Y},\boldsymbol{Y})}(\mathbf{b}, \mathbf{b})\big). \tag{8.24}$$

Remark 8. *We keep making the slight abuse of notation to consider all OT kernels with $\overline{h}$, but we explicit the loss Λ for a Gromov-Wasserstein loss. We note the ground cost $C^{n,p,1}$ and $C^{n,p,2}$ as C^1 and C^2 for sake of readability. With the $\mathcal{GW}$ kernel, the loss Λ is equal to:*

$$\begin{aligned}\Lambda^{h,m}_{w,P,C^1(\boldsymbol{X},\boldsymbol{X}),C^2(\boldsymbol{Y},\boldsymbol{Y})}(\mathbf{a}, \mathbf{b}) &:= \overline{h}^m_{w,P,C^1(\boldsymbol{X},\boldsymbol{X}),C^2(\boldsymbol{Y},\boldsymbol{Y})}(\mathbf{a}, \mathbf{b}) \\ &- \frac{1}{2}\big(\overline{h}^m_{w,P,C^1(\boldsymbol{X},\boldsymbol{X}),C^1(\boldsymbol{X},\boldsymbol{X})}(\mathbf{a}, \mathbf{a}) + \overline{h}^m_{w,P,C^2(\boldsymbol{Y},\boldsymbol{Y}),C^2(\boldsymbol{Y},\boldsymbol{Y})}(\mathbf{b}, \mathbf{b})\big),\end{aligned} \tag{8.25}$$

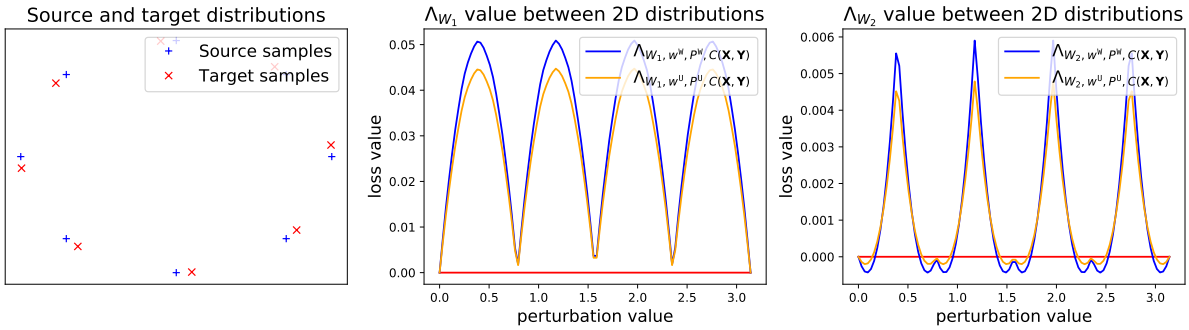


Figure 8.5: Positivity counter example. (Left) source and target distributions for a given perturbation. (Middle and right) Comparison of different estimator values for Λ_{W_1} and Λ_{W_2} with an Euclidean ground cost between the distributions. The red line is the y-axis equals to 0.

It is straight forward to see that $\Lambda^{h,m}_{w,P}(\mathbf{b}, \mathbf{b}) = 0$. A similar loss has been proposed in [Salimans 2018] as a generalized energy distance using the entropic Wasserstein distance as metric. While their loss debiased the minibatch bias, it still had a bias from the entropic regularization. They then relied on the energy distance properties to argue positiveness. The downside of this loss is that it needs to rely on a metric to be positive, however the entropic regularized optimal transport is not a metric between probability distributions as $W^{\varepsilon}(\mathbf{a}, \mathbf{a}) \neq 0$.

We bring insights to this debiased loss and compare its differences to the minibatch OT losses both mathematically and empirically. We use our loss $\Lambda^{h,m}_{w,P}$ with the Wasserstein distance and Sinkhorn divergence because they respect the distance separability axiom. Unfortunately, we prove that even if we consider the Wasserstein distance, this loss is not positive and we will give counter examples.

8.2.3 Positivity: a negative counter example

The loss function $\Lambda^{h,m}_{w,P}$ is composed of three terms of the form of $\overline{h}^m_{w,P}$, then it is possible to estimate it with the different estimators $\overline{h}^m_{\mathtt{W}}$ and $\overline{h}^m_{\mathtt{U}}$ we defined in Section 8.1.3. When estimated with $\overline{h}^m_{\mathtt{W}}$ (resp $\overline{h}^m_{\mathtt{U}}$), we denote $\Lambda^{h,m}_{\mathtt{W}}$ (resp. $\Lambda^{h,m}_{\mathtt{U}}$). Let us consider 8 points on the unit circle equally distributed. Then let us add a perturbation as a rotation to each point position, where the rotation vary from 0 to π. The perturbed distribution becomes our target distribution. When computing the quantity $\Lambda_{W_p}(\boldsymbol{u}, \boldsymbol{u})$, with $p \geq 2$ and an Euclidean ground cost, it can return a negative value. We give the variations of the debiased minibatch OT losses in function of the pertubartion in Figure 8.5 for both the estimators $\overline{h}^m_{\mathtt{W}}$ and $\overline{h}^m_{\mathtt{U}}$.

The loss function might not be always positive for particular case but in practice, we always had a positive loss and it performed better than the biased minibatch OT losses. Furthermore, while we have been able to find counter examples for $p \geq 2$, we have not found any counter example for W_1. Hence we conjecture that Λ_{W_1} might be a positive loss function, and the proof is left as future work, as we did not succeed to formally proving it.

8.3 Learning with minibatch Optimal Transport: concentration bounds and gradients

In this section, we present different deviation bounds for minibatch OT. We study the case where input probability measures have compact supports or are sub-Gaussian probability measures. We also study the case where we have a bounded cost. We then review the optimization properties of minibatch OT and see if it can be minimized with modern optimization techniques.

8.3.1 Concentration bounds between estimators and their expectations

We want to find similar results to the U-statistics concentration bounds (see Theorem 7.3.2) for our estimator $\widetilde{h}^{m,k}_{w,P}(\mathbf{a},\mathbf{b})$ and its expectation $\mathbb{E}\widetilde{h}^{m,k}_{w,P}(\mathbf{a},\mathbf{b})$. We will give a maximal deviation bound for several scenarios. For bounded measures, we will prove that we have a Hoeffding-type inequality. Then we relax the boundness condition on probability measures to give a concentration bound for sub-Gaussian probability measures. Finally, we give a deviation bound in the case of a bounded cost and general probability measures.

In this context, the probability vectors $(\mathbf{a}^{(n)})$ and $(\mathbf{b}^{(n)})$ are sequences which depend on the number of data n. More precisely $(\mathbf{a}^{(n)})$ and $(\mathbf{b}^{(n)})$ are sequences of vectors such that for each $n \in \mathbb{N}$, $\mathbf{a}^{(n)}, \mathbf{b}^{(n)} \in \Sigma_n$, we denote the space of these sequences as $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)}) \in \Sigma$. The sequences of probability vectors $(\mathbf{a}^{(n)})$ and $(\mathbf{b}^{(n)})$ can not be taken arbitrarily if we want to guarantee convergence. Indeed, they appear in the expectations over the index m-tuples in $\overline{h}^{m}_{w,P}$ and we need to bound them to apply Hoeffding's lemma. Hence we rely on local constraints that the probability vectors $\mathbf{a}$ and $\mathbf{b}$ must verify.

Definition 25 (Local averages conditions)**.** *Let $(\mathbf{a}^{(n)}) \in \Sigma$ and two integers $n, m \in \mathbb{N}^*$ such as $n \geq m$. We say that $(\mathbf{a}^{(n)})$ verifies the local mean condition if there exists a constant $D > 0$ and $\gamma \in (0, 1]$ such that for any $n \in \mathbb{N}^*$ and $I \subset [\![n]\!]$ with $|I| = m$ we have:*

$$\frac{1}{m}\sum_{i\in I}\mathbf{a}_i^{(n)} \leq \frac{D}{n^\gamma}. \tag{8.26}$$

We write that $(\mathbf{a}^{(n)})$ satisfies $\texttt{Loc}_\texttt{A}(m,\gamma,D))$ (or $\texttt{Loc}_\texttt{A}(m,\gamma)$) when the constant D is implicit).
(ii) *Analogously, $(\mathbf{a}^{(n)})$ is said to verify the local geometric mean condition if there exists a constant $D > 0$ and $\gamma > 0$ such that for any $n \in \mathbb{N}^*$ and $I \in [\![n]\!]^m$ we have*

$$\left(\Pi_{i\in I}\mathbf{a}_i^{(n)}\right)^{\frac{1}{m}} \leq \frac{D}{n^\gamma}. \tag{8.27}$$

We write that $(\mathbf{a}^{(n)})$ verifies $\texttt{Loc}_\texttt{G}(m,\gamma,D))$ (or $\texttt{Loc}_\texttt{G}(m,\gamma)$) when the constant D is implicit).

Intuitively, these conditions require that the product or the sum of any m elements of $(\mathbf{a}^{(n)})$ do not get too big. A straight forward example is the uniform vector $\boldsymbol{u}^{(n)}$ which respects $\texttt{Loc}_\texttt{A}(m,1,1)$ for the local mean condition and $\texttt{Loc}_\texttt{G}(m,1,1)$ for the local product condition. Thus Eq.(8.26) naturally extends and quantifies the fact that a sequence has uniformly controlled m-averages. We also observe that for any generic sequence $(\mathbf{a}^{(n)})$ in Σ verifies $\texttt{Loc}_\texttt{A}(m,0,\frac{1}{m})$. Regarding the local product condition, Eq.(8.27) extends the fact that a sequence has uniformly controlled m-products. We illustrate the local constraints on the simplex in Figure 8.6 with python ternary [Harper 2017]. We have the following result about the local constraints:

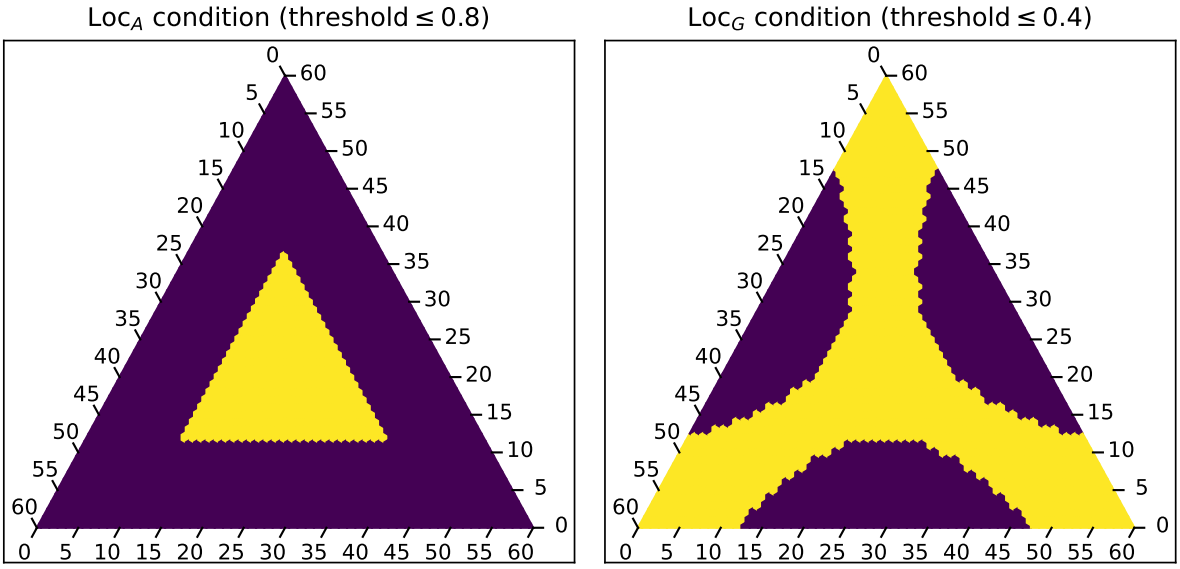


Figure 8.6: $\texttt{Loc}_\texttt{A}$ and $\texttt{Loc}_\texttt{G}$ local constraints illustrations on the simplex with $m = 2$ and $n = 3$.

Lemma 8.3.1. *Let $m \in \mathbb{N}^*$, $\gamma > 0$ and $D > 0$. Let $(\mathbf{a}^{(n)}) \in \Sigma$ be a sequence of probability vectors. The following statements hold:*
(i) *If $(\mathbf{a}^{(n)})$ verifies $\texttt{Loc}_\texttt{A}(m, \gamma, D)$ or $\texttt{Loc}_\texttt{G}(m, \gamma, D)$ then $\gamma \leq 1$.*
(ii) *If $(\mathbf{a}^{(n)})$ is $\texttt{Loc}_\texttt{A}(m, \gamma, D)$ then $(\mathbf{a}^{(n)})$ is $\texttt{Loc}_\texttt{G}(m, \gamma, D)$.*

Bounded data. For bounded data, we show that in order to obtain reasonable convergence properties of the estimators $\overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ we need to ensure that the sequences $(\mathbf{a}^{(n)})$ and $(\mathbf{b}^{(n)})$ verify the *local condition* with enough decay, e.g. $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ are $\texttt{Loc}_\texttt{A}(m, \gamma)$ or $\texttt{Loc}_\texttt{G}(m, \gamma)$ for a γ sufficiently close to 1.

Theorem 8.3.1 (Maximal deviation bound for compactly supported distributions)**.** *Let $\delta \in (0, 1)$, $k \geqslant 1$ an integer and $m \geqslant 1$ be a fixed integer. Let $C = C^{n,p}$ be as in (8.2). Consider two compactly supported distributions α, β, two n-tuples of empirical* i.i.d. *data $\boldsymbol{X} \sim \alpha^{\otimes n}, \boldsymbol{Y} \sim \beta^{\otimes n}$ and a kernel $h \in \{W_p, W_p^p, \mathfrak{L}^\varepsilon, S^\varepsilon, \mathcal{GW}\}$. Let the reweighting function w and the probability law over m-tuple P be as in Examples 1, 2, 3, 4. Let the sequences of probability vectors $(\mathbf{a}^{(n)}) \in \Sigma$ and $(\mathbf{b}^{(n)}) \in \Sigma$ satisfy $\texttt{Loc}_\texttt{A}(m, \gamma, D)$ and let $D > 0$ and $\gamma \in (\frac{3}{4}, 1]$. We have a deviation bound for the sampling without replacement between $\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ and $\mathbb{E}\overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ depending on the number of empirical data n and the number of batches k:*

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\Big(2\frac{D^2 m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}\sqrt{2\log(\frac{2}{\delta})} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\Big)\right) \leq \delta, \tag{8.28}$$

where M is a constant depending on the diameters of distribution supports. And for the sampling with replacement, let the sequences of probability vectors $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ verify $\texttt{Loc}_\texttt{G}(m, \gamma, D)$ for some

$\gamma \in (1 - \frac{1}{4m}, 1]$ *and* $D > 0$*, we have:*

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^{m}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\Big(2\frac{D^{2m} m^{\frac{1}{2}}}{n^{2m(\frac{1}{4m}-1+\gamma)}}\sqrt{2\log(\frac{2}{\delta})} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\Big)\right) \leq \delta. \tag{8.29}$$

Remark 9. *In the case of uniform measures, we recover the sampling without replacement bounds of Section 8.1.1 for both sampling with or without replacement:*

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{\mathtt{W}}(\boldsymbol{u}^{(n)}, \boldsymbol{u}^{(n)}) - \mathbb{E}\overline{h}^{m}_{\mathtt{W}}(\boldsymbol{u}^{(n)}, \boldsymbol{u}^{(n)})| \geq M\Big(2\sqrt{2\frac{m}{n}\log(2/\delta)} + \sqrt{\frac{2\log(2/\delta)}{k}}\Big)\right) \leq \delta,$$

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{\mathtt{U}}(\boldsymbol{u}^{(n)}, \boldsymbol{u}^{(n)}) - \mathbb{E}\overline{h}^{m}_{\mathtt{U}}(\boldsymbol{u}^{(n)}, \boldsymbol{u}^{(n)})| \geq M\Big(2\sqrt{2\frac{m}{n}\log(2/\delta)} + \sqrt{\frac{2\log(2/\delta)}{k}}\Big)\right) \leq \delta.$$

The proof is based on the U-statistics concentration bound proof [Hoeffding 1963] and can be found in Appendix A.1.2 with the proof of constant M. The proof idea is to rewrite our minibatch OT losses as a sum of independent terms and then to apply Hoeffding's lemma to the rewritten sum. The local constraints were necessary for a generalization of the concentration bounds to non uniform probability vectors $\mathbf{a}^{(n)}$ and $\mathbf{b}^{(n)}$. These concentration bounds are also valid for our debiased minibatch OT loss as it is composed of three terms of the form $\overline{h}$. Furthermore, it is possible to extend this concentration inequality with an expectation over the batch couples and empirical data.

Corollary 1. *With the same hypothesis and notations as in Theorem 8.3.1. The following inequality holds:*

$$\mathbb{E}[|\widetilde{h}^{m,k}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^{m}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})|] \leqslant 20 \cdot M \max\left(2\sqrt{2}D^2\frac{m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}, \sqrt{\frac{2}{k}}\right), \tag{8.30}$$

$$\mathbb{E}[|\widetilde{h}^{m,k}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^{m}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})|] \leqslant 20 \cdot M \max\left(2\sqrt{2}D^{2m}\frac{m^{\frac{1}{2}}}{n^{1/2-2m+2m\gamma}}, \sqrt{\frac{2}{k}}\right). \tag{8.31}$$

And for our debiased minibatch OT loss:

$$\mathbb{E}[|\widetilde{\Lambda}^{h,m,k}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\widetilde{\Lambda}^{h,m,k}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})|] \leqslant 40 \cdot M \max\left(2\sqrt{2}D^2\frac{m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}, \sqrt{\frac{2}{k}}\right), \tag{8.32}$$

$$\mathbb{E}[|\widetilde{\Lambda}^{h,m,k}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\widetilde{\Lambda}^{h,m,k}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})|] \leqslant 40 \cdot M \max\left(2\sqrt{2}D^{2m}\frac{m^{\frac{1}{2}}}{n^{1/2-2m+2m\gamma}}, \sqrt{\frac{2}{k}}\right). \tag{8.33}$$

This deviation bound shows that if we increase the number of data n and batches k while keeping the minibatch size fixed, we get closer to the expectation. Remarkably for all OT kernel h, the bound does not depend on the dimension of $\mathcal{X}$, which is an appealing property when data lie in high dimensional space. A similar property was proven but only for W_p^p (see proposition 20, [Weed 2019]). Another nice property of the bounds above is that for a fixed minibatch size m, if one chooses k proportional to the number of samples, the convergence of $\widetilde{h}^{m,k}_{w,P}$ to its mean is in $O(n^{-1/2})$ for a $O(n)$ computational complexity.

Now let us consider a small experiments. To illustrate the dependence to the dimension, we consider 2 empirical data n-tuple, $\boldsymbol{X}$ and $\boldsymbol{Y} \sim \alpha^{\otimes n}$, where α is the uniform distribution on the unit cube $[0,1]^d$, and compute $\Lambda_h(\boldsymbol{u}, \boldsymbol{u})$ as a function of n. For a first experiment, we fix the batch size $m = 128$ and we consider several values of dimension d. For a second experiment, we now fix the dimension d and consider several batch sizes. Both experiments highlight no dependence of $\Lambda_h(\boldsymbol{u}, \boldsymbol{u})$ to the dimension. To the best

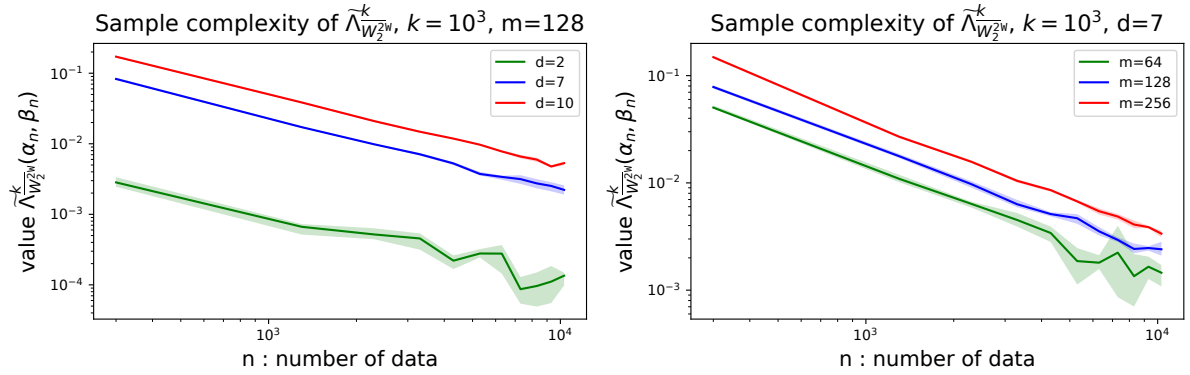


Figure 8.7: $\widetilde{\Lambda}^{h,m,k}_{\mathtt{W}}(\mathbf{a}, \mathbf{b})$ as a function of n in log-log space. Here $(\mathbf{a}, \mathbf{b})$ are two probability vectors associated to $\boldsymbol{X}$ and $\boldsymbol{Y} \sim \alpha^{\otimes n}$, where α is the uniform distribution on the unit cube $[0,1]^d$. (Left) $\widetilde{\Lambda}^{h,m,k}_{\mathtt{W}}(\mathbf{a}, \mathbf{b})$ is tested for several values of $d \in \{2, 7, 10\}$ and a fix $m = 128$ or (right) $\widetilde{\Lambda}^{h,m,k}_{\mathtt{W}}(\mathbf{a}, \mathbf{b})$ is tested for several values of $m \in \{64, 128, 256\}$ and a fix $d = 7$. The experiments were run 5 times and the shaded bar corresponds to the 20% and 80% percentiles.

of our knowledge, it is the first time that a loss using the exact Wasserstein distance has no dependence on the dimension, making it a good candidate for learning problems.

We gave concentration bounds in the bounded data case and now we extend these results to the unbounded data case.

Unbounded data. We supposed in the previous results that the distributions have a bounded support. We can relax this condition by supposing they have light tails, i.e., they are sub-Gaussian. We consider the Euclidean norm ($\|.\|_2$) and give a formal definition:

Definition 26 (sub-Gaussian random vectors)**.** *A random vector $\boldsymbol{x} \in \mathbb{R}^d$ is sub-Gaussian, if there exists $\sigma \in \mathbb{R}$ so that:*

$$\mathbb{E}e^{\langle \mathbf{v}, \boldsymbol{x} - \mathbb{E}\boldsymbol{x} \rangle} \leq e^{\frac{\|\boldsymbol{y}\|^2 \sigma^2}{2}}, \quad \forall \boldsymbol{y} \in \mathbb{R}^d$$

The proof uses a related class of sub-Gaussian random vectors and a discussion of the difference is available in appendix. In the case of sub-Gaussian data, we can not rely on the Hoeffding inequality anymore as the data are not bounded. However we are able to get a similar concentration inequality. Hereafter we write $A \ll_\gamma B$ for $A, B > 0$ if there exists a large constant $\tau = \tau(\gamma) > 0$ such that $A \leq \tau B$.

Theorem 8.3.2 (Concentration inequality sub-Gaussian data)**.** *Let the cost $C = C^{n,p}$ be defined as in (8.2). Let $(\boldsymbol{x}_i)_{1 \leq i \leq n}$ and $(\boldsymbol{y}_i)_{1 \leq i \leq n}$ be two i.i.d. sequences of random vectors such that $\boldsymbol{x}_1 \in \mathrm{normSG}(\rho_{\boldsymbol{x}}, \sigma^2_{\boldsymbol{x}})$ and $\boldsymbol{y}_1 \in \mathrm{normSG}(\rho_{\boldsymbol{y}}, \sigma^2_{\boldsymbol{y}})$ with $\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{y}} > 0$ and $\rho_{\boldsymbol{x}}, \rho_{\boldsymbol{y}} \in \mathbb{R}^d$. Let us introduce*

$$\sigma := \min(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{y}})$$
$$\rho := \|\rho_{\boldsymbol{x}} - \rho_{\boldsymbol{y}}\|_2$$

Let the sequence probability vectors $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ verify $\mathtt{Loc}_{\mathtt{A}}(m, \gamma, D)$ for some $\gamma \in (\frac{3}{4}, 1]$ and $D > 0$. We assume that n verifies the following condition:

$$n \geq \tau(m, \sigma, \rho, D, p). \tag{8.34}$$

Consider $m \geqslant 1$ be a fixed integer and a kernel $h \in \{W_p, W^\varepsilon, S^\varepsilon\}$. Let the reweighting function $w^{\mathtt{W}}$ and the probability law over m-tuple $P^{\mathtt{W}}$ be as in Examples 2 and 4. Then we have the following concentration bound for the sampling without replacement:

$$\mathbb{P}\left(\left|\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})\right| \geq (2^{3p+4}m)^{\frac{1}{2}}\sigma^p D^2 \cdot \frac{\log(4n)^{\frac{p+1}{2}}}{n^{2(\gamma-\frac{3}{4})}}\right) \leq 4n^{-\frac{1}{2^p}}, \tag{8.35}$$

The proof can be found in Appendix A.1.3. It uses a truncation argument where we split data which lie in a compact and data which do not. We can remark the following facts about our bounds in the unbounded case:

- The decay and the constants in (8.35) are artificial consequences of the constants chosen in the proof.
- The condition (8.34) seems to be necessary. In the bounded case of Theorem 8.3.1 such a condition was not needed.
- We loose a $\sqrt{\log(n)}$ factor between (8.35) and the deviation bound between the complete estimator $\overline{h}$ and its mean from the compactly supported data case. We report to a longer discussion on this comparison in Appendix A.1.3.

Bounded cost. We finish this section by discussing the case where we have a bounded cost. In this case, it is possible to get deviation bounds of the form of Theorem 8.3.1 for *any* distribution. This case is not unrealistic because it is easy to to bound a cost. For instance, consider an unbounded ground cost $c(\boldsymbol{x}, \boldsymbol{y})$, then you can consider the ground cost $\mathfrak{c}(\boldsymbol{x}, \boldsymbol{y}) = \frac{c(\boldsymbol{x},\boldsymbol{y})}{1+c(\boldsymbol{x},\boldsymbol{y})}$ as a new ground cost which is bounded. Furthermore if c is a distance, then $\mathfrak{c}$ is also a distance. Thus we have:

Theorem 8.3.3 (Maximal deviation bound for bounded cost)**.** *Let $\delta \in (0,1)$, $k \geqslant 1$ an integer and $m \geqslant 1$ be a fixed integer. Consider two distributions α, β possibly unbounded, two n-tuples of empirical data $\boldsymbol{X} \sim \alpha^{\otimes n}, \boldsymbol{Y} \sim \beta^{\otimes n}$ and a kernel $h \in \{W_p, W_p^p, \mathfrak{L}^\varepsilon, S^\varepsilon, \mathcal{G}W\}$ with a bounded ground cost C. Let the reweighting function w and the probability law over m-tuple P be as in Examples 1, 2, 3, 4. Let the sequences of probability vectors $(\mathbf{a}^{(n)}) \in \Sigma$ and $(\mathbf{b}^{(n)}) \in \Sigma$ satisfy $\mathtt{Loc}_{\mathtt{A}}(m, \gamma, D)$ and let $D > 0$ and $\gamma \in (\frac{3}{4}, 1]$. We have a deviation bound for the sampling without replacement between $\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ and $\mathbb{E}\overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ depending on the number of empirical data n and the number of batches k:*

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\Big(2\frac{D^2 m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}\sqrt{2\log(\frac{2}{\delta})} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\Big)\right) \leq \delta, \tag{8.36}$$

where M is an upper bound on the ground cost. And for the sampling with replacement, let the sequences of probability vectors $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ verify $\mathtt{Loc}_{\mathtt{G}}(m, \gamma, D)$ for some $\gamma \in (1 - \frac{1}{4m}, 1]$ and $D > 0$, we have:

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\Big(2\frac{D^{2m} m^{\frac{1}{2}}}{n^{2m(\frac{1}{4m}-1+\gamma)}}\sqrt{2\log(\frac{2}{\delta})} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\Big)\right) \leq \delta. \tag{8.37}$$

After studying concentration bounds for the minibatch OT loss in the bounded, unbounded data and bounded cost cases, we give similar bounds for the minibatch OT plan.

Minibatch Transport plan. As discussed before an interesting output of Minibatch Wasserstein is the minibatch OT plan $\overline{\Pi}^{h,m}_{w,P}$, but since it is hard to compute in practice, we instead use $\widetilde{\Pi}^{h,m,k}_{w,P}$ and we investigate the error on the marginal constraints. In what follows, we denote $\Pi_{(i)}$ the i-th row of matrix Π and $\mathbf{1} \in \mathbb{R}^n$ the vector whose entries are all equal to 1.

Theorem 8.3.4 (Distance to marginals)**.** *Let $\delta \in (0,1)$, two integers $m \leq n$ and consider two sequences of probability vectors $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)}) \in \Sigma$. Let a ground cost $C = C^{m,p}$ be as in (8.2) for some $p \geq 1$. Consider an OT kernel $h \in \{W_p, W_p^p, W^\varepsilon, S^\varepsilon, \mathcal{G}W\}$.. Suppose now that the probability law over m-tuples P and the reweighting function w, as defined in (8.12) and (8.13), satisfy the admissibility condition (8.21). For all integers $k \geqslant 1$ and all integers $1 \leqslant i \leqslant n$, we have:*

$$\mathbb{P}\left(\left|\widetilde{\Pi}^{h,m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})_{(i)}\mathbf{1} - a_i^{(n)}\right| \geq \sqrt{\frac{2\log(2/\delta)}{k}}\right) \leq \delta \tag{8.38}$$

The proof uses the convergence of $\widetilde{\Pi}^{h,m,k}_{w,P}$ to $\overline{\Pi}^{h,m}_{w,P}$ and the fact that $\overline{\Pi}^{h,m}_{w,P}$ is a transport plan and respects the marginals. It is far easier to get this result as we always have bounded transport plan. Let us now study the convergence of stochastic algorithm when minimizing MBOT.

8.3.2 Unbiased gradients for stochastic optimization

Thanks to the statistical properties of our estimator, we now know that minibatch OT losses concentrate well around their expectation. We now study their behaviour with modern optimization techniques. We consider the same data fitting problem as in Section 7.2.2. We recall that we aim at finding the minimum

$$\hat{\theta} = \arg\min_{\theta \in \theta} \quad h(\alpha_n, \beta_\theta, c), \tag{8.39}$$

where α_n is an empirical measure, β_θ a parametrized measure, h an OT loss and c a given ground cost. One way to compute the estimator $\hat{\theta}$ given by (8.39), is to use a stochastic solver for semi-discrete optimal transport [Peyré 2019, Chapter 5]. This strategy is unfortunately not efficient in practice [Genevay 2016, Seguy 2018]. A common alternative approach is to use stochastic gradient descent which were computed between minibatches sampled from α, β_θ [Genevay 2018, Salimans 2018]. We want to prove that the SGD algorithm would converge towards the correct $\hat{\theta}$. Unfortunately, OT is not differentiable with respect to the ground cost $\boldsymbol{C}$ as shown in Section 7.2.2. We thus rely on Clarke gradients defined in Section 7.2.2. The full formal statement and proof of those theorems can be found in Appendix A.1.5.

Theorem 8.3.5. *Let $\mathbf{a}, \mathbf{b} \in \Sigma_m$. Let $\boldsymbol{X}$ be a $\mathbb{R}^{dm}$-valued random variable, and $\{\boldsymbol{Y}_\theta\}$ a family of $\mathbb{R}^{dm}$-valued random variables defined on the same probability space, indexed by $\theta \in \Theta$, where $\Theta \subset \mathbb{R}^q$ is open. Assume that $\theta \mapsto \boldsymbol{Y}_\theta$ is C^1. Denote $C = C^{m,p}$ for some $p \geq 1$ and let $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon\}$. Then the function $\theta \mapsto -h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$ is Clarke regular and for all $1 \leq i \leq q$ we have:*

$$\begin{aligned}\partial_{\theta_i} h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta)) = \overline{co}\{-tr(P \cdot D^T) \cdot (\nabla_{\theta_i} Y) : P \in \Pi(h, C(\boldsymbol{X}, \boldsymbol{Y}_\theta), \mathbf{a}, \mathbf{b}),\\ D \in \mathbb{R}^{m,m}, D_{j,k} \in \partial_Y C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta)\}\end{aligned} \tag{8.40}$$

where ∂_{θ_i} is the Clare subdifferential with respect to θ_i, $\partial_Y C_{j,k}$ is the subdifferential of the cell $C_{j,k}$ of the cost matrix with respect to Y, $\overline{co}$ denotes the closed convex hull and $\mathrm{Opt}(h, C, \mathbf{a}, \mathbf{b})$ is defined in Definition 18. For $h = \mathcal{G}W$, when the cost matrix is differentiable (that is $p > 1$), the function $-h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{X}), C(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta))$ is also regular, and an analogous formula holds.

Theorem 8.3.6. *Let $\mathbf{a}, \mathbf{b}, \boldsymbol{X}, \boldsymbol{Y}$ be as in Theorem 8.3.5, and assume in addition that the random variables $\boldsymbol{X}, \{Y_\theta\}_{\theta\in\Theta}$ have finite p-moments. For $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon\}$, under an additional integrability assumption, we have:*

$$\partial_\theta \mathbb{E}\left[h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right] = \mathbb{E}\left[\partial_\theta h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right]. \tag{8.41}$$

with both expectation being finite. Furthermore the function $\theta \mapsto -\mathbb{E}\left[h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right]$ is also Clarke regular. An analogous result holds for $h = \mathcal{G}W$, given that the cost is differentiable (that is $p > 1$) and random variables $\boldsymbol{X}, \{Y_\theta\}$ have finite $2p$-moments.

Remark 10. *If the cost matrix in Theorem 8.3.5 is differentiable with respect to $\boldsymbol{Y}$ (that is for $p > 1$) and $h = W^\varepsilon$, then all the Clarke derivatives in (8.40), (8.41) are sets consisting of one element, which is the gradient of respective functions. In that case we may deduce for $h = S^\varepsilon$ a formula for the gradient from Theorem 8.3.5 and an interchange of expectation and integration from Theorem 8.3.6.*

Suppose that in the above theorem the random variable $\boldsymbol{X}$ is distributed according to $\alpha^{\otimes m}$, each random variable $\boldsymbol{Y}_\theta$ is distributed according to $\beta^{\otimes m}$ and $\boldsymbol{X}$ is independent of family of variables $\{\boldsymbol{Y}_\theta\}_{\theta\in\Theta}$. Then Theorem 8.3.5 implies that it is easy to compute unbiased stochastic gradients of a Minibatch OT loss, defined as follows:

Definition 27 (Minibatch OT)**.** *Let $\alpha, \beta \in \mathcal{P}_p(\mathbb{R}^n)$ be two measures on an Euclidean space with finite p-moments, for $p \geq 1$. Chose an integer $m \in \mathbb{N}$ and let $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon, S^\varepsilon\}$. Given the ground cost the ground cost $C^{m,p}$ defined in Eq.(8.2), we define the following quantity:*

$$E_h^m(\mathbf{a}, \mathbf{b}, \alpha, \beta) := \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim\alpha^{\otimes m}\otimes\beta^{\otimes m}}\left[\overline{h}^m_{w,P,C^{m,p}(\boldsymbol{X},\boldsymbol{Y})}(\mathbf{a}, \mathbf{b})\right] \tag{8.42}$$

for any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$. We define an analogous quantity For $h = \mathcal{G}W$. Assuming that $\alpha, \beta \in \mathcal{P}_{2p}(\mathbb{R}^n)$, we denote

$$E_h^m(\mathbf{a}, \mathbf{b}, \alpha, \beta) := \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim\alpha^{\otimes m}\otimes\beta^{\otimes m}}\left[\overline{h}^m_{w,P,C^{m,p}(\boldsymbol{X},\boldsymbol{X}),C^{m,p}(\boldsymbol{Y},\boldsymbol{Y})}(\mathbf{a}, \mathbf{b})\right] \tag{8.43}$$

for any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$.

The fact that the above is well defined follows trivially from the assumption that measures α, β have finite p-moments (or finite $2p$-moments for $h = GW$) and a standard bound (A.92) used in the proof of Theorem 8.3.6. In fact, the finiteness of (8.42) and (8.43) is shown in that proof. We finish this section by noting, that Theorem 8.3.6 implies that if we use the Minibatch OT loss with $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon, GW\}$ (or $h = S^\varepsilon$ for $p > 1$) as a contrast function in (7.5), then the objective function is minus Clarke regular. In this case, it is known that SGD with decreasing step sizes converges almost surely to the set of critical points of Clarke generalized derivative [Davis 2020], [Majewski 2018].

Accelerating stochastic methods Other solvers than SGD have been discussed to optimize U-statistics in [Papa 2015]. It is not realistic in our problem to rely on variance reduced algorithms such as SAG [Schmidt 2017b]. The later needs to store the past gradients, which leads to high memory cost, while the former needs to compute the full gradients on a regular basis, which leads to a high computation cost. However as we have a huge number of terms, one might hope to accelerate its convergence by considering a smaller number of data but it leads to poorer results at convergence.

8.4 Numerical experiments on generative modelling, color transfer and minibatch Gromov-Wasserstein

After presenting the formalism of minibatch OT, studied its statistical and optimization properties and defining a new loss function, we now explore different applications of our methods. To compare the minibatch OT losses and their debiased counter parts, we set three qualitative experiments and two quantitative ones. The first experiment is a gradient flow between male and female images and the second is a Monge map estimation between male and female images. The quantitative experiment consists in learning a GAN where we investigate the inception score of several minibatch OT losses. Our fourth experiment is a color transfer experiment, we also investigate the sparsity degree of the resulting minibatch OT plan. Finally, our two last experiments are dedicated to the minibatch Gromov-Wasserstein loss where we investigate the inherited properties from the Gromov-Wasserstein distance. As our experiments are learning scenarios, we have uniform measures and consider the reweighting function $w^{\mathtt{U}}$, regarding the probability laws on tuples, we investigate both $P^{\mathtt{W}}$ and $P^{\mathtt{U}}$. Note that $w^{\mathtt{U}}$ and $P^{\mathtt{W}}$ check the admissibility condition from Proposition (12). Finally, experiments were computed on a single GTX Titan GPU.

8.4.1 Gradient flows between human faces with minibatch Optimal Transport

The first experiment we conducted is a gradient flow of a source distribution towards a target distribution. It corresponds to the nonparametric setting of a data fitting experiments such as GANs. For two given probability vectors $\mathbf{a}$ and $\mathbf{b}$, and support $\boldsymbol{X}$, $\boldsymbol{Y}$ associated to $\mathbf{b}$, the goal of gradient flows is to model a support $\boldsymbol{x}_t$ associated to $\mathbf{a}$ which at each iteration follows the loss gradient $\boldsymbol{x}_t \mapsto h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}_t, \boldsymbol{Y}))$. This experiment has been investigated in [Liutkus 2019, Peyré 2015]. We apply it between male and female images from the celebA dataset [Liu 2015] where we seek a natural evolution along iterations. CelebA is a large-scale face attribute dataset with 202,599 face images, 5 landmark locations, and 40 binary attribute annotations per image. We only considered 5000 male images and 5000 female images. We build the training dataset by cropping and scaling the aligned images to 64 x 64 pixels.

Following the procedure in [Feydy 2019], the gradient flow algorithm uses an Euler scheme and we start from an initial distribution at time $t = 0$. At each iteration we numerically integrate the ordinary differential equation:

$$\dot{\boldsymbol{X}}(t) = -\nabla_{\boldsymbol{x}} \widetilde{\Lambda}^{h,m,k}_{w,P,C(\boldsymbol{X}(t),\boldsymbol{Y})}(\mathbf{a}, \mathbf{b}).$$

As our losses take probability vectors as inputs, we need to correct an inherent scaling when we calculate the gradient. The scaling comes from the sample weights a_i, which is equal to $1/m$ in our case. To correct the scaling, we apply a re-scaling to the gradient equal to m. Finally, for a n-tuple of data $\boldsymbol{X}$ we integrate:

$$\dot{\boldsymbol{X}}(t) = -m\nabla_{\boldsymbol{x}} \left[\widetilde{h}^{m,k}_{w,P,C(\boldsymbol{X}(t),\boldsymbol{Y})}(\mathbf{a}, \mathbf{b}) - \frac{1}{2}\left(\widetilde{h}^{m,k}_{w,P,C(\boldsymbol{X}(t),\boldsymbol{X}(t))}(\mathbf{a}, \mathbf{a}) + \widetilde{h}^{m,k}_{w,P,C(\boldsymbol{Y},\boldsymbol{Y})}(\mathbf{b}, \mathbf{b}) \right) \right]. \tag{8.44}$$

We conducted gradient flow experiments for both minibatch OT loss and debiased minibatch OT loss with the Wasserstein distance as OT kernel and probability laws on m-tuples $P^{\mathtt{W}}$ and $P^{\mathtt{U}}$. However, as our images lie in high dimension, the Euclidean ground cost is not meaningful anymore, that is why we followed the experiments of [Liutkus 2019] where they considered gradient flows in the latent space of a pre-trained AutoEncoder. We considered a pre-trained DFC-VAE [Hou 2017] with 64×64 image and perform gradient flow in the encoder's latent space. In addition of the typical [Kingma 2014] loss, DFC-VAE considers the difference between features of the input image and the reconstructed image through a pre-trained neural network. In our case, we considered a pre-trained VGG-19 network and the

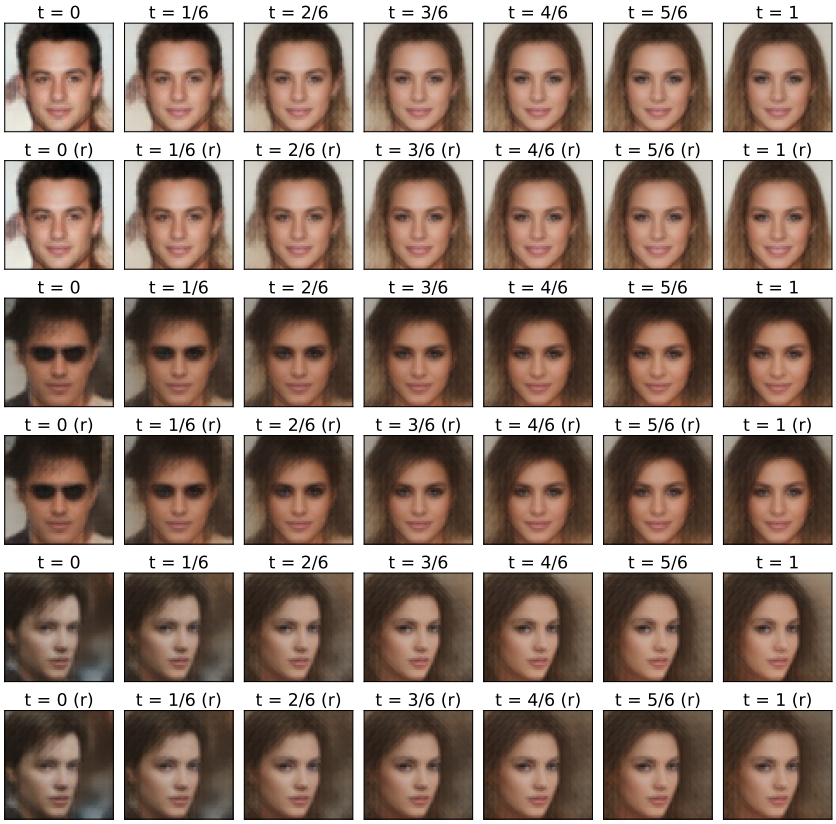


Figure 8.8: minibatch OT gradient flow on the CelebA dataset in a DFC-VAE latent space. Source data are 5000 male images while target data are 5000 female images. The batch size m is set to 200 and the number of minibatch k is set to 10. (r) means that the probability law on m-tuple is $P^{\mathtt{U}}$ otherwise it is $P^{\mathtt{W}}$.

layers 1-2-3. We trained the DFC-VAE with a batch size of 64 for 5 epochs over the training dataset and use Adam method for optimization with initial learning rate of 0.0005, see [Hou 2017] for more details. With the feature extraction, we are able to improve the quality of final distribution's images.

The minibatch OT loss produces blurred images at the end of the flow as shown in Figure 8.8, especially at the back of the image where all details are lost. This is due to the fact that the minibatch OT shrinks the distribution. On the contrary, the debiased minibatch OT reported in Figure 8.9 produced images with high background details, quality and coherence with respect to the original background. Moreover, the evolution seems more natural between the source and the target distributions.

8.4.2 Mapping estimation between human faces with minibatch Optimal Transport

While the previous application focused on updating samples, the second application is a continuous mapping estimation between source and target distributions that will allow transforming new samples that are not in the original training data. The map is parametrized by a neural network $f_\varphi : \mathcal{X} \to \mathbb{R}^p$ between the source data and the target data. The objective is to minimize the loss:

$$\min_{\varphi} \Lambda^{h,m}_{w,P,C(f_\varphi(\boldsymbol{X}),\boldsymbol{Y})}(\boldsymbol{u},\boldsymbol{u}), \tag{8.45}$$

Figure 8.9: Unbiased minibatch Wasserstein gradient flow on the CelebA dataset in a DFC-VAE latent space. Source data are 5000 male images while target data are 5000 female images. The batch size m is set to 200 and the number of minibatch k is set to 10. (r) means that the probability law on m-tuple is $P^{\mathtt{U}}$ otherwise it is $P^{\mathtt{W}}$.

Where $f_\varphi(\boldsymbol{X}) = \{f_\varphi(\boldsymbol{x}_1), \cdots, f_\varphi(\boldsymbol{x}_n)\}$. We apply this problem on the celebA dataset [Liu 2015]. We considered 5000 male and 5000 female images. The image size is 64×64. The goal is to learn how to transform a male image into a female one. Unfortunately, in order to avoid blurry images, we once again relied on the latent space, of dimension 100, of a pre-trained DFC-VAE [Hou 2017]. We used the same setting as described in the Gradient Flow section. We performed the training in the latent space and then we decoded the transformed samples. We consider a 4 dense layer neural network with ReLu activation function ($100 \to 1024 \to 1024 \to 512 \to 100$). The minibatch size m is set to 128, and we used the Adam optimizer [Kingma 2015] with a step size of $1e^{-4}$ and the coefficients $\beta_1 = 0$ and $\beta_2 = 0.9$.

We conducted the experiments for minibatch Wasserstein loss and for the debiased loss Λ_h. We spotted once again that the transformed samples with the minibatch Wasserstein losses are blurred (Figure 8.10). However, the results with the unbiased minibatch Wasserstein loss are more diverse and more realistic. It shows the effectiveness of the unbiased loss to debiased the minibatch Wasserstein losses (Figure 8.10). It is interesting to note that the estimated mapping are quite different on some images between losses which use a sampling with or without replacement.

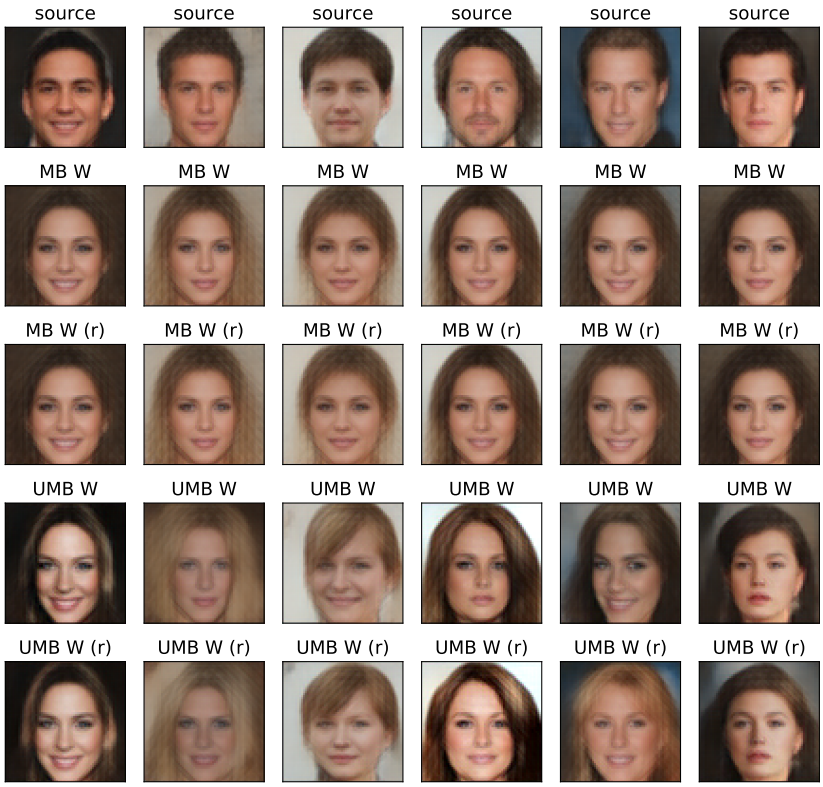


Figure 8.10: Map learning between 5000 male source images and 5000 female target images. The batch size m is set to 128 and the number of batch couple k is set to 1. (First row) Source data. (Second and third rows) Respectively minibatch Wasserstein without replacement and with replacement (r) mapping on the CelebA dataset in a DFC-VAE latent space. (Fourth and fifth rows) Respectively unbiased minibatch Wasserstein without and with replacement (r) mapping on the CelebA dataset in a DFC-VAE latent space.

8.4.3 Generative adversarial networks on Cifar-10 with minibatch Optimal Transport

In this experiment, we want to learn a GAN using our new loss function which is a debiased version of minibatch SD. We recall that the objective of a GAN is to train a neural network G_θ that can generate realistic data which are close to real data $\boldsymbol{X}$. For the ground cost of the Wasserstein distance, we could rely on an Euclidean cost between images. Unfortunately, using an Euclidean cost on high dimensional images generates blurred versions of the real images [Aggarwal 2001, Liwei Wang 2005, Kulis 2013a]. Hence, we will learn adversarially a critic network f_φ which extracts meaningful feature vectors for input images. Then we will apply the Euclidean distance between the encoded generated data $f_\varphi(G_\theta(\boldsymbol{z}_j))$ and encoded real data $f_\varphi(\boldsymbol{x}_i)$. Other methods relied on a feature extractor such as MMD GAN [Li 2017a] of Sinkhorn GAN [Genevay 2018]. We can summarize our learning problem as the following:

$$\min_\theta \max_\varphi \quad \mathbb{E}\Lambda^{h,m}_{w,P,C_\varphi(\boldsymbol{X},G_\theta(\boldsymbol{Z}))}(\boldsymbol{u},\boldsymbol{u}), \tag{8.46}$$

$$\text{where} \quad C_\varphi(\boldsymbol{x}_i,\boldsymbol{y}_j) \overset{\text{def.}}{=} \|f_\varphi(\boldsymbol{x}_i) - f_\varphi(\boldsymbol{y}_j)\|_2 \quad \text{and} \quad f_\varphi : \mathcal{X} \to \mathbb{R}^p.$$

Generator	Critic
INPUTS: 100	INPUTS: $3 \times 32 \times 32$
Conv2D$^\intercal$ nc=256 k=4 stride=1, BN, ReLU	Conv2D nc=64 k=4 stride=2, LReLU(slope=0.2)
Conv2D$^\intercal$ nc=128 k=4 stride=2, BN, ReLU	Conv2D nc=128 k=4 stride=2, LReLU(slope=0.2)
Conv2D$^\intercal$ nc=64 k=4 stride=2, BN, ReLU	Conv2D nc=256 k=4 stride=2, LReLU(slope=0.2)
Conv2D$^\intercal$ nc=3 k=4 stride=2, TanH	Conv2D nc=100 k=4 stride=1, LReLU(slope=0.2)

Table 8.1: Generator (left) and critic (right) 4 convolutional layer architectures used in our experiments to generate CIFAR10 data.

Where $G_\theta(\boldsymbol{Z}) = \{G_\theta(\boldsymbol{z}_1), \cdots, G_\theta(\boldsymbol{z}_m)\}$ and $\boldsymbol{u} \in \mathbb{R}^m$. We train GAN for image generation of CIFAR-10 data [Krizhevsky]. The number of data is 50K of size 32×32. Regarding the implementation detail, we consider the same setting as [Li 2017a, Genevay 2018]. The input noise is of dimension 100. The generator and the critic have 4 convolution layers (full detail in Table 8.1). We clip the parameters of the critic in order to have a Lipschitz constant bounded by 1 as done in [Li 2017a, Genevay 2018]. The batch size m we considered is 64 and we set the number of batch couple to $k = 1$ for each SGD update. The optimizer we used is RMSProp [Tieleman 2012a] with a learning rate of 5.10^{-5}. Regarding the entropic regularization parameter for the Sinkhorn divergence, we set it in $\{10, 100, 1000\}$. We update the discriminator 5 times before one update of the generator. The code can be found here*.

We compare our method to 4 different methods: WGAN-GP [Gulrajani 2017], Sinkhorn GAN, OT-GAN [Salimans 2018] and MMD GAN. Regarding Sinkhorn GAN we use a batch size of 256 as done in their work. We also compared our GAN to the effective WGAN-GP, we considered the same architecture as above but we used the hyperparameters described in their paper [Gulrajani 2017]. Finally for OT-GAN [Salimans 2018], we used an entropic regularization parameter set to 500 and for fair comparison with other methods, we set the batch size to 256. In their paper authors used batch size of 8000 images to get a more stable training, however this method is not reproducible in our setting with a single GPU. We report Inception scores in Table 8.2. As we can see, the debiased minibatch Sinkhorn divergence gives the best Inception score showing the relevance of this new loss function. Comparing to the typical Sinkhorn GAN, the debiased strategy increases the inception score by 1 point. Furthermore, it seems that regularizing the problem with the entropic regularization helps to get better performance as already suggested in previous work [Genevay 2018]. We also report in Figure 8.11 some generated examples from MBSD, UMBSD and WGAN models and we can see that UMBSD leads to slightly more detailed samples than MBSD and more realistic than WGAN-GP.

8.4.4 Large scale color transfer between images

The purpose of color transfer is to transform the color of a source image so that it follows the color of a target image. Optimal transport is a well known method to solve this problem and has been studied before in [Ferradans 2013, Blondel 2018]. Images are represented by point clouds in the RGB color space. Then by calculating the transport plan between the two point clouds, we get a transfer color mapping by using a barycentric projection. As the number of pixels might be huge, previous work selected a subset of pixels using k-means clusters for each point cloud. This strategy allows to make the problem memory tractable but looses some information to the quantification. With MB optimal transport, we can compute a barycentric mapping for all pixels in the image by incrementally updating the full transported vector at

*https://github.com/kilianFatras/unbiased_minibatch_sinkhorn_GAN

Methods	Inception score
WGAN-GP	4.59 ± 0.07
MBSD ($\varepsilon = 10$)	3.57 ± 0.03
MBSD ($\varepsilon = 100$)	3.61 ± 0.05
MBSD ($\varepsilon = 1000$)	3.83 ± 0.05
OT-GAN ($\varepsilon = 500$)	4.13 ± 0.08
MMD	4.29 ± 0.06
UMBW (ours)	4.38 ± 0.08
UMBSD ($\varepsilon = 10$) (ours)	4.72 ± 0.07
UMBSD ($\varepsilon = 100$) (ours)	$\mathbf{4.76} \pm 0.08$
UMBSD ($\varepsilon = 1000$) (ours)	4.67 ± 0.08

Table 8.2: Inception Scores on CIFAR10 for several GAN variants trained with a batch size of 64. Biggest score is in bold.

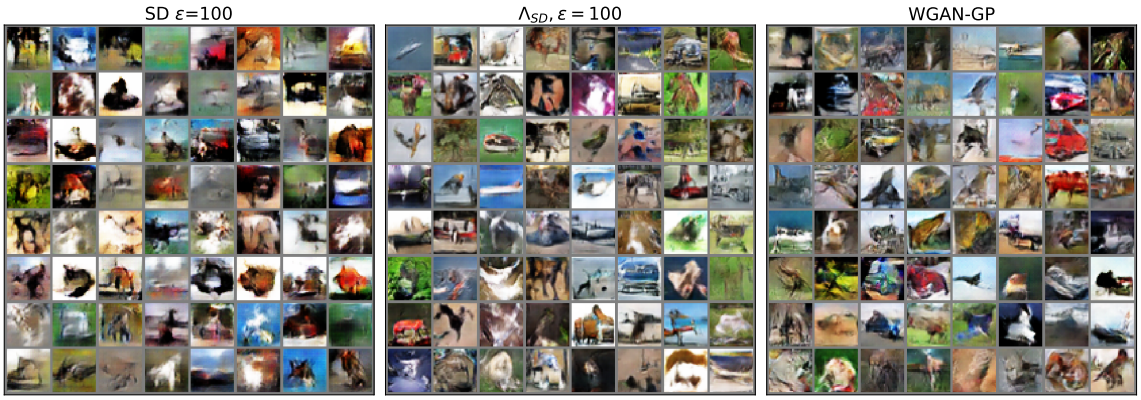


Figure 8.11: Generated samples with the same latent vectors from different GANs.

each minibtach. When one selects a source indices m-tuple I_1 and a target m-tuple I_2, she just needs to update the transported vector between the considered minibatches as $Q_{I_1} Y_s = \sum_{I_2} \Pi^m_{I_1,I_2} Q_{I_2} X_t$, with matrix Q_{I_1} and Q_{I_2} defined as in Definition 18. Indeed, the incremental computation can be rewritten as:

$$Y_s = n_s \widetilde{\Pi}_{\mathtt{W}}^{W_2^2,m,k}(\boldsymbol{u}, \boldsymbol{u}) X_t, \tag{8.47}$$

when we use the incomplete MBOT plan $\widetilde{\Pi}_{\mathtt{W}}^{W_2^2,m,k}$. To the best of our knowledge, it is the first time that a barycentric mapping algorithm has been scaled up to 1M pixel images. About the required memory for experiments, the memory cost to store data is $O(n)$. The minibatch OT calculus requires $O(m^2)$ because we need to store the ground cost and the OT plan. The marginal experiment requires $O(n)$, as we just need to average the marginals of the plan. Finally, the memory cost is $O(n)$ while exact OT would be $O(n^2)$.

The source image has (943000, 3) RGB dimension and the target image has RGB dimension (933314, 3). For this experiments, we compare the results between the minibatch framework with the Wasserstein distance for several m and k. We used batch of size 10, 100 and 1000. We selected k so as to obtain a good visual quality and observed that a smaller k was needed when using large minibatches. Also

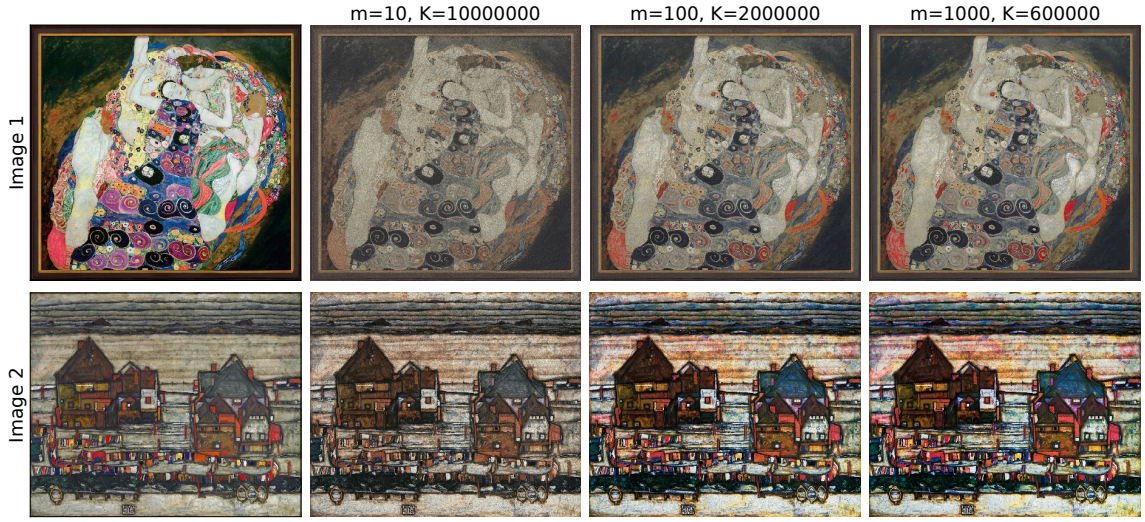


Figure 8.12: Color transfer between full images for different batch size and number of batches. (Top) color transfert from image 1 to image 2. (Bottom) color transfer from image 2 to image 1.

note that performing MB optimal transport can be done in parallel and can be greatly speed-up on multi-CPU architectures. The code can be found here[†]. One can see in Figure 8.12 the color transfer (in both directions) provided with our method. We can see that the diversity of colors falls when the batch size is too small as the entropic solver would do for a large regularization parameter. However, even for 1M pixels, a batch size of 1000 is enough to keep a good diversity of colors.

From now on for speed constraints, we consider a selected subset of 1000 pixels using k-means clusters for each point cloud. We reproduced empirically the results of Theorem 8.3.4 about the marginal errors, as shown in Figure 8.13. We recover the $O(k^{-1/2})$ convergence rate on the marginal with a constant depending on the batch size m.

As we stated above, minibatch Wasserstein loss increases the number of connections similarly to regularized OT variants. Hence, we want to conduct a sparsity experiment of the minibatch Wasserstein transport plan and we report it for several settings. We considered batch sizes of 50, 100, 200, 350 and 500 and computed the sparsity of the incomplete minibatch OT plan with respect to several number of minibatches k. The results are gathered in Figure 8.13. We see that as m gets smaller, the degree of sparsity decreases and that the sparsity reaches a limit as the number of minibatches increases. Intuitively, it is expected as when m gets smaller, the number of connections increases. The results can be justified with the following facts. When the minibatch size between the source and target batches is the same and with uniform weights, then m coefficients of the transport matrix will be non null for the exact Wasserstein distance. As we draw k batch couples, such as $k.m < n$ and if we suppose that the batches define a disjoint union of the samples $\boldsymbol{X}$ and $\boldsymbol{Y}$, then we have at most km coefficients of $\widetilde{\Pi}_{\mathtt{W}}^{W_2^2,m,k}(\boldsymbol{u},\boldsymbol{u})$ non zero. In the case of non uniform weights $\mathbf{a},\mathbf{b}$, the positive linear program has a solution with at most $2m-1$ non zero coefficients. Then we have at most $k.(2m-1)$ coefficients of $\widetilde{\Pi}_{\mathtt{W}}^{W_2^2,m,k}(\mathbf{a},\mathbf{b})$ non zero.

[†] https://github.com/kilianFatras/minibatch_Wasserstein

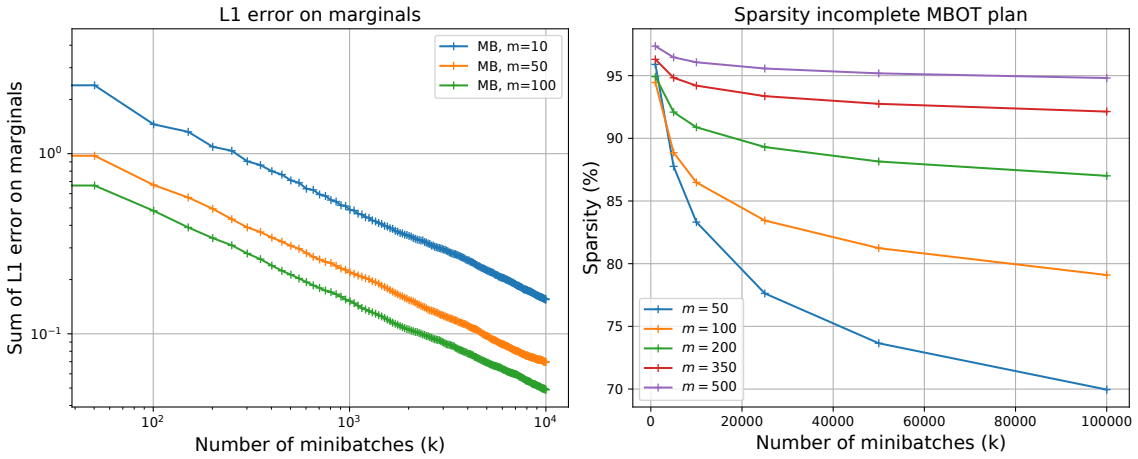


Figure 8.13: (left) L1 error on both marginals (log-log scale). We selected 1000 points from original images and computed the error on marginals for several m and k (log-log scale). (Right) Sparsity of incomplete minibatch OT plan $\widetilde{\Pi}_{\mathtt{W}}^{W_2^2,m,k}(\boldsymbol{u},\boldsymbol{u})$. We selected 1000 points from original images and computed the sparsity of $\widetilde{\Pi}_{\mathtt{W}}^{W_2^2,m,k}(\boldsymbol{u},\boldsymbol{u})$ for several k and m.

8.4.5 Minibatch Gromov-Wasserstein rotation and translation invariance

The Gromov-Wasserstein distance has the nice properties to be rotational and translation invariant, so in this section we study if the minibatch Gromov-Wasserstein loss (MBGW) shares the same properties. As shown in the previous section, our statistical results can be extended to the Gromov-Wasserstein distance. We start with a spiral experiment where we compute the value of the MBGW loss for several rotations of the spirals. Then, we aim at checking if the MBGW loss is able to recover the motion of a galloping horse on a dataset containing a sequence of shapes [Solomon 2016].

Rotational invariance. Our first result shows the stability of rotation and translation invariances with minibatches. We have the following results:

Proposition 14 (Invariance)**.** *The minibatch Gromov-Wasserstein* $\overline{\mathcal{GW}}^m_{w_1,w_2,P^1,P^2,C^1,C^2}$ *is rotation and translation invariant.*

Proof. Let $\mathbf{a}$ and $\mathbf{b}$ be two probability vectors with support $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively. Consider now the support $\boldsymbol{Y}'$ which is a rotation and a translation of $\boldsymbol{Y}$. Consider three ground costs $C^1 = C(\boldsymbol{X},\boldsymbol{X})$, $C^2 = C(\boldsymbol{Y},\boldsymbol{Y})$ and $C^3 = C(\boldsymbol{Y}',\boldsymbol{Y}')$. For fixed minibatches I and J, as $\boldsymbol{Y}'$ is a translation and rotations of $\boldsymbol{Y}$, we have:

$$\mathcal{GW}\Big(w_1(\mathbf{a},I), w_2(\mathbf{b},J), C^1_{I,I}, C^2_{J,J}\Big) = \mathcal{GW}\Big(w_1(\mathbf{a},I), w_2(\mathbf{b},J), C^1_{I,I}, C^3_{J,J}\Big),$$

summing over all minibatch couples finishes the proof.

□

Empirically, distances which are rotation invariant return a constant when comparing rotated distributions. To support the proposition, we consider a small spiral experiment for different rotations of the target distribution. We follow the procedure in [Vayer 2019b]. The source and the target distributions are spirals taken from the scikit-learn spiral dataset [Pedregosa 2011b]. We compute Gromov-Wasserstein

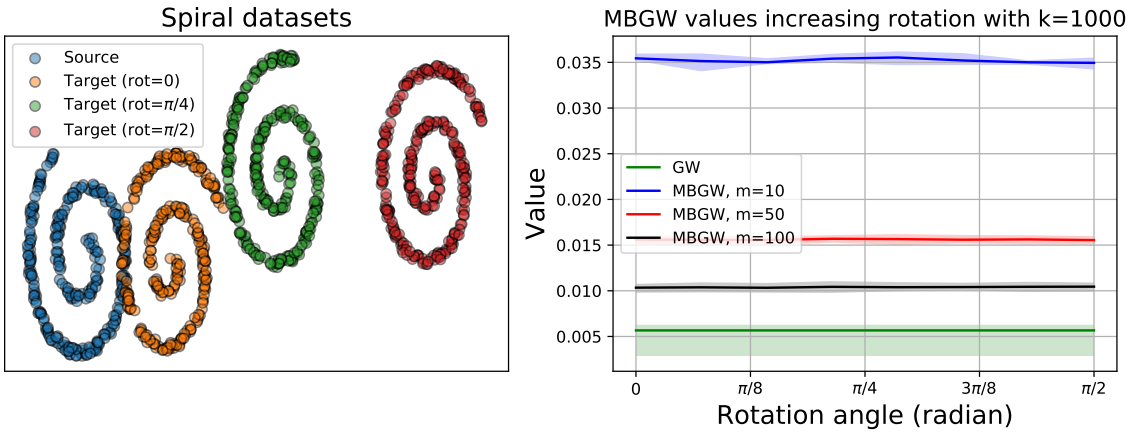


Figure 8.14: Average value of MBGW and GW losses as a function of rotation angle on 2D spirals. Colored areas correspond to the 20% and 80% percentiles. Experiments were run 10 times.

distance and the MBGW loss on $n = 300$ samples. We report in Figure 8.14 the average values of the GW and MBGW losses for a varying angle and we can see that it is in practice invariant to rotation. From the Figure 8.14, one recovers that the MBGW loss returns a constant, depending on the minibatch size m, and hence is rotation invariant.

Meshes comparison In the context of computer graphics, Gromov-Wasserstein distance can be used to measure similarities between two meshes [Peyré 2016, Solomon 2015, Vayer 2019b]. It can also be used for shape matching, search, exploration or organization of databases. As minibatch GW loss and its debiased counter parts are not distances, we want to know if they are meaningful for use in a context of meshes comparison. From a time series of 45 meshes representing the motion of a galloping horse, we compute a multidimensional scaling (MDS) of the pairwise distances with minibatch GW losses, that allows plotting each mesh as a 2D point. Each horse mesh is composed of approximately 9, 000 vertices. The results can be found in Figure 8.15. As one can observe in Figure 8.15, the cyclical nature of this motion is successfully recovered in this 2D plot for both MBGW loss and its debiased counter parts.

Running time comparison Our last experiment is the time computation of minibatch Gromov-Wasserstein. We compare it to Gromov-Wasserstein distance, the entropic regularized Gromov-Wasserstein, the Sliced Gromov-Wasserstein and its rotational invariant variant [Vayer 2019b]. Unfortunately, the Sliced variant can only be computed for square Euclidean ground cost unlike the MBGW and is not rotational invariant. We calculate these distances between two 100-D random measures of $n \in 10^2, ..., 10^4$ points. For the minibatch Gromov-Wasserstein we consider two settings. The first setting is with a fixed number of minibatch couples ($k = 5000$) and the second is a linear setting where k grows linearly according to n ($k = \frac{n}{10}$). The latter is due to our concentrations bounds which decreases linearly in the number of samples if we consider a number of minibatch couples proportional to the number of samples (see Theorem 8.3.1). We use the Python Optimal Transport (POT) toolbox to compute GW distance on CPU. For entropic-GW we use the POT implementation with a regularization parameter of $\varepsilon = 0.01$. We were not able to compute transport plan for a bigger number of data than 10^4 for both GW and its entropic variant, with respect to our limited memory and time budget.

We see that MBGW enjoys a constant time computation. The sliced Gromov-Wasserstein and its

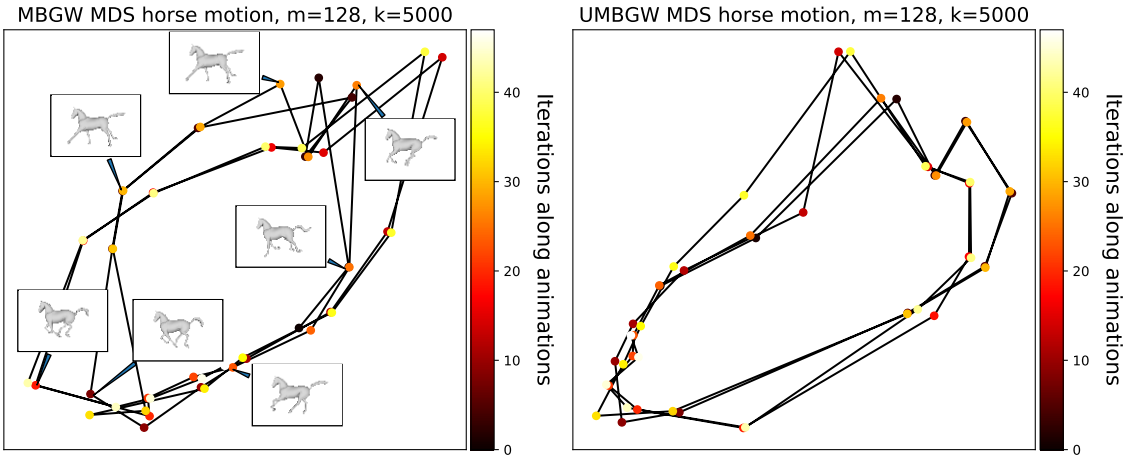


Figure 8.15: MDS on the galloping horse animation with MB Gromov-Wasserstein loss and its debiased variant. Each sample in this Figure corresponds to a mesh and is colored by the corresponding time iteration. One can see that the cyclical nature of the motion is recovered.

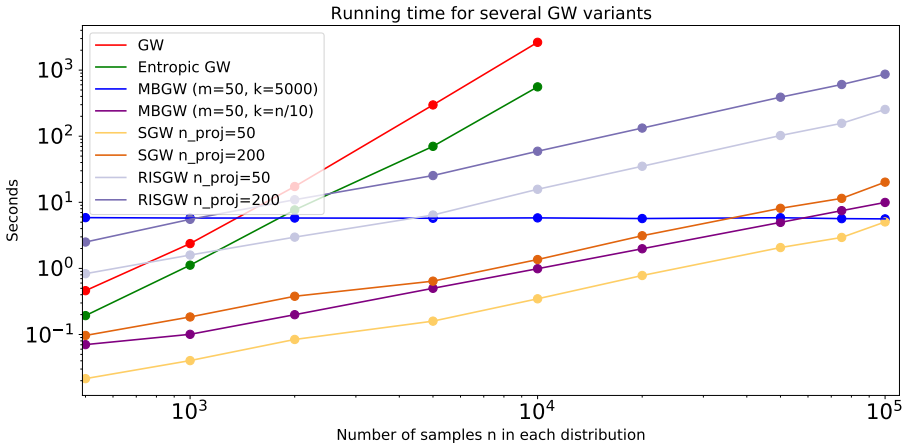


Figure 8.16: Runtimes comparison between MBGW, SGW, RISGW, GW, entropic-GW between two 100-D random distributions with varying number of points from 0 to 10^4 in log-log scale. The time includes the calculation of the pair-to-pair distances.

rotational variant grow in $\mathcal{O}(n\log(n))$ making it slower than the minibatch GW for large scale dataset. Regarding GW and its entropic counter part, we see that for $10e^4$ points, the MBGW is 100 time faster than GW.

8.5 Discussion: non-optimal connections consequences

In this chapter, we introduced a rigorous formalism of minibatch optimal transport. We first designed an estimator in the simple case of uniform empirical measures. Following this definition, we expressed a closed-form formula when data lie in 1D and illustrate the consequences on the transport plan. We showed that minibatch OT creates a bigger of connections between distributions as illustrated in Figure 8.1. From the formalism for uniform probability vectors, we extended the formalism for general probability vectors

as well as different sampling. We show the effectiveness of this general formalism to generate misclassified data for a pre-trained classifier as in ARWGAN from Chapter 6. Then we reviewed the basic metric properties and we introduced a new loss function which respects the separability axiom. Unfortunately, this loss is not always positive but always outperformed minibatch OT in practice. We also studied the deviation bounds of our estimators towards their expectation and their optimization properties. We indeed showed that we can optimize them using SGD. Finally, we extensively evaluated minibatch OT losses for generative modelling and color transfer. We also showed that minibatch Gromov-Wasserstein enjoys many properties of the Gromov-Wasserstein distance.

As illustrated in Figures 8.1 and 8.2, the number of connections increases with minibatch OT. Theses non optimal connections can connect two samples of different clusters and this is similar to a regularization effect. Furthermore, the intensity of these connections tend to be uniform as the minibatch size decreases. In the next chapter, we show that these non optimal connections can lead to undesirable effects for specific applications and we propose a simple manner to mitigate them.

Chapter 9

Unbalanced Minibatch Optimal Transport

Contents

In this chapter, we discuss and prove some limits of the minibatch OT framework due to the marginal constraints and the minibatch sampling which create non-optimal connections. We then give empirical and theoretical evidences that using unbalanced optimal transport at the minibatch level can help mitigating the aforementioned non optimal connections. We show that the deviation bounds and the optimization properties, proved in the last chapter, remain true for unbalanced minibatch OT despite the fact that the UOT transport plans are possibly unbounded. Finally, we extensively evaluate our method on gradient flow, domain adaptation and partial domain adaptation experiments and show that our method improves the results from a state-of-the-art algorithm based on minibatch OT. These contributions have been published to the The Thirty-eighth International Conference on Machine Learning [Fatras 2021b].

9.1 Robustness to outliers and sampling

In this section, we discuss the limits and sensitivities of minibatch OT. We then justify that using a relaxed marginal OT variant can help mitigating these weaknesses. Finally, we theoretically show that unbalanced optimal transport is robust to outliers contrary to exact optimal transport.

Limits of minibatch OT. First, we discuss the limitations of combining balanced OT with the minibatch framework. OT is sensitive to the geometry of distributions. When those distributions are tainted by outliers, OT is forced to transport them due to the marginal constraints, inducing an undesirable extra transportation cost. Minibatch OT averages several OT terms related to subsamples of the original distributions, thus sharing this sensitivity. The problem is even worse as two minibatches do not necessarily

share samples that would lie in the support of the full OT plan, hence forced to match samples that could be, at the level of a minibatch, considered as outliers. Take as an example two distributions with clustered samples. While in the full OT plan clusters can be matched exactly, those clusters are likely to appear as imbalanced in the minibatches, especially if the size of the minibatch is small and does not respect the statistics of the original distribution. Due to the marginal constraints, samples from one cluster are likely to be matched to unrelated clusters, as depicted in Figure 8.2. This explains why in practice previous works relied on large minibatches to mitigate this sampling effect [Damodaran 2018]. To overcome this issue, we propose the natural solution of relaxing the *marginal constraints* at the minibatch level. The expected outcome is twofold: *i)* mitigating the effect of subsampling in the minibatch strategy and *ii)* providing a natural and scalable robust optimal transport computation strategy at the global level. In the following paragraph, we present some robust OT formulations.

Relaxed marginals: Unbalanced and Robust Optimal Transport variants. Unbalanced optimal transport is empirically known to be more robust to outliers than OT as it does not need to meet the marginals [Chizat 2017, Liero 2017]. Indeed, if an outlier is too expensive to move, UOT would not transport it. Several other formulations make optimal transport robust for practical and statistical reasons. Partial OT can be adapted for partial matchings problem with applications for positive-unlabeled learning [Chapel 2020]. A line of work proposes 'distributionnally robust' models, where models are trained in a Wasserstein ball around the empirical distribution in the space of probabilities [Mohajerin Esfahani 2018, Kuhn 2019]. In a similar approach, several variants relax the OT marginal constraints with a ball constraint, and consider several penalties such as integral probability metrics [Nath 2020], total variation or Csiszàr divergences for outlier detection [Mukherjee 2020, Balaji 2020]. Such relaxations allow to derive statistical guarantees w.r.t. noise and outliers. Another idea to ensure robustness consists in learning the cost adversarially, and is formulated as a max-min problem where the cost is modeled by an Euclidean embedding [Genevay 2018], a compact space of matrices [Dhouib 2020a] or a projection on a lower dimensional subspace [Paty 2019].

We chose to rely on *unbalanced optimal transport* with an entropic regularization at the minibatch level to design a new loss function that we called *Unbalanced Minibatch Optimal Transport*. The entropic-regularized unbalanced formulation of optimal transport can be easily computed with a Sinkhorn algorithm as discussed in Section 2.4.2. We discuss in the following some theoretical considerations to support Unbalanced OT's robustness.

Theoretical analysis: impact of an outlier. We start by examining the impact of an outlier in the behaviors of OT and UOT. The following lemma illustrates the relations between those two quantities.

Lemma 9.1.1. *Take (α, β) two probability distributions. For $\zeta \in [0,1]$, write $\tilde{\alpha} = \zeta\alpha + (1-\zeta)\delta_{\boldsymbol{z}}$ a distribution perturbed by a Dirac outlier located at some z outside of the support of (α, β). Take the unregularized OT loss $\mathrm{OT}^{\tau,0}_{KL}$ with KL entropy and cost C. Write $m(\boldsymbol{z}) = \int C(\boldsymbol{z}, \boldsymbol{y}) d\beta(\boldsymbol{y})$. One has:*

$$\mathrm{OT}^{\tau,0}_{KL}(\tilde{\alpha}, \beta, C) \leq \zeta\, \mathrm{OT}^{\tau,0}_{KL}(\alpha, \beta, C) + 2\tau(1-\zeta)(1 - e^{-m(\boldsymbol{z})/2\tau}) \tag{9.1}$$

Now take the unregularized, balanced OT loss $\mathfrak{L}$ with cost C. Write (f, g) the optimal dual potentials (i.e. functions) of $\mathfrak{L}(\alpha, \beta)$, and y^* in β's support. Then:

$$\mathfrak{L}(\tilde{\alpha}, \beta, C) \geq \zeta\mathfrak{L}(\alpha, \beta, C) + (1-\zeta)\Big(C(\boldsymbol{z}, y^*) - g(y^*) + \int g d\beta\Big) \tag{9.2}$$

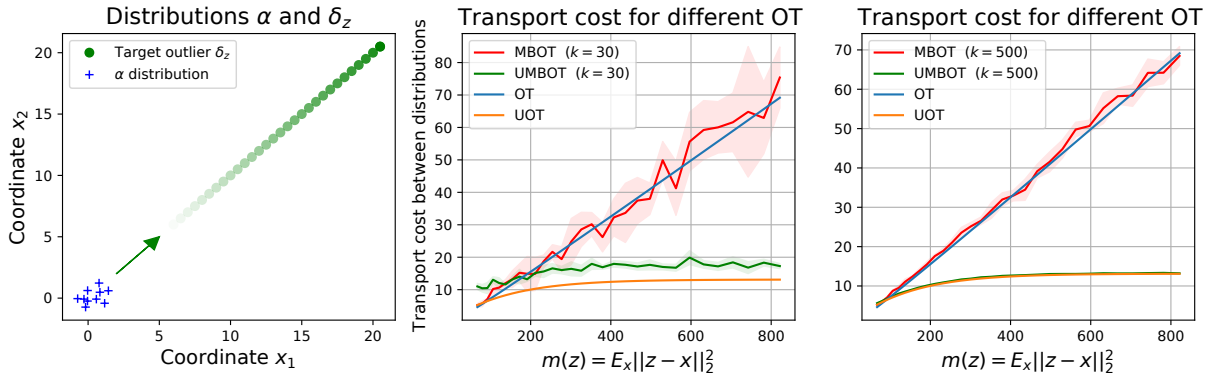


Figure 9.1: Several OT costs between 2D distributions with $n = 10$ samples and $m = 5$. Target distribution is equal to the source distribution tainted with a moving outlier (green dot). The shaded area represent the variance of subsample MBOT on 5 run.

Equation (9.2) shows that when $\boldsymbol{z}$ gets further from the supports of (α, β), the OT loss increases. However for UOT the upper bound (9.1) tends to saturate as $\boldsymbol{z}$ gets further away. What remains is the UOT loss between distributions whose outliers are removed, with a cost of removing the outlier proportional to its mass.

We illustrate Lemma 9.1.1 with a toy example in Figure 9.1. We consider a probability distribution α tainted with an outlier (green dot) to get a target probability distribution $\alpha' = \frac{1}{n+1}(n.\alpha + \delta_{\boldsymbol{z}})$. We then move away the outlier from α's support, as shown with the green arrow, and we calculate several OT costs. The minibatch size is set to $m = 5$ and the total cost is the average of k OT costs between those minibatches. We see that OT variants are not robust to the outlier as their loss increases along the outlier displacement unlike UOT variants which reach a plateau as predicted by Lemma 9.1.1. Each computation is done 5 times to show that variance is lower for bigger k and that UMBOT has a lower variance than MBOT for $k = 30$ and $k = 500$.

We consider now an example in 2D, akin to Figure 8.2, where our goal is to illustrate the OT plan between two empirical distributions of 10 samples in Figure 9.2. We use two 2D empirical distributions where the samples belong to a certain cluster depending on a related class (color information). The source data are equally distributed between classes while the target data have different proportions, 3 samples belong to the red class while 7 samples belong to the green class. Different proportions between domains are ubiquitous for real-world data. We compare unbalanced minibatch OT, minibatch OT, entropic OT and UOT. For UOT, the divergence D_ϕ equals to `KL` divergence and for the minibatch variant, the minibatch size is $m = 2$. We can see from the OT plans in the first row of the figure that the cluster structure is more or less recovered. However OT and minibatch OT tend to connect samples from different classes. This configuration would lead, for instance, to negative transfer in a context of domain adaptation applications, *i.e.,* matching of samples between different domains. This is less true for UOT, where the pairings between different classes is diminished and tend to disappear when we reduce the penalty τ. In the next section, we show that under reasonable hypotheses, the main theoretical results from Chapter 8 remain true.

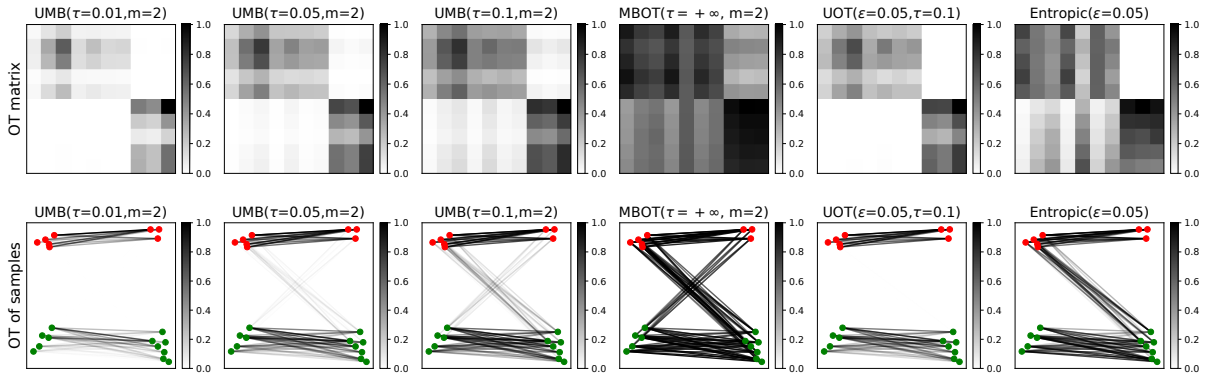


Figure 9.2: Several OT plans, normalized by their maximum value, between 2D distributions with $n = 10$ samples. The first row shows the minibatch OT plans $\overline{\Pi}^{h,m}$ for different values of m and different OT kernels, the second row provides an equivalent geometric interpretation of the OT plans, where the mass transportation is depicted as connections between samples.

9.2 Statistical and optimization properties

For sake of simplicity and due to the experimental setting where we have empirical measures, we consider the uniform formalism of minibatch OT developed in Section 8.1.1. In this section, we show that the deviation bounds on the transport plan and the Unbalanced minibatch OT loss remain true, despite the fact that the mass of the transport plans are possibly unbounded. We demonstrate that under reasonable hypotheses, the UOT cost can be bounded and the associated set of optimal transport plans is compact. Then we demonstrate that under the same hypotheses, we can exchange Clarke gradients and expectations to justify SGD's convergence.

9.2.1 Deviation bounds

To generalize the concentration bounds from Section 8.3.1, we first need to assure that our estimators are bounded. From now on, we suppose that the measures α and β are compactly supported. Our first lemma intends to show that the UOT cost is finite and that the optimal transport plan is bounded.

Lemma 9.2.1 (Bounded UOT and optimal transport plan)**.** *Let C be a ground cost and $\mathbf{a}, \mathbf{b}$ two positive vectors in $\mathbb{R}_+^n$ such that $m_{\mathbf{a}} = \|\mathbf{a}\|_1 > 0$ and $m_{\mathbf{b}} = \|\mathbf{b}\|_1 > 0$. Assume that $\langle \mathbf{a}\mathbf{b}^\top, C\rangle < +\infty$. Consider $h = \mathrm{OT}_\phi^{\tau,\varepsilon}$ and assume $\varepsilon > 0$ or $\phi'_\infty > 0$. Then $h(\mathbf{a}, \mathbf{b}, C)$ is finite and the set of optimal transport plan is a compact set.*

It is straightforward to prove boundedness of $h = S_\phi^{\tau,\varepsilon}$ from Lemma 9.2.1. We can now turn to establish deviation bounds for both incomplete estimators $\widetilde{h}^{m,k}$ and $\widetilde{\Pi}^{h,m,k}$.

Theorem 9.2.1 (Maximal deviation bound)**.** *Let $\delta \in (0,1)$, three integers $k \geq 1$ and $m \leq n$ be fixed. Consider two n-tuples $\boldsymbol{X} \sim \alpha^{\otimes n}$ and $\boldsymbol{Y} \sim \beta^{\otimes n}$ and a kernel $h \in \{\mathrm{OT}_\phi^{\tau,\varepsilon}, S_\phi^{\tau,\varepsilon}\}$. We have a maximal deviation bound between $\widetilde{h}^{m,k}(\boldsymbol{X}, \boldsymbol{Y})$ and E_h^m depending on the number of samples n and the number*

of batches k. With probability at least $1-\delta$ on the draw of $\boldsymbol{X}, \boldsymbol{Y}$ and D_k we have:

$$|\widetilde{h}^{m,k}(\boldsymbol{X},\boldsymbol{Y}) - E_h^m| \leq M\left(\sqrt{\frac{\log(\frac{2}{\delta})}{2\lfloor\frac{n}{m}\rfloor}} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\right),$$

where M is the UOT upper bound. Furthermore, for $h = \mathrm{OT}_\phi^{\tau,\varepsilon}$, let $\mathfrak{M}_\Pi^\infty$ be the maximum mass of minibatch transport plans. For all $k \geqslant 1$, all $1 \leqslant i \leqslant n$, with probability at least $1-\delta$ on the draw of $\boldsymbol{X}, \boldsymbol{Y}$ and D_k we have:

$$|\widetilde{\Pi}^{h,m,k}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1}_n - \overline{\Pi}^{h,m}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1}_n| \leqslant \mathfrak{M}_\Pi^\infty\sqrt{\frac{2\log(2/\delta)}{k}},$$

where we denote by $\Pi_{(i)}$ the i-th row of matrix Π and by $\mathbf{1}_n \in \mathbb{R}^n$ the vector whose entries are all equal to 1.

The rate $\frac{m}{n}$ is the same as in Chapter 8.3.1. The main difference is the upper bound M which bounds UOT. Note that the bound does not depend on the dimension of $\mathcal{X}$ unlike original unbalanced OT [Séjourné 2019]. Regarding OT plans, the constant $\mathfrak{M}_\Pi$ represents the maximum mass of minibatch transport plans which would be 1 for OT. In the next section, we study the convergence of SGD using Clarke gradients when the contrast function is unbalanced minibatch OT.

9.2.2 Unbiased Clarke gradients

In Section 8.3.2, we proved that minibatch OT does not suffer from biased gradients contrary to optimal transport. In this section we show that this property can be generalized for minibatch UOT, including unregularized UOT. Like in Section 8.3.2, we also achieve this point by relying on Clarke regularity. It is not immediate because UOT is not differentiable as we do not have a unique optimal transport plan when $\varepsilon = 0$. We have the following theorem.

Theorem 9.2.2. *Let $\hat{\boldsymbol{X}}, \{\hat{\boldsymbol{Y}}_\theta\}_{\theta\in\Theta}$ be two m-tuples of random vectors compactly supported, $h \in \{\mathrm{OT}^{\tau,\varepsilon}, S^{\tau,\varepsilon}\}$ and C^m a $\mathbf{C}^1$ cost. Under an additional integrability assumption, we have:*

$$\partial_\theta\mathbb{E}[h(\boldsymbol{u},\boldsymbol{u},C^m(\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}}_\theta))] = \mathbb{E}[\partial_\theta h(\boldsymbol{u},\boldsymbol{u},C^m(\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}}_\theta))],$$

with both expectation being finite. Furthermore the function $\theta \mapsto -\mathbb{E}[h(\boldsymbol{u},\boldsymbol{u},C^m(\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}}_\theta))]$ is also Clarke regular.

To prove this theorem, we used a similar proof which was used for proving the minibatch OT case. We also needed to prove that UOT is Lipschitz in the ground cost $\boldsymbol{C}$, which is possible because the transport plan is bounded by Lemma 9.2.1. Theorem 9.2.2 implies that if we use the Minibatch UOT loss with $h \in \{\mathrm{OT}_\phi^{\tau,\varepsilon}, S_\phi^{\tau,\varepsilon}\}$ as a loss function, then the minus objective function is Clarke regular. Furthermore, Stochastic gradient with decreasing step sizes converges almost surely to the set of critical points of Clarke generalized derivative [Davis 2020, Majewski 2018]. As a consequence, it is justified to use SGD with minibatch UOT.

Lazy gradients. We finish this section with a phenomenon that we call lazy gradients. We show that for small values of τ, the norm of the partial gradient with respect to the parameters is small as well, which leads to slow training phase in practice. We first express the partial derivative of unbalanced minibatch

OT. For $h = \mathrm{OT}_{\phi}^{\tau,\varepsilon}$ and for all $1 \leq i \leq q$ we have:

$$\partial_{\theta_i} h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta)) = \overline{\mathrm{co}}\{-\langle \Pi \cdot D \rangle \cdot (\nabla_{\theta_i} Y) : \Pi \in \mathrm{Opt}_h(\boldsymbol{X}, \boldsymbol{Y}), \\ D \in \mathbb{R}^{m,m}, D_{j,k} = \nabla_Y C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta)\}. \tag{9.3}$$

Where ∂_{θ_i} is the Clarke subdifferential with respect to θ_i, $\nabla_Y C_{j,k}$ is the differential of the cell $C_{j,k}$ of the cost matrix with respect to Y, $\mathrm{Opt}_h(\boldsymbol{X}, \boldsymbol{Y}_\theta)$ is the set of optimal transport plan and $\overline{\mathrm{co}}$ denotes the closed convex hull. Note that when $\varepsilon > 0$ the set $\mathrm{Opt}_h(\boldsymbol{X}, \boldsymbol{Y}_\theta)$ is reduced to a singleton, the notation $\overline{\mathrm{co}}$ is then superfluous and it reduces to

$$\partial_{\theta_i} h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta)) = -\langle \Pi^\star \cdot D \rangle \cdot (\nabla_{\theta_i} Y), \tag{9.4}$$

where $\Pi^\star$ is the optimal plan. As we are in finite dimension, all norms are equivalent and we can consider $\|\cdot\|$ to be a submultiplicative norm. We can then bound the partial gradient as:

$$\|\partial_{\theta_i} h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\| = \| - \langle \Pi^\star \cdot D \rangle \cdot (\nabla_{\theta_i} Y)\| \leq \|\Pi^\star\| \|D\| \|\nabla_{\theta_i} Y\|. \tag{9.5}$$

So for small values of τ, the mass of the optimal transport plan $\|\Pi^\star\|$ is small. Hence the gradient norm is small as well which leads to slow down the training phase. Thus a trade-off is needed on the value of τ in order to have transport plan with big enough mass and which do not transport outliers. Further analysis can be made by using the closed-form solution of entropic-regularized UOT between Gaussian distributions [Janati 2020] and we left such analysis as future work.

9.3 Numerical experiments on gradient flows and domain adaptation

In this section, we illustrate the practical behavior of unbalanced minibatch OT for gradient flow and for domain adaptation experiments. We relied on the POT package [Flamary 2021] to compute the exact OT solver or the entropic UOT loss and the Geomloss package [Feydy 2019] for the Unbalanced Sinkhorn divergence. The experiments were designed in PyTorch [Paszke 2017] and all the code can be found here*.

9.3.1 Unbalanced MiniBatch OT gradient flow: a qualitative example

We consider the same experiments as in Section 8.4.1 but with toy 2D data. Consider a given target distribution α, we recall that the purpose of gradient flows is to model a distribution $\beta(t)$ which at each iteration follows the gradient direction minimizing the loss $\beta_t \mapsto h(\alpha, \beta_t)$ [Peyré 2015, Liutkus 2019].

For α and $\beta(0)$ we generate 10000 2D points divided in 2 imbalanced clusters with number of samples in each cluster provided in Figure 9.3. We consider the (unbalanced) sinkhorn divergence, a squared Euclidean cost, a learning rate of 0.02, 5000 iterations, m equals 64 or 128 and $k = 1$. We show the gradient flow of the upper clusters to the lower clusters in Figure 9.3. From the experiment, we can see that the minibatch OT is not robust to imbalanced classes on the contrary to the minibatch UOT. Indeed there are data from the upper left cluster which converge to the down right cluster and we can also see an overlap between the classes. Due to OT marginal constraints, the loss forces to transport all data in the batch which results in breaking the target shapes. This is not the case for minibatch UOT, which better respects the shape of target distributions.

*https://github.com/kilianFatras/JUMBOT

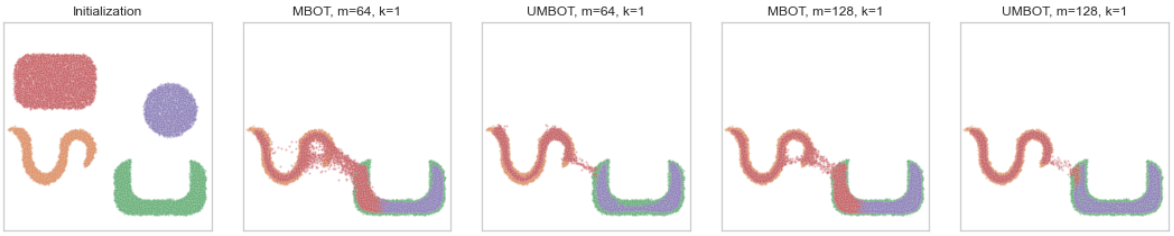


Figure 9.3: (Best viewed in colors) Minibatch UOT gradient flow on a 2D dataset. Source data and target data are divided in 2 imbalanced clusters, source left (red) and target right (green) shapes have 6400 samples while source right (purple) and target left (orange) shapes have 3600 samples. The batch size m is set to $\{64, 128\}$ and the number of minibatch k is set to 1, meaning that the explicit Euler integration step is conducted for each batch. Results are computed with the (unbalanced) minibatch Sinkhorn divergence losses.

9.3.2 JUMBOT: a new approach for domain adaptation

In this section, we evaluate Unbalanced minibatch OT in domain adaptation. Based on the state-of-the-art algorithm DEEPJDOT, we replace OT by UOT at the minibatch level. We consider vanilla domain adaptation and partial domain adaptation experiments.

Domain adaptation.

JUMBOT. To solve the domain adaptation problem, we use an alignment strategy with a cross entropy on the source domain and a transfer term. The transfer term of our method is based on [Damodaran 2018] and aims at finding a joint distribution map between the source and the target distributions by taking into account a term on a neural network embedding space and on the label space. Formally, let g_θ be an embedding function where the input is mapped into the latent space and f_λ which maps the latent space to the label space on the target domain. The embedding space is in our experiment the penultimate layer of a neural network. For a given minibatch, embedding g_θ and classification map f_λ, the transfer term is:

$$\bar{h}^m_{C_{\theta,\lambda}}((\boldsymbol{X}^s, \boldsymbol{Y}^s), (\boldsymbol{X}^t, f_\lambda(g_\theta(\boldsymbol{X}^t)))), \text{with} \tag{9.6}$$
$$(C_{\theta,\lambda})_{i,j} = \eta_1 \|g_\theta(\boldsymbol{x}^s_i) - g_\theta(\boldsymbol{x}^t_j)\|_2^2 + \eta_2 \mathcal{L}(\boldsymbol{y}^s_i, f_\lambda(g_\theta(\boldsymbol{x}^t_j))),$$

where $\mathcal{L}(.,.)$ is the cross-entropy loss and η_1, η_2 are positive constants. Basically, this specific transportation cost combines a distance between the representation of the data through the neural network g_θ and a loss function between the associated labels [Courty 2017b]. Taking $k = 1$ led to state-of-the-art results. The Csiszàr divergence ϕ is the Kullback-Leibler divergence KL. We also add a cross entropy term on the source data. Hence our optimization problem is:

$$\min_{\theta,\lambda} \sum_i \mathcal{L}(f_\lambda(g_\theta(\boldsymbol{x}^s_i)), \boldsymbol{y}^s_i) + \eta_3 \widetilde{h}^{m,k}_{C_{\theta,\lambda}}((\boldsymbol{X}^s, \boldsymbol{Y}^s), (\boldsymbol{X}^t, f_\lambda(g_\theta(\boldsymbol{X}^t)))). \tag{9.7}$$

Our method is called JUMBOT and stands for Joint Unbalanced MiniBatch OT. It is related to DEEPJDOT [Courty 2017b, Damodaran 2018] at the notable exceptions that we use minibatch UOT, which can also handle partial domain adaptation as suggested by our experiments.

Datasets. We start with **digits** datasets. Following the evaluation protocol of [Damodaran 2018] we experiment on three adaptation scenarios: USPS to MNIST (U↦M), MNIST to M-MNIST (M↦MM),

Methods/DA	U ↦ M	M ↦ MM	S ↦ M	Avg
DANN(*)	92.2 ± 0.3	96.1 ± 0.6	88.7 ± 1.2	92.3
CDAN-E(*)	**99.2 ± 0.1**	95.0 ± 3.4	90.9 ± 4.8	95.0
ALDA(*)	97.0 ± 1.4	96.4 ± 0.3	96.1 ± 0.1	96.5
DEEPJDOT(*)	96.4 ± 0.3	91.8 ± 0.2	95.4 ± 0.1	94.5
E-DEEPJDOT(*)	97.1 ± 0.3	94.2 ± 0.1	97.6 ± 0.1	96.3
JUMBOT	98.2 ± 0.1	**97.0 ± 0.3**	**98.9 ± 0.1**	**98.0**

Table 9.1: Summary table of DA results on digit datasets. Experiments were run three times. (*) denotes the reproduced methods.

and SVHN to MNIST (S↦M). MNIST [LeCun 2010] contains 60,000 images of handwritten digits, M-MNIST contains the 60,000 MNIST images with color patches [Ganin 2016] and USPS [Hull 1994] contains 7,291 digit images. Street View House Numbers (SVHN) [Netzer 2011] consists of 73, 257 images with digits and numbers in natural scenes. We report the evaluation results on the test target datasets. **Office-Home** [Venkateswara 2017] is a difficult dataset for unsupervised domain adaptation (UDA), it has 15,500 images from four different domains: Artistic images (A), Clip Art (C), Product images (P) and Real-World (R). For each domain, the dataset contains images of 65 object categories that are common in office and home scenarios. We evaluate all methods in 12 adaptation scenarios. **VisDA-2017** [Peng 2017] is a large-scale dataset for UDA from simulation to real. VisDA contains 152,397 synthetic images as the source domain and 55,388 real-world images as the target domain. 12 object categories are shared by these two domains. Following [Long 2018, Chen 2020], we evaluate all methods on VisDA validation set.

Setup. First note that for all datasets, JUMBOT uses a stratified sampling on source minibatches as done in DEEPJDOT [Damodaran 2018]. Stratified sampling means that each class has the same number of samples in the minibatches. This is a realistic setting as labels are available in the source dataset.

For Digits datasets, we used the 9 CNN layers architecture and the 1 dense layer classification proposed in [Damodaran 2018]. We trained our neural network on the source domain during 10 epochs before applying JUMBOT. We used Adam optimier with a learning rate of $2e^{-4}$ with a minibatch size of 500. Regarding competitors, we use the official implementations with the considered architecture and training procedure.

For Office-Home and VisDA, we employed ResNet-50 as generator. ResNet-50 is pretrained on ImageNet and our discriminator consists of two fully connected layers with dropout, which is the same as previous works [Ganin 2016, Long 2018, Chen 2020]. As we train the classifier from scratch, we set its learning rates to be 10 times that of the generator. We train the model with Stochastic Gradient Descent optimizer with a momentum of 0.9. We schedule the learning rate with the strategy in [Ganin 2016], it is adjusted by $\chi_p = \frac{\chi_0}{(1+\mu q)^\nu}$, where q is the training progress linearly changing from 0 to 1, $\chi_0 = 0.01$, $\mu = 10$, $\nu = 0.75$.

We compare JUMBOT against recent domain adaptation papers, namely DANN [Ganin 2016], CDAN-E [Long 2018], ALDA [Chen 2020], DEEPJDOT [Damodaran 2018] and ROT [Balaji 2020] on all considered datasets. We reproduced their scores and contrary to these papers we do not report the best classification on the test along the iterations but at the end of training, which explains why there might be a difference between reported results and reproduced results. We sincerely believe that the evaluation shall only be done at the end of training as labels are not available in the target domain. But we also report the maximum accuracy along epochs for the Office-Home DA task in Table 9.3 and it shows that our method

	Method	A-C	A-P	A-R	C-A	C-P	C-R	P-A	P-C	P-R	R-A	R-C	R-P	avg
DA	RESNET-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
	DANN (*)	44.3	59.8	69.8	48.0	58.3	63.0	49.7	42.7	70.6	64.0	51.7	78.3	58.3
	CDAN-E(*)	52.5	71.4	76.1	59.7	69.9	71.5	58.7	50.3	77.5	70.5	57.9	**83.5**	66.6
	DEEPJDOT (*)	50.7	68.6	74.4	59.9	65.8	68.1	55.2	46.3	73.8	66.0	54.9	78.3	63.5
	E-DEEPJDOT (*)	50.6	68.9	74.4	59.3	65.1	69.0	56.2	46.5	74.5	65.1	54.7	78.1	63.5
	ALDA (*)	52.2	69.3	76.4	58.7	68.2	71.1	57.4	49.6	76.8	70.6	57.3	82.5	65.8
	ROT (*)	47.2	71.8	76.4	58.6	68.1	70.2	56.5	45.0	75.8	69.4	52.1	80.6	64.3
	JUMBOT	**55.2**	**75.5**	**80.8**	**65.5**	**74.4**	**74.9**	**65.2**	**52.7**	**79.2**	**73.0**	**59.9**	83.4	**70.0**
PDA	RESNET-50	46.3	67.5	75.9	59.1	59.9	62.7	58.2	41.8	74.9	67.4	48.2	74.2	61.4
	DEEPJDOT(*)	48.2	66.2	76.6	56.1	57.8	64.5	58.3	42.7	73.5	65.7	48.2	73.7	60.9
	E-DEEPJDOT(*)	47.6	67.0	77.3	57.1	57.9	65.4	58.1	41.3	74.4	66.4	47.7	75.1	61.3
	PADA	51.9	67.0	78.7	52.2	53.8	59.0	52.6	43.2	78.8	73.7	56.6	77.1	62.1
	ETN	59.2	77.0	79.5	62.9	65.7	75.0	68.3	55.4	84.4	75.7	57.7	**84.5**	70.4
	BA3US(*)	56.7	76.0	**84.8**	73.9	67.8	**83.7**	72.7	56.5	84.9	77.8	64.5	83.8	73.6
	JUMBOT	**62.7**	**77.5**	84.4	**76.0**	**73.3**	80.5	**74.7**	**60.8**	**85.1**	**80.2**	**66.5**	83.9	**75.5**

Table 9.2: Summary table of DA and Partial DA results on Office-Home (ResNet-50). (*) denotes the reproduced methods.

Method	A-C	A-P	A-R	C-A	C-P	C-R	P-A	P-C	P-R	R-A	R-C	R-P	avg
RESNET-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN(*)	46.2	65.2	73.0	54.0	61.0	65.2	52.0	43.6	72.0	64.7	52.3	79.2	60.7
CDAN-E(*)	52.8	71.4	76.1	59.7	70.6	71.5	59.8	50.8	77.7	71.4	58.1	**83.5**	67.0
ALDA(*)	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
ROT(*)	47.2	70.8	77.6	61.3	69.9	72.0	55.4	41.4	77.6	69.9	50.4	81.5	64.6
DEEPJDOT(*)	53.4	71.7	77.2	62.8	70.2	71.4	60.2	50.2	77.1	67.7	56.5	80.7	66.6
JUMBOT	**55.3**	**75.5**	**80.8**	**65.5**	**74.4**	**74.9**	**65.4**	**52.7**	**79.3**	**74.2**	**59.9**	83.4	**70.1**

Table 9.3: Vanilla domain adaptation experiments on Office-Home dataset with maximum classification along training iterations. (ResNet50)

is above all of the competitors by a safe margin of 3%.

For Office-Home, we made 10000 iterations with a batch size of 65 and for VisDA, we made 10000 iterations with a batch size of 72. For fair comparison we used our minibatch size and number of iterations to evaluate competitors. The hyperparameters used in our experiments are as follows $\eta_1 = 0.1, \eta_2 = 0.1,\ \eta_3 = 1, \tau = 1, \varepsilon = 0.1$ for the digits and for Office-Home datasets $\eta_1 = 0.01, \eta_2 = 0.5,\ \eta_3 = 1, \tau = 0.5, \varepsilon = 0.01$. For VisDA, $\eta_1 = 0.005, \eta_2 = 1,\ \eta_3 = 1, \varepsilon = 0.01$ and τ was set to 0.3.

Results. The results on digit datasets can be found in Table 9.1 where (*) denotes reproduced results. We conducted each experiment three times and report their average results and variance. For fair comparisons, we only resize and normalize the image without data augmentation. We see that our method performs best with a margin of at least 1.5 points. Furthermore, we see an important 4% increase of the performance compared to DEEPJDOT. A deeper analysis of this difference is considered in the next paragraph. Office-Home results are gathered in Table 9.2 and VisDA are reported in Table 9.4. For fair comparison with previous work, we used a similar data pre-processing and we used the ten-crop technique [Long 2018, Chen 2020] for testing our methods. Experiments in [Balaji 2020] follow a different setup explaining the difference of performance between their reported score and our reproduced score. JUMBOT achieves the best accuracy on average and on 11 of the 12 scenarios on the Office-Home dataset and achieves the best accuracy on VisDA at the end of training with a margin of 4% and 2% respectively.

Ablation. The main difference between JUMBOT and DEEPJDOT is the use of a different OT solver. JUMBOT uses entropic regularized unbalanced OT and DEEPJDOT uses original OT. We have conducted an

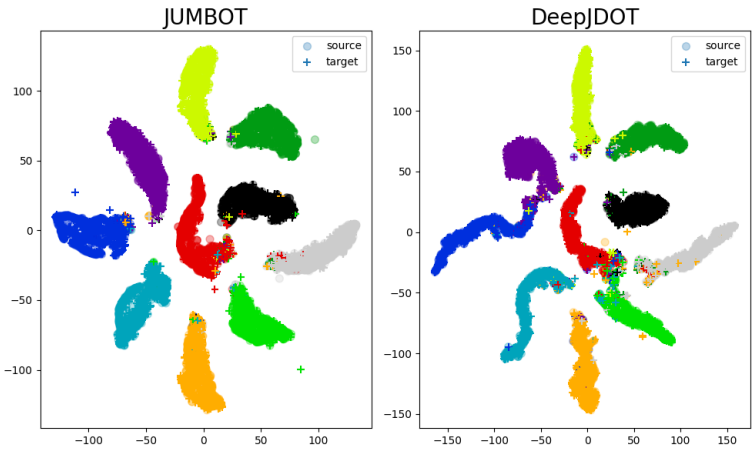


Figure 9.4: T-SNE embeddings of 10000 test samples for MNIST (source) and MNIST-M(target) for DEEPJDOT and our method. It shows the ability of the methods to discriminate classes (samples are colored w.r.t. their classes).

Methods	Accuracy (in %)
CDAN-E(*)	70.1
ALDA(*)	70.5
DEEPJDOT(*)	68.0
E-DEEPJDOT(*)	69.2
ROBUST OT(*)	66.3
JUMBOT	**72.5**

Table 9.4: Summary table of DA results on VisDA datasets. (*) denotes the reproduced methods.

ablation study in all our domain adaptation experiments to measure the effect of the entropic regularization and the unbalanced formulation. The use of entropic regularized OT in place of original OT in DEEPJDOT is denoted E-DEEPJDOT. We can see that for the digits experiments, the entropic regularization helps to get better performances of 0.6% on the DA task USPS $\mapsto$ MNIST and more than 2% on SVHN $\mapsto$ MNIST. For the last digit task MNIST $\mapsto$ M-MNIST, the grid search on ε needed to be reduced due to some numerical instabilities. Overall, the performances of E-DEEPJDOT are still lower than the entropic regularized Unbalanced OT of JUMBOT by more than 1%. A similar performance gain was observed on the VisDA dataset, where E-DEEPJDOT was 1.2% higher than DEEPJDOT but still more than 3% below JUMBOT. On the Office-Home experiments, the entropic regularization alone did not get better results.

Analysis. In this paragraph, we study the difference of behavior between DEEPJDOT and JUMBOT. Along JUMBOT's training on the DA task USPS to MNIST, we measured the percentage of mass between data with different labels at each iteration. In average along training about 7% of DEEPJDOT connections are between data with different labels while this percentage decreases to 0.7% for JUMBOT as shown in Figure 9.6. So DEEPJDOT transfers wrong labels to target data which will decrease the overall accuracy.

We also plot a TSNE embedding of our method and DEEPJDOT (see Figure 9.4), we can see that there are some overlaps between clusters for DEEPJDOT unlike our method. This is probably due to the minibatch smoothing effect which would tend to bring clusters of different classes closer.

We also provide a *sensitivity analysis* to the batch size, ε and τ parameters. All results are gathered in Figure 9.5. When τ is too small, JUMBOT creates negative transfer because of the entropic regularization.

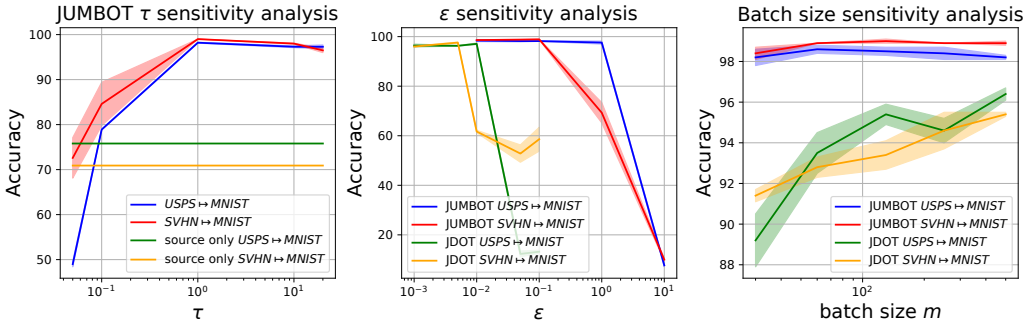


Figure 9.5: (Best viewed in colors) DEEPJDOT and JUMBOT sensitivity analysis. We report the classification accuracy of DEEPJDOT and JUMBOT on the DA tasks USPS ↦ MNIST and SVHN ↦ MNIST for several hyperparameter variations. We consider the marginal coeffifient τ, the entropic coefficient ε and the batch size m.

When τ increases, we see that JUMBOT accuracy increases and it reaches its maximum around $\tau = 1$. However when τ is too high, we recover entropic OT.

When ε varies for JUMBOT and the entropic variant of DEEPJDOT, we see that entropy helps getting slightly better results. However when the entropic regularization coefficient is too big, the accuracy falls. We conjecture that entropic regularized OT regularizes the neural network because the target prediction is matched to a smoothed source label (see a similar discussion in [Damodaran 2019]). And it is well known that label smoothing creates class clusters in the penultimate layer of the neural network [Müller 2019].

We now discuss the minibatch size. While JUMBOT has a constant accuracy along all batch size, the DEEPJDOT accuracy falls of 4% for SVHN ↦ MNIST and 6% for USPS ↦ MNIST. The benefits of our method over DEEPJDOT are twofold, it is more robust to small batch sizes and it is performant for small computation budget unlike DEEPJDOT.

Finally, we provide a classification accuracy along training which demonstrates the network overfitting with DEEPJDOT and not with JUMBOT, the results are gathered in Figure 9.7. One can see that DEEPJDOT starts overfitting from epoch 30 on each class. There are some classes which are more affected by overfitting than others. The accuracy on each class is reduced of several points. This behaviour is not shared with our method JUMBOT. Indeed it is more stable, it does not show any sign of overfitting and it has a higher accuracy. This shows the relevance of using our method JUMBOT.

Partial domain adaptation.

Partial DA. Finally we consider the Partial DA (PDA) application. In PDA, the target labels are a subset of the source labels, *i.e.*, $\mathcal{Y}_t \subset \mathcal{Y}_s$. Samples belonging to these missing classes become outliers which can produce negative transfer. We want to investigate the robustness of our method in such an extreme scenario. We evaluate our method on partial Office-Home, where we follow [Cao 2018] to select the first 25 categories (in alphabetic order) in each domain as a partial target domain.

Set up. We compare our method against state-of-the-art PDA methods: PADA [Cao 2018], ETN [Cao 2019] and BA3US [Jian 2020]. For fair comparison we followed the experimental setting of PADA, ETN and BA3US. We considered a neural network architecture and a training procedure similar as in the domain adapation experiments which also corresponds to the setting in [Jian 2020]. Our hyperparameters are set as follows : $\tau = 0.06, \eta_1 = 0.003, \eta_2 = 0.75$ and finally η_3 was set to 10. Regarding training procedure, we made 5000 iterations with a batch size of 65 and for optimization procedure, we used the same as in [Jian 2020]. We do not use the ten crop technique to evaluate our model on the test set as we

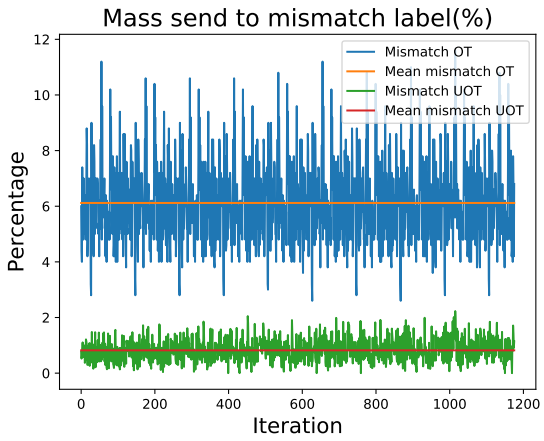


Figure 9.6: Percentage of mass between data with different labels for JDOT and JUMBOT during the USPS to MNIST DA task.

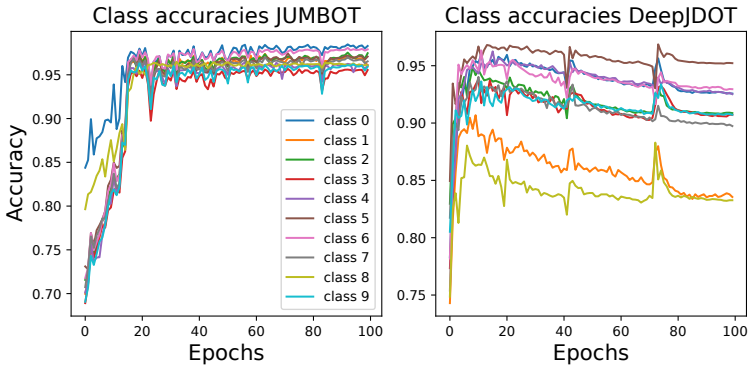


Figure 9.7: (Best viewed in colors) DEEPJDOT and JUMBOT class accuracies along training. We report the class accuracies along training of DEEPJDOT and JUMBOT on the DA task MNIST $\mapsto$ M-MNIST for optimal hyper-parameters. Each color represents a different class.

were not able to reproduce the results from ENT and PADA. Furthermore, we do not know if the reported results ENT and PADA were evulated at the end of optimization or during training, but our reported scores are above their scores by at least 5% on average.

Results. The final performances are gathered in the lower part of Table 9.2. We can see that JUMBOT is state-of-the-art on 9 out of the 12 domain adaptation tasks and is on average 2% above competitors. Finally, we also evaluate DEEPJDOT [Damodaran 2018] and its entropic variant E-DEEPJDOT on the PDA task to compare the performances of regularized unbalanced OT against regularized and unregularized OT. E-DEEPJDOT gives similar results to DEEPJDOT and they both produce negative transfer. JUMBOT is on average 15% higher on this problem, showing the clear advantages of our strategy in the presence of unbalanced classes.

9.4 Conclusion

In this chapter, we presented some weaknesses of minibatch optimal transport. Due to the marginal constraints, minibatch OT, like OT, is sensitive to outliers. Furthermore, two minibatches do not necessarily share samples that would lie in the support of the full OT plan. Hence it would force OT to match samples that could be, at the level of a minibatch, considered as outliers. To mitigate these issues, we proposed to use unbalanced OT at the minibatch level. We proved that contrary to OT, UOT is robust to these outliers. Then we also proved that the deviation bounds and optimization properties showed in the previous chapter remain true for unbalanced minibatch OT. Finally we demonstrated the relevance of our method on gradient flows, domain adaptation and partial domain adaptation experiments. We show that by using unbalanced minibatch OT in DEEPJDOT, nonetheless it outperforms the original DEEPJDOT based on minibatch OT but also outperforms all recent state-of-the-art domain adaptation algorithms.

Chapter 10

Conclusion

Contents

10.1 Overview of the contributions

In this thesis, we developed new optimal transport based methods for deep learning and we theoretically and empirically studied a commonly used optimal transport approximation in deep learning.

Optimal Transport for label noise. We first focused on the label noise problem, which is a supervised learning setting where a proportion of labels are corrupted. Based on the intuition that label noise can be mitigated if the predicted classification is uniform among clusters of data, we used an optimal transport loss in the *Virtual Adversarial Algorithm*. The VAT algorithm promotes a local uniform prediction by penalizing large local changes in the prediction thanks to a divergence. We replaced the divergence with optimal transport and crafted different ground costs in order to change the geometry of the regularization. These ground costs allowed us to have complex boundaries between similar classes and smooth boundaries between non similar classes. One ground cost uses semantic distances based on word embeddings such as *word2vec*, as similarly done in [Frogner 2015] and another uses the distance between embedded data. We then showed the relevance of using our method on standard and real-world datasets including remote sensing images. We also showed the effectivness of our methods on open-set noise, where images that do not correspond to given classes are in the dataset.

Generating natural misclassified data. Our second contribution uses generative adversarial networks to generate misclassified data. Based on a pre-trained classifier, where we have access to the output, and to training data, we developed a new weighting strategy to define a new probability measure for training data. The weighting strategy is based on a softmax function and gives bigger weights to misclassified data than correctly classified data. We then give this probability measure to Wasserstein GAN [Arjovsky 2017, Gulrajani 2017], an optimal transport variant of GAN, in order to generate misclassified data. We evaluated empirically our method on different remote sensing data, where we showed that the generated data were similar to training data and misclassified by the pre-trained classifier.

We also demonstrated that a large proportion of misclassified data for the pre-trained classifier are also misclassified for other classifiers, showing the transfer property of our method. Finally, we also showed our method can modify training data to make them misclassified and that we can fool state-of-the-art detector.

Theoretical and empirical study of minibatch Optimal Transport. After using optimal transport as a loss function in deep learning, we studied a standard optimal transport approximation. To fasten training of deep learning, loss functions are commonly computed between minibatches of training data. While it is a competitive method in practice [Genevay 2018, Damodaran 2018], a theoretical explanation and study were lacking. We thus developed a rigorous formalism that works for general probability measures. After defining new empirical estimators, we found a closed-form solution for 1D data. We also reviewed the metric properties, and in order to fix the separability axiom loss, we introduced a new loss function based on minibatch OT. Then we reviewed deviation bounds between our estimators and their expectations, which is the quantity we aim to minimize. We also proved the convergence of SGD in a data fitting scenario when the contrast loss is MBOT. We showed that minibatch OT can solve large-scale color transfer and kept a high diversity of color. Finally, we empirically showed the practical success of our new loss function over minibatch OT on generating data.

Mitigating non-optimal connections. Our last contribution studied the practical consequences of the non-optimal connections in a classification scenario. The standard strategy to solve the domain adaptation problem is to align the embeddings of source and target data which share the same class. Unfortunately, minibatch OT creates non-optimal connections between the domains due to the minibatch sampling and the OT marginal constraints. Thus, the non-optimal connections align domains of data which have different classes. In order to alleviate these connections, we proposed to rely on unbalanced optimal transport. We showed that this variant is indeed robust to outliers unlike original optimal transport. We also showed that our theoretical contributions to minibatch OT remained true when unbalanced optimal transport was used at the minibatch level. Finally, we empirically demonstrated the success of our method on domain adaptation experiments including in extreme imbalanced scenario. We also showed that the performance of unbalanced minibatch optimal transport remains constant when the minibatch size decreases contrary to exact optimal transport.

10.2 Perspectives of future works

10.2.1 Perspectives on our contributions

The different contributions developed in this manuscript lead to several open questions.

Perspectives on the Wasserstein Adversarial Regularization. An important question to improve WAR is to study new ground costs. The two ground costs introduced for our regularizations have been empirically crafted and thus might be crude approximations. New ground costs could be for instance derived from auxiliary tasks that do not involve labels, or by inspecting the confusion matrix score, obtained on the noisy validation set, along the learning process. A dynamic update of the class matrix is also a relevant research avenue to be studied in the case the noise statistics of the data change over time. Recently, several works were published were authors tried to learn the ground cost for a given machine learning problem [Cuturi 2014, Heitz 2020]. We believe that this strategy can also be relevant

for WAR. Finally, we recall that WAR is based on the VAT algorithm, which was designed to solve the semi-supervised learning setting. While preliminary results showed that WAR do not outperform VAT on this task, it might be due to non-insightful ground cost. We also foresee an application for imbalanced classes where the ground cost could allow to have complex boundaries for imbalanced classes and smooth boundaries for others. Finally, WAR can also be relevant on other tasks were perturbation based regularization were successful.

Perspectives on minibatch OT. The recent work on minibatch OT suggests new open questions. On the theoretical side, new concentration bounds could be derived for minibatch OT using the Talagrand concentration inequality [Talagrand 1995]. A recent work [Nguyen 2021] introduced a new minibatch OT cost based on a different sampling. Instead of averaging the optimal transport cost between minibatches, they see the minibatches as their new measures to transport. They create a ground cost which measures the pairwise distance of minibatches. They proved that their loss is a metric between probability measures and that their transport plan is empirically more sparse. However, it is not proven that they indeed transport all the input masses. While appealing, this strategy remains heavy in practice as you need to compute a lot of minibatch distances as a pre-processing step in order to get your ground cost. The empirical gain in deep learning is also not clear and needs to be demonstrated.

Perspectives on Unbalanced Optimal Transport. The use of unbalanced optimal transport at the minibatch level to improve classification score in domain adaptation, highlighted some limits of UOT. In practice, when the marginal coefficient τ is too small, the mass of the optimal transport plan is small as well. This leads to gradients with possibly small norms which slow down the learning process. Studying empirically and theoretically this effect is an important element for the use of UOT in deep learning. This subject could be investigated with closed-form solutions of unbalanced OT, as done for the Gaussian case in [Janati 2020]. In order to get a transport plan with a bigger mass, we could also learn adversarially the probability measures of our minibatches, similarly to [Balaji 2020]. Intuitively, we might want to give small weights to transport data with a high cost and bigger weights to data which are cheap to transport.

10.2.2 Perspectives on Optimal Transport in Machine Learning

Several new Machine Learning applications were recently published and we make a small overview of domains where optimal transport has been or can be applied.

Normalizing Flow. Among them, we note that optimal transport has been applied to normalizing flow. A normalizing flow is an invertible map between an arbitrary probability distribution and a standard normal distribution. The flow can be used for density estimation, statistical inference or generative modelling. Several variants of normalizing flow rely on the dynamical formulation of optimal transport [Onken 2021, Finlay 2020], on the sliced Wasserstein distance formulation [Dai 2021] or the Brenier theorem [Huang 2021], where the optimal transport map is the gradient of a convex function for the squared euclidean distance [Peyré 2019]. Among all these variants, a minibatch computation of the primal might bring interesting elements such a more stable training as for GANs [Genevay 2018].

Out-of-distribution sample. Deep neural networks have a tendency to make spurious correlations that do not hold outside a specific training distribution. For instance, [Beery 2018] trained a neural network to classify camels from cows. Most of the training pictures of cows had green pastures, while

most pictures of camels had a desert landscape. Unfortunately, the neural network associated a green pasture to the cows and incorrectly classified cows on a beach. The main method to solve this problem is the *Invariant Risk Minimization* [Arjovsky 2020]. It aims at building a classifier that performs well across many unseen environments. As optimal transport has been successful in Domain Adaptation to create cluster of data between domains [Damodaran 2018, Fatras 2021b]. It would be interesting to see if a similar approach could lead to create cluster of training data and improvements on classifying OOD samples.

Annex A

Appendix

Contents

In this appendix, we gather the different proofs of the results we claimed in the previous chapter. In Appendix A.1 we gather the proofs of the different results from Chapter 8. In Appendix A.2 we gather the proofs of results from Chapter 9.

A.1 Proofs of Chapter 8

Outline. This Appendix is organized as follows:

- Appendix A.1.1 gives the proofs of our general minibatch OT distances formalism. In particular, it proves under what conditions the minibatch OT matrix is a minibatch OT plan.
- Appendix A.1.2 provides the proofs of concentration bounds for compactly supported distributions. We generalize the U-statistic proof to know under what conditions our estimator is close to its mean.
- Appendix A.1.3 provides the proofs of concentration bounds for sub-Gaussian distributions. Based on the compactly supported case, we use a truncation argument to provide a more general concentration bound for unbounded distributions.
- Appendix A.1.4 gives the proofs of concentration bounds for the minibatch OT plan. We provide a concentration bound of our incomplete minibatch OT plan around the input marginals.

- Appendix A.1.5 details the optimization proofs. We prove that we can exchange Clarke gradients over parameters and expectations, which justifies the convergence of SGD when we use MBOT as a contrast function. Notably, this proof includes the Wasserstein distance and the Gromov-Wasserstein distance.
- Appendix A.1.6 discusses the 1D case. We detail the calculus of the 1D minibatch Wasserstein distance closed-form.

A.1.1 Formalism

In this appendix, we show results which justify our formalism and then, we show how we can upper bound our minibatch OT loss. We first show that Example 4 defines a probability law on m-tuples without replacement.

Example 4 (Drawing indices "without replacement")**.** *Given a discrete probability distribution $\mathbf{a} \in \Sigma_n$, it is also possible to draw distinct indices $i_\ell \in [\![n]\!]$, $1 \leq \ell \leq m$, by defining $P^{\mathtt{W}}_{\mathbf{a}}(I) = 0$ if the m-tuple I has repeated indices, otherwise*

$$P^{\mathtt{W}}_{\mathbf{a}}(I) = \frac{1}{m}\frac{(n-m)!}{(n-1)!}\sum_{i\in I} a_i. \tag{A.1}$$

Denote by $\mathcal{P}^m$ the set of all m-tuples without repeated elements. Let us check, that equation (A.1) defines a probability distribution on $\mathcal{P}^m$. Observe that $\sum_{i=1}^n a_i = 1$ and that for each $1 \leq i \leq n$

$$\begin{aligned}\sharp\{I \in \mathcal{P}^m : i \in I\} &= \sharp\{I \in \mathcal{P}^m : n \in I\} \\ &= \sharp\{I = (i_1, \ldots, i_m) \in \mathcal{P}^m : i_1 = n\} + \ldots + \sharp\{I = (i_1, \ldots, i_m) \in \mathcal{P}^m : i_m = n\} \\ &= m \cdot \sharp\{I = (i_1, \ldots, i_m) \in \mathcal{P}^m : i_m = n\}. \end{aligned} \tag{A.2}$$

Since $\sharp\{I = (i_1, \ldots, i_m) \in \mathcal{P}^m : i_m = n\}$ is the number of $(m-1)$-tuples without repeated indices of $[\![1, n-1]\!]$, $(n-1)!/(n-m)!$, it follows that

$$\frac{m(n-1)!}{(n-m)!} \cdot \sum_{I\in\mathcal{P}^m} P^{\mathtt{W}}_{\mathbf{a}}(I) = \sum_{I\in\mathcal{P}^m}\sum_{i\in I} a_i = \sum_{i=1}^n a_i \cdot \sharp\{I \in \mathcal{P}^m : i \in I\} = m \cdot \frac{(n-1)!}{(n-m)!}. \tag{A.3}$$

This shows that $\sum_{I\in\mathcal{P}^m} P^{\mathtt{W}}_{\mathbf{a}}(I) = 1$.

We now prove that if Equation (8.21) is respected, then the minibatch OT matrix defines a transport plan. We also prove that for a W_p^p kernel, minibatch W_p^p, *i.e.,* $\overline{W_p^p}$, is an upper bound of W_p^p.

Proposition 12. *If the reweighting function w and the parametric distribution on m-tuples $P_{\mathbf{c}}$ satisfy the following admissibility condition*

$$\mathbb{E}_{I\sim P_{\mathbf{c}}} Q_I^\top w(\mathbf{c}, I) = \mathbf{c}, \qquad \forall \mathbf{c} \in \Sigma_n \tag{A.4}$$

Then with the notations of Definition 19, the averaged minibatch transport matrix $\overline{\Pi}^h_{w,P}$ is an admissible transport plan between the discrete probabilities $\mathbf{a}, \mathbf{b} \in \Sigma_n$ in the sense that $\overline{\Pi}^h_{w,P}\mathbf{1}_n = \mathbf{a}$ and $\mathbf{1}_n^\top \overline{\Pi}^h_{w,P} = \mathbf{b}^\top$. Considering the Wasserstein kernel $h = W_p^p$, the minibatch loss defined in Definition (21), as the associated coupling $\overline{\Pi}^h_{w,P}$ is not the optimal coupling of the full OT problem, it satisfies

$$\overline{h}^m_{w,P}(\mathbf{a}, \mathbf{b}) = \langle \overline{\Pi}^h_{w,P}, C\rangle_F \geq h(\mathbf{a}, \mathbf{b}). \tag{A.5}$$

Under assumption (A.4) one can safely call $\overline{\Pi}^h_{w,P}(\mathbf{a},\mathbf{b})$ an averaged minibatch transport *plan*.

Proof. By the definition of Π^m_h and the properties $Q_J\mathbf{1}_n = \mathbf{1}_m$ and $\Pi^m_{I,J}\mathbf{1}_m = \mathbf{a}_I = w(\mathbf{a}, I)$ we have

$$\overline{\Pi}^h_{w,P}\mathbf{1}_n = \mathbb{E}_{I,J}Q_I^\top \Pi^m_{I,J}Q_J\mathbf{1}_n = \mathbb{E}_{I,J}Q_I^\top \Pi^m_{I,J}\mathbf{1}_m = \mathbb{E}_{I,J}Q_I^\top \mathbf{a}_I = \mathbb{E}_I Q_I^\top \mathbf{a}_I = \mathbb{E}_{I\sim P_\mathbf{a}} Q_I^\top w(\mathbf{a}, I) = \mathbf{a}.$$

The proof that $\mathbf{1}_n^\top \overline{\Pi}^h_{w,P} = \mathbf{b}^\top$ is similar. This establishes that $\overline{\Pi}^h_{w,P} \in U(\mathbf{a},\mathbf{b})$ is an admissible transport plan between the discrete probabilities $\mathbf{a}$ and $\mathbf{b}$.

We now prove (A.5) for the Wasserstein distance $h = W_p^p$ ($\varepsilon = 0$, $1 \le p < \infty$). Since $\overline{\Pi}^h_{w,P}$ is an admissible transport plan we have:

$$h(\mathbf{a},\mathbf{b}) = \min_{\Pi\in U(\mathbf{a},\mathbf{b})} \langle \Pi, C\rangle \le \langle \overline{\Pi}^h_{w,P}, C\rangle.$$

Further, by definition of the average minibatch transport plan $\overline{\Pi}^h_{w,P}$, and observing that the matrices Q_I, Q_J from Definition 23 are such that $C_{I,J} = Q_I C Q_J^\top$, we obtain

$$\langle \overline{\Pi}^h_{w,P}, C\rangle = \langle \mathbb{E}_{I,J}\Pi_{I,J}, C\rangle = \mathbb{E}_{I,J}\langle \Pi_{I,J}, C\rangle = \mathbb{E}_{I,J}\langle Q_I^\top \Pi^m_{I,J} Q_J, C\rangle = \mathbb{E}_{I,J}\langle \Pi^m_{I,J}, C_{I,J}\rangle.$$

Now observe that by definition of the minibatch transport plans $\Pi^m_{I,J}$ (cf Definition 23) we have,

$$\langle \Pi^m_{I,J}, C_{I,J}\rangle = h\Big(w_1(\mathbf{a}, I), w_2(\mathbf{b}, J), C_{(I,J)}\Big).$$

For the Wasserstein distance $h = W_p^p$, combining all of the above we obtain

$$h(\mathbf{a},\mathbf{b}) \le \langle \overline{\Pi}^h_{w,P}, C\rangle = \mathbb{E}_{I,J}\langle \Pi^m_{I,J}, C_{I,J}\rangle = \mathbb{E}_{I,J} h\Big(w_1(\mathbf{a}, I), w_2(\mathbf{b}, J), C_{(I,J)}\Big).$$

□

We prove some associations of reweighting functions and parametric laws on m-tuple which respects the marginal constraints (A.4).

Lemma 8.1.1 (Admissibility)**.** *The uniform reweighting function $w^{\mathtt{U}}$ and the parametric law "with replacement" $P^{\mathtt{U}}$ satisfy the admissibility condition. The admissibility condition also holds for the parametric law without replacement $P^{\mathtt{W}}$ with the normalized reweighting function $w^{\mathtt{W}}$.*
In contrast for $w^{\mathtt{U}}, P^{\mathtt{W}}$ when $\mathbf{a}$ is not uniform, the resulting OT matrix is not a transportation plan.

Proof. Consider first $w^{\mathtt{W}}$ and draws with the probability law $P^{\mathtt{W}}$. This law only allows to draw m-tuples without repeated entries. Since the probability of drawing a tuple without repeated indices such that $\sum_{j\in I} a_j = 0$ is zero, without loss of generality we consider a draw I such that $\sum_{j\in I} a_j > 0$ as $a_i > 0$ with $1 \le i \le n$. Given $1 \le i \le n$, we distinguish several cases: if $i \notin I$ then $Q_I^\top w^{\mathtt{W}}(\mathbf{a}, I) = 0$; otherwise there exists $1 \le k \le m$ such that $i = i_k$, hence

$$(Q_I^\top w^{\mathtt{W}}(\mathbf{a}, I))_i = w^{\mathtt{W}}_k(\mathbf{a}, I) = \frac{a_{i_k}}{\sum_{p=1}^m a_{i_p}} = \frac{a_i}{\sum_{j\in I} a_j}.$$

As a result

$$\mathbb{E}_{I\sim P^{\mathtt{W}}_\mathbf{a}}(Q_I^\top w^{\mathtt{W}}(\mathbf{a}, I))_i = \mathbb{E}_{I\sim P^{\mathtt{W}}_\mathbf{a}} \frac{a_i}{\sum_{j\in I} a_j}\mathbf{1}_I(i) = a_i \mathbb{E}_{I\sim P^{\mathtt{W}}_\mathbf{a}} \frac{\mathbf{1}_I(i)}{\sum_{j\in I} a_j}.$$

If $a_i = 0$ the right hand side equals a_i. Assuming now $a_i > 0$, we have $\sum_{j\in I} a_j > 0$ for each I that contains i, and we prove that $\mathbb{E}_{I\sim P^{\mathtt{W}}_{\mathbf{a}}} \frac{\mathbf{1}_I(i)}{\sum_{j\in I} a_j} = 1$. Indeed, by definition of $P^{\mathtt{W}}$ we have

$$\begin{aligned}\mathbb{E}_{I\sim P^{\mathtt{W}}_{\mathbf{a}}} \frac{\mathbf{1}_I(i)}{\sum_{j\in I} a_j} &= \sum_{I\in\mathcal{P}^m} P^{\mathtt{W}}_{\mathbf{a}}(I) \frac{\mathbf{1}_I(i)}{\sum_{j\in I} a_j} = \sum_{I\in\mathcal{P}^m, I\ni i} P^{\mathtt{W}}_{\mathbf{a}}(I) \frac{1}{\sum_{j\in I} a_j} = \sum_{I\in\mathcal{P}^m, I\ni i} \frac{(n-m)!}{m(n-1)!} \\ &= \frac{(n-m)!}{m(n-1)!} \cdot \sharp\{I\in\mathcal{P}^m, i\in I\} = 1.\end{aligned}$$

Where the last equality is from (A.2). We can conclude that $\mathbb{E}_{I\sim P^{\mathtt{W}}_{\mathbf{a}}} Q_I^\top w^{\mathtt{W}}(\mathbf{a}, I) = \mathbf{a}$ for every $\mathbf{a}$.

To show that admissibility does not hold with $w^{\mathtt{U}}$ and $P^{\mathtt{W}}$, we similarly obtain

$$\mathbb{E}_{I\sim P^{\mathtt{W}}_{\mathbf{a}}} (Q_I^\top w^{\mathtt{U}}(\mathbf{a}, I))_i = \mathbb{E}_{I\sim P^{\mathtt{W}}_{\mathbf{a}}} \tfrac{1}{m} \mathbf{1}_I(i) = \tfrac{1}{m} \mathbb{E}_{I\sim P^{\mathtt{W}}_{\mathbf{a}}} \mathbf{1}_I(i).$$

When $\mathbf{a}$ is not uniform, by the pigeonhole principle there is an index i such that $a_i > 1/m$. Since the right hand side above cannot exceed $1/m$, we conclude that $\mathbb{E}_{I\sim P^{\mathtt{W}}_{\mathbf{a}}} (Q_I^\top w^{\mathtt{U}}(\mathbf{a}, I)) \neq \mathbf{a}$.

Consider now the pair $(w^{\mathtt{U}}, P^{\mathtt{U}})$. For an m-tuple $I = (i_1, \dots, i_m)$ we denote $m_j = m_j(I)$ the multiplicity of index $1 \le j \le n$ and observe that $m_1 + \cdots + m_n = m$, and $\Pi_{j\in I} a_j = \Pi_{k=1}^n a_k^{m_k}$. Vice-versa, given integers $(m_1, \dots, m_n)$ such that $m_1 + \cdots + m_n = m$ there are $m!/(m_1! \cdots m_n!)$ m-tuples I with the corresponding multiplicity. Given $1 \le i \le n$, reasoning as above we obtain

$$\begin{aligned}\mathbb{E}_{I\sim P^{\mathtt{U}}_{\mathbf{a}}} (Q_I^\top w^{\mathtt{U}}(\mathbf{a}, I))_i &= \mathbb{E}_{I\sim P^{\mathtt{U}}_{\mathbf{a}}} \frac{m_i}{m} \mathbf{1}_I(i) = \sum_{I\in[\![n]\!]^m} P^{\mathtt{U}}_{\mathbf{a}}(I) \frac{m_i}{m} \mathbf{1}_I(i) = \sum_{I\in[\![n]\!]^m} (\Pi_{j\in I} a_j) \frac{m_i}{m} \mathbf{1}_I(i) \\ &= \sum_{m_1+\cdots+m_n=m} \frac{m!}{m_1! \cdots m_n!} (\Pi_{k=1}^n a_k^{m_k}) \frac{m_i}{m} \mathbf{1}(m_i \ge 1) \\ &= \sum_{\substack{m_1+\cdots+m_n=m \\ m_i\ge 1}} \frac{m!}{m_1! \cdots m_i! \cdots m_n!} \frac{m_i}{m} \Pi_{k=1}^n a_k^{m_k} \\ &= a_i \sum_{m_1'+\cdots+m_n'=m-1} \frac{(m-1)!}{m_1'! \cdots m_i'! \cdots m_n'!} \Pi_{k=1}^n a_k^{m_k'} = a_i \left(\sum_{i=1}^n a_i\right)^{m-1} = a_i.\end{aligned}$$

In the last line, we used Newton's multinomial theorem and the fact that $\sum_{i=1}^n a_i = 1$. □

Upper bound. We now give a upper bound of minibatch optimal transport. We have access to empirical data and the distance between each data can be bounded by the maximum distance between data, *i.e.,* for two random data $\boldsymbol{x}$ and $\boldsymbol{y}$, we have : $\|\boldsymbol{x} - \boldsymbol{y}\|_2^p \leqslant 2 \max_{1\leqslant i,j\leqslant n} \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^p$.

Lemma A.1.1 (Upper bounds on OT kernels)**.** *Let $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$, $\boldsymbol{Y} = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n)$ be two n-tuple of vectors in $\mathbb{R}^d$ and C the ground cost matrix. Let $\mathbf{a}$ and $\mathbf{b}$ be two probability vectors, w_1 and w_2 be two reweighting functions and let (I, J) be two m tuples. Then, we have the following bounds for kernel OT $h \in \{\mathfrak{L}, W_p, W_p^p, \mathfrak{L}^\varepsilon, S^\varepsilon\}$:*

$$h\big(w_1(\mathbf{a}, I), w_2(\mathbf{b}, J), C_{(I,J)}(\boldsymbol{X}, \boldsymbol{Y})\big) \leqslant 2 \max_{1\leqslant i,j\leqslant n} \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^p. \tag{A.6}$$

And for $h = \mathcal{GW}$, let $C^1 = C(\boldsymbol{X}, \boldsymbol{X})$ and $C^2 = C(\boldsymbol{Y}, \boldsymbol{Y})$. Then,

$$h\big(w_1(\mathbf{a}, I), w_2(\mathbf{b}, J), C^1_{(I,I)}, C^2_{(J,J)}\big) \leqslant \max_{1\leqslant i,j,k,l\leqslant n} \|C^1_{i,k} - C^2_{j,l}\|_2^p. \tag{A.7}$$

Proof. We start with the case $h = W_\varepsilon$ for $\varepsilon \geq 0$. Note that with our choice of cost matrix $C = (C_{i,j})_{1 \leq i,j \leq n}$ one has $0 \leqslant C_{i,j} \leqslant 2 \max_{1 \leqslant i,j \leqslant n} \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^p$. Denote the optimal transport plan between I and J as $\Pi^\star = (\Pi_{i,j})$ (with respect to the cost matrix $C_{I,J}$), consider the transport plan $w_1(\mathbf{a}, I) \otimes w_2(\mathbf{b}, J)$, we directly have:

$$
\begin{aligned}
&|\langle \Pi^\star, C_{I,J} \rangle - \varepsilon H(\Pi^\star | w_1(\mathbf{a}, I) \otimes w_2(\mathbf{b}, J))|, \\
&\leq \langle w_1(\mathbf{a}, I) \otimes w_2(\mathbf{b}, J), C_{I,J} \rangle + \varepsilon (H(w_1(\mathbf{a}, I) \otimes w_2(\mathbf{b}, J) | w_1(\mathbf{a}, I) \otimes w_2(\mathbf{b}, J)), \\
&\leq 2 \max_{1 \leqslant i,j \leqslant n} \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^p. \qquad \text{(A.8)}
\end{aligned}
$$

As the second term is equal to zero in first inequality's right hand side expression. The extension to $h = S_\varepsilon$ is direct as it is a weighted sum of three terms of the form W_ε. Lastly, a similar argument gives the desired bound for the Gromov-Wasserstein distance. Let $C^1 = C(\boldsymbol{X}, \boldsymbol{X})$ and $C^2 = C(\boldsymbol{Y}, \boldsymbol{Y})$, for 2 m-tuples I, J, one can write:

$$
|\sum_{i,j,k,l} \|(C^1_{I,I})_{i,k} - (C^2_{J,J})_{j,l}\|_2^p \pi_{i,j} \pi_{k,l}| \leq \max_{1 \leqslant i,j,k,l \leqslant n} \|(C^1_{I,I})_{i,k} - (C^2_{J,J})_{j,l}\|_2^p. \qquad \text{(A.9)}
$$

Finally, in the case of data lying in a compact, the quantity $2 \max_{1 \leqslant i,j \leqslant n} \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^p$ is upper bounded by a constant $M = 2(\mathrm{diam}(\mathrm{supp}(\alpha)) \cup \mathrm{diam}(\mathrm{supp}(\beta)))^p$ and $\max_{1 \leqslant i,j,k,l \leqslant n} \|C^1_{i,k} - C^2_{j,l}\|_2^p$ is upper bounded by $M = (\mathrm{diam}(\mathrm{supp}(\alpha)) \cup \mathrm{diam}(\mathrm{supp}(\beta)))^{p^2}$.

□

In the next section, we use the upper bound on optimal transport to show a deviation bound between the empirical estimators of MBOT and their mean value in the case of compactly supported measures.

A.1.2 Concentration theorem (compactly supported measures)

In this section, we consider two compactly supported probability distributions α, β. In what follows, we are interested in concentration bounds with $m \in \mathbb{N}^*$ fixed. For $n \in \mathbb{N}^*$ and $n \geq m$, we denote by $\boldsymbol{u}$ the element of Σ_n such that $\boldsymbol{u}_i = \frac{1}{n} (1 \leq i \leq n)$. We will also often omit the dependence of various quantities (the minibatch procedure $\overline{h}$, the reweighting function w etc.) in the asymptotic parameter n. The purpose of this appendix is to prove Theorem 8.3.1. The appendix is structured as follows. In a first section, we prove the deviation between the complete estimator $\overline{h}$ and its mean. Afterwards, we provide the deviation between the complete estimator $\overline{h}$ and its incomplete counter part $\widetilde{h}$ in a second section. The third section gathers all previous propositions and lemmas to prove Theorem 8.3.1 and corollaries.

Deviation between the complete estimator $\overline{h}$ and its mean.

We focus on the first ingredient of our proof: the deviation between the complete estimator $\overline{h}$ and its mean. This proof is based on the U-statistics concentration inequality proof but needs to be adapted due to the non-uniform probability vectors $\mathbf{a}$ and $\mathbf{b}$.

From now on, the probability vectors $(\mathbf{a}^{(n)})$ and $(\mathbf{b}^{(n)})$ are sequences which depend on the number of data n. More precisely $(\mathbf{a}^{(n)})_{n \in \mathbb{N}}$ and $(\mathbf{b}^{(n)})_{n \in \mathbb{N}}$ are sequences of vectors of size n such that for each $n \in \mathbb{N}$, $\mathbf{a}^{(n)}, \mathbf{b}^{(n)} \in \Sigma_n$, we denote the space of these sequences as $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)}) \in \Sigma$. The sequence of probability vectors $(\mathbf{a}^{(n)})$ and $(\mathbf{b}^{(n)})$ can not be taken arbitrarily if we want to guarantee convergence. Hence we rely on local constraints that we defined in Chapter 8. We recall them:

Definition 25 (Local averages conditions)**.** *Let* $(\mathbf{a}^{(n)}) \in \Sigma$ *and two integers* $n, m \in \mathbb{N}^*$ *such as* $n \geq m$.
(i) *We say that* $(\mathbf{a}^{(n)})$ *satisfies the local arithmetic mean condition if there exists a constant* $D > 0$ *and* $\gamma > 0$ *such that for any* $n \in \mathbb{N}$ *and* $I \in [\![n]\!]^m$ *we have*

$$\frac{1}{m} \sum_{i \in I} \mathbf{a}_i^{(n)} \leq \frac{D}{n^\gamma}. \tag{A.10}$$

We write that $(\mathbf{a}^{(n)})$ *satisfies* $\texttt{Loc}_\texttt{A}(m, \gamma, D))$ *(or* $\texttt{Loc}_\texttt{A}(m, \gamma)$*) when the constant* D *is implicit).*
(ii) *Analogously,* $(\mathbf{a}^{(n)})$ *is said to verify the local geometric mean condition if there exists a constant* $D > 0$ *and* $\gamma > 0$ *such that for any* $n \in \mathbb{N}^*$ *and* $I \in [\![n]\!]^m$ *we have*

$$\left(\Pi_{i \in I} \mathbf{a}_i^{(n)}\right)^{\frac{1}{m}} \leq \frac{D}{n^\gamma}. \tag{A.11}$$

We write that $(\mathbf{a}^{(n)})$ *verifies* $\texttt{Loc}_\texttt{G}(m, \gamma, D))$ *(or* $\texttt{Loc}_\texttt{G}(m, \gamma)$*) when the constant* D *is implicit).*

We record the following properties of the $\texttt{Loc}_\texttt{A}$ and $\texttt{Loc}_\texttt{G}$ conditions.

Lemma 8.3.1. *Let* $m \in \mathbb{N}^*$, $\gamma > 0$ *and* $D > 0$. *Let* $(\mathbf{a}^{(n)}) \in \Sigma$ *be a sequence of probability vectors. The following statements hold:*
(i) *If* $(\mathbf{a}^{(n)})$ *verifies* $\texttt{Loc}_\texttt{A}(m, \gamma, D)$ *or* $\texttt{Loc}_\texttt{G}(m, \gamma, D)$ *then* $\gamma \leq 1$.
(ii) *If* $(\mathbf{a}^{(n)})$ *is* $\texttt{Loc}_\texttt{A}(m, \gamma, D)$ *then* $(\mathbf{a}^{(n)})$ *is* $\texttt{Loc}_\texttt{G}(m, \gamma, D)$.

Proof. We first prove (i). Let $(\mathbf{a}^{(n)}) \in \Sigma$ which verifies $\texttt{Loc}_\texttt{A}(m, \gamma, D)$ for some $m \in \mathbb{N}$, $D > 0$ and $\gamma > 1$. Fix $1 \leq \ell \leq n$ an integer. By choosing $I = (\ell, \cdots, \ell) \in [\![n]\!]^m$, we have by (A.10)

$$a_\ell^{(n)} \leq \frac{D}{n^\gamma}. \tag{A.12}$$

Hence, we find by summing (A.12) for $1 \leq \ell \leq n$

$$1 \leq D n^{1-\gamma},$$

which contradicts $(\mathbf{a}^{(n)}) \in \Sigma$. The proof is similar if $(\mathbf{a}^{(n)})$ verifies $\texttt{Loc}_\texttt{G}(m, \gamma, D)$. Lastly, (ii) follows from the arithmetic-geometric mean inequality. □

We show that in order to obtain concentration properties of the estimators $\overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ we need to ensure that the sequences $(\mathbf{a}^{(n)})$ and $(\mathbf{b}^{(n)})$ verify the *local condition* with enough decay, e.g. $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ are $\texttt{Loc}_\texttt{A}(m, \gamma)$ or $\texttt{Loc}_\texttt{G}(m, \gamma)$ for a γ sufficiently close to 1.

Hereafter, we denote by τ an absolute (and possibly large) constant. We also writes $\tau = \tau(\gamma)$ to denote constants which depend on some parameter γ.

Proposition 15 (Generalized U-statistics concentration bound)**.** *Let* $\delta \in (0, 1)$ *and* $m \geq 1$ *be a fixed integer. Consider two compactly supported distributions* α, β*, two n-tuples of empirical data* $\boldsymbol{X} \sim \alpha^{\otimes n}, \boldsymbol{Y} \sim \beta^{\otimes n}$ *and a kernel* $h \in \{W_p, W_\varepsilon, S_\varepsilon, \mathcal{G}W_p\}$ *with a ground cost* C *as in 8.2. Let the reweighting function* w *and the probability law* P *over* m*-tuple be as in Examples 1, 2, 3, 4. Let* $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)}) \in \Sigma$ *satisfy* $\texttt{Loc}_\texttt{A}(m, \gamma, D)$ *for some* $\gamma \in (\frac{3}{4}, 1]$ *and* $D > 0$. *We have the following concentration bound for the sampling without replacement*

$$\mathbb{P}\left(\left|\overline{h}^m_\texttt{W}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_\texttt{W}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})\right| \geq 2MD^2 \frac{m^{\frac{1}{2}}}{n^{2(\gamma - \frac{3}{4})}} \sqrt{2\log(2/\delta)}\right) \leq \delta, \tag{A.13}$$

where $M = \tau(\mathrm{diam}(\mathrm{supp}(\alpha)) \cup \mathrm{diam}(\mathrm{supp}(\beta)))$. *And for the sampling with replacement, let the sequence probability vectors* $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ *verify* $\mathtt{Loc}_{\mathtt{G}}(m, \gamma, D)$ *for some* $\gamma \in (1 - \frac{1}{4m}, 1]$ *and* $D > 0$. *We have the following concentration bound*

$$\mathbb{P}\left(\left|\overline{h}^m_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})\right| \geq 2M \frac{D^{2m} m^{\frac{1}{2}}}{n^{2m(\gamma - 1 + \frac{1}{4m})}} \sqrt{2\log(2/\delta)}\right) \leq \delta, \tag{A.14}$$

where $M = \tau(\mathrm{diam}(\mathrm{supp}(\alpha)) \cup \mathrm{diam}(\mathrm{supp}(\beta)))$.

Proof. The proof is inspired by the two-sample U-statistic proof from [Hoeffding 1963, section 5]. We start with the sampling without replacement case.

Sampling without replacement : We first consider the case of Example 4, i.e, when the law P is given by (8.18). The proof is based on two-sample U-statistic Hoeffding inequalities and we give it for $h \in \{W_p, W_\varepsilon, S_\varepsilon\}$ as the $\mathcal{G}W_p$ follows the same principle. The goal is to rewrite $\overline{h}^m_{\mathtt{W}}$ as a sum of terms where each term is a sum of independent random variables. Let $(\mathbf{a}^{(n)})$ and $(\mathbf{b}^{(n)})$ be as in the above. To ease the notations, the dependence in n will be implicit.

We fix $r = \lfloor n/m \rfloor$. Let $0 \leq k \leq r-1$, we define the set $I^k := \{km + 1, \cdots, km + m\}$. Then we define the function V as :

$$V(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) = \frac{1}{r} \sum_{k=0}^{r-1} P^{\mathtt{W}}_{\mathbf{a}}(I^k) P^{\mathtt{W}}_{\mathbf{b}}(I^k) h\big(w^{\mathtt{W}}(\mathbf{a}, I^k), w^{\mathtt{W}}(\mathbf{b}, I^k), C_{(I^k, I^k)}\big) \tag{A.15}$$

We recall the implicit dependence in $\boldsymbol{X}$ and $\boldsymbol{Y}$ in the right-hand-side of (A.15) through the ground cost C. In the summation below, σ_x or σ_y denotes a generic permutation of $\{1, \cdots, n\}$. We compute :

$$\frac{1}{n!^2} \sum_{\sigma_x, \sigma_y} V(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}) \tag{A.16}$$

$$= \frac{(n-m)!^2}{n!^2} \sum_{I \in \mathcal{P}^m} \sum_{J \in \mathcal{P}^m} P^{\mathtt{W}}_{\mathbf{a}}(I) P^{\mathtt{W}}_{\mathbf{a}}(J) h\Big(w^{\mathtt{W}}(\mathbf{a}, I), w^{\mathtt{W}}(\mathbf{b}, J), C_{(I,J)}\Big) \tag{A.17}$$

$$= \frac{(n-m)!^2}{n!^2} \overline{h}^m_{\mathtt{W}}(\mathbf{a}, \mathbf{b}) \tag{A.18}$$

Finally, we have :

$$\overline{h}^m_{\mathtt{W}}(\mathbf{a}, \mathbf{b}) = \frac{1}{n!^2} \sum_{\sigma_x, \sigma_y} V'(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}), \tag{A.19}$$

where the function V' is defined by

$$V' = \left(\frac{n!}{(n-m)!}\right)^2 V$$

Let us define

$$T(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) = V'(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) - \mathbb{E}\big[V'(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n)\big],$$

We have

$$\overline{h}^m_{\mathtt{W}}(\mathbf{a}, \mathbf{b}) - \mathbb{E}\big[\overline{h}^m_{\mathtt{W}}(\mathbf{a}, \mathbf{b})\big] = \frac{1}{n!^2} \sum_{\sigma_x, \sigma_y} T(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}) \tag{A.20}$$

Note that T may be rewritten as a sum as in (A.15) with $h\big(w^{\mathtt{W}}(\mathbf{a}, I^k), w^{\mathtt{W}}(\mathbf{b}, I^k), C_{(I^k, I^k)}\big)$ replaced by $h\big(w^{\mathtt{W}}(\mathbf{a}, I^k), w^{\mathtt{W}}(\mathbf{b}, I^k), C_{(I^k, I^k)}\big) - \mathbb{E}\big[h\big(w^{\mathtt{W}}(\mathbf{a}, I^k), w^{\mathtt{W}}(\mathbf{b}, I^k), C_{(I^k, I^k)}\big)\big]$ for each k.

More precisely, we write $T(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}) = \frac{1}{r}\sum_{k=0}^{r-1} T_k^{\sigma_x, \sigma_y}$ for σ_x, σ_y two permutations of $[\![n]\!]$. Here, $T_k^{\sigma_x, \sigma_y}$ are independent and centered random variables such that

$$\begin{aligned}\big|T_k^{\sigma_x, \sigma_y}\big| &\leq 2M \left\{\frac{1}{m}\frac{(n-m)!}{(n-1)!}\right\}^2 \left(\frac{n!}{(n-m)!}\right)^2 \sum_{i=km+1}^{(k+1)m} a_{\sigma_x(i)} \sum_{i=km+1}^{(k+1)m} b_{\sigma_y(i)} \\ &\leq 2MD^2 n^{2(1-\gamma)} \end{aligned} \tag{A.21}$$

thanks to the $\mathtt{Loc}_{\mathtt{A}}(m, \gamma, D)$-condition and to the constant $M = \tau(\mathrm{diam}(\mathrm{supp}(\alpha)) \cup \mathrm{diam}(\mathrm{supp}(\beta)))$. In what follows we write $T_\sigma = T\big((\boldsymbol{x}_{\sigma_x(i)})_i, (\boldsymbol{y}_{\sigma_y(i)})_i\big)$ for simplicity. From (A.20), we get

$$\begin{aligned}\mathbb{P}\left(\overline{h}_{\mathtt{W}}^m(\mathbf{a}, \mathbf{b}) - \mathbb{E}\big[\overline{h}_{\mathtt{W}}^m(\mathbf{a}, \mathbf{b})\big] \geq t\right) &\leq e^{-\lambda t}\mathbb{E}\left[e^{\lambda(\overline{h}_{\mathtt{W}}^m(\mathbf{a}, \mathbf{b}) - \mathbb{E}[\overline{h}_{\mathtt{W}}^m(\mathbf{a}, \mathbf{b})])}\right] \\ &= e^{-\lambda t}\mathbb{E}\left[e^{\lambda \frac{1}{n!^2}\sum_\sigma T_\sigma}\right] \leq e^{-\lambda t}\frac{1}{(n!)^2}\sum_\sigma \mathbb{E}\left[e^{\lambda T_\sigma}\right] \leq e^{-\lambda t}\max_\sigma \mathbb{E}\left[e^{\lambda T_\sigma}\right],\end{aligned}$$

where in the first and second inequalities, we used Markov's and Jensen's inequalities respectively. Furthermore, for any $T = T_\sigma$ we have from Lemma 7.3.1 along with (A.21),

$$\mathbb{E}\big[e^{\lambda T}\big] = \prod_{k=0}^{r-1}\mathbb{E}\big[e^{\frac{\lambda}{r}T_k}\big] \leq \prod_{k=0}^{r-1}\mathbb{E}\big[e^{\frac{2\lambda^2}{r^2}M^2D^4n^{4(1-\gamma)}}\big] = \exp\left(2\lambda^2M^2D^4mn^{3-4\gamma}\right) \tag{A.22}$$

Hence,

$$\mathbb{P}\left(\overline{h}_{\mathtt{W}}^m(\mathbf{a}, \mathbf{b}) - \mathbb{E}\big[\overline{h}_{\mathtt{W}}^m(\mathbf{a}, \mathbf{b})\big] \geq t\right) \leq \exp\Big(-\lambda t + 2\lambda^2M^2D^4m \cdot n^{3-4\gamma}\Big)$$

Optimizing the latter over $\lambda \in \mathbb{R}_+$ and following a similar reasoning for $-\big(\overline{h}_{\mathtt{W}}^m(\mathbf{a}, \mathbf{b}) - \mathbb{E}\big[\overline{h}_{\mathtt{W}}^m(\mathbf{a}, \mathbf{b})\big]\big)$ gives (A.13).

The $\mathcal{G}W_p$ proof follows the same reasoning but differs in the definition of V which would be equal to:

$$V(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) = \frac{1}{r}\sum_{k=0}^{r-1} P_{\mathbf{a}}^{\mathtt{W}}(I^k)P_{\mathbf{b}}^{\mathtt{W}}(I^k)\mathcal{G}W\big(w^{\mathtt{W}}(\mathbf{a}, I^k), w^{\mathtt{W}}(\mathbf{b}, I^k), C^1_{(I^k, I^k)}, C^2_{(I^k, I^k)}\big),$$

with ground costs $C^1 = C^{n,p}(\boldsymbol{X}, \boldsymbol{X})$ and $C^2 = C^{n,p}(\boldsymbol{Y}, \boldsymbol{Y})$.

Sampling with replacement Let us now consider the sampling with replacement $P^{\mathtt{U}}$ with the reweighting function $w^{\mathtt{U}}$. We follow the same procedure as in section 5.C from [Hoeffding 1963]. For the sake of simplicity, we abbreviate $w^{\mathtt{U}}(\mathbf{a}, I)$ as $\boldsymbol{u} \in \Sigma_m$ and give the proof for $h \in \{W_p, W_\varepsilon, S_\varepsilon\}$ as the $\mathcal{G}W$ case can be deduced from it. In this case, it is possible to rewrite $\overline{h}_{\mathtt{U}}^m$ as a sum over m-tuples without replacement, i.e.,

$$\overline{h}_{\mathtt{U}}^m(\mathbf{a}, \mathbf{b}) = \sum_{I_1 \in [\![n]\!]^m, I^2 \in [\![n]\!]^m} P_{\mathbf{a}}^{\mathtt{U}}(I_1)P_{\mathbf{b}}^{\mathtt{U}}(I_2)h\Big(\boldsymbol{u}, \boldsymbol{u}, C_{(I_1, I_2)}\Big) \tag{A.23}$$

$$= \left(n^m\frac{(n-m)!}{n!}\right)^2 \sum_{I_1 \in \mathcal{P}^m, I_2 \in \mathcal{P}^m} h^\star(I_1, I_2). \tag{A.24}$$

Where $h^\star$ is a weighted arithmetic mean of certain values of h. Let us take an example with $m = 2$,

we consider $(i_1, i_2) \in \mathcal{P}^2$ and $(j_1, j_2) \in \mathcal{P}^2$ we have:

$$\begin{aligned} h^\star\left((i_1,i_2),(j_1,j_2)\right) =& \Big(\frac{n-1}{n}\Big)^2 P^{\mathtt{U}}_{\mathbf{a}}((i_1,i_2))P^{\mathtt{U}}_{\mathbf{b}}((j_1,j_2))h(\boldsymbol{u},\boldsymbol{u},C_{(i_1,i_2),(j_1,j_2)}) \\ &+ \frac{n-1}{n^2}P^{\mathtt{U}}_{\mathbf{a}}((i_1,i_1))P^{\mathtt{U}}_{\mathbf{b}}((j_1,j_2))h(\boldsymbol{u},\boldsymbol{u},C_{(i_1,i_1),(j_1,j_2)}) \\ &+ \frac{n-1}{n^2}P^{\mathtt{U}}_{\mathbf{a}}((i_1,i_2))P^{\mathtt{U}}_{\mathbf{b}}((j_1,j_1))h(\boldsymbol{u},\boldsymbol{u},C_{(i_1,i_2),(j_1,j_1)}) \\ &+ \frac{1}{n^2}P^{\mathtt{U}}_{\mathbf{a}}((i_1,i_1))P^{\mathtt{U}}_{\mathbf{b}}((j_1,j_1))h(\boldsymbol{u},\boldsymbol{u},C_{(i_1,i_1),(j_1,j_1)}) \end{aligned}$$

More examples for V-statistics can be found in (section 5.C, [Hoeffding 1963]). Following the above example, we see that we have the bounds : $0 \leq h^\star \leq \max_{I_1} \max_{I_2} P^{\mathtt{U}}_{\mathbf{a}}(I_1)P^{\mathtt{U}}_{\mathbf{b}}(I_2)M$. From now, the proof is like the sampling without replacement proof and we only give the main differences.

We fix $r = \lfloor n/m \rfloor$. Let $0 \leq k \leq r-1$, we define the set $I^k := \{km+1, \cdots, km+m\}$. Then we define the function V as :

$$V(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) = \frac{1}{r}\sum_{k=0}^{r-1} h^\star\big(I^k, I^k\big) \tag{A.25}$$

We recall the implicit dependence in $\boldsymbol{X}$ and $\boldsymbol{Y}$ in the right-hand-side of (A.25) through the ground costs C. In the summation below, σ_x or σ_y denotes a generic permutation of $\{1, \cdots, n\}$. We compute :

$$(\frac{n^m}{n!})^2 \sum_{\sigma_x,\sigma_y} V(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}) \tag{A.26}$$

$$= (n^m)^2 \frac{(n-m)!^2}{n!^2} \sum_{I^1 \in \mathcal{P}^m} \sum_{I^2 \in \mathcal{P}^m} h^\star\Big(I^1, I^2\Big) \tag{A.27}$$

$$= \overline{h}^m_{\mathtt{U}}(\mathbf{a}, \mathbf{b}) \tag{A.28}$$

Finally, we have :

$$\overline{h}^m_{\mathtt{U}}(\mathbf{a}, \mathbf{b}) = \frac{1}{n!^2} \sum_{\sigma_x,\sigma_y} V'(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}), \tag{A.29}$$

where the function V' is defined by $V' = (n^m)^2 V$. Let us define

$$T(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) = V'(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) - \mathbb{E}\big[V'(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n)\big],$$

We have

$$\overline{h}^m_{\mathtt{U}}(\mathbf{a}, \mathbf{b}) - \mathbb{E}\big[\overline{h}^m_{\mathtt{U}}(\mathbf{a}, \mathbf{b})\big] = \frac{1}{n!^2} \sum_{\sigma_x,\sigma_y} T(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}) \tag{A.30}$$

Note that T may be rewritten as a sum as in (A.25) with $h\big(\boldsymbol{u}, \boldsymbol{u}, C_{(I^k,I^k)}\big)$ replaced by $h\big(\boldsymbol{u}, \boldsymbol{u}, C_{(I^k,I^k)}\big) - \mathbb{E}\big[h\big(\boldsymbol{u}, \boldsymbol{u}, C_{(I^k,I^k)}\big)\big]$ for each k.
More precisely, we write $T(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)}) = \frac{1}{r}\sum_{k=0}^{r-1} T_k^{\sigma_x,\sigma_y}$ for σ_x, σ_y two permutations of $[\![n]\!]$. Here, $T_k^{\sigma_x,\sigma_y}$ are independent and centered random variables such that

$$\begin{aligned} |T_k^{\sigma_x,\sigma_y}| &\leq 2Mn^{2m} \max_{I_1} P^{\mathtt{U}}_{\mathbf{a}}(I_1) \max_{I_2} P^{\mathtt{U}}_{\mathbf{b}}(I_2) \\ &\leq 2MD^{2m} n^{2m(1-\gamma)} \end{aligned} \tag{A.31}$$

With $M = \tau(\text{diam}(\text{supp}(\alpha)) \cup \text{diam}(\text{supp}(\beta)))$ and where the second inequality uses Definition 8.27. In what follows we write $T_\sigma = T\big((\boldsymbol{x}_{\sigma_x(i)})_i, (\boldsymbol{y}_{\sigma_y(i)})_i\big)$ for simplicity. From (A.30), we get

$$\mathbb{P}\left(\overline{h}^m_{\mathfrak{U}}(\mathbf{a},\mathbf{b}) - \mathbb{E}\big[\overline{h}^m_{\mathfrak{U}}(\mathbf{a},\mathbf{b})\big] \geq t\right) \leq e^{-\lambda t} \max_\sigma \mathbb{E}\left[e^{\lambda T_\sigma}\right],$$

Furthermore, for any $T = T_\sigma$ we have from Lemma 7.3.1 along with (A.31),

$$\mathbb{E}\big[e^{\lambda T}\big] \leq \prod_{k=0}^{r-1} \mathbb{E}\big[e^{2\frac{\lambda^2}{r^2}M^2D^4}\big] = \exp\left(2\frac{\lambda^2}{r}M^2D^{4m}n^{4m(1-\gamma)}\right) \tag{A.32}$$

Hence, $\mathbb{P}\left(\overline{h}^m_{\mathfrak{U}}(\mathbf{a},\mathbf{b}) - \mathbb{E}\big[\overline{h}^m_{\mathfrak{U}}(\mathbf{a},\mathbf{b})\big] \geq t\right) \leq \exp\Big(-\lambda t + 2\lambda^2 M^2 D^{4m} m n^{4m-1-4m\gamma}\Big)$

Optimizing the latter over $\lambda \in \mathbb{R}_+$ and following a similar reasoning for $-\big(\overline{h}^m_{\mathfrak{U}}(\mathbf{a},\mathbf{b}) - \mathbb{E}\big[\overline{h}^m_{\mathfrak{U}}(\mathbf{a},\mathbf{b})\big]\big)$ gives (A.14) gives the desired results. □

The sampling with replacement bounds show that when the minibatch size m gets bigger, $\mathbf{a}$ and $\mathbf{b}$ must have a γ close to 1. Now that we have a deviation between the complete estimator and its mean, we focus on the approximation of the complete estimator with its incomplete counter part.

Deviation between the incomplete and complete estimator $\overline{h}$.

We are now ready to give the second and last ingredient of our proof: a deviation between the complete and the incomplete estimators. And in order to prove it, we rely on the Hoeffding inequality.

Lemma A.1.2 (Hoeffding's inequality)**.** *Let $X_1, \ldots, X_n$ be independent random variables such that X_i takes its values in $[a_i, b_i]$ almost surely for all $i \leq n$. Let the random variable*

$$S = \sum_{i=1}^{n} (X_i - \mathbb{E}X_i).$$

Then for every $t > 0$, we have:

$$P\{S \geq t\} \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right) \tag{A.33}$$

The next reformulation of the incomplete estimator is useful to prove deviation bounds between the complete and the incomplete estimators. See Lemma A.1.3 and Theorem 8.3.4 below.

Remark 11. *Let m, $n \geq m$ and k be positive integers. Let $\big(\mathfrak{b}^{\mathbf{a},\mathbf{b}}_\ell(I,J)\big)_{I,J\in[\![n]\!]^m, 1\leq\ell\leq k}$ be a sequence of mutually independent Bernoulli variables of parameter $P_{\mathbf{a}}(I)P_{\mathbf{b}}(J)$ such that*

$$\mathfrak{b}^{P_{\mathbf{a}},P_{\mathbf{b}}}_\ell(I,J) = \begin{cases} 1 \text{ if } (I,J) \text{ has been selected in the } \ell\text{-th draw} \\ 0 \text{ otherwise.} \end{cases}$$

We then can write

$$\widetilde{h}^{m,k}_{w,P}(\mathbf{a},\mathbf{b}) = \frac{1}{k}\sum_{\ell=1}^{k} \sum_{I,J\in[\![n]\!]^m} \mathfrak{b}^{P_{\mathbf{a}},P_{\mathbf{b}}}_\ell(I,J) h\Big(w(\mathbf{a},I), w(\mathbf{b},J), C_{(I,J)}\Big)$$

$$\widetilde{\Pi}^{h,k}_{w,P}(\mathbf{a},\mathbf{b}) = \frac{1}{k}\sum_{\ell=1}^{k} \sum_{I,J\in[\![n]\!]^m} \mathfrak{b}^{P_{\mathbf{a}},P_{\mathbf{b}}}_\ell(I,J) \Pi_{I,J}.$$

Now we are ready to bound the deviation between the two quantities and the following lemma gives us the wanted deviation:

Lemma A.1.3 (Deviation bound)**.** *Consider two compactly supported distributions α, β, two n-tuples of empirical data $\boldsymbol{X} \sim \alpha^{\otimes n}, \boldsymbol{Y} \sim \beta^{\otimes n}$. Let $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)}) \in \Sigma$ be two sequences of probability vectors, let $\delta \in (0,1)$ and an integer $k \geqslant 1$. Consider a reweighting function w, a probability law over m-tuple P as in Examples 1, 2, 3, 4 and an OT kernel $h \in \{W_p, W_\varepsilon, S_\varepsilon, \mathcal{G}W_p\}$ with a ground cost C as in 8.2. We have a deviation bound between $\widetilde{h}^{m,k}_{w,P}(\mathbf{a}, \mathbf{b})$ and $\overline{h}^{m}_{w,P}(\mathbf{a}, \mathbf{b})$ depending on the number of minibatches k.*

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \overline{h}^{m}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\sqrt{\frac{2\log(2/\delta)}{k}}\right) \leq \delta, \tag{A.34}$$

where $M = \tau(\mathrm{diam}(\mathrm{supp}(\alpha)) \cup \mathrm{diam}(\mathrm{supp}(\beta)))$

Proof. Thanks to Remark 11 we have

$$\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \overline{h}^{m}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) = \frac{1}{k}\sum_{\ell=1}^{k} \omega_l$$

where $\omega_l = \sum_{I,J \in [\![n]\!]^m} \left(\mathfrak{b}_\ell^{P_{\mathbf{a}^{(n)}}, P_{\mathbf{b}^{(n)}}}(I,J) - P_{\mathbf{a}^{(n)}}(I) P_{\mathbf{b}^{(n)}}(J)\right) h\Big(w(\mathbf{a}^{(n)}, I), w(\mathbf{b}^{(n)}, J), C_{(I,J)}\Big)$. Conditioned upon $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ and $\boldsymbol{Y} = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n)$, the variables ω_l are independent, centered and bounded by $2M$ with $M = \tau(\mathrm{diam}(\mathrm{supp}(\alpha)) \cup \mathrm{diam}(\mathrm{supp}(\beta)))$ thanks to Lemma A.6. Using Hoeffding's inequality yields

$$\begin{aligned}
&\mathbb{P}(|\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \overline{h}^{m}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| > \varepsilon) \\
&\quad = \mathbb{E}[\mathbb{P}(|\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \overline{h}^{m}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| > \varepsilon | \boldsymbol{X}, \boldsymbol{Y})] \\
&\quad = \mathbb{E}[\mathbb{P}(|\frac{1}{k}\sum_{l=1}^{k} \omega_l)| > \varepsilon | \boldsymbol{X}, \boldsymbol{Y})] \\
&\quad \leqslant \mathbb{E}[2e^{\frac{-k\varepsilon^2}{2M^2}}] = 2e^{\frac{-k\varepsilon^2}{2M^2}}
\end{aligned}$$

which concludes the proof. □

Proof of Theorem 8.3.1.

We have now the three ingredients to prove Theorem 8.3.1 :

Theorem 8.3.1 (Maximal deviation bound for compactly supported distributions)**.**
Let $\delta \in (0,1)$, $k \geqslant 1$ an integer and $m \geqslant 1$ be a fixed integer. Consider two compactly supported distributions α, β, two n-tuples of empirical data $\boldsymbol{X} \sim \alpha^{\otimes n}, \boldsymbol{Y} \sim \beta^{\otimes n}$ and a kernel $h \in \{W_p, W_p^p, W_\varepsilon, S_\varepsilon, \mathcal{G}W_p\}$ with a ground cost C as in 8.2. Let the reweighting function w and the probability law over m-tuple P be as in Examples 1, 2, 3, 4. Let the sequences of probability vectors $(\mathbf{a}^{(n)}) \in \Sigma$ and $(\mathbf{b}^{(n)}) \in \Sigma$ satisfy $\mathtt{Loc}_{\mathtt{A}}(m, \gamma, D)$ and let $D > 0$ and $\gamma \in (\frac{3}{4}, 1]$. We have a deviation bound for the sampling without replacement between $\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ and $\mathbb{E}\overline{h}^{m}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ depending on the number of empirical data n and the number of batches k:

$$\mathbb{P}\left(|\widetilde{h}^{k}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^{m}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\Big(2\frac{D^2 m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}\sqrt{2\log(\frac{2}{\delta})} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\Big)\right) \leq \delta. \tag{A.35}$$

where $M = \tau(\mathrm{diam}(\mathrm{supp}(\alpha)) \cup \mathrm{diam}(\mathrm{supp}(\beta)))$. *And for the sampling with replacement, let the sequences of probability vectors* $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ *verify* $\mathtt{Loc}_{\mathtt{G}}(m, \gamma, D)$ *for some* $\gamma \in (1 - \frac{1}{4m}, 1]$ *and* $D > 0$.

$$\mathbb{P}\left(|\widetilde{h}^k_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\Big(2\frac{D^{2m}m^{\frac{1}{2}}}{n^{2m(\gamma-1+\frac{1}{4m})}}\sqrt{2\log(\frac{2}{\delta})} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\Big)\right) \leq \delta, \tag{A.36}$$

Proof. With the triangle inequality, we have:

$$\begin{aligned}&|\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \\ &\quad \leq |\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| + |\overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})|\end{aligned} \tag{A.37}$$

Thanks to Lemma A.1.3 and 15 we get the desired results. □

Corollary. It is also possible to have a bound on the expectation over the batch couples and the data.

Corollary 8.31. *With the same hypothesis and notations as in Theorem 8.3.1. The following inequality holds:*

$$\mathbb{E}[|\widetilde{h}^k_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})|] \leqslant 20 \cdot M \max\left(2\sqrt{2}D^2\frac{m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}, \sqrt{\frac{2}{k}}\right) \tag{A.38}$$

$$\mathbb{E}[|\widetilde{h}^k_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})|] \leqslant 20 \cdot M \max\left(2\sqrt{2}D^{2m}\frac{m^{\frac{1}{2}}}{n^{2m(\gamma-1+\frac{1}{4m})}}, \sqrt{\frac{2}{k}}\right) \tag{A.39}$$

Proof. Once again, we give the proof for the sampling without replacement and the proof for the sampling with replacement follows the same steps. The proof for the debiased minibatch is straight forward as we have three terms of the form $\widetilde{h}^k_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$.

Let us recall the formula : for a real random variable X. If $X \geqslant 0$ then $\mathbb{E}[X] = \int_0^\infty \mathbb{P}(X > \lambda)d\lambda$. We denote by X the random variable $|\widetilde{h}^k_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})|$. The last Theorem 8.3.1 writes

$$\mathbb{P}\left(X > C\sqrt{\log(\frac{2}{\delta})}\right) \leqslant \delta$$

where $C := 2M \max(2\sqrt{2}D^2\frac{m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}, \sqrt{\frac{2}{k}})$. We can rewrite it as

$$\mathbb{P}(X > \lambda) \leqslant \exp\left(-\frac{\lambda^2}{C^2}\right)$$

Thus, using the formula above:

$$\begin{aligned}\mathbb{E}[X] &\leqslant \int_0^\infty \exp\left(-\frac{\lambda^2}{C^2}\right)d\lambda \\ &= C\int_0^\infty \exp(-u^2)du \leqslant 10C\end{aligned}$$

as announced. □

Bounded cost. In this paragraph we also prove the case for bounded cost.

Theorem 8.3.3 (Maximal deviation bound for bounded cost)**.** *Let* $\delta \in (0, 1)$, $k \geqslant 1$ *an integer and* $m \geqslant 1$ *be a fixed integer. Consider two distributions* α, β *possibly unbounded, two* n*-tuples of empirical data* $\boldsymbol{X} \sim \alpha^{\otimes n}, \boldsymbol{Y} \sim \beta^{\otimes n}$ *and a kernel* $h \in \{W_p, W_p^p, \mathfrak{L}^\varepsilon, S^\varepsilon, \mathcal{G}W\}$ *with a ground cost* C *as in 8.2. Let*

the reweighting function w and the probability law over m-tuple P be as in Examples 1, 2, 3, 4. Let the sequences of probability vectors $(\mathbf{a}^{(n)}) \in \Sigma$ and $(\mathbf{b}^{(n)}) \in \Sigma$ satisfy $\mathtt{Loc_A}(m, \gamma, D)$ and let $D > 0$ and $\gamma \in (\frac{3}{4}, 1]$. We have a deviation bound for the sampling without replacement between $\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ and $\mathbb{E}\overline{h}^{m}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})$ depending on the number of empirical data n and the number of batches k:

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^{m}_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\Big(2\frac{D^2 m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}\sqrt{2\log(\frac{2}{\delta})} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\Big)\right) \leq \delta, \tag{A.40}$$

where M is an upper bound on the ground cost. And for the sampling with replacement, let the sequences of probability vectors $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ verify $\mathtt{Loc_G}(m, \gamma, D)$ for some $\gamma \in (1 - \frac{1}{4m}, 1]$ and $D > 0$, we have:

$$\mathbb{P}\left(|\widetilde{h}^{m,k}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^{m}_{\mathtt{U}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})| \geq M\Big(2\frac{D^{2m} m^{\frac{1}{2}}}{n^{2m(\frac{1}{4m}-1+\gamma)}}\sqrt{2\log(\frac{2}{\delta})} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}}\Big)\right) \leq \delta. \tag{A.41}$$

Proof. The proof is the same as for Theorem 8.3.1. The only difference is that we do not need to have compactly supported data to bound the OT kernel in (A.22), as the OT kernel is bounded by assumption on the ground cost. □

A.1.3 Concentration theorem (sub-Gaussian)

In this section we relax the assumption of bounded data and give a proof for Theorem 8.3.2. The proof for the sub-Gaussian case rely on a truncation argument between data that lie in some compact data which do not. We start by recalling the sub-Gaussian data definition:

Definition 28 (sub-Gaussian random vectors)**.** *A random vector $\boldsymbol{x} \in \mathbb{R}^d$ is sub-Gaussian, if there exists $\sigma \in \mathbb{R}$ so that:*

$$\mathbb{E}e^{\langle \boldsymbol{y}, \boldsymbol{x} - \mathbb{E}\boldsymbol{x}\rangle} \leq e^{\frac{\|\boldsymbol{y}\|^2\sigma^2}{2}}, \quad \forall \boldsymbol{y} \in \mathbb{R}^d$$

We write the class of sub-Gaussian random vectors as $\mathrm{SG}(\sigma)$. For the sake of simplicity, we consider the following class of random vectors:

Definition 29 (Norm sub-Gaussian data [Jin 2019])**.** *Let $\boldsymbol{x} \in \mathbb{R}^d$ be a random vector and $\rho_{\boldsymbol{x}} = \mathbb{E}\boldsymbol{x}$. We say that $\boldsymbol{x} \in \mathrm{normSG}(\rho_{\boldsymbol{x}}, \sigma^2)$ for some $\sigma > 0$ if the following inequality holds*

$$\mathbb{P}\big(\|\boldsymbol{x} - \rho_{\boldsymbol{x}}\| \geq t\big) \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right). \tag{A.42}$$

Norm sub-Gaussian random vectors are a generalization of both sub-Gaussian random vectors and norm bounded random vectors. They show tighter concentration bounds than sub-Gaussian random vectors. We also have the following inclusion: $\mathrm{SG}(\sigma/\sqrt{d}) \subset \mathrm{normSG}(\sigma) \subset \mathrm{SG}(\sigma)$, see [Jin 2019] for a detailed review of the connections between these two random vector classes.

Theorem 8.3.2 (Concentration inequality sub-Gaussian data)**.** *Let the cost $C = C^{n,p}$ be defined as in (8.2). Let $(\boldsymbol{x}_i)_{1\leq i\leq n}$ and $(\boldsymbol{y}_i)_{1\leq i\leq n}$ be two i.i.d. sequences of random vectors such that $\boldsymbol{x}_1 \in \mathrm{normSG}(\rho_{\boldsymbol{x}}, \sigma^2_{\boldsymbol{x}})$ and $\boldsymbol{y}_1 \in \mathrm{normSG}(\rho_{\boldsymbol{y}}, \sigma^2_{\boldsymbol{y}})$ with $\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{y}} > 0$ and $\rho_{\boldsymbol{x}}, \rho_{\boldsymbol{y}} \in \mathbb{R}^d$. Let us introduce*

$$\sigma := \min(\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{y}})$$
$$\rho := \|\rho_{\boldsymbol{x}} - \rho_{\boldsymbol{y}}\|_2$$

Let the sequence probability vectors $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)})$ *verify* $\mathtt{Loc}_{\mathtt{A}}(m,\gamma,D)$ *for some* $\gamma \in (\frac{3}{4}, 1]$ *and* $D > 0$. *We assume that* n *verifies the following condition:*

$$n \geq \tau(m, \sigma, \rho, D, p). \tag{A.43}$$

Consider $m \geqslant 1$ *be a fixed integer and a kernel* $h \in \{W_p, W^{\varepsilon}, S^{\varepsilon}\}$. *Let the reweighting function* $w^{\mathtt{W}}$ *and the probability law over* m*-tuple* $P^{\mathtt{W}}$ *be as in Examples 2 and 4. Then we have the following concentration bound for the sampling without replacement:*

$$\mathbb{P}\left(\left|\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})\right| \geq (2^{3p+4}m)^{\frac{1}{2}}\sigma^p D^2 \cdot \frac{\log(4n)^{\frac{p+1}{2}}}{n^{2(\gamma-\frac{3}{4})}}\right) \leq 4n^{-\frac{1}{2^p}}, \tag{A.44}$$

Proof. Let us fix $\varepsilon > 0$ to be chosen later. We use the following notations for $\delta > 0$

$$\Delta := \overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}), \tag{A.45}$$

$$A_\delta := \bigcap_{i=1}^{n} \{\|\boldsymbol{x}_i - \rho_{\boldsymbol{x}}\|_2 \leq \delta\} \cap \{\|\boldsymbol{y}_i - \rho_{\boldsymbol{y}}\|_2 \leq \delta\} \tag{A.46}$$

We estimate,

$$\mathbb{P}[|\Delta| \geq \varepsilon] \leq \mathbb{P}\big[|\Delta|\mathbf{1}_{A_\delta} \geq \frac{\varepsilon}{2}\big] + \mathbb{P}\big[|\Delta|\mathbf{1}_{A^c_\delta} \geq \frac{\varepsilon}{2}\big], \tag{A.47}$$

First, by the union bound, we have

$$\mathbb{P}\big[|\Delta|\mathbf{1}_{A^c_\delta} \geq \frac{\varepsilon}{2}\big] \leq \mathbb{P}\big[A^c_\delta\big] \leq 4n \exp\left(-\frac{\delta^2}{2\sigma^2}\right) \tag{A.48}$$

Next, we claim that

$$\begin{aligned}\mathbb{P}\big[|\Delta|\mathbf{1}_{A_\delta} \geq \frac{\varepsilon}{2}\big] \leq 2\exp\Big(-\frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+2}D^4(\rho+\delta)^{2p}}\varepsilon^2\Big) \\ + 4n\exp\big(-\frac{\delta^2}{2\sigma^2}\big)\end{aligned} \tag{A.49}$$

assuming the following conditions together with (A.43),

$$\varepsilon \geq 4n^{1-2\gamma} \cdot n^{\frac{1}{5}} \tag{A.50}$$

$$\delta \geq \sigma\sqrt{2\log(n)}. \tag{A.51}$$

Let us show how (A.49) comes from a slight modification of the (proof) of (A.13). Using the same notations as in the proof of (A.13) we have (in place of (A.20)):

$$\Delta\mathbf{1}_{A_\delta} = \frac{1}{n!^2}\sum_{\sigma_x,\sigma_y} T(\boldsymbol{x}_{\sigma_x(1)}, \cdots, \boldsymbol{x}_{\sigma_x(n)}, \boldsymbol{y}_{\sigma_y(1)}, \cdots, \boldsymbol{y}_{\sigma_y(n)})\mathbf{1}_{A_\delta},$$

and

$$T(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n) = \frac{1}{r}\sum_{k=0}^{r-1} T_k\big((\boldsymbol{x}_i)_{i\in I^k}, (\boldsymbol{y}_i)_{i\in I^k}\big). \tag{A.52}$$

We emphasize that for each $1 \leq k \leq r-1$, T_k depends only on the set $I^k := \{km+1, \cdots, km+m\}$. Moreover, the variables $(T_k)_k$ are mutually independent.

As in the proof of Proposition 15 is suffices to estimate $\mathbb{P}(\Delta\mathbf{1}_{A_\delta} \geq t)$ up to a factor 2. We have,

$$\mathbb{P}(\Delta\mathbf{1}_{A_\delta} \geq t) \leq e^{-\lambda t}\max_{\sigma}\mathbb{E}\left[e^{\lambda T_\sigma \mathbf{1}_{A_\delta}}\right]. \tag{A.53}$$

For simplicity, we assume that the maximum in (A.53) is attained at $\sigma = \mathrm{Id}$. Observe the following equality

$$\mathbf{1}_{A_\delta} = \prod_{i=1}^{n} \mathbf{1}_{B(\rho_{\boldsymbol{x}},\delta)}(X_i)\mathbf{1}_{B(\rho_{\boldsymbol{y}},\delta)}(Y_i) \tag{A.54}$$

We write $T = T_{\mathrm{Id}}$ and insert the indicator function $\mathbf{1}_{B(\rho_{\boldsymbol{x}},\delta)^c}(X_1)$ using (A.54) and (A.52),

$$\begin{aligned}\mathbb{E}\left[e^{\lambda T\mathbf{1}_{A_\delta}}\right] &= \mathbb{E}\Big[\exp\Big(\frac{\lambda}{r}\sum_{k=1}^{r-1} T_k\mathbf{1}_{A_\delta}\Big)\Big] \\ &= \mathbb{E}\big[\mathbf{1}_{B(\rho_{\boldsymbol{x}},\delta)^c}(X_1)\big] + \mathbb{E}\Big[\exp\Big(\frac{\lambda}{r}\sum_{k=1}^{r-1} T_k\mathbf{1}_{A_\delta}\Big)\mathbf{1}_{B(\rho_{\boldsymbol{x}},\delta)}(X_1)\Big].\end{aligned}$$

Hence, by repeating this procedure we find, using the tail estimate on the variables X_k, Y_k $(1 \le k \le n)$:

$$\mathbb{E}\left[e^{\lambda T_{\mathrm{Id}}\mathbf{1}_{A_\delta}}\right] \le 4n\exp\big(-\frac{\delta^2}{2\sigma^2}\big) + \mathbb{E}\Big[\exp\Big(\frac{\lambda}{r}\sum_{k=1}^{r-1} T_k\Big)\mathbf{1}_{A_\delta}\Big]. \tag{A.55}$$

The first term in the right on side is the same as in (A.48) so this loss is acceptable. In the following, we estimate $\mathbb{E}\Big[\exp\Big(\frac{\lambda}{r}\sum_{k=1}^{r-1} T_k\Big)\mathbf{1}_{A_\delta}\Big]$. We denote, for each $0 \le k \le r-1$, by $A_\delta(k)$ the set of data from $\boldsymbol{X}(I^k)$ which belong to the ball of radius δ around their mean:

$$A_\delta(k) := \bigcap_{i\in I^k} \{\|\boldsymbol{x}_i - \rho_{\boldsymbol{x}}\|_2 \le \delta\} \cap \{\|\boldsymbol{y}_i - \rho_{\boldsymbol{y}}\|_2 \le \delta\},$$

and the set of the remaining $n - rm$ data:

$$A_\delta(r) := \bigcap_{i=rm+1}^{n} \{\|\boldsymbol{x}_i - \rho_{\boldsymbol{x}}\|_2 \le \delta\} \cap \{\|\boldsymbol{y}_i - \rho_{\boldsymbol{y}}\|_2 \le \delta\}.$$

Using the independence properties of the T_k's recalled above we get

$$\begin{aligned}\mathbb{E}\Big[\exp\Big(\frac{\lambda}{r}\sum_{k=1}^{r-1} T_k\Big)\mathbf{1}_{A_\delta}\Big] &= \mathbb{E}\Big[\prod_{k=1}^{r-1}\Big(\exp\big(\frac{\lambda}{r}T_k\big)\mathbf{1}_{A_\delta(k)}\Big)\mathbf{1}_{A_\delta(r)}\Big] \\ &= \prod_{k=1}^{r-1}\mathbb{E}\Big[\exp\big(\frac{\lambda}{r}T_k\big)\mathbf{1}_{A_\delta(k)}\Big]\mathbb{E}\big[\mathbf{1}_{A_\delta(r)}\big] \\ &\le \prod_{k=1}^{r-1}\mathbb{E}\Big[\exp\big(\frac{\lambda}{r}T_k\big)\mathbf{1}_{A_\delta(k)}\Big].\end{aligned} \tag{A.56}$$

Hence, it suffices to estimate $\mathbb{E}\Big[\exp\big(\frac{\lambda}{r}T_k\big)\mathbf{1}_{A_\delta(k)}\Big]$ for a fixed $0 \le k \le r-1$. Let $0 \le k \le r-1$. From Lemma A.6, the $\mathtt{Loc}_{\mathtt{A}}(m,\gamma,D)$ property of our variables, we have for $\omega \in A_\delta(k)$:

$$\big|T_k^\omega\big((\boldsymbol{x}_i)_{i\in I^k},(\boldsymbol{y}_i)_{i\in I^k}\big)\big| \le \big(2(\rho+\delta)\big)^p D^2 n^{2(1-\gamma)} =: \mathrm{M}_\delta. \tag{A.57}$$

Unfortunately, we can not use Lemma 7.3.1 on T_k^ω because we need to have a bounded random variable. To overcome this issue we introduce:

$$T_k^\delta := T_k\mathbf{1}_{|T_k|\le \mathrm{M}_\delta} + M_\delta\mathbf{1}_{|T_k|>\mathrm{M}_\delta}. \tag{A.58}$$

which is bounded by M_δ. Note that by construction the following equality holds

$$\mathbb{E}\Big[\exp\big(\frac{\lambda}{r}T_k\big)\mathbf{1}_{A_\delta(k)}\Big] = \mathbb{E}\Big[\exp\big(\frac{\lambda}{r}T_k^\delta\big)\mathbf{1}_{A_\delta(k)}\Big].$$

Thus we can compute using Lemma 7.3.1:

$$\begin{aligned}\mathbb{E}\Big[\exp\big(\frac{\lambda}{r}T_k\big)\mathbf{1}_{A_\delta(k)}\Big] &= \mathbb{E}\Big[\exp\big(\frac{\lambda}{r}(T_k^\delta - \mathbb{E}\big[T_k^\delta\big])\big)\mathbf{1}_{A_\delta(k)}\Big]\exp\big(\frac{\lambda}{r}\mathbb{E}\big[T_k^\delta\big]\big)\\ &\leq \exp\big(\frac{\lambda^2 M_\delta^2}{8r^2}\big)\exp\big(\frac{\lambda}{r}\mathbb{E}\big[T_k^\delta\big]\big). \end{aligned} \tag{A.59}$$

Hence, taking the products in (A.59) for $0 \leq k \leq r$, we have

$$(\text{A.56}) \leq \prod_{k=0}^{r-1}\exp\big(\frac{\lambda^2 M_\delta^2}{8r^2}\big)\cdot\prod_{k=0}^{r-1}\exp\big(\frac{\lambda}{r}\mathbb{E}\big[T_k^\delta\big]\big). \tag{A.60}$$

From Lemma A.1.4 proven below, we have

$$\big|\mathbb{E}\big[T_k^\delta\big]\big| \leq n^{1-2\gamma+\frac{1}{5}} \tag{A.61}$$

Thus, combining (A.60) and (A.61) we get by optimizing in λ the inequalities,

$$\begin{aligned}(\text{A.53}) &\leq \exp\Big(\frac{\lambda^2 m M_\delta^2}{4n} - \lambda(t - n^{1-2\gamma}\cdot n^{\frac{1}{5}})\Big)\\ &\leq \exp\Big(-\frac{n}{m\mathrm{M}_\delta^2}(t - n^{1-2\gamma}\cdot n^{\frac{1}{5}})^2\Big)\\ &\leq \exp\Big(-\frac{n}{4m\mathrm{M}_\delta^2}t^2\Big),\\ &= \exp\Big(-\frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+1}mD^4(\rho+\delta)^{2p}}t^2\Big), \end{aligned} \tag{A.62}$$

assuming

$$t \geq 2n^{1-2\gamma}\cdot n^{\frac{1}{5}}, \tag{A.63}$$

which comes from (A.50) since $t = \frac{\varepsilon}{2}$. Together with (A.55), this shows the claim (A.49).

Thus, combining (A.48) and (A.49) gives

$$\begin{aligned}(\text{A.47}) &\leq 2\exp\Big(-\frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+2}D^4(\rho+\delta)^{2p}}\varepsilon^2\Big) + 8n\exp\Big(-\frac{\delta^2}{2\sigma^2}\Big)\\ &=: \Xi_1 + \Xi_2 \end{aligned} \tag{A.64}$$

Setting $\delta = \delta_n$ such that $\Xi_1 = \Xi_2$ we find that δ_n satisfies the equation given by:

$$P(\delta) := a\delta^2(\rho+\delta)^{2p} + b(\rho+\delta)^{2p} + c = 0 \tag{A.65}$$

with

$$a := \frac{1}{2\sigma^2}, \tag{A.66}$$

$$b := -\log(4n), \tag{A.67}$$

$$c := -\frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+2}mD^4}\varepsilon^2 \tag{A.68}$$

Such a δ_n exists by the intermediate value theorem since $P(0) < 0$ and $\lim_{\delta \to +\infty} P(\delta) = +\infty$. Note that (A.65) writes

$$\frac{1}{2\sigma^2}\delta_n^2(\rho+\delta_n)^{2p} = \log(4n)(\rho+\delta_n)^{2p} + \frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+2}mD^4}\varepsilon^2, \tag{A.69}$$

which implies

$$\delta_n \geq \sigma\sqrt{2\log(4n)} \tag{A.70}$$

We are interested in getting an upper bound of our quantity. Let us assume that we have:

$$\frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+2}mD^4}\varepsilon^2 \leq \log(4n)(\rho+\delta_n)^{2p}. \tag{A.71}$$

If (A.71) holds then we have

$$\frac{1}{2\sigma^2}\delta_n^2(\rho+\delta_n)^{2p} \leq 2\log(4n)(\rho+\delta_n)^{2p},$$

and hence

$$\delta_n \leq 2\sigma\sqrt{\log(4n)}, \tag{A.72}$$

showing that (A.70) is essentially sharp. We now investigate under which condition (A.71) holds. From (A.70) the condition (A.71) holds if we have

$$\frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+2}mD^4}\varepsilon^2 \leq \log(4n)\big(\rho+\sigma\sqrt{2\log(4n)}\big)^{2p}.$$

It suffices to have,

$$\frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+2}mD^4}\varepsilon^2 \leq \log(4n)(2\sigma^2\log(4n))^p. \tag{A.73}$$

We thus choose the parameter $\mathbf{e}$ as follows

$$\varepsilon^2 = 2^{3p+4}m\sigma^{2p}D^4 \cdot \frac{\log(4n)^{p+1}}{n^{4(\gamma-\frac{3}{4})}}. \tag{A.74}$$

Note that the condition (A.74) implies that (A.50) is verified for n large enough (i.e. under (A.43)). If (A.74) holds, then the condition (A.73) is verified and hence (A.72) holds. We now compute,

$$\begin{aligned}
(A.64) &= 2\Xi_1 \\
&\leq 4\exp\Big(-\frac{n^{4(\gamma-\frac{3}{4})}}{4^{p+2}mD^4(\rho+\delta)^{2p}}\varepsilon^2\Big) \\
&\leq 4\exp\Big(-\frac{(2\sigma^2)^p\log(4n)^{p+1}}{(\rho+\delta)^{2p}}\Big) \\
&\leq 4\exp\Big(-\frac{\log(4n)}{2^p}\Big) \\
&\leq 4n^{-\frac{1}{2^p}}
\end{aligned}$$

□

Lemma A.1.4. *Assuming the conditions (A.50), (A.51) and (A.43) we have the following bound:*

$$\big|\mathbb{E}\big[T_k^\delta\big]\big| \leq n^{1-2\gamma+\frac{1}{5}}$$

Proof. We write

$$\begin{aligned}\mathbb{E}\big[T_k^\delta\big] &= \mathbb{E}\big[T_k^\delta \mathbf{1}_{A_\delta(k)}\big] + \mathbb{E}\big[T_k^\delta \mathbf{1}_{A_\delta(k)^c}\big] \\ &=: \mathrm{I} + \mathrm{II}\end{aligned} \tag{A.75}$$

From (A.58) we observe using the definition of M_δ in (A.57), the union bound and the inequality $e^{-x} = e^{-\frac{1}{5}x} \cdot e^{-\frac{4}{5}x} \leq \frac{\tau}{x^p} \cdot e^{-\frac{4}{5}x}$ for $x, p \geq 1$ and some constant $\tau = \tau(p)$.

$$\begin{aligned}|\mathrm{II}| &\leq \mathrm{M}_\delta \cdot \mathbb{P}\big(A_\delta(k)^c\big) \\ &\leq \big(2(\rho+\delta)\big)^p D^2 n^{2(1-\gamma)} \cdot 4m \exp\left(-\frac{\delta^2}{2\sigma^2}\right) \\ &\leq \tau(D,m,\sigma,p) \cdot \delta^p n^{2(1-\gamma)} \cdot \frac{\tau(p)}{\delta^{2p}} \exp\left(-\frac{4}{5} \cdot \frac{\delta^2}{2\sigma^2}\right) \\ &\leq \tau(D,m,\sigma,p) \cdot \frac{1}{\log(n)^{\frac{p}{2}}} n^{2(1-\gamma)} \cdot n^{-\frac{4}{5}} \\ &\leq \frac{1}{2} n^{1-2\gamma+\frac{1}{5}}.\end{aligned} \tag{A.76}$$

In the above, we used the assumptions (A.51) and (A.43) in the last two inequalities.

We now estimate the contribution of I. Recalling the definition of T_k^δ in (A.58) we observe the following using the definition of M_δ in (A.57) and the fact that T_k is a mean-zero random variable,

$$\begin{aligned}\mathrm{I} &= \mathbb{E}\big[T_k \mathbf{1}_{|T_k| \leq \mathrm{M}_\delta} \mathbf{1}_{A_\delta(k)}\big] = \mathbb{E}\big[T_k \mathbf{1}_{A_\delta(k)}\big] \\ &= -\mathbb{E}\big[T_k \mathbf{1}_{A_\delta(k)^c}\big].\end{aligned} \tag{A.77}$$

Note that the the indicator function $\mathbf{1}_{A_\delta(k)^c}$ can be expressed as the superposition of at most 4^m functions of the form

$$f(J_1, J_2) := \prod_{k_1 \in J_1} \mathbf{1}_{B(\rho_{\boldsymbol{x}},\delta)^c}(\boldsymbol{x}_{k_1}) \cdot \prod_{k_2 \in J_2} \mathbf{1}_{B(\rho_{\boldsymbol{y}},\delta)^c}(\boldsymbol{y}_{k_2}), \tag{A.78}$$

where $J_1, J_2 \subset I^k$ with $|J_1| + |J_2| \geq 1$. Consider a positive real numbers r and a vector $\boldsymbol{z}$, we denote an annulus around the vector $\boldsymbol{z}$ as $\mathbb{A}(\boldsymbol{z}, r)$, *i.e.,* $\boldsymbol{x} \in \mathbb{A}(\boldsymbol{z}, r)$ if $r \leq \|\boldsymbol{x} - \boldsymbol{z}\|_2 \leq 2r$. For $i \in \{1, 2\}$ let $t_i = |J_i|$ and write $J_i = \{j_1(i), \cdots, j_{t_i}(i)\}$. By decomposing each indicator function in (A.78) dyadically we get from the (mutual) independence of the variables $\{\boldsymbol{x}_i, \boldsymbol{y}_j : (i,j) \in J_1 \times J_2\}$ and Lemma A.1.1,

$$\begin{aligned}&\big|\mathbb{E}\big[T_k f(J_1, J_2)\big]\big| = \\ &= \left|\mathbb{E}\left[T_k \prod_{i_1=1}^{t_1} \Big(\sum_{\ell(j_{i_1}(1)) \geq 0} \mathbf{1}_{\mathbb{A}(\rho_{\boldsymbol{x}}, 2^{\ell(j_{i_1}(1))}\delta)}(\boldsymbol{x}_{j_{i_1}(1)})\Big) \times \prod_{i_2=1}^{t_2} \Big(\sum_{\ell(j_{i_2}(2)) \geq 0} \mathbf{1}_{\mathbb{A}(\rho_{\boldsymbol{y}}, 2^{\ell(j_{i_2}(2))}\delta)}(\boldsymbol{y}_{j_{i_2}(2)})\Big)\right]\right| \qquad \text{(A.79)} \\ &= \left| \sum_{\substack{\ell(j_1(1)), \cdots \ell(j_{t_1}(1)) \geq 0 \\ \ell(j_1(2)), \cdots \ell(j_{t_2}(2)) \geq 0}} \mathbb{E}\left[T_k \prod_{\substack{1 \leq i_1 \leq t_1 \\ 1 \leq i_2 \leq t_2}} \mathbf{1}_{\mathbb{A}(\rho_{\boldsymbol{x}}, 2^{\ell(j_{i_1}(1))}\delta)}(\boldsymbol{x}_{j_{i_1}(1)}) \mathbf{1}_{\mathbb{A}(\rho_{\boldsymbol{y}}, 2^{\ell(j_{i_2}(2))}\delta)}(\boldsymbol{y}_{j_{i_2}(2)})\right]\right| \\ &\leq \sum_{\substack{\ell(j_1(1)), \cdots \ell(j_{t_1}(1)) \geq 0 \\ \ell(j_1(2)), \cdots \ell(j_{t_2}(2)) \geq 0}} D^2 n^{2(1-\gamma)} (\rho + 2^{\ell_{\max}+1}\delta)^p \prod_{\substack{1 \leq i_1 \leq t_1 \\ 1 \leq i_2 \leq t_2}} \mathbb{E}\Big[\mathbf{1}_{\mathbb{A}(\rho_{\boldsymbol{x}}, 2^{\ell(j_{i_1}(1))}\delta)}(\boldsymbol{x}_{j_{i_1}(1)})\Big] \mathbb{E}\Big[\mathbf{1}_{\mathbb{A}(\rho_{\boldsymbol{y}}, 2^{\ell(j_{i_2}(2))}\delta)}(\boldsymbol{y}_{j_{i_2}(2)})\Big],\end{aligned} \tag{A.80}$$

where $\ell_{max} = \max\big(\ell(j_1(1)), \cdots \ell(j_{t_1}(1)), \ell(j_1(2)), \cdots \ell(j_{t_2}(2)\big)$. Hence, the sub-Gaussian assumption on the family $\{\boldsymbol{x}_i, \boldsymbol{y}_j : (i,j) \in J_1 \times J_2, \}$ yields using again the inequality $e^{-x} = e^{-\frac{1}{5}x} \cdot e^{-\frac{4}{5}x} \leq \frac{C}{x^p} \cdot e^{-\frac{4}{5}x}$

for $x, p \geq 1$ and some constant $\tau = \tau(p)$,

$$\begin{aligned}
(\text{A.80}) &\leq D^2 n^{2(1-\gamma)} \sum_{\substack{\ell(j_1(1)),\cdots \ell(j_{t_1}(1))\geq 0 \\ \ell(j_1(2)),\cdots \ell(j_{t_2}(2))\geq 0}} (\rho + 2^{\ell_{\max}+1}\delta)^p \\
&\quad \times \prod_{\substack{1\leq i_1\leq t_1 \\ 1\leq i_2\leq t_2}} \exp\Big(-\frac{(2^{\ell(j_{i_1}(1))}\delta)^2}{2\sigma^2}\Big) \exp\Big(-\frac{(2^{\ell(j_{i_2}(2))}\delta)^2}{2\sigma^2}\Big) \\
&\leq \tau(p,m,\sigma,D)\cdot n^{2(1-\gamma)} \exp\Big(-\frac{2\delta^2}{5\sigma^2}(t_1+t_2)\Big) \\
&\quad \times \sum_{\substack{\ell(j_1(1)),\cdots \ell(j_{t_1}(1))\geq 0 \\ \ell(j_1(2)),\cdots \ell(j_{t_2}(2))\geq 0}} (2^{\ell_{\max}}\delta)^p \cdot \prod_{1\leq i_1\leq t_1} \frac{1}{(2^{\ell(j_{i_1}(1))}\delta)^{2p}} \cdot \prod_{1\leq i_2\leq t_2} \frac{1}{(2^{\ell(j_{i_2}(2))}\delta)^{2p}} \\
&\leq \tau(p,m,\sigma,D)\cdot n^{2(1-\gamma)} \exp\Big(-\frac{2\delta^2}{5\sigma^2}(t_1+t_2)\Big) \cdot \frac{\delta^p}{\delta^{2(t_1+t_2)}} \cdot \Big(\sum_{\ell\geq 0} 2^{-p\ell}\Big)^{t_1+t_2} \\
&\leq \tau(p,m,\sigma,D)\cdot n^{2(1-\gamma)} \frac{1}{\delta} \exp\Big(-\frac{2\delta^2}{5\sigma^2}\Big) \leq \tau(p,m,\sigma,D)\cdot n^{2(1-\gamma)} \frac{1}{n^{\frac{4}{5}}\sqrt{\log(n)}}. \qquad (\text{A.81})
\end{aligned}$$

In the second inequality, we used the fact that ρ is upper bounded by $2^{\ell_{\max}+1}\delta$ due to the hypotheses (A.51) and (A.43). In the last inequality we used that $1 \leq t_1 + t_2 \leq 2m$. Hence, from (A.78) and (A.81) we estimate

$$\begin{aligned}
|\mathrm{I}| &\leq \max_{(J_1,J_2)} \big|\mathbb{E}\big[T_k f(J_1,J_2)\big]\big| \\
&\leq 4^m \cdot \tau(p,m,\sigma,D)\cdot n^{2(1-\gamma)} \frac{1}{n^{\frac{4}{5}}\sqrt{\log(n)}} \leq \frac{1}{2} n^{1-2\gamma}\cdot n^{\frac{1}{5}}, \qquad (\text{A.82})
\end{aligned}$$

for n as in (A.43). Combining (A.81) and (A.76) with (A.75) yields the desired result. □

Now that we have bound the deviation between the complete estimator and its expectation, let us bound the deviation between the complete estimator and its incomplete counter part.

Remark 12. *Theorem 8.3.2 holds when the distributions are compactly supported. Suppose we have probability sequences* $(\mathbf{a}^{(n)})$ *and* $(\mathbf{b}^{(n)})$*. Setting* $\delta = n^{-\frac{1}{2p}}$ *in* (A.13) *gives:*

$$\mathbb{P}\left(\big|\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)},\mathbf{b}^{(n)}) - \mathbb{E}\overline{h}^m_{\mathtt{W}}(\mathbf{a}^{(n)},\mathbf{b}^{(n)})\big| \geq 2^{\frac{3}{2}-\frac{p}{2}} M D^2 \frac{m^{\frac{1}{2}}}{n^{2(\gamma-\frac{3}{4})}}\sqrt{\log(n)}\right) \leq n^{-\frac{1}{2p}}$$

Hence we essentially lose a $\log(n)^{\frac{p}{2}}$ *factor in comparison with* (A.44)*.*

We now discuss the difference between the deviation bounds of the estimator $\overline{h}$ and its mean in the bounded and unbounded data cases.

Remark 13. *The proof of Lemma A.1.3 also yields*

$$\mathbb{P}\left(\big|\widetilde{h}^{m,k}_{w,P}(\mathbf{a}^{(n)},\mathbf{b}^{(n)}) - \overline{h}^m_{w,P}(\mathbf{a}^{(n)},\mathbf{b}^{(n)})\big| \geq M\sqrt{\frac{2\log(2/\delta)}{k}}\Big|\boldsymbol{x}_1,\boldsymbol{y}_1,\cdots,\boldsymbol{x}_n,\boldsymbol{y}_n\right) \leq \delta \qquad (\text{A.83})$$

for sub-Gaussian data $\boldsymbol{x}_1,\boldsymbol{y}_1,\cdots,\boldsymbol{x}_n,\boldsymbol{y}_n$ *and any* $\delta > 0$*.*

A.1.4 Distance to marginals

In this appendix, we give the details of the proof of Theorem 8.3.4. In what follows, we denote by $\Pi_{(i)}$ the i-th row of matrix Π. Let us denote by $\mathbf{1} \in \mathbb{R}^n$ the vector whose entries are all equal to 1.

Theorem 8.3.4 (Distance to marginals)**.** *Let $\delta \in (0,1)$, two integers $m \leq n$ and consider two sequences of probability vectors $(\mathbf{a}^{(n)}), (\mathbf{b}^{(n)}) \in \Sigma$. Let a ground cost $C = C^{m,p}$ for some $p \geq 1$ as in 8.2. Consider an OT kernel $h \in \{W_p, W_p^p, W^\varepsilon, S^\varepsilon, \mathcal{G}W\}$. Suppose now that the probability law over m-tuples P and the reweighting function w, as defined in (8.12) and (8.13), satisfy the admissibility condition (8.21). For all integers $k \geqslant 1$ and all integers $1 \leqslant i \leqslant n$, we have:*

$$\mathbb{P}\left(\left|\widetilde{\Pi}^{h,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})_{(i)}\mathbf{1} - a_i^{(n)}\right| \geq \sqrt{\frac{2\log(2/\delta)}{k}}\right) \leq \delta \tag{A.84}$$

Proof. Let us recall that thanks to the admissibility condition (8.21), $\overline{\Pi}^h_{w,P}$ is a transport plan between the input probability vectors $\mathbf{a}^{(n)}$ and $\mathbf{b}^{(n)}$ and hence, it verifies the marginal constraints, i.e $(\overline{\Pi}^h_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}))_i \times \mathbf{1} = a_i$. Thanks to Remark 11 we have

$$\widetilde{\Pi}^{h,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})_{(i)}\mathbf{1} = \frac{1}{k}\sum_{\ell=1}^{k} \omega_\ell$$

where $\omega_\ell = \sum_{I,J \in ([\![n]\!]^m)^2} \sum_{j=1}^n (\Pi_{I,J})_{i,j} \mathfrak{b}_\ell^{P_{\mathbf{a}^{(n)}}, P_{\mathbf{b}^{(n)}}}(I,J)$. Conditioned upon $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ and $\boldsymbol{Y} = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n)$, the random vectors ω_p are independent, and bounded by 1. Moreover, one can observe that $\mathbb{E}[\widetilde{\Pi}^{h,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})_i\mathbf{1}] = \overline{\Pi}^h_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})_i\mathbf{1}$. Using Hoeffding's inequality yields

$$\begin{aligned}\mathbb{P}(|\widetilde{\Pi}^{h,k}_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})_i\mathbf{1} - \overline{\Pi}^h_{w,P}(\mathbf{a}^{(n)}, \mathbf{b}^{(n)})_i\mathbf{1})| > \varepsilon) &= \mathbb{E}[\mathbb{P}(|\frac{1}{k}\sum_{p=1}^{k}\omega_p - \mathbb{E}[\frac{1}{k}\sum_{p=1}^{k}\omega_p])| > \varepsilon|\boldsymbol{X}, \boldsymbol{Y})] \\ &\leqslant 2e^{-2k\varepsilon^2}\end{aligned}$$

which concludes the proof. □

A.1.5 Optimization

This appendix is dedicated to providing the full statements and proofs of Theorem 8.3.5 and Theorem 8.3.6:

Theorem 8.3.5. *Let $\mathbf{a}, \mathbf{b} \in \Sigma_m$. Let $\boldsymbol{X}$ be a $\mathbb{R}^{dm}$-valued random variable, and $\{\boldsymbol{Y}_\theta\}$ a family of $\mathbb{R}^{dm}$-valued random variables defined on the same probability space, indexed by $\theta \in \Theta$, where $\Theta \subset \mathbb{R}^q$ is open. Assume that $\theta \mapsto \boldsymbol{Y}_\theta$ is C^1. Denote $C = C^{m,p}$ for some $p \geq 1$ and let $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon\}$. Then the function $\theta \mapsto -h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$ is Clarke regular and for all $1 \leq i \leq q$ we have:*

$$\begin{aligned}\partial_{\theta_i} h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta)) = \overline{co}\{-tr(P \cdot D^T) \cdot (\nabla_{\theta_i}\boldsymbol{Y}) : &P \in \mathrm{Opt}(h, C(\boldsymbol{X}, \boldsymbol{Y}_\theta), \mathbf{a}, \mathbf{b}), \\ &D \in \mathbb{R}^{m,m}, D_{j,k} \in \partial_Y C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta)\}\end{aligned} \tag{A.85}$$

where ∂_{θ_i} is the Clare subdifferential with respect to θ_i, $\partial_Y C_{j,k}$ is the subdifferential of the cell $C_{j,k}$ of the cost matrix with respect to Y, $\overline{co}$ denotes the closed convex hull and $\mathrm{Opt}(h, C, \mathbf{a}, \mathbf{b})$ is defined in Definition 18.

For $h = \mathcal{GW}$ and $p > 1$, the function $-h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{X}), C(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta))$ is also Clarke regular, and we have:

$$\partial_{\theta_i} h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{X}), C(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta)) = \{- \sum_{j_1, j_2, k_1, k_2 = 1}^{m} D_{j_1,k_1,j_2,k_2} P_{j_1,j_2} P_{k_1,k_2} \cdot (\nabla_{\theta_i} Y) : \tag{A.86}$$

$$P \in \mathrm{Opt}(h, C(\boldsymbol{X}, \boldsymbol{X}), C(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta), \mathbf{a}, \mathbf{b}) \tag{A.87}$$

$$D \in (\mathbb{R}^m)^4,\ D_{j_1,k_1,j_2,k_2} = \nabla_Y C_{j_1,k_1,j_2,k_2}(\boldsymbol{X}, \boldsymbol{Y}_\theta)\}$$

where $C_{j_1,k_1,j_2,k_2} = \|C_{j_1,k_1}(\boldsymbol{X}, \boldsymbol{X}) - C_{j_2,k_2}(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta)\|^p$.

Proof. We start with the case $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon\}$. The function $Y \mapsto C_{j,k}(X, Y)$ is equal to $\|\boldsymbol{x}_j - \boldsymbol{y}_k\|_2^p$. It is therefore convex, and thus Clarke regular by Proposition 2.3.6(b) [Clarke 1990]. Since $\theta \mapsto \boldsymbol{Y}_\theta$ is C^1, from Theorem 2.3.10 [Clarke 1990] it follows that $\theta \mapsto C_{j,k}^{m,p}(\boldsymbol{X}, \boldsymbol{Y}_\theta)$ is Clarke regular, and:

$$\partial_{\theta_i} C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta) = \{D_{j,k} \cdot \nabla_{\theta_i} Y_\theta : D_{j,k} \in \partial_Y C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta)\}.$$

Note, that the set of admissible transport plans for any marginals $\mathbf{a}, \mathbf{b} \in \Sigma^m$ is compact. Furthermore, the transport cost for a given plan P is a linear function of cost matrix C. Therefore, from Danskin's Theorem (Proposition B.25 [Bertsekas 1997]) it follows that for $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon\}$ the function $C \mapsto -h(\mathbf{a}, \mathbf{b}, C)$ is convex, and it's subderivative is equal to $\mathrm{Opt}(h, C, \mathbf{a}, \mathbf{b})$. Therefore from Theorem 2.3.9(i) and Proposition 2.3.1 (for $s = 1$) in [Clarke 1990] it follows that $\theta \mapsto -h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$ is Clarke regular for $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon\}$, and that (A.85) holds.

Assume now that $h = \mathcal{GW}$ and that $p > 1$. The proof is analogous. In this case, the function $Y \mapsto C_{j_1,k_1,j_2,k_2}(\boldsymbol{X}, \boldsymbol{Y})$ is differentiable. Therefore the function $\theta \mapsto C_{j_1,k_1,j_2,k_2}(\boldsymbol{X}, \boldsymbol{Y}_\theta)$ is differentiable. Again the set of admissible transport plans is compact and for a given transport plan, the transport cost is a linear function of the four dimensional tensor C_{j_1,k_1,j_2,k_2}. Therefore, using Danskin's Theorem (Proposition B.25 [Bertsekas 1997]), as well as Theorem 2.3.10 and Proposition 2.3.1 in [Clarke 1990] we get that $-h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{X}), C(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta))$ is Clarke regular and the formula (A.86) holds. □

Theorem 8.3.6. *Let $\mathbf{a}, \mathbf{b}, \boldsymbol{X}, \boldsymbol{Y}, C$ be as in theorem 8.3.5, $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon\}$, and assume in addition that the random variables $\boldsymbol{X}, \{Y_\theta\}_{\theta \in \Theta}$ have finite p-moments. If for all $\theta \in \Theta$ there exists an open neighbourhood U, $\theta \in U \subset \Theta$, and a random variable $K_U : \Omega \to \mathbb{R}$ with finite expected value, such that*

$$\|C(\boldsymbol{X}(\omega), \boldsymbol{Y}_{\theta_1}(\omega)) - C(\boldsymbol{X}(\omega), \boldsymbol{Y}_{\theta_2}(\omega))\| \leq K_U(\omega)\|\theta_1 - \theta_2\| \tag{A.88}$$

then we have

$$\partial_\theta \mathbb{E}\left[h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right] = \mathbb{E}\left[\partial_\theta h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right]. \tag{A.89}$$

with both expectation being finite. Furthermore the function $\theta \mapsto -\mathbb{E}\left[h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right]$ is also Clarke regular.

For $h = \mathcal{GW}$, assume that $p > 1$ and that random variables $\boldsymbol{X}, \{\boldsymbol{Y}_\theta\}$ have finite $2p$-moments. Assume also that for each $\theta \in \Theta$ there exists an open neighbourhood U, $\theta \in U \subset \Theta$, and a random variable $K_U : \Omega \to \mathbb{R}$ with finite expected value, such that

$$\|\tilde{C}(\boldsymbol{X}(\omega), \boldsymbol{Y}_{\theta_1}(\omega)) - \tilde{C}(\boldsymbol{X}(\omega), \boldsymbol{Y}_{\theta_2}(\omega))\| \leq K_U(\omega)\|\theta_1 - \theta_2\| \tag{A.90}$$

where $\tilde{C}(\boldsymbol{X}, \boldsymbol{Y}) = \|C(\boldsymbol{X}, \boldsymbol{X}) - C(\boldsymbol{Y}, \boldsymbol{Y})\|^p$. Then we have

$$\partial_\theta \mathbb{E}\left[h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{X}), C(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta))\right] = \mathbb{E}\left[\partial_\theta h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{X}), C(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta))\right]. \tag{A.91}$$

with both expectation being finite. Furthermore the function $\theta \mapsto -\mathbb{E}\left[h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{X}), C(\boldsymbol{Y}_\theta, \boldsymbol{Y}_\theta))\right]$ is also Clarke regular.

Proof. We start with the case $h \in \{\mathfrak{L}, \mathfrak{L}^\varepsilon\}$. Suppose that $U \subset \Theta$ is open and K_U is a function for which (A.88) is satisfied. Then the same bound is also satisfied for the function $h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}(\omega), \boldsymbol{Y}_\theta(\omega)))$, since the function $C \mapsto h(\mathbf{a}, \mathbf{b}, C)$ is 1-Lipshitz. Hence, given the regularity of $-h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$, the interchange (A.89) and regularity of $\theta \mapsto -\mathbb{E}[h(\mathbf{a}, \mathbf{b}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))]$ will follow from Theorem 2.7.2 and Remark 2.3.5 [Clarke 1990], once we establish that the expectation on the left hand side is finite. This follows trivially from the standard bound:

$$\|\boldsymbol{x} - \boldsymbol{y}\|^p \leq 2^{p-1}(\|\boldsymbol{x}\|^p + \|\boldsymbol{y}\|^p) \tag{A.92}$$

and the assumption that $\boldsymbol{X}, \boldsymbol{Y}_\theta$ have finite p-moments. The same argument applies to the case when $h = \mathcal{GW}$. The $\mathcal{GW}$ cost depends on the four-dimensional tensor C_{j_1,k_1,j_2,k_2} defined in the proof of Theorem 8.3.5 in a Lipshitz manner, since it's supdifferential is bounded. Again, the thesis for $h = GW$ will follow from Theorem 2.7.2 and Remark 2.3.5 [Clarke 1990], once we establish that the expectation on the left hand side of (A.91) is finite. This follows from applying the bound (A.92) twice and the assumption of finite $2p$-moments. □

A.1.6 Minibatch OT closed-form solution for 1D data

We now give the closed-form combinatorial calculus for the 1D data. We start by sorting all the data and give to each of them an index which represents their position after the sorting phase. Then we select and sort all the minibatches. x_j can not be at a position superior to its index j inside a batch. For a fixed x_j, a simple combinatorial arguments tells you that there are $C^i_{x_j}$ sets where x_j is at the i-th position:

$$C^{m,n}_{i,x_j} = \binom{j-1}{i-1}\binom{n-j}{m-i} \tag{A.93}$$

Suppose that x_j is transported to a y_k points in the target mini batch. Then, they both share the same positions i in their respective minibatch. As there are several i where x_j is transported to y_k, we sum over all those possible positions. Hence our current transport matrix coefficient $\Pi_{j,k}$ can be calculated as :

$$\Pi_{j,k} = \sum_{i=i_{\min}}^{i_{\max}} C^{m,n}_{i,x_j} C^{m,n}_{i,y_k} \tag{A.94}$$

Where $i_{\min} = \max(0, m - n + j, m - n + k)$ and $i_{\max} = \min(j, k)$. $i_{\min}$ and $i_{\max}$ represent the sorting constraints. Furthermore, as we have uniform weight histograms, we will transport a mass of $\frac{1}{m}$ and averaged it by the total number of transport. So finally, our transport matrix coefficient $\Pi_{j,k}$ are:

$$\Pi_{j,k} = \frac{1}{m\binom{n}{m}^2} \sum_{i=i_{\min}}^{i_{\max}} C^{m,n}_{i,x_j} C^{m,n}_{i,y_k} \tag{A.95}$$

The sampling with replacement case is much more complex and highly computationally costly. Following the same strategy as above, we sort all minibatches. In this case, x_i might appears several times more or less than y_j and we need to take that into account. We denote m_i the number of repetitions of the i-th element and the summation $|\text{vec}m| = \sum_{k=1}^m m_k$. We denote $N^{i,j}_{m_1,m_2}$ the number of times that x_i and y_j share the same position in their respective minibatches m_1 and m_2, after sorting. For the coefficient $\pi_{i,j}$, we have :

$$\pi_{i,j} = \frac{1}{m}\left(\frac{1}{mn^{m-1}}\right)^2 \sum_{|\text{vec}m_s|=m} \sum_{|\text{vec}m_t|=m} N^{i,j}_{m_s,m_t} \tag{A.96}$$

A.2 Proofs of Chapter 9

While many proofs are similar to the proofs of Chapter 8, we recall them for readers interested only for the uniform measure case. We start with the basic properties of minibatch OT proof in Section A.2.1. Then we prove several UOT properties in Section A.2.2. The UOT properties are at the heart of the main statistical and optimization proofs presented in Section A.2.3.

A.2.1 Basic properties

Proposition 16 (Positivity, symmetry and bias)**.** *The minibatch UOT are positive and symmetric losses. However, they are not definites,* i.e., $\overline{h}^m(\boldsymbol{X}, \boldsymbol{X}) > 0$ *for non trivial* $\boldsymbol{X}$*,* $\tau > 0$ *and* $1 < m < n$*.*

Proof. The first two properties are inherited from the classical UOT cost. Consider a uniform probability vector and random 3-data tuple $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ with distinct vectors $(\boldsymbol{x}_i)_{1 \leq i \leq 3}$. As $\overline{h}^m$ is an average of positive terms, it is equal to 0 if and only if each of its term is 0. But consider the minibatch term $I_1 = (i_1, i_2)$ and $I_2 = (i_1, i_3)$, then obviously $h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}(I_1), \boldsymbol{X}(I_2))) \neq 0$ as $\boldsymbol{x}_2 \neq \boldsymbol{x}_3$, where $\boldsymbol{X}(I_1)$ denotes the data minibatch corresponding to indices in I_1. □

We now give the proof for our claim "A simple combinatorial argument assures that the sum of $\boldsymbol{u}_m$ over all m-tuples I gives $\boldsymbol{u}_n$."

Proposition 17 (Averaged distributions)**.** *Let* $\boldsymbol{u}_m$ *be a uniform vector of size* m*. The average over* m*-tuples* $I \in \mathcal{P}^m$ *for a given index of* $\boldsymbol{u}_m$ *is equal to* $\frac{m_{\mathbf{a}}}{n}$*, i.e.,* $\forall i \in [\![1, n]\!], \sum_{I \in \mathcal{P}^m} (\boldsymbol{u}_m)_i = (\boldsymbol{u}_n)_i = \frac{m_{\mathbf{a}}}{n}$*.*

Proof. We recall that $\mathcal{P}^m$ denotes the set of all m-tuples without repeated elements. Let us check we recover the initial weights $(\boldsymbol{u}_n)_i = \frac{m_{\mathbf{a}}}{n}$. Observe that $\sum_{i=1}^n a_i = m_{\mathbf{a}}$ and that for each $1 \leq i \leq n$

$$\begin{aligned}
\sharp\{I \in \mathcal{P}^m : i \in I\} &= \sharp\{I \in \mathcal{P}^m : n \in I\} \\
&= \sharp\{I = (i_1, \ldots, i_m) \in \mathcal{P}^m : i_1 = n\} + \ldots + \sharp\{I = (i_1, \ldots, i_m) \in \mathcal{P}^m : i_m = n\} \\
&= m \cdot \sharp\{I = (i_1, \ldots, i_m) \in \mathcal{P}^m : i_m = n\}. \qquad \text{(A.97)}
\end{aligned}$$

Since $\sharp\{I = (i_1, \ldots, i_m) \in \mathcal{P}^m : i_m = n\}$ is the number of $(m-1)$-tuples without repeated indices of $[\![1, n-1]\!]$, $(n-1)!/(n-m)!$, it follows that

$$\frac{(n-m)!}{n!} \cdot \sum_{I \in \mathcal{P}^m} \frac{m_{\mathbf{a}}}{m} 1_I(i) = \frac{(n-m)!}{n!} \sum_{I \in \mathcal{P}^m, i \in I} \frac{m_{\mathbf{a}}}{m} = \frac{(n-m)!}{n!} \frac{m_{\mathbf{a}}}{m} \cdot \sharp\{I \in \mathcal{P}^m : i \in I\} \qquad \text{(A.98)}$$

$$= \frac{(n-m)!}{n!} \frac{m_{\mathbf{a}}}{m} m \cdot \frac{(n-1)!}{(n-m)!} = \frac{m_{\mathbf{a}}}{n} \qquad \text{(A.99)}$$

□

A.2.2 Unbalanced Optimal Transport properties

In this appendix we prove the UOT properties necessary to prove the minibatch statistical and optimization theorems. We start with the robustness property and we follow with upper bound, bounded transport plan and Lipschitz properties.

We recall the definition of Csiszàr divergences. Consider a convex, positive, lower-semicontinuous function such that $\phi(1) = 0$. Define its recession constant as $\phi'_\infty = \lim_{\boldsymbol{x} \to +\infty} \phi(\boldsymbol{x})/\boldsymbol{x}$. The Csiszàr

divergence between positively weighted vectors $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}_+^d$ reads

$$D_\phi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{y}_i \neq 0} \boldsymbol{y}_i \phi\Big(\frac{\boldsymbol{x}_i}{\boldsymbol{y}_i}\Big) + \phi'_\infty \sum_{\boldsymbol{y}_i = 0} \boldsymbol{x}_i.$$

It allows to generalize OT programs. We retrieve common penalties such as Total Variation and Kullback-Leibler divergence by respectively taking $\phi(\boldsymbol{x}) = |\boldsymbol{x} - 1|$ and $\phi(\boldsymbol{x}) = (\boldsymbol{x} \log \boldsymbol{x} - \boldsymbol{x} + 1)$. We provide a generalized definition of all OT programs as

$$\mathrm{OT}_\phi^{\tau,\varepsilon}(\mathbf{a}, \mathbf{b}, C) = \min_{\Pi \in \mathbb{R}_+^{n \times n}} \mathcal{F}(\Pi, C) = \min_{\Pi \in \mathbb{R}_+^{n \times n}} \langle C, \Pi \rangle + \tau D_\phi(\Pi \mathbf{1}_n | \mathbf{a}) + \tau D_\phi(\Pi^T \mathbf{1}_n | \mathbf{b}) + \varepsilon \mathtt{KL}(\Pi | \mathbf{a} \otimes \mathbf{b}).$$

Where $\mathcal{F}$ denotes the UOT energy.

Robustness

We start by showing the robustness properties Lemma 9.1.1 that we split in two different lemmas. Lemma 9.1.1.a shows that the UOT cost is robust to an outlier while Lemma 9.1.1.b shows that OT is not robust to an outlier.

Lemma 9.1.1.a. *Take (μ, ν) two probability measures with compact support, and z outside of ν's support. Recall the Gaussian-Hellinger distance [Liero 2017] between two positive measures as*

$$\mathrm{GH}_\tau(\mu, \nu) = \inf_{\pi \geq 0} \int C(x, y) d\pi(x, y) + \tau KL(\pi_1 | \mu) + \tau KL(\pi_2 | \nu).$$

For $\zeta \in [0, 1]$, write $\tilde{\mu} = \zeta\mu + (1-\zeta)\delta_z$ a measure perturbed by a Dirac outlier. Write $m(z) = \int C(z, y) d\nu(y)$. One has

$$\mathrm{GH}_\tau(\tilde{\mu}, \nu) \leq \zeta\, \mathrm{GH}_\tau(\mu, \nu) + 2\tau(1 - \zeta)(1 - e^{-m(z)/2\tau}). \tag{A.100}$$

In particular, with the notation $\mathrm{OT}_{KL}^{\tau,0}$ it reads

$$\mathrm{OT}_{KL}^{\tau,0}(\tilde{\mu}, \nu, C) \leq \zeta\, \mathrm{OT}_{KL}^{\tau,0}(\mu, \nu, C) + 2\tau(1 - \zeta)(1 - e^{-m(\boldsymbol{z})/2\tau}).$$

Proof. Write π the optimal plan for $\mathrm{OT}_\phi^{\tau,0}(\mu, \nu)$. We consider a suboptimal plan for $\mathrm{OT}_\phi^{\tau,0}(\tilde{\mu}, \nu)$ of the form

$$\tilde{\pi} = \zeta\pi + (1 - \zeta)\kappa\delta_z \otimes \nu,$$

where κ is mass parameter which will be optimized after. Note that the marginals of the plan $\tilde{\pi}$ are $\tilde{\pi}_1 = \zeta\pi_1 + (1 - \zeta)\kappa\delta_z$ and $\tilde{\pi}_2 = \zeta\pi_2 + (1 - \zeta)\kappa\nu$. Note that $\mathtt{KL}$ is jointly convex, thus one has

$$\begin{aligned}
\mathtt{KL}(\tilde{\pi}_1 | \tilde{\mu}) &\leq \zeta \mathtt{KL}(\pi_1 | \mu) + (1 - \zeta)\mathtt{KL}(\kappa\delta_z | \delta_z), \\
\mathtt{KL}(\tilde{\pi}_2 | \tilde{\mu}) &\leq \zeta \mathtt{KL}(\pi_2 | \nu) + (1 - \zeta)\mathtt{KL}(\kappa\nu | \nu).
\end{aligned}$$

Thus a convex inequality yields

$$\begin{aligned}
\mathrm{OT}_\phi^{\tau,0}(\tilde{\mu}, \nu) \leq \zeta \Big[\int ||x - y|| d\pi(x, y) + \tau \mathtt{KL}(\pi_1 | \mu) + \tau \mathtt{KL}(\pi_2 | \nu) \Big] \\
+ (1 - \zeta) \Big[\kappa m(z) + \tau \mathtt{KL}(\kappa\delta_z | \delta_z) + \tau \mathtt{KL}(\kappa\nu | \nu) \Big].
\end{aligned}$$

We optimize now the upper bound w.r.t. κ. Both KL terms are equal to $\phi(\kappa) = \kappa \log \kappa - \kappa + 1$, thus differentiating w.r.t. κ yields

$$m(z) + 2\tau \log \kappa = 0 \Rightarrow \kappa = e^{-m(z)/2\tau}.$$

Reusing this expression of κ in the upper bound yields Equation (A.100). □

Lemma 9.1.1.b. *Take (μ, ν) two probability measures with compact support, and z outside of ν's support. Define the Wasserstein distance between two probabilities as*

$$\mathrm{W}(\mu, \nu) = \sup_{f(\boldsymbol{x})+g(\boldsymbol{y}) \leq C(\boldsymbol{x},\boldsymbol{y})} \int f(\boldsymbol{x}) d\mu(\boldsymbol{x}) + \int g(\boldsymbol{y}) d\nu(\boldsymbol{y}). \tag{A.101}$$

For $\zeta \in [0, 1]$, write $\tilde{\mu} = \zeta\mu + (1-\zeta)\delta_z$ a measure perturbed by a Dirac outlier. Write (f, g) the optimal dual potentials of $\mathrm{W}(\mu, \nu)$, and y^* a point in ν's support. One has

$$\mathrm{W}(\tilde{\mu}, \nu) \geq \zeta\, \mathrm{W}(\mu, \nu) + (1-\zeta)\Big(||\boldsymbol{z} - \boldsymbol{y}^*||^2 - g(\boldsymbol{y}^*) + \int g d\nu\Big). \tag{A.102}$$

In particular, with the notation $\mathfrak{L}$ it reads

$$\mathfrak{L}(\tilde{\mu}, \nu) \geq \zeta\mathfrak{L}(\mu, \nu) + (1-\zeta)\Big(C(\boldsymbol{z}, \boldsymbol{y}^*) - g(\boldsymbol{y}^*) + \int g d\beta\Big).$$

Proof. We consider a suboptimal pair $(\tilde{f}, \tilde{g})$ of potentials for $\mathfrak{L}(\tilde{\mu}, \nu)$. On the support of (μ, ν) we take the optimal potentials pair (f, g) for (μ, ν), i.e. $\tilde{f} = f$ and $\tilde{g} = g$. We need to extend $\tilde{f}$ at z. To do so we take the c-transform of g, i.e.

$$\tilde{f}(z) = \inf_{y \in spt(\nu)} ||\boldsymbol{z} - \boldsymbol{y}||^2 - g(\boldsymbol{y}) = ||\boldsymbol{z} - \boldsymbol{y}^*||^2 - g(\boldsymbol{y}^*),$$

where the infimum is attained at some y^* since ν has compact support. the pair $(\tilde{f}, \tilde{g})$ is suboptimal, thus

$$\begin{aligned}
\mathfrak{L}(\tilde{\mu}, \nu) &\geq \int \tilde{f}(\boldsymbol{x}) d\tilde{\mu}(\boldsymbol{x}) + \int \tilde{g}(\boldsymbol{y}) d\nu(\boldsymbol{y}), \\
&\geq \zeta \int f(\boldsymbol{x}) d\mu(\boldsymbol{x}) + (1-\zeta)\tilde{f}(\boldsymbol{z}) + \int \tilde{g}(\boldsymbol{y}), \\
&\geq \zeta\mathfrak{L}(\mu, \nu) + (1-\zeta)\Big[||\boldsymbol{z} - \boldsymbol{y}^*||^2 - g(\boldsymbol{y}^*) + \int g(\boldsymbol{y}) d\nu(\boldsymbol{y})\Big].
\end{aligned}$$

Hence the resulted given by Equation (A.102). □

UOT properties

Now let us present results which will be useful for concentration bounds. A key element is to have a bounded plan and a finite UOT cost in order to derive a Hoeffding-type bound. We start this appendix by proving Lemma 9.2.1. We split it in two, Lemma 9.2.1.a proves that the UOT cost is finite and provides an upper bound while Lemma 9.2.1.b proves that the UOT plans exist and belong to a compact set.

Lemma 9.2.1.a (Upper bounds)**.** *Let $(\mathbf{a}, \mathbf{b})$ be two positive vectors and assume that $\langle \mathbf{a}\mathbf{b}^\top, C\rangle < +\infty$, then the UOT cost is finite. Furthermore, we have the following bound for $h = \mathrm{OT}_\phi^{\tau,\varepsilon}$, one has $|h(\mathbf{a}, \mathbf{b}, C)| \leqslant M^h_{\mathbf{a},\mathbf{b}}$, where*

$$M^h_{\mathbf{a},\mathbf{b}} = M m_{\mathbf{a}} m_{\mathbf{b}} + \tau m_{\mathbf{a}} \phi(m_{\mathbf{b}}) + \tau m_{\mathbf{b}} \phi(m_{\mathbf{a}}). \tag{A.103}$$

Regarding $h = S_\phi^{\tau,\varepsilon}$, one has $|h(\mathbf{a}, \mathbf{b}, C)| \leqslant M^S_{\mathbf{a},\mathbf{b}}$, where

$$M^S_{\mathbf{a},\mathbf{b}} = 2M m_{\mathbf{a}} m_{\mathbf{b}} + \tau m_{\mathbf{a}} \phi(m_{\mathbf{b}}) + \tau m_{\mathbf{b}} \phi(m_{\mathbf{a}}) + \tau m_{\mathbf{a}} \phi(m_{\mathbf{a}}) + \tau m_{\mathbf{b}} \phi(m_{\mathbf{b}}) + \frac{\varepsilon}{2}(m_{\mathbf{a}} - m_{\mathbf{b}})^2. \tag{A.104}$$

Proof. As $\langle \mathbf{a}\mathbf{b}^\top, C\rangle < +\infty$ is finite, one can bound the ground cost C as $0 \leqslant C_{i,j} \leqslant M$. Consider the OT kernel $h \in \{\mathrm{OT}_\phi^{\tau,\varepsilon}\}$ for any $\varepsilon \geq 0$. Let us consider the transport plan $\Pi = \mathbf{a}\mathbf{b}^\top = (a_i b_j)$ (with respect to the cost matrix C). Because all terms are positive, we have:

$$\begin{aligned}
|h| &\leqslant \langle \mathbf{a}\mathbf{b}^\top, C\rangle + \varepsilon \mathtt{KL}(\mathbf{a}\mathbf{b}^\top|\mathbf{a}\mathbf{b}^\top) + \tau D_\phi\left((\mathbf{a}\mathbf{b}^\top)\mathbf{1}_n|\mathbf{a}\right) + \tau D_\phi\left((\mathbf{b}\mathbf{a}^\top)\mathbf{1}_n|\mathbf{b}\right), \\
&\leqslant M\sum_{i,j} a_i b_j + \tau m_{\mathbf{a}}\phi(m_{\mathbf{b}}) + \tau m_{\mathbf{b}}\phi(m_{\mathbf{a}}), \\
&\leqslant M m_{\mathbf{a}} m_{\mathbf{b}} + \tau m_{\mathbf{a}}\phi(m_{\mathbf{b}}) + \tau m_{\mathbf{b}}\phi(m_{\mathbf{a}}). \qquad \text{(A.105)}
\end{aligned}$$

Defining $M^h_{\mathbf{a},\mathbf{b}}$ as the last upper bound finishes the proof. The case $h = S_\phi^{\tau,\varepsilon}$, is the sum of three terms of the form $\mathrm{OT}_\phi^{\tau,\varepsilon}$ Thus the sum $M^h_{\mathbf{a},\mathbf{b}} + \frac{1}{2}M^h_{\mathbf{a},\mathbf{a}} + \frac{1}{2}M^h_{\mathbf{b},\mathbf{b}}$ is an upper bound of S_ε. □

We now bound the UOT plan.

Lemma 9.2.1.b (locally compact optimal transport plan)**.** *Assume that $\langle \mathbf{a}\mathbf{b}^\top, C\rangle < +\infty$. Consider regularized or unregularized UOT with entropy ϕ and penalty D_ϕ such that one has $\phi'_\infty > 0$. Then there exists an open neighbhourhood U around C, and a compact set K, such that the set of optimal transport plan for any $\tilde{C} \in U$ is in K,* i.e., $\mathrm{Opt}(h, \tilde{C}, \mathbf{a}, \mathbf{b}) \subset K$. *Furthermore, if all costs are uniformly bounded such that $0 \leq C \leq M < \infty$, then the compact K can be taken global, i.e. independent of C.*

Proof. We identify the mass of a positive measure with its L1 norm, i.e. $m_{\mathbf{a}} = \sum \mathbf{a}_i = ||\mathbf{a}||_1$. We first consider the case where $0 \leq C \leq M < \infty$. The OT cost is finite because the plan $\pi = \mathbf{a}\mathbf{b}^\top$ is suboptimal and yields $\mathrm{OT}_\phi^{\tau,\varepsilon}(\mathbf{a}, \mathbf{b}, C) \leq M m_{\mathbf{a}} m_{\mathbf{b}} + \tau m_{\mathbf{a}}\phi(m_{\mathbf{b}}) + \tau m_{\mathbf{b}}\phi(m_{\mathbf{a}}) < +\infty$.

Take a sequence Π_t approaching the infimum. Note that thanks to the Jensen inequality, one has $D_\phi(\boldsymbol{x}, \boldsymbol{y}) \geq m_{\boldsymbol{y}}\phi(m_{\boldsymbol{x}}/m_{\boldsymbol{y}})$ (see [Liero 2017]). Write $m_\Pi = \sum \Pi_{t,ij}$. One has for any t

$$\begin{aligned}
&\langle \Pi_t, C\rangle + \tau D_\phi(\Pi_{t,1}\mathbf{1}_n|\mathbf{a}) + \tau D_\phi(\Pi_{t,2}^\top \mathbf{1}_n|\mathbf{b}) + \varepsilon \mathrm{KL}(\Pi_t|\mathbf{a}\mathbf{b}^\top), \\
&\quad \geq m_\Pi\left[\min C_{ij} + \tau\frac{m_{\mathbf{a}}}{m_\Pi}\phi\big(\frac{m_\Pi}{m_{\mathbf{a}}}\big) + \tau\frac{m_{\mathbf{b}}}{m_\Pi}\phi\big(\frac{m_\Pi}{m_{\mathbf{b}}}\big) + \varepsilon\frac{m_{\mathbf{a}}m_{\mathbf{b}}}{m_\Pi}\phi_{KL}\big(\frac{m_\Pi}{m_{\mathbf{a}}m_{\mathbf{b}}}\big)\right], \\
&\quad \geq m_\Pi L(m_\Pi).
\end{aligned}$$

If $||\Pi_t||_1 = m_\Pi \to +\infty$, then $L(m_\Pi) \to +\infty$ if $\varepsilon > 0$ and $L(m_\Pi) \to \min C_{ij} + 2\phi'_\infty > 0$ otherwise. In both cases, as $t \to \infty$ and $||\Pi_t||_1 = m_\Pi \to +\infty$, we are supposed to approach the infimum but its lower bound goes to $+\infty$, which contradicts the fact that the optimal OT cost is finite.

More precisely, there exists a large enough value $\tilde{M}$ such that for $m_\Pi > \tilde{M}$, the lower bound is superior to the upper bound $M m_{\mathbf{a}} m_{\mathbf{b}} + \tau m_{\mathbf{a}}\phi(m_{\mathbf{b}}) + \tau m_{\mathbf{b}}\phi(m_{\mathbf{a}})$ and thus necessarily not optimal. Furthermore, $\tilde{M}$ depends on $(m_{\mathbf{a}}, m_{\mathbf{b}}, M)$ since $0 \leq C \leq M$. Thus, there exists $\tilde{M} > 0$ and some t_0 such that for $t \geq t_0$ any plan approaching the optimum statisfies $||\Pi_t||_1 \leq \tilde{M}$. The sequence $(\Pi_t)_t$ is in a finite dimensional, bounded, and closed set, i.e. a compact set. One can extract a converging subsequence whose limit is a plan attaining the minimum. Thus any optimal plan is necessarily in a compact set.

To generalize to local compactness, we consider $\delta > 0$ and a neighbourhood U of C such that for any $\tilde{C} \in U$ one has $0 \leq \tilde{C} \leq \max C + \delta$. Reusing the above proof yields the existence of $\tilde{M}$ such that for any $\tilde{C} \in U$, any plan approaching the optimum satisfies $||\Pi_t||_1 \leq \tilde{M}$, but this time $\tilde{M}$ depends on $(m_{\mathbf{a}}, m_{\mathbf{b}}, \max C + \delta)$, which is independent of C in its neighbourhood. □

As each element Π of $\mathrm{Opt}(h, C, \mathbf{a}, \mathbf{b})$ is bounded by a constant M, $\mathrm{Opt}(h, C, \mathbf{a}, \mathbf{b})$ is a compact space of $\mathcal{M}_+(\mathcal{X})$. We denote the maximal constant M which bounds all elements of $\mathrm{Opt}(h, C, \mathbf{a}, \mathbf{b})$ as $\mathfrak{M}_\Pi$. We

now prove that the set of optimal transport plan is convex , which will be useful for the optimization section.

Lemma A.2.1 (optimal transport plan convexity)**.** *Consider regularized or unregularized UOT with entropy ϕ and penalty D_ϕ. The set of all optimal transport plans* $\mathrm{Opt}(h, C, \mathbf{a}, \mathbf{b})$ *is a convex set.*

Proof. It is a general property of convex analysis. Take a convex function f and two points $(\boldsymbol{x}, \boldsymbol{y})$ that both attain the minimum over a convex set E. Write $\boldsymbol{z} = t\boldsymbol{x} + (1-t)\boldsymbol{y}$ for $t \in [0,1]$. By convexity and suboptimality of $\boldsymbol{z}$ one has $\min_E f \leq f(\boldsymbol{z}) \leq tf(\boldsymbol{x}) + (1-t)f(\boldsymbol{y}) = \min_E f$. Thus z is also optimal, hence the set of minimizers is convex. The losses $\mathrm{OT}_\phi^{\tau,\varepsilon}$ fall under this setting. □

Finally, we provide a final result about UOT cost which is also useful for the optimization properties.

Lemma A.2.2 (UOT is Lipschitz in the cost C)**.** *The map $C \mapsto h(\boldsymbol{u}, \boldsymbol{u}, C)$ is locally Lipshitz. Furthermore, if the costs are uniformly bounded ($0 \leq C \leq M$) then the loss is globally Lipschitz.*

Proof. We recall that $h(\boldsymbol{u}, \boldsymbol{u}, C) = \min_{\Pi \in \mathbb{R}_+^{n \times n}} \mathcal{F}(\Pi, C)$. Let C_1 and C_2 be two ground costs. Let Π_1 and Π_2 be the optimal solutions of $h(\boldsymbol{u}, \boldsymbol{u}, C_1)$ and $h(\boldsymbol{u}, \boldsymbol{u}, C_2)$, *i.e.,* $h(\boldsymbol{u}, \boldsymbol{u}, C_1) = \mathcal{F}(\Pi_1, C_1)$. Then we have:

$$\mathcal{F}(\Pi_1, C_1) - \mathcal{F}(\Pi_1, C_2) \leq h(\boldsymbol{u}, \boldsymbol{u}, C_1) - h(\boldsymbol{u}, \boldsymbol{u}, C_2) \leq \mathcal{F}(\Pi_2, C_1) - \mathcal{F}(\Pi_2, C_2). \tag{A.106}$$

Thus we have

$$\begin{aligned} h(\boldsymbol{u}, \boldsymbol{u}, C_1) - h(\boldsymbol{u}, \boldsymbol{u}, C_2) &\leq \mathcal{F}(\Pi_2, C_1) - \mathcal{F}(\Pi_2, C_2), \\ &= \langle \Pi_2, C_1 - C_2 \rangle, \qquad (A.107) \\ &\leq \|\Pi_2\| \|C_1 - C_2\|. \qquad (A.108) \end{aligned}$$

Where the last inequality uses the Cauchy-Schwarz inequality. Following the same logic we get a bound for minus the left hand term

$$h(\boldsymbol{u}, \boldsymbol{u}, C_2) - h(\boldsymbol{u}, \boldsymbol{u}, C_1) \leq \mathcal{F}(\Pi_1, C_2) - \mathcal{F}(\Pi_1, C_1) \leq \|\Pi_1\| \|C_1 - C_2\|. \tag{A.109}$$

It remains to bound $\|\Pi_i\|$. When we study the local Lipschitz property, without loss of generality, we fix C_1 and take C_2 in a local neighbourhood of C_1. Thus Lemma 9.2.1.b, gives that $\|\Pi_i\| \leq \tilde{M}$, where $\tilde{M}$ only depends on $(\phi, \tau, \varepsilon, \mathbf{a}, \mathbf{b}, \max C)$, with $\mathbf{a} = \mathbf{b} = \boldsymbol{u}$, i.e. it is locally independant of C in its neighbourhood, hence the local Lipschitz property. When $0 \leq C \leq M$, then $\tilde{M}$ is independent of the cost, hence the bound is global and the map is globally Lipschitz. □

A.2.3 Statistical and optimization proofs

We consider a positive, symmetric, definite and $\mathbf{C}^1$ ground cost and without loss of generality, we consider our ground cost to be the squared Euclidean distance. We recall our definitions and hypotheses. As the distributions α and β are compactly supported, there exists a constant $M > 0$ such that for any $1 \leqslant i, j \leqslant n$, $c(\boldsymbol{x}_i, \boldsymbol{y}_j) \leqslant M$ with $M := \mathrm{diam}(\mathrm{Supp}(\alpha) \cup \mathrm{Supp}(\beta))^2$. We also furthermore suppose that the input masses $m_{\mathbf{a}}$ and $m_{\mathbf{b}}$ of positive vectors are strictly positive and finite, i.e., $0 < m_{\mathbf{a}} < \infty$. These hypotheses assure us that the UOT cost is finite and that the UOT plan is bounded.

Proof of Theorem 9.2.1

We now give the details of the proof of Theorem 9.2.1. We separate Theorem 9.2.1 in two sub theorems: Theorem 9.2.1.a and Theorem 9.2.1.b. In the Theorem 9.2.1.a, we show the deviation bound between $\widetilde{h}^{m,k}$ and E_h and in Theorem 9.2.1.b, we show the deviation bound between $\widetilde{\Pi}^{h,m,k}$ and $\overline{\Pi}^{h,m}$. For Theorem 9.2.1.a, we rely on two lemmas. The first lemma bounds the deviation between the complete estimator $\overline{h}^m$ and its expectation E_h. We denote the floor function as $\lfloor x \rfloor$ which returns the biggest integer smaller than x.

Lemma A.2.3 (U-statistics concentration bound)**.** *Let $\delta \in (0,1)$, three integers $k \geq 1$ and $m \leq n$ be fixed, and two compactly supported distributions α, β. Consider two n-tuples $\boldsymbol{X} \sim \alpha^{\otimes n}$ and $\boldsymbol{Y} \sim \beta^{\otimes n}$ and a kernel $h \in \{\mathrm{OT}_\phi^{\tau,\varepsilon}, S_\phi^{\tau,\varepsilon}\}$ with the squared Euclidean distance as ground cost. We have a concentration bound between $\overline{h}^m(\boldsymbol{X}, \boldsymbol{Y})$ and the expectation over minibatches E_h depending on the number of empirical data n:*

$$|\overline{h}^m(\boldsymbol{X}, \boldsymbol{Y}) - E_h| \leq M^h_{\boldsymbol{u},\boldsymbol{u}} \sqrt{\frac{\log(2/\delta)}{2\lfloor n/m \rfloor}}, \tag{A.110}$$

with probability at least $1-\delta$ and where $M^h_{\boldsymbol{u},\boldsymbol{u}}$ is an upper bound defined in Lemma 9.2.1.a.

Proof. $\overline{h}^m(\boldsymbol{X}, \boldsymbol{Y})$ is a two-sample U-statistic of order $2m$ and E_h is its expectation as $\boldsymbol{X}$ and $\boldsymbol{Y}$ are *i.i.d.* random variables. $\overline{h}^m(\boldsymbol{X}, \boldsymbol{Y})$ is a sum of dependant variables and it is possible to rewrite $\overline{h}^m(\boldsymbol{X}, \boldsymbol{Y})$ as a sum of independent random variables. As α, β are compactly supported by hypothesis, the UOT loss is bounded thanks to Lemma 9.2.1.a. Thus, we can apply Theorem 7.3.2 to our U-statistic and get the claimed bound. □

The second lemma bounds the deviation between the incomplete estimator $\widetilde{h}^{m,k}$ and the complete estimator $\overline{h}^m$.

Lemma A.2.4 (Deviation bound)**.** *Let $\delta \in (0,1)$, three integers $k \geq 1$ and $m \leq n$ be fixed, and two compactly supported distributions α, β. Consider two n-tuples $\boldsymbol{X} \sim \alpha^{\otimes n}$ and $\boldsymbol{Y} \sim \beta^{\otimes n}$ and a kernel $h \in \{\mathrm{OT}_\phi^{\tau,\varepsilon}, S_\phi^{\tau,\varepsilon}\}$ with the squared Euclidean distance as ground cost. We have a deviation bound between $\widetilde{h}^{m,k}(\boldsymbol{X}, \boldsymbol{Y})$ and $\overline{h}^m(\boldsymbol{X}, \boldsymbol{Y})$ depending on the number of batches k:*

$$|\widetilde{h}^{m,k}(\boldsymbol{X}, \boldsymbol{Y}) - \overline{h}^m(\boldsymbol{X}, \boldsymbol{Y})| \leqslant M^h_{\boldsymbol{u},\boldsymbol{u}} \sqrt{\frac{2\log(2/\delta)}{k}}, \tag{A.111}$$

with probability at least $1-\delta$ and where $M^h_{\boldsymbol{u},\boldsymbol{u}}$ is an upper bound defined in Lemma 9.2.1.a.

Proof. First note that $\widetilde{h}^{m,k}(\boldsymbol{X}, \boldsymbol{Y})$ is a subsample quantity of $\overline{h}^m(\boldsymbol{X}, \boldsymbol{Y})$. Let us consider the sequence of random variables $((\mathfrak{b}_l(I,J)_{(I,J)\in\mathcal{P}^m})_{1\leqslant l\leqslant k}$ such that $\mathfrak{b}_l(I,J)$ is equal to 1 if (I,J) has been selected at the $l-$th draw and 0 otherwise. By construction of $\widetilde{h}^{m,k}$, the aforementioned sequence is an i.i.d sequence of random vectors and the $\mathfrak{b}_l(I,J)$ are Bernoulli random variables of parameter $1/|\Gamma|$. We then have

$$\widetilde{h}^{m,k}(\boldsymbol{X}, \boldsymbol{Y}) - \overline{h}^m(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{k}\sum_{l=1}^{k} \omega_l, \tag{A.112}$$

where $\omega_l = \sum_{(I,J)\in\mathcal{P}^m}(\mathfrak{b}_l(I,J) - \frac{1}{|\Gamma|})h(I,J)$. Conditioned upon $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ and $\boldsymbol{Y} = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n)$, the variables ω_l are independent, centered and bounded by $2M^h_{\boldsymbol{u},\boldsymbol{u}}$ thanks to Lemma 9.2.1.a. Using

Hoeffding's inequality yields

$$\mathbb{P}(|\widetilde{h}^{m,k}(\boldsymbol{X},\boldsymbol{Y}) - \overline{h}^{m}(\boldsymbol{X},\boldsymbol{Y})| > \varepsilon) = \mathbb{E}[\mathbb{P}(|\widetilde{h}^{m,k}(\boldsymbol{X},\boldsymbol{Y}) - \overline{h}^{m}(\boldsymbol{X},\boldsymbol{Y})| > \varepsilon | X, Y)], \tag{A.113}$$

$$= \mathbb{E}[\mathbb{P}(|\frac{1}{k}\sum_{l=1}^{k}\omega_l)| > \varepsilon | X, Y)], \tag{A.114}$$

$$\leqslant \mathbb{E}[2e^{\frac{-k\varepsilon^2}{2(M^h_{\boldsymbol{u},\boldsymbol{u}})^2}}] = 2e^{\frac{-k\varepsilon^2}{2(M^h_{\boldsymbol{u},\boldsymbol{u}})^2}}, \tag{A.115}$$

which concludes the proof. □

We are now ready to prove Theorem 9.2.1.a.

Theorem 9.2.1.a (Maximal deviation bound)**.** *Let $\delta \in (0,1)$, three integers $k \geq 1$ and $m \leq n$ be fixed and two compactly supported distributions α, β. Consider two n-tuples $\boldsymbol{X} \sim \alpha^{\otimes n}$ and $\boldsymbol{Y} \sim \beta^{\otimes n}$ and a kernel $h \in \{\mathrm{OT}^{\tau,\varepsilon}_{\phi}, S^{\tau,\varepsilon}_{\phi}\}$ with the squared Euclidean distance as ground cost. We have a maximal deviation bound between $\widetilde{h}^{m,k}(\boldsymbol{X},\boldsymbol{Y})$ and the expectation over minibatches E_h depending on the number of empirical data n and the number of batches k*

$$|\widetilde{h}^{m,k}(\boldsymbol{X},\boldsymbol{Y}) - E_h| \leq M^h_{\boldsymbol{u},\boldsymbol{u}}\sqrt{\frac{\log(2/\delta)}{2\lfloor n/m \rfloor}} + M^h_{\boldsymbol{u},\boldsymbol{u}}\sqrt{\frac{2\log(2/\delta)}{k}}, \tag{A.116}$$

with probability at least 1 - δ and where $M^h_{\boldsymbol{u},\boldsymbol{u}}$ is an upper bound defined in Lemma 9.2.1.a.

Proof. Thanks to Lemma A.2.4 and A.2.3 we get

$$|\widetilde{h}^{m,k}(\boldsymbol{X},\boldsymbol{Y}) - E_h| \leq |\widetilde{h}^{m,k}(\boldsymbol{X},\boldsymbol{Y}) - \overline{h}^{m}(\boldsymbol{X},\boldsymbol{Y})| + |\overline{h}^{m}(\boldsymbol{X},\boldsymbol{Y}) - E_h|, \tag{A.117}$$

$$\leq M^h_{\boldsymbol{u},\boldsymbol{u}}\sqrt{\frac{\log(2/\delta)}{2\lfloor n/m \rfloor}} + M^h_{\boldsymbol{u},\boldsymbol{u}}\sqrt{\frac{2\log(2/\delta)}{k}}, \tag{A.118}$$

with probability at least $1 - (\frac{\delta}{2} + \frac{\delta}{2}) = 1 - \delta$. □

We now give the details of the proof of Theorem 9.2.1.b. In what follows, we denote by $\Pi_{(i)}$ the i-th row of matrix Π. Let us denote by $\mathbf{1} \in \mathbb{R}^n$ the vector whose entries are all equal to 1.

Theorem 9.2.1.b (Distance to marginals)**.** *Let $\delta \in (0,1)$, two integers $m \leq n$ be fixed. Consider two n-tuples $\boldsymbol{X} \sim \alpha^{\otimes n}$ and $\boldsymbol{Y} \sim \beta^{\otimes n}$ and the kernel $h = \mathrm{OT}^{\tau,\varepsilon}_{\phi}$ with the squared Euclidean distance as ground cost. For all integer $k \geq 1$, all $1 \leqslant i \leqslant n$, with probability at least $1 - \delta$ on the draw of $\boldsymbol{X}, \boldsymbol{Y}$ and D_k we have*

$$|\widetilde{\Pi}^{h,m,k}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1} - \overline{\Pi}^{h,m}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1}| \leqslant \mathfrak{M}^{\infty}_{\Pi}\sqrt{\frac{2\log(2/\delta)}{k}}, \tag{A.119}$$

where $\mathfrak{M}^{\infty}_{\Pi}$ denotes an upper bound of all minibatch UOT plan.

Proof. Let us consider the sequence of random variables $((\mathfrak{b}_p(I,J)_{(I,J)\in\Gamma})_{1\leqslant p\leqslant k}$ such that $\mathfrak{b}_p(I,J)$ is equal to 1 if (I,J) has been selected at the $p-$th draw and 0 otherwise. By construction of $\widetilde{\Pi}^{h,m,k}(\boldsymbol{X},\boldsymbol{Y})$, the aforementioned sequence is an i.i.d sequence of random vectors and the $\mathfrak{b}_p(I,J)$ are bernoulli random variables of parameter $1/|\Gamma|$. We then have:

$$\widetilde{\Pi}^{h,m,k}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1} = \frac{1}{k}\sum_{p=1}^{k}\omega_p, \tag{A.120}$$

where $\omega_p = \sum_{(I,J)\in\Gamma}\sum_{j=1}^{n}(\Pi_{I,J})_{i,j}\mathfrak{b}_p(I,J)$. Conditioned upon $\boldsymbol{X} = (\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)$ and $\boldsymbol{Y} = (\boldsymbol{y}_1,\cdots,\boldsymbol{y}_n)$, the random vectors ω_p are independent, and thanks to Lemma 9.2.1.b, they are bounded by a constant $\mathfrak{M}_\Pi$ which is the maximum mass of all optimal minibatch unbalanced plan in $\mathrm{Opt}(h, C(\boldsymbol{X}(I),\boldsymbol{Y}(J)), \boldsymbol{u}_m, \boldsymbol{u}_m)$. We denote the maximum upper bound $\mathfrak{M}_\Pi$ of all minibatch UOT plan as $\mathfrak{M}_\Pi^\infty$. Moreover, one can observe that $\mathbb{E}[\widetilde{\Pi}^{h,m,k}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1}] = \overline{\Pi}^{h,m}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1}$. Using Hoeffding's inequality yields

$$\mathbb{P}(|\widetilde{\Pi}^{h,m,k}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1} - \overline{\Pi}^{h,m}(\boldsymbol{X},\boldsymbol{Y})_{(i)}\mathbf{1})| > \varepsilon) = \mathbb{E}[\mathbb{P}(|\frac{1}{k}\sum_{p=1}^{k}\omega_p - \mathbb{E}[\frac{1}{k}\sum_{p=1}^{k}\omega_p])| > \varepsilon|X,Y)], \tag{A.121}$$

$$\leqslant 2e^{-2\frac{k\varepsilon^2}{(\mathfrak{M}_\Pi^\infty)^2}}, \tag{A.122}$$

which concludes the proof. □

Note that the unbalanced Sinkhorn divergence $S_\phi^{\tau,\varepsilon}$ involves three terms of the form $\mathrm{OT}_\phi^{\tau,\varepsilon}$, hence three transport plans, which explains why we do not attempt to define an associated averaged minibatch transport matrix.

Proof of Theorem 9.2.2

To prove the exchange of gradients and expectations over minibatches we rely on Clarke differential. We need to use this non-smooth analysis tool as unregularized UOT is not differentiable. It is not differentiable because the set of optimal solutions might not be a singleton. Clarke differential are generalized gradients for locally Lipschitz function and non necessarily convex. A similar strategy was developed in Section A.1. The key element of this appendix is to rewrite the original UOT problem $\mathrm{OT}_\phi^{\tau,\varepsilon}$ as:

$$\mathrm{OT}_\phi^{\tau,\varepsilon}(\mathbf{a},\mathbf{b},C) = \min_{\Pi\in\mathbb{R}_+^{n\times n}} \langle C,\Pi\rangle + \varepsilon\mathtt{KL}(\Pi|\mathbf{a}\otimes\mathbf{b}) + \tau D_\phi(\Pi\mathbf{1}_n|\mathbf{a}) + \tau D_\phi(\Pi^\top\mathbf{1}_n|\mathbf{b}), \tag{A.123}$$

$$= \min_{\Pi\in\mathrm{Opt}(\mathrm{OT}_\phi^{\tau,\varepsilon},C,\mathbf{a},\mathbf{b})} \langle C,\Pi\rangle + \varepsilon\mathtt{KL}(\Pi|\mathbf{a}\otimes\mathbf{b}) + \tau D_\phi(\Pi\mathbf{1}_n|\mathbf{a}) + \tau D_\phi(\Pi^\top\mathbf{1}_n|\mathbf{b}), \tag{A.124}$$

Where $\mathrm{Opt}(\mathrm{OT}_\phi^{\tau,\varepsilon},C,\mathbf{a},\mathbf{b})$ is a compact set of the set of measures $\mathcal{M}_+(\mathcal{X})$. The compact set is a key element for using Danskin like theorem (Proposition B.25 [Bertsekas 1997]).

We start by recalling a basic proposition for Clarke regular function:

Proposition 18. *A $\mathbf{C}^1$ or convex map is Clarke regular.*

Proof. see Proposition 2.3.6 [Clarke 1990] □

We first give a lemma which gives the Clarke regularity of the UOT cost with respect to a parametrized random vector.

Lemma A.2.5. *Let $\boldsymbol{u}$ be a uniform probability vector. Let $\boldsymbol{X}$ be a $\mathbb{R}^{dm}$-valued random variable, and $\{\boldsymbol{Y}_\theta\}$ a family of $\mathbb{R}^{dm}$-valued random variables defined on the same probability space, indexed by $\theta\in\Theta$, where $\Theta\subset\mathbb{R}^q$ is open. Assume that $\theta\mapsto\boldsymbol{Y}_\theta$ is $\mathbf{C}^1$. Consider a $\mathbf{C}^1$ cost C and let $h\in\{\mathrm{OT}_\phi^{\tau,\varepsilon}, S_\phi^{\tau,\varepsilon}\}$. Then the function $\theta\mapsto -h(\boldsymbol{u},\boldsymbol{u},C(\boldsymbol{X},\boldsymbol{Y}_\theta))$ is Clarke regular. Furthermore, for $h=\mathrm{OT}_\phi^{\tau,\varepsilon}$ and for all $1\leq i\leq q$ we have:*

$$\partial_{\theta_i} h(\boldsymbol{u},\boldsymbol{u},C(\boldsymbol{X},\boldsymbol{Y}_\theta)) = \overline{co}\{-\langle\Pi\cdot D\rangle\cdot(\nabla_{\theta_i}Y) : \Pi\in\mathrm{Opt}(h,C(\boldsymbol{X},\boldsymbol{Y}),\boldsymbol{u},\boldsymbol{u}), \tag{A.125}$$
$$D\in\mathbb{R}^{m,m}, D_{j,k} = \nabla_Y C_{j,k}(\boldsymbol{X},\boldsymbol{Y}_\theta)\},$$

where ∂_{θ_i} is the Clarke subdifferential with respect to θ_i, $\nabla_Y C_{j,k}$ is the differential of the cell $C_{j,k}$ of the cost matrix with respect to Y, $\mathrm{Opt}(h, C(\boldsymbol{X}, \boldsymbol{Y}_\theta), \boldsymbol{u}, \boldsymbol{u})$ is the set of optimal transport plan and $\overline{\mathrm{co}}$ denotes the closed convex hull. Note that when $\varepsilon > 0$ the set $\mathrm{Opt}(h, C(\boldsymbol{X}, \boldsymbol{Y}_\theta), \boldsymbol{u}, \boldsymbol{u})$ is reduced to a singleton, and the notation $\overline{\mathrm{co}}$ is superfluous.

Proof. We start with the regularity of $\theta \mapsto -\mathrm{OT}_\phi^{\tau,\varepsilon}(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$. To prove the Clarke regularity of this map, we rely on a chain rule argument. Consider the function $\boldsymbol{Y} \mapsto C_{j,k}(\boldsymbol{X}, \boldsymbol{Y})$, it is Clarke regular because it is $\mathbf{C}^1$. Since $\theta \mapsto \boldsymbol{Y}_\theta$ is $\mathbf{C}^1$, it follows by the chain rule that $\theta \mapsto C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta)$ is $\mathbf{C}^1$ and thus Clarke regular. The Unbalanced OT cost $\mathrm{OT}_\phi^{\tau,\varepsilon}$ is a minimization of an energy which is linear in C, and it is thus concave in C, hence $-\mathrm{OT}_\phi^{\tau,\varepsilon}$ is Clarke regular by convexity. Therefore from Theorem 2.3.9(i) and Proposition 2.3.1 (for $s = 1$) in [Clarke 1990] it follows that $\theta \mapsto -\mathrm{OT}_\phi^{\tau,\varepsilon}(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$ is Clarke regular.

We now furnish the gradients associated to $\theta \mapsto -\mathrm{OT}_\phi^{\tau,\varepsilon}(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$. By chain rule, the gradient of $\theta \mapsto C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta)$ reads

$$\nabla_{\theta_i} C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta) = \nabla_Y C_{j,k}(\boldsymbol{X}, \boldsymbol{Y}_\theta) \cdot \nabla_{\theta_i} Y_\theta.$$

We now deal with the gradient of the map $C \mapsto \mathrm{OT}_\phi^{\tau,\varepsilon}(\boldsymbol{u}, \boldsymbol{u}, C)$ by verifying the assumptions of Danskin's theorem [Clarke 1975, Theorem 2.1]. We use in particular the remark below [Clarke 1975, Theorem 2.1] which states that the hypothesis on the map are verified if the map is *u.s.c* in both variables (Π, C) and convex in C. We recall that $\mathrm{Opt}(h, C(\boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{u}, \boldsymbol{u})$ is a compact and a convex set, thanks to Lemma A.2.1 and Lemma A.2.2. Furthermore, the energy associated to $h = \mathrm{OT}_\phi^{\tau,\varepsilon}$ is concave in the cost C and *l.s.c* in (Π, C) [Liero 2017, Lemma 3.9]. From [Clarke 1975, Theorem 2.1] it follows that the subderivatives of the convex function $C \mapsto -h(\boldsymbol{u}, \boldsymbol{u}, C)$ are equal to $\mathrm{Opt}(h, C(\boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{u}, \boldsymbol{u})$, due to the energy's linearity in C. Thus combining the formulas of the Danskin theorem with the Chain rule yields Equation (A.125). When $\varepsilon > 0$ the set $\mathrm{Opt}(h, C(\boldsymbol{X}, \boldsymbol{Y}_\theta), \boldsymbol{u}, \boldsymbol{u})$ is reduced to a singleton, and the notation $\overline{\mathrm{co}}$ is superfluous.

We now give the proof for the regularity of the map $\theta \mapsto S_\phi^{\tau,\varepsilon}(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$ with $\varepsilon > 0$ as when $\varepsilon = 0$, we get the unregularized UOT treated in the above paragraph. We recall that $S_\phi^{\tau,\varepsilon}$ is the summation of three terms of the form $\mathrm{OT}_\phi^{\tau,\varepsilon}$. For $\varepsilon > 0$ and each term of the sum, the set of optimal plans $\mathrm{Opt}(\mathrm{OT}_\phi^{\tau,\varepsilon}, C(\boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{u}, \boldsymbol{u})$ is reduced to a unique element and the differential (A.125) is also a singleton, thus $\mathrm{OT}_\phi^{\tau,\varepsilon}$ is differentiable. Then $S_\phi^{\tau,\varepsilon}$ is differentiable as a difference of differentiable functions. Furthermore $S_\phi^{\tau,\varepsilon}$ is also Clarke regular as a difference of differentiable functions. □

We finally prove Theorem 9.2.2.

Theorem 9.2.2. *Let $\boldsymbol{u}$ be a uniform probability vector and let $\boldsymbol{X}, \boldsymbol{Y}, C$ be as in Lemma A.2.5, $h \in \{\mathrm{OT}_\phi^{\tau,\varepsilon}, S_\phi^{\tau,\varepsilon}\}$, and assume in addition that the random variables $\boldsymbol{X}, \{Y_\theta\}_{\theta\in\Theta}$ are compactly supported. If for all $\theta \in \Theta$ there exists an open neighbourhood U, $\theta \in U \subset \Theta$, and a random variable $K_U : \Omega \to \mathbb{R}$ with finite expected value, such that*

$$\|C(\boldsymbol{X}(\omega), \boldsymbol{Y}_{\theta_1}(\omega)) - C(\boldsymbol{X}(\omega), \boldsymbol{Y}_{\theta_2}(\omega))\| \leq K_U(\omega)\|\theta_1 - \theta_2\|, \tag{A.126}$$

then we have

$$\partial_\theta \mathbb{E}\left[h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right] = \mathbb{E}\left[\partial_\theta h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right]. \tag{A.127}$$

with both expectation being finite. Furthermore the function $\theta \mapsto -\mathbb{E}\left[h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right]$ is also Clarke regular.

Proof. Suppose that $U \subset \Theta$ is open and K_U is a function for which (A.126) is satisfied. As data lie in compacts the ground cost C, which is $\mathbf{C}^1$, is in a compact K_C and as the map $C \mapsto h(\boldsymbol{u}, \boldsymbol{u}, C)$ is locally Lipshitz by Lemma A.2.2, there exists a uniform constant which makes the map $C \mapsto h(\boldsymbol{u}, \boldsymbol{u}, C)$ globally Lipshitz on the compact K_C. Thus, a similar bound to (A.126) is also satisfied for the function $h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}(\omega), \boldsymbol{Y}_\theta(\omega)))$. Thanks to Lemma A.2.5, $-h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))$ is Clarke regular, the interchange (A.127) and regularity of $\theta \mapsto -\mathbb{E}[h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))]$ will follow from Theorem 2.7.2 and Remark 2.3.5 [Clarke 1990], once we establish that the expectation on the left hand side is finite. This is direct as we suppose we have compactly supported distributions and C is a $\mathbf{C}^1$ cost. Indeed consider the function which is equal to $M^h_{\boldsymbol{u},\boldsymbol{u}}$ on the distributions's support and which is set to 0 everywhere else. Taking the expectation on this function is finite as $M^h_{\boldsymbol{u},\boldsymbol{u}}$ is finite. □

List of Figures

List of Tables

Bibliography

[Abadi 2015] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu et Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015. Software available from tensorflow.org. (Cited on pages 4 and 2.)

[Abid 2018] Brahim Khalil Abid et Robert Gower. *Stochastic algorithms for entropy-regularized optimal transport problems.* In Amos Storkey et Fernando Perez-Cruz, editeurs, Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, volume 84 of *Proceedings of Machine Learning Research*, pages 1505–1512, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR. (Cited on page 21.)

[Aggarwal 2001] Charu C. Aggarwal, Alexander Hinneburg et Daniel A. Keim. *On the Surprising Behavior of Distance Metrics in High Dimensional Space.* In Lecture Notes in Computer Science, pages 420–434. Springer, 2001. (Cited on page 128.)

[Alaya 2019] Mokhtar Z. Alaya, Maxime Berar, Gilles Gasso et Alain Rakotomamonjy. *Screening Sinkhorn Algorithm for Regularized Optimal Transport.* In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. (Cited on page 21.)

[Aljundi 2016] Rahaf Aljundi et Tinne Tuytelaars. *Lightweight Unsupervised Domain Adaptation by Convolutional Filter Reconstruction.* In Gang Hua et Hervé Jégou, editeurs, Computer Vision – ECCV 2016 Workshops, pages 508–515, Cham, 2016. Springer International Publishing. (Cited on page 34.)

[Altschuler 2017] Jason Altschuler, Jonathan Niles-Weed et Philippe Rigollet. *Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration.* In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et R. Garnett, editeurs, Advances in Neural Information Processing Systems 30, pages 1964–1974. Curran Associates, Inc., 2017. (Cited on pages 5, 3, 21, 55 and 62.)

[Alvarez-Melis 2019] David Alvarez-Melis, Stefanie Jegelka et Tommi S. Jaakkola. *Towards Optimal Transport with Global Invariances.* In Kamalika Chaudhuri et Masashi Sugiyama, editeurs, Proceedings of Machine Learning Research, volume 89 of *Proceedings of Machine Learning Research*, pages 1870–1879. PMLR, 16–18 Apr 2019. (Cited on page 26.)

[Arjovsky 2017] Martin Arjovsky, Soumith Chintala et Léon Bottou. *Wasserstein Generative Adversarial Networks*. In Proceedings of the 34th ICML, Proceedings of Machine Learning Research. PMLR, 2017. (Cited on pages 5, 6, 3, 4, 16, 23, 39, 40, 92 and 151.)

[Arjovsky 2020] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani et David Lopez-Paz. *Invariant Risk Minimization*, 2020. (Cited on page 154.)

[Arpit 2017] D. Arpit, S.K. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M.S. Kanwal, T. Maharaj, A. Fischer, A.C. Courville, Y. Bengio et S. Lacoste-Julien. *A closer look at memorization in deep networks*. In The International Conference on Machine Learning, 2017. (Cited on page 52.)

[Audebert 2018] N. Audebert, B. Le Saux et S. Lefèvre. *Generative adversarial networks for realistic synthesis of hyperspectral samples*. In 2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Juillet 2018. (Cited on page 69.)

[Audebert 2019] N. Audebert, B. Le Saux et S. Lefevre. *Deep Learning for Classification of Hyperspectral Data: A Comparative Review*. IEEE Geoscience and Remote Sensing Magazine, vol. 7, no. 2, 2019. (Cited on page 77.)

[Badrinarayanan 2017] Vijay Badrinarayanan, Alex Kendall et Roberto Cipolla. *Segnet: A deep convolutional encoder-decoder architecture for image segmentation*. IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pages 2481–2495, 2017. (Cited on page 81.)

[Baktashmotlagh 2013] Mahsa Baktashmotlagh, Mehrtash T. Harandi, Brian C. Lovell et Mathieu Salzmann. *Unsupervised Domain Adaptation by Domain Invariant Projection*. In 2013 IEEE International Conference on Computer Vision, pages 769–776, 2013. (Cited on page 34.)

[Balaji 2020] Yogesh Balaji, Rama Chellappa et Soheil Feizi. *Robust Optimal Transport with Applications in Generative Modeling and Domain Adaptation*. In Advances in Neural Information Processing Systems, 2020. (Cited on pages 5, 3, 138, 144, 145 and 153.)

[Bansal 2018] Aayush Bansal, Shugao Ma, Deva Ramanan et Yaser Sheikh. *Recycle-GAN: Unsupervised Video Retargeting*. In ECCV, 2018. (Cited on page 41.)

[Bassetti 2006] Federico Bassetti, Antonella Bodini et Eugenio Regazzini. *On minimum Kantorovich distance estimators*. Statistics & Probability Letters, vol. 76, 2006. (Cited on page 15.)

[Beery 2018] Sara Beery, Grant Van Horn et Pietro Perona. *Recognition in Terra Incognita*. In Proceedings of the European Conference on Computer Vision (ECCV), September 2018. (Cited on page 153.)

[Belkin 2004] Mikhail Belkin et Partha Niyogi. *Semi-Supervised Learning on Riemannian Manifolds*. Machine Learning, vol. 56, no. 1-3, pages 209–239, 2004. (Cited on page 48.)

[Bellemare 2017] Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer et Rémi Munos. *The Cramer Distance as a Solution to Biased Wasserstein Gradients*. CoRR, vol. abs/1705.10743, 2017. (Cited on page 95.)

[Bellet 2015] Aurélien Bellet, Amaury Habrard et Marc Sebban. Metric Learning, volume 9 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan and Claypool Publishers (USA), Synthesis Lectures on Artificial Intelligence and Machine Learning, pp 1-151, Janvier 2015. (Cited on page 100.)

[Ben-David 2007] Shai Ben-David, John Blitzer, Koby Crammer et Fernando Pereira. *Analysis of Representations for Domain Adaptation.* In B. Schölkopf, J. Platt et T. Hoffman, editeurs, Advances in Neural Information Processing Systems, volume 19. MIT Press, 2007. (Cited on page 34.)

[Bengio 2013] Yoshua Bengio, Aaron Courville et Pascal Vincent. *Representation Learning: A Review and New Perspectives.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pages 1798–1828, 2013. (Cited on page 36.)

[Bernton 2019] Espen Bernton, Pierre E Jacob, Mathieu Gerber et Christian P Robert. *On parameter estimation with the Wasserstein distance.* Information and Inference: A Journal of the IMA, vol. 8, no. 4, pages 657–676, 2019. (Cited on pages 95 and 103.)

[Berthelot 2019] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver et Colin A Raffel. *MixMatch: A Holistic Approach to Semi-Supervised Learning.* In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. (Cited on page 48.)

[Bertsekas 1973] D. P. Bertsekas. *Stochastic optimization problems with nondifferentiable cost functionals.* Journal of Optimization Theory and Applications, vol. 12, no. 2, pages 218–231, Aug 1973. (Cited on page 96.)

[Bertsekas 1997] Dimitri P Bertsekas. *Nonlinear programming.* Journal of the Operational Research Society, vol. 48, no. 3, pages 334–334, 1997. (Cited on pages 20, 175 and 184.)

[Bertsimas 1997] Dimitris Bertsimas et John N. Tsitsiklis. Introduction to linear optimization. Numeéro 6 de Athena scientific series in optimization and neural computation. Athena Scientific, Belmont, Mass. [u.a.], 1997. (Cited on page 16.)

[Bińkowski 2018] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel et Arthur Gretton. *Demystifying MMD GANs.* In International Conference on Learning Representations, 2018. (Not cited)

[Blondel 2018] Mathieu Blondel, Vivien Seguy et Antoine Rolet. *Smooth and Sparse Optimal Transport.* In Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, 2018. (Cited on pages 24, 109 and 129.)

[Bonnotte 2013] Nicolas Bonnotte. *Unidimensional and Evolution Methods for Optimal Transportation.* PhD thesis, Université de Paris-Sud, 2013. (Cited on pages 17 and 18.)

[Boser 1992] Bernhard E. Boser, Isabelle M. Guyon et Vladimir N. Vapnik. *A Training Algorithm for Optimal Margin Classifiers.* In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery. (Cited on page 73.)

[Bottou 2010] Léon Bottou. *Large-scale machine learning with stochastic gradient descent.* In in COMPSTAT, 2010. (Cited on pages 2 and 32.)

[Bottou 2018] Léon Bottou, Frank E. Curtis et Jorge Nocedal. *Optimization Methods for Large-Scale Machine Learning.* SIAM Review, vol. 60, no. 2, pages 223–311, 2018. (Cited on page 95.)

[Brock 2019] Andrew Brock, Jeff Donahue et Karen Simonyan. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. In International Conference on Learning Representations, 2019. (Cited on pages 4, 2, 40 and 188.)

[Brodley 1999] Carla E. Brodley et Mark A. Friedl. *Identifying Mislabeled Training Data*. Journal of Artificial Intelligence Research, 1999. (Cited on page 52.)

[Brown 2018] Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano et Ian Goodfellow. *Unrestricted adversarial examples*. arXiv preprint arXiv:1809.08352, 2018. (Cited on page 47.)

[Bunne 2019] Charlotte Bunne, David Alvarez-Melis, Andreas Krause et Stefanie Jegelka. *Learning Generative Models across Incomparable Spaces*. In Kamalika Chaudhuri et Ruslan Salakhutdinov, editeurs, Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 851–861, Long Beach, California, USA, 09–15 Jun 2019. PMLR. (Cited on pages 5, 3 and 26.)

[Burnel 2020] Jean-Christophe Burnel, Kilian Fatras et Nicolas Courty. *Generating natural adversarial hyperspectral examples with a modified Wasserstein GAN*. In C&ESAR 2020, 2020. (Cited on pages 7 and 5.)

[Burnel 2021] Jean-Christophe Burnel, Kilian Fatras, Rémi Flamary et Nicolas Courty. *Generating natural adversarial Remote Sensing Images*. IEEE Transactions on Geoscience and Remote Sensing, 2021. (Cited on pages 6, 4 and 69.)

[Cao 2018] Zhangjie Cao, Lijia Ma, Mingsheng Long et Jianmin Wang. *Partial Adversarial Domain Adaptation*. In Proceedings of the European Conference on Computer Vision (ECCV), September 2018. (Cited on page 147.)

[Cao 2019] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang et Qiang Yang. *Learning to Transfer Examples for Partial Domain Adaptation*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. (Cited on page 147.)

[Caramanis 2011] Constantine Caramanis, Shie Mannor et Huan Xu. Optimization for machine learning, chapitre Robust Optimization in Machine Learning. The MIT Pres, 2011. (Cited on page 53.)

[Chan 2019] Caroline Chan, Shiry Ginosar, Tinghui Zhou et Alexei A Efros. *Everybody Dance Now*. In IEEE International Conference on Computer Vision (ICCV), 2019. (Cited on page 41.)

[Chapel 2020] Laetitia Chapel, Mokhtar Z. Alaya et Gilles Gasso. *Partial Optimal Transport with Applications on Positive-Unlabeled Learning*. In Advances in Neural Information Processing Systems, 2020. (Cited on pages 5, 3 and 138.)

[Chapelle 2006] Olivier Chapelle, Bernhard Schölkopf et Alexander Zien, editeurs. Semi-supervised learning. The MIT Press, 2006. (Cited on page 48.)

[Chen 2019] P. Chen, B.B. Liao, G. Chen et S. Zhang. *Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels*. In The International Conference on Machine Learning, 2019. (Cited on page 52.)

[Chen 2020] Minghao Chen, Shuai Zhao, Haifeng Liu et Deng Cai. *Adversarial-Learned Loss for Domain Adaptation.* arXiv, vol. abs/2001.01046, 2020. (Cited on pages 34, 144 and 145.)

[Chizat 2017] Lénaïc Chizat. *Unbalanced Optimal Transport : Models, Numerical Methods, Applications.* PhD thesis, Université Paris sciences et lettres, 2017. (Cited on pages 27 and 138.)

[Chizat 2018a] Lenaïc Chizat, Gabriel Peyré, Bernhard Schmitzer et François-Xavier Vialard. *Scaling algorithms for unbalanced optimal transport problems.* Math. Comput., vol. 87, no. 314, pages 2563–2609, 2018. (Cited on pages 21 and 28.)

[Chizat 2018b] Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer et François-Xavier Vialard. *Unbalanced optimal transport: Dynamic and Kantorovich formulations.* Journal of Functional Analysis, vol. 274, no. 11, pages 3090–3123, 2018. (Cited on page 27.)

[Choi 2018a] K. Choi, G. Fazekas, K. Cho et M. Sandler. *The Effects of Noisy Labels on Deep Convolutional Neural Networks for Music Tagging.* IEEE Transactions on Emerging Topics in Computational Intelligence, 2018. (Cited on page 52.)

[Choi 2018b] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim et Jaegul Choo. *StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. (Cited on page 41.)

[Chowdhury 2019] Samir Chowdhury et Facundo Mémoli. *The Gromov–Wasserstein distance between networks and stable network invariants.* Information and Inference: A Journal of the IMA, vol. 8, no. 4, pages 757–787, 2019. (Cited on page 27.)

[Clarke 1975] H. Frank Clarke. *Generalized Gradients and Applications.* Transactions of The American Mathematical Society, pages 247–247, 1975. (Cited on page 185.)

[Clarke 1990] Frank H Clarke. Optimization and nonsmooth analysis. SIAM, 1990. (Cited on pages 9, 7, 97, 98, 99, 175, 176, 184, 185 and 186.)

[Clémençon 2016] Stephan Clémençon, Igor Colin et Aurélien Bellet. *Scaling-up Empirical Risk Minimization: Optimization of Incomplete U-statistics.* Journal of Machine Learning Research, vol. 17, no. 76, pages 1–36, 2016. (Cited on page 100.)

[Clémençon 2008] Stéphan Clémençon, Gábor Lugosi et Nicolas Vayatis. *Ranking and Empirical Minimization of U-statistics.* The Annals of Statistics, vol. 36, no. 2, pages 844 – 874, 2008. (Cited on page 100.)

[Cortes 2014] Corinna Cortes et Mehryar Mohri. *Domain adaptation and sample bias correction theory and algorithm for regression.* Theoretical Computer Science, vol. 519, pages 103–126, 2014. Algorithmic Learning Theory. (Cited on page 34.)

[Courty 2014] Nicolas Courty, Rémi Flamary et Devis Tuia. *Domain Adaptation with Regularized Optimal Transport.* In Machine Learning and Knowledge Discovery in Databases, 2014. (Cited on pages 24 and 34.)

[Courty 2017a] N. Courty, R. Flamary, D. Tuia et A. Rakotomamonjy. *Optimal Transport for Domain Adaptation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 9, pages 1853–1865, Sep. 2017. (Cited on pages 5, 3, 24, 34, 35 and 187.)

[Courty 2017b] Nicolas Courty, Rémi Flamary, Amaury Habrard et Alain Rakotomamonjy. *Joint distribution optimal transportation for domain adaptation.* In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. (Cited on pages 14, 34, 35, 100 and 143.)

[Csiszar 1975] I. Csiszar. *I-Divergence Geometry of Probability Distributions and Minimization Problems.* The Annals of Probability, vol. 3, no. 1, pages 146 – 158, 1975. (Cited on pages 5 and 3.)

[Cuturi 2013] Marco Cuturi. *Sinkhorn Distances: Lightspeed Computation of Optimal Transport.* In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani et K. Q. Weinberger, editeurs, Advances in Neural Information Processing Systems 26, pages 2292–2300. Curran Associates, Inc., 2013. (Cited on pages 5, 3, 19, 21, 55 and 57.)

[Cuturi 2014] Marco Cuturi et David Avis. *Ground metric learning.* The Journal of Machine Learning Research, vol. 15, no. 1, pages 533–564, 2014. (Cited on pages 56 and 152.)

[Dai 2021] Biwei Dai et Uros Seljak. *Sliced Iterative Normalizing Flows.* In ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models, 2021. (Cited on page 153.)

[Damodaran 2018] Bharath Bhushan Damodaran, Benjamin Kellenberger, Remi Flamary, Devis Tuia et Nicolas Courty. *DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation.* In ECCV 2018 - 15th European Conference on Computer Vision, volume 11208 of *LNCS*, pages 467–483, Munich, Germany, Septembre 2018. Springer. European Conference on Computer Vision 2018 (ECCV-2018). (Cited on pages 5, 7, 3, 35, 36, 61, 92, 138, 143, 144, 148, 152, 154 and 187.)

[Damodaran 2019] Bharath Bhushan Damodaran, Rémi Flamary, Viven Seguy et Nicolas Courty. *An Entropic Optimal Transport Loss for Learning Deep Neural Networks under Label Noise in Remote Sensing Images.* In Computer Vision and Image Understanding, 2019. (Cited on page 147.)

[Damodaran 2020] Bharath Bhushan Damodaran, Rémi Flamary, Vivien Seguy et Nicolas Courty. *An Entropic Optimal Transport loss for learning deep neural networks under label noise in remote sensing images.* Computer Vision and Image Understanding, vol. 191, page 102863, 2020. (Cited on page 61.)

[Dantzig 1997] George B. Dantzig et Mukund N. Thapa. Linear programming 1: Introduction. Springer-Verlag, Berlin, Heidelberg, 1997. (Cited on page 18.)

[Davis 2020] Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade et Jason D Lee. *Stochastic subgradient method converges on tame functions.* Foundations of computational mathematics, vol. 20, no. 1, pages 119–154, 2020. (Cited on pages 124 and 141.)

[Defazio 2014] Aaron Defazio, Francis Bach et Simon Lacoste-Julien. *SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives.* In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence et K. Q. Weinberger, editeurs, Advances in Neural Information Processing Systems, volume 27. Curran Associates, Inc., 2014. (Cited on pages 21 and 22.)

[Deng 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li et Li Fei-Fei. *Imagenet: A large-scale hierarchical image database.* In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009. (Cited on pages 4 and 2.)

[Dhouib 2020a] Sofien Dhouib, Ievgen Redko, Tanguy Kerdoncuff, Rémi Emonet et Marc Sebban. *A swiss army knife for minimax optimal transport.* In International Conference on Machine Learning, pages 2504–2513, 2020. (Cited on page 138.)

[Dhouib 2020b] Sofien Dhouib, Ievgen Redko et Carole Lartizien. *Margin-aware Adversarial Domain Adaptation with Optimal Transport.* In Hal Daumé III et Aarti Singh, editeurs, Proceedings of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 2514–2524. PMLR, 13–18 Jul 2020. (Cited on page 35.)

[Duan 2018] Y. Duan, X. Tao, M. Xu, C. Han et J. Lu. *GAN-NL: Unsupervised Representation Learning for Remote Sensing Image Classification.* In 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 375–379, Nov 2018. (Cited on page 70.)

[Dubey 2018] A. Dubey, O. Gupta, P. Guo, R. Raskar, R. Farrell et N. Naik. *Pairwise Confusion for Fine-Grained Visual Classification.* In The European Conference on Computer Vision, 2018. (Cited on page 52.)

[Dudley 1969] R. M. Dudley. *The Speed of Mean Glivenko-Cantelli Convergence.* Ann. Math. Statist., vol. 40, no. 1, pages 40–50, 02 1969. (Cited on pages 5, 3 and 93.)

[Dvurechensky 2018] Pavel Dvurechensky, Alexander Gasnikov et Alexey Kroshnin. *Computational Optimal Transport: Complexity by Accelerated Gradient Descent Is Better Than by Sinkhorn's Algorithm.* In Jennifer Dy et Andreas Krause, editeurs, Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 1367–1376. PMLR, 10–15 Jul 2018. (Cited on page 21.)

[Fatras 2020a] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval et Nicolas Courty. *Divergence Wasserstein par lots.* In 52 èmes Journées de Statistiques de la Société Française de Statistique, 2020. (Cited on pages 8 and 5.)

[Fatras 2020b] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval et Nicolas Courty. *Learning with minibatch Wasserstein: asymptotic and gradient properties.* In AISTATS, 2020. (Cited on pages 7, 5 and 105.)

[Fatras 2021a] Kilian Fatras, Bharath Bushan, Sylvain Lobry, Remi Flamary, Devis Tuia et Nicolas Courty. *Wasserstein Adversarial Regularization for learning with label noise.* IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–1, 2021. (Cited on pages 6, 7, 4 and 51.)

[Fatras 2021b] Kilian Fatras, Thibault Séjourné, Nicolas Courty et Rémi Flamary. *Unbalanced minibatch Optimal Transport; applications to Domain Adaptation.* In Proceedings of the 38th International Conference on Machine Learning, 2021. (Cited on pages 7, 5, 29, 35, 92, 137 and 154.)

[Fatras 2021c] Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval et Nicolas Courty. *Minibatch optimal transport distances; analysis and applications.* CoRR, 2021. (Cited on pages 7, 5, 40 and 105.)

[Feinman 2017] Reuben Feinman, Ryan R Curtin, Saurabh Shintre et Andrew B Gardner. *Detecting adversarial samples from artifacts.* arXiv preprint arXiv:1703.00410, 2017. (Cited on page 46.)

[Fernando 2013] Basura Fernando, Amaury Habrard, Marc Sebban et Tinne Tuytelaars. *Unsupervised Visual Domain Adaptation Using Subspace Alignment.* In 2013 IEEE International Conference on Computer Vision, pages 2960–2967, 2013. (Cited on page 34.)

[Ferradans 2013] Sira Ferradans, Nicolas Papadakis, Julien Rabin, Gabriel Peyré et Jean-François Aujol. *Regularized Discrete Optimal Transport.* In Scale Space and Variational Methods in Computer Vision. Springer Berlin Heidelberg, 2013. (Cited on page 129.)

[Feydy 2019] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve et Gabriel Peyré. *Interpolating between Optimal Transport and MMD using Sinkhorn Divergences.* In Kamalika Chaudhuri et Masashi Sugiyama, editeurs, Proceedings of Machine Learning Research, volume 89 of *Proceedings of Machine Learning Research*, pages 2681–2690. PMLR, 16–18 Apr 2019. (Cited on pages 23, 24, 25, 93, 125 and 142.)

[Figalli 2010a] Alessio Figalli. *The Optimal Partial Transport Problem.* Archive for Rational Mechanics and Analysis, vol. 195, pages 533–560, 02 2010. (Cited on page 27.)

[Figalli 2010b] Alessio Figalli et Nicola Gigli. *A new transportation distance between non-negative measures, with applications to gradients flows with Dirichlet boundary conditions.* Journal de mathématiques pures et appliquées, vol. 94, no. 2, pages 107–130, 2010. (Cited on page 27.)

[Finlay 2020] Chris Finlay, Augusto Gerolin, Adam M Oberman et Aram-Alexandre Pooladian. *Learning normalizing flows from Entropy-Kantorovich potentials*, 2020. (Cited on page 153.)

[Flamary 2021] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong et Titouan Vayer. *POT: Python Optimal Transport.* Journal of Machine Learning Research, vol. 22, no. 78, pages 1–8, 2021. (Cited on pages 8, 6, 22, 142 and 187.)

[Forrow 2019] Aden Forrow, Jan-Christian Hütter, Mor Nitzan, Philippe Rigollet, Geoffrey Schiebinger et Jonathan Weed. *Statistical Optimal Transport via Factored Couplings.* In Kamalika Chaudhuri et Masashi Sugiyama, editeurs, Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, volume 89 of *Proceedings of Machine Learning Research*, pages 2454–2465. PMLR, 16–18 Apr 2019. (Cited on pages 25 and 91.)

[Frisch 2002] Uriel Frisch, Sabino Matarrese, Roya Mohayaee et Andrei Sobolevski. *A reconstruction of the initial conditions of the Universe by optimal mass transportation.* Nature, vol. 417, no. 6886, page 260–262, May 2002. (Cited on pages 5 and 3.)

[Frogner 2015] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya et Tomaso A Poggio. *Learning with a Wasserstein Loss.* In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama et R. Garnett, editeurs, Advances in Neural Information Processing Systems 28, pages 2053–2061. Curran Associates, Inc., 2015. (Cited on pages 5, 3, 27, 32, 55, 56 and 151.)

[Ganin 2015] Yaroslav Ganin et Victor Lempitsky. *Unsupervised Domain Adaptation by Backpropagation.* In Francis Bach et David Blei, editeurs, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR. (Cited on pages 4 and 34.)

[Ganin 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March et Victor Lempitsky. *Domain-Adversarial Training of Neural Networks*. Journal of Machine Learning Research, vol. 17, no. 59, pages 1–35, 2016. (Cited on pages 2, 34 and 144.)

[Genevay 2016] Aude Genevay, Marco Cuturi, Gabriel Peyré et Francis Bach. *Stochastic Optimization for Large-scale Optimal Transport*. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon et R. Garnett, editeurs, Advances in Neural Information Processing Systems 29, pages 3440–3448. Curran Associates, Inc., 2016. (Cited on pages 22 and 123.)

[Genevay 2018] Aude Genevay, Gabriel Peyre et Marco Cuturi. *Learning Generative Models with Sinkhorn Divergences*. In Amos Storkey et Fernando Perez-Cruz, editeurs, Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, volume 84 of *Proceedings of Machine Learning Research*, pages 1608–1617, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR. (Cited on pages 5, 7, 3, 14, 21, 25, 39, 40, 55, 92, 95, 123, 128, 129, 138, 152 and 153.)

[Genevay 2019] Aude Genevay, Lénaïc Chizat, Francis Bach, Marco Cuturi et Gabriel Peyré. *Sample Complexity of Sinkhorn Divergences*. In Kamalika Chaudhuri et Masashi Sugiyama, editeurs, Proceedings of Machine Learning Research, volume 89 of *Proceedings of Machine Learning Research*, pages 1574–1583. PMLR, 16–18 Apr 2019. (Cited on pages 5, 3, 93 and 94.)

[Gerber 2017] Samuel Gerber et Mauro Maggioni. *Multiscale Strategies for Computing Optimal Transport*. Journal of Machine Learning Research, 2017. (Cited on page 91.)

[Germain 2013] Pascal Germain, Amaury Habrard, François Laviolette et Emilie Morvant. *A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear Classifiers*. In Sanjoy Dasgupta et David McAllester, editeurs, Proceedings of the 30th International Conference on Machine Learning, volume 28 of *Proceedings of Machine Learning Research*, pages 738–746, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. (Cited on page 34.)

[Ghosh 2017] A. Ghosh, H. Kumar et P.S. Sastry. *Robust Loss Functions under Label Noise for Deep Neural Networks*. In Association for the Advancement of Artificial Intelligence, 2017. (Cited on page 52.)

[Gidel 2019a] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent et Simon Lacoste-Julien. *A Variational Inequality Perspective on Generative Adversarial Networks*. In International Conference on Learning Representations, 2019. (Cited on page 39.)

[Gidel 2019b] Gauthier Gidel, Reyhane Askari Hemmat, Mohammad Pezeshki, Rémi Le Priol, Gabriel Huang, Simon Lacoste-Julien et Ioannis Mitliagkas. *Negative Momentum for Improved Game Dynamics*. In Kamalika Chaudhuri et Masashi Sugiyama, editeurs, Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, volume 89 of *Proceedings of Machine Learning Research*, pages 1802–1811. PMLR, 16–18 Apr 2019. (Cited on page 39.)

[Goibert 2019] M. Goibert et E. Dohmatob. *Adversarial Robustness via Adversarial Label-Smoothing*. arXiv, vol. 1906.11567, 2019. (Cited on page 54.)

[Golub 2000] Gene H. Golub et Henk A. van der Vorst. *Eigenvalue computation in the 20th century.* Journal of Computational and Applied Mathematics, vol. 123, no. 1, pages 35 – 65, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra. (Cited on page 49.)

[Gong 2012] Boqing Gong, Yuan Shi, Fei Sha et Kristen Grauman. *Geodesic flow kernel for unsupervised domain adaptation.* In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 2066–2073, 2012. (Cited on page 34.)

[Gong 2016] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour et Bernhard Schölkopf. *Domain Adaptation with Conditional Transferable Components.* In Maria Florina Balcan et Kilian Q. Weinberger, editeurs, Proceedings of The 33rd International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, pages 2839–2848, New York, New York, USA, 20–22 Jun 2016. PMLR. (Cited on page 34.)

[Goodfellow 2014a] Ian Goodfellow*et al. Generative adversarial nets.* In Advances in neural information processing systems, 2014. (Cited on pages 39 and 46.)

[Goodfellow 2014b] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville et Yoshua Bengio. *Generative Adversarial Nets.* In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence et K. Q. Weinberger, editeurs, Advances in Neural Information Processing Systems 27, pages 2672–2680. Curran Associates, Inc., 2014. (Cited on pages 5, 6 and 4.)

[Goodfellow 2015] I. Goodfellow, J. Shlens et C. Szegedy. *Explaining and Harnessing Adversarial Examples.* In The International Conference on Learning Representations, 2015. (Cited on pages 45, 46, 47, 48, 53 and 188.)

[Gretton 2012] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf et Alexander Smola. *A Kernel Two-Sample Test.* Journal of Machine Learning Research, vol. 13, no. 25, pages 723–773, 2012. (Cited on pages 3, 24, 93, 95 and 115.)

[grs] 2018 IEEE GRSS Data Fusion Contest. Online: `http://www.grss-ieee.org/community/technical-committees/data-fusion`. (Cited on page 75.)

[Gulrajani 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin et Aaron C Courville. *Improved Training of Wasserstein GANs.* In Advances in Neural Information Processing Systems 30. 2017. (Cited on pages 6, 4, 23, 39, 40, 92, 129 and 151.)

[Haeusser 2017] Philip Haeusser, Thomas Frerix, Alexander Mordvintsev et Daniel Cremers. *Associative Domain Adaptation.* In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2784–2792, 2017. (Cited on page 34.)

[Han 2018] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang et M. Sugiyama. *Co-teaching: Robust training of deep neural networks with extremely noisy labels.* In Advances in Neural Information Processing Systems, 2018. (Cited on pages 52, 59 and 60.)

[Hanin 1992] L. Hanin. *Kantorovich-Rubinstein norm and its application in the theory of Lipschitz spaces.* 1992. (Cited on page 27.)

[Harper 2017] Marc Harper, Bryan Weinstein, tgwoodcock, Cory Simon, chebee7i, Wiley Morgan, Vince Knight, Nick Swanson-Hysell, Matthew Evans, jl bernal, The Gitter Badger, SaxonAnglo, Maximiliano Greco et Guido Zuidhof. *marcharper/python-ternary: New Features and Bug Fixes*, Août 2017. (Cited on page 118.)

[He 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren et Jian Sun. *Deep Residual Learning for Image Recognition*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. (Cited on pages 4, 2 and 61.)

[Hearst 1998] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt et Bernhard Scholkopf. *Support vector machines*. IEEE Intelligent Systems and their applications, vol. 13, no. 4, pages 18–28, 1998. (Cited on page 73.)

[Heitz 2020] Matthieu Heitz, Nicolas Bonneel, David Coeurjolly, Marco Cuturi et Gabriel Peyré. *Ground Metric Learning on Graphs*. Journal of Mathematical Imaging and Vision, vol. 63, no. 1, page 89–107, Oct 2020. (Cited on page 152.)

[Hendrycks 2017] Dan Hendrycks et Kevin Gimpel. *Early Methods for Detecting Adversarial Images*. arXiv: Learning, 2017. (Cited on page 46.)

[Hendrycks 2018] D. Hendrycks, M. Mazeika, D. Wilson et K. Gimpel. *Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise*. In Advances in Neural Information Processing Systems, 2018. (Cited on page 52.)

[Hoeffding 1948] Wassily Hoeffding. *A Class of Statistics with Asymptotically Normal Distribution*. The Annals of Mathematical Statistics, vol. 19, no. 3, pages 293 – 325, 1948. (Cited on pages 9, 7 and 99.)

[Hoeffding 1963] Wassily Hoeffding. *Probability Inequalities for Sums of Bounded Random Variables*. Journal of the American Statistical Association, vol. 58, no. 301, pages 13–30, March 1963. (Cited on pages 9, 7, 99, 101, 102, 120, 161, 162 and 163.)

[Hongxin 2020] Wei Hongxin, Feng Lei, Chen Xiangyu et An Bo. *Combating Noisy Labels by Agreement: A Joint Training Method with Co-Regularization*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. (Cited on page 52.)

[Hou 2017] Xianxu Hou, Linlin Shen, Ke Sun et Guoping Qiu. *Deep Feature Consistent Variational Autoencoder*. In Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, pages 1133–1141. IEEE, 2017. (Cited on pages 125, 126 and 127.)

[Hu 2015] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang et Hengchao Li. *Deep convolutional neural networks for hyperspectral image classification*. Journal of Sensors, vol. 2015, 2015. (Cited on page 76.)

[Huang 2016] Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha et Kilian Q Weinberger. *Supervised word mover's distance*. In Advances in Neural Information Processing Systems, pages 4862–4870, 2016. (Cited on page 56.)

[Huang 2021] Chin-Wei Huang, Ricky T. Q. Chen, Christos Tsirigotis et Aaron Courville. *Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization*. In International Conference on Learning Representations, 2021. (Cited on page 153.)

[Hull 1994] Jonathan Hull. *Database for handwritten text recognition research.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 16, 1994. (Cited on page 144.)

[Isola 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou et Alexei A. Efros. *Image-to-Image Translation with Conditional Adversarial Networks.* In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5967–5976, 2017. (Cited on page 41.)

[J Lee 2019] A J Lee. *U-statistics : theory and practice / A. J. Lee.* SERBIULA (sistema Librum 2.0), 06 2019. (Cited on pages 99, 112 and 115.)

[Janati 2020] Hicham Janati, Boris Muzellec, Gabriel Peyré et Marco Cuturi. *Entropic Optimal Transport between Unbalanced Gaussian Measures has a Closed Form.* In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan et H. Lin, editeurs, Advances in Neural Information Processing Systems, volume 33, pages 10468–10479. Curran Associates, Inc., 2020. (Cited on pages 142 and 153.)

[Jian 2020] Liang Jian, Wang Yunbo, Hu Dapeng, He Ran et Feng Jiashi. *A Balanced and Uncertainty-aware Approach for Partial Domain Adaptation.* In European Conference on Computer Vision (ECCV), August 2020. (Cited on page 147.)

[Jiang 2018] L. Jiang, Z. Zhou, T. Leung, L.-J. Li et F.-F. Li. *MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels.* In The International Conference on Machine Learning, 2018. (Cited on page 52.)

[Jiang 2019] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu et J. Jiang. *Edge-Enhanced GAN for Remote Sensing Image Superresolution.* IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 8, pages 5799–5812, Aug 2019. (Cited on page 70.)

[Jin 2019] Chi Jin, Praneeth Netrapalli, R. Ge, Sham M. Kakade et Michael I. Jordan. *A Short Note on Concentration Inequalities for Random Vectors with SubGaussian Norm.* ArXiv, vol. abs/1902.03736, 2019. (Cited on page 167.)

[Jo 2019] Youngjoo Jo et Jongyoul Park. *SC-FEGAN: Face Editing Generative Adversarial Network with User's Sketch and Color.* arXiv preprint arXiv:1902.06838, 2019. (Cited on pages 40 and 82.)

[Johnson 2013] Rie Johnson et Tong Zhang. *Accelerating Stochastic Gradient Descent using Predictive Variance Reduction.* In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani et K. Q. Weinberger, editeurs, Advances in Neural Information Processing Systems 26, pages 315–323. Curran Associates, Inc., 2013. (Cited on pages 21 and 22.)

[Jumper 2021] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin **v**Zídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli et Demis Hassabis. *Highly accurate protein structure prediction with AlphaFold.* Nature, 2021. (Accelerated article preview). (Cited on pages 4 and 2.)

[Kang 2006] Feng Kang, Rong Jin et R. Sukthankar. *Correlated Label Propagation with Application to Multi-label Learning.* In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 1719–1726, 2006. (Cited on page 32.)

[Kantorovich 1942] L. Kantorovich. *On the translocation of masses.* C.R. (Doklady) Acad. Sci. URSS (N.S.), vol. 37, pages 199–201, 1942. (Cited on page 14.)

[Karras 2019] Tero Karras, Samuli Laine et Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks.* In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4396–4405, 2019. (Cited on page 40.)

[Kerdoncuff 2020] Tanguy Kerdoncuff, Rémi Emonet et Marc Sebban. *Metric Learning in Optimal Transport for Domain Adaptation.* In Christian Bessiere, editeur, Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pages 2162–2168. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track. (Cited on page 35.)

[Kingma 2014] Diederik P. Kingma et Max Welling. *Auto-Encoding Variational Bayes.* In Yoshua Bengio et Yann LeCun, editeurs, 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014. (Cited on pages 36, 37 and 125.)

[Kingma 2015] Diederik P. Kingma et Jimmy Ba. *Adam: A Method for Stochastic Optimization.* In Yoshua Bengio et Yann LeCun, editeurs, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. (Cited on pages 4, 2 and 127.)

[Klenke 2008] Achim Klenke. Probability theory: A comprehensive course. Springer, 2008. (Cited on page 96.)

[Kolouri 2016] Soheil Kolouri, Yang Zou et Gustavo K Rohde. *Sliced Wasserstein kernels for probability distributions.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. (Cited on pages 18 and 92.)

[Kolouri 2017] Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev et Gustavo K. Rohde. *Optimal Mass Transport: Signal processing and machine-learning applications.* IEEE Signal Processing Magazine, vol. 34, no. 4, pages 43–59, 2017. (Cited on pages 5 and 3.)

[Kolouri 2018] Soheil Kolouri, Gustavo Kunde Rohde et Heiko Hoffmann. *Sliced Wasserstein Distance for Learning Gaussian Mixture Models.* 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3427–3436, 2018. (Cited on pages 18 and 92.)

[Kolouri 2019a] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau et Gustavo Rohde. *Generalized Sliced Wasserstein Distances.* In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox et R. Garnett, editeurs, Advances in Neural Information Processing Systems 32, pages 261–272. Curran Associates, Inc., 2019. (Cited on pages 18 and 92.)

[Kolouri 2019b] Soheil Kolouri, Phillip E. Pope, Charles E. Martin et Gustavo K. Rohde. *Sliced Wasserstein Auto-Encoders.* In International Conference on Learning Representations, 2019. (Cited on page 38.)

[Krause 2016] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin et F.-F. Li. *The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition.* The European Conference on Computer Vision, 2016. (Cited on page 52.)

[Krizhevsky] Alex Krizhevsky, Vinod Nair et Geoffrey Hinton. *CIFAR-10 (Canadian Institute for Advanced Research).* (Cited on page 129.)

[Krizhevsky 2009] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Rapport technique, University of Toronto, 2009. (Cited on page 59.)

[Krizhevsky 2012] Alex Krizhevsky, Ilya Sutskever et Geoffrey E Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. In F. Pereira, C. J. C. Burges, L. Bottou et K. Q. Weinberger, editeurs, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012. (Cited on pages 4 and 2.)

[Kuhn 2019] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen et Soroosh Shafieezadeh Abadeh. *Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning*. INFORMS TutORials in Operations Research, 2019. (Cited on page 138.)

[Kulis 2013a] Brian Kulis. *Metric Learning: A Survey*. Foundations and Trends® in Machine Learning, vol. 5, no. 4, pages 287–364, 2013. (Cited on page 128.)

[Kulis 2013b] Brian Kulis*et al. Metric learning: A survey*. Foundations and Trends® in Machine Learning, vol. 5, no. 4, pages 287–364, 2013. (Cited on page 100.)

[Kun 2019a] Yi Kun et Wu Jianxin. *Probabilistic End-to-end Noise Correction for Learning with Noisy Labels*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. (Cited on page 52.)

[Kun 2019b] Yi Kun et Wu Jianxin. *Probabilistic End-to-end Noise Correction for Learning with Noisy Labels*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. (Cited on pages 59 and 63.)

[Le 2019] Tam Le, Makoto Yamada, Kenji Fukumizu et Marco Cuturi. *Tree-Sliced Variants of Wasserstein Distances*. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox et R. Garnett, editeurs, Advances in Neural Information Processing Systems 32, pages 12304–12315. Curran Associates, Inc., 2019. (Cited on page 26.)

[LeCun 1989] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard et L. D. Jackel. *Backpropagation Applied to Handwritten Zip Code Recognition*. Neural Computation, vol. 1, no. 4, pages 541–551, 12 1989. (Cited on pages 3 and 2.)

[Lecun 1998] Y. Lecun, L. Bottou, Y. Bengio et P. Haffner. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, vol. 86, no. 11, pages 2278–2324, 1998. (Cited on pages 3 and 2.)

[LeCun 2010] Yann LeCun et Corinna Cortes. *MNIST handwritten digit database*. 2010. (Cited on page 144.)

[Ledig 2017] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang et W. Shi. *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 105–114, July 2017. (Cited on pages 40, 41 and 188.)

[Lee 2018] K.-H. Lee, X. He, L. Zhang et L. Yang. *CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise*. In The IEEE Conference on Computer Vision and Pattern Recognition, 2018. (Cited on page 52.)

[Li 2017a] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang et Barnabás Póczos. *MMD GAN: Towards Deeper Understanding of Moment Matching Network*. arXiv preprint arXiv:1705.08584, 2017. (Cited on pages 39, 128 and 129.)

[Li 2017b] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo et L.-J. Li. *Learning from Noisy Labels with Distillation*. In The International Conference on Computer Vision, 2017. (Cited on page 52.)

[Li 2019] Ruilin Li, Xiaojing Ye, Haomin Zhou et Hongyuan Zha. *Learning to Match via Inverse Optimal Transport*. Journal of Machine Learning Research, vol. 20, no. 80, pages 1–37, 2019. (Cited on page 56.)

[Liero 2017] Matthias Liero, Alexander Mielke et Giuseppe Savaré. *Optimal Entropy-Transport problems and a new Hellinger–Kantorovich distance between positive measures*. Inventiones mathematicae, vol. 211, no. 3, page 969–1117, Dec 2017. (Cited on pages 27, 28, 138, 178, 180 and 185.)

[Lin 2017] D. Lin, K. Fu, Y. Wang, G. Xu et X. Sun. *MARTA GANs: Unsupervised Representation Learning for Remote Sensing Image Classification*. IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 11, pages 2092–2096, Nov 2017. (Cited on page 70.)

[Liu 2014] Tongliang Liu et Dacheng Tao. *Classification with Noisy Labels by Importance Reweighting*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, 11 2014. (Cited on page 52.)

[Liu 2015] Ziwei Liu, Ping Luo, Xiaogang Wang et Xiaoou Tang. *Deep Learning Face Attributes in the Wild*. In Proceedings of International Conference on Computer Vision (ICCV), December 2015. (Cited on pages 125 and 127.)

[Liu 2017] Ming-Yu Liu, Thomas Breuel et Jan Kautz. *Unsupervised Image-to-Image Translation Networks*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. (Cited on page 34.)

[Liu 2018] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao et Bryan Catanzaro. *Image Inpainting for Irregular Holes Using Partial Convolutions*, 2018. (Cited on page 40.)

[Liutkus 2019] Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus et Fabian-Robert Stöter. *Sliced-Wasserstein Flows: Nonparametric Generative Modeling via Optimal Transport and Diffusions*. In Proceedings of the 36th International Conference on Machine Learning, 2019. (Cited on pages 18, 92, 125 and 142.)

[Liwei Wang 2005] Liwei Wang, Yan Zhang et Jufu Feng. *On the Euclidean distance of images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pages 1334–1339, 2005. (Cited on page 128.)

[Long 2014] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun et Philip S. Yu. *Transfer Joint Matching for Unsupervised Domain Adaptation*. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 1410–1417, 2014. (Cited on page 34.)

[Long 2015a] Jonathan Long, Evan Shelhamer et Trevor Darrell. *Fully convolutional networks for semantic segmentation*. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3431–3440, 2015. (Cited on page 32.)

[Long 2015b] Mingsheng Long, Yue Cao, Jianmin Wang et Michael I. Jordan. *Learning Transferable Features with Deep Adaptation Networks.* In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, pages 97–105, 2015. (Cited on page 34.)

[Long 2018] Mingsheng Long, Zhangjie Cao, Jianmin Wang et Michael I Jordan. *Conditional adversarial domain adaptation.* In Advances in Neural Information Processing Systems, pages 1645–1655, 2018. (Cited on pages 34, 144 and 145.)

[Luise 2018] G. Luise, A. Rudi, M. Pontil et C. Ciliberto. *Differential Properties of Sinkhorn Approximation for Learning with Wasserstein Distance.* In Advances in Neural Information Processing Systems, 2018. (Cited on pages 55 and 57.)

[Luo 2017] Zelun Luo, Yuliang Zou, Judy Hoffman et Li F Fei-Fei. *Label Efficient Learning of Transferable Representations acrosss Domains and Tasks.* In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. (Cited on page 34.)

[Ma 2018] X. Ma, Y. Wang, M.E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema et J. Bailey. *Dimensionality-Driven Learning with Noisy Labels.* In The International Conference on Machine Learning, 2018. (Cited on pages 52, 59 and 63.)

[Majewski 2018] Szymon Majewski, Błażej Miasojedow et Eric Moulines. *Analysis of nonsmooth stochastic approximation: the differential inclusion approach.* arXiv preprint arXiv:1805.01916, 2018. (Cited on pages 124 and 141.)

[Makhzani 2016] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly et Ian Goodfellow. *Adversarial Autoencoders.* In International Conference on Learning Representations, 2016. (Cited on page 37.)

[Mémoli 2011] Facundo Mémoli. *Gromov–Wasserstein Distances and the Metric Approach to Object Matching.* Found. Comput. Math., vol. 11, no. 4, page 417–487, Août 2011. (Cited on pages 5, 3, 25 and 93.)

[Mena 2019] Gonzalo Mena et Jonathan Niles-Weed. *Statistical bounds for entropic optimal transport: sample complexity and the central limit theorem.* In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox et R. Garnett, editeurs, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. (Cited on pages 5, 3 and 94.)

[Mengxue 2020] Li Mengxue, Zhai Yi Ming, Luo You Wei, Ge Peng Fei et Chuan Xian Ren. *Enhanced Transport Distance for Unsupervised Domain Adaptation.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. (Cited on page 35.)

[Menon 2015] Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong et Bob Williamson. *Learning from Corrupted Binary Labels via Class-Probability Estimation.* In The International Conference on Machine Learning, 2015. (Cited on page 52.)

[Mensch 2020] Arthur Mensch et Gabriel Peyré. *Online Sinkhorn: Optimal Transport distances from sample streams.* In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan et H. Lin, editeurs, Advances in Neural Information Processing Systems, volume 33, pages 1657–1667. Curran Associates, Inc., 2020. (Cited on page 92.)

[Mescheder 2017] Lars Mescheder, Sebastian Nowozin et Andreas Geiger. *Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks*. In International Conference on Machine Learning (ICML), 2017. (Cited on page 37.)

[Mikolov 2013] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado et J. Dean. *Distributed Representations of Words and Phrases and their Compositionality*. In Advances in Neural Information Processing Systems, 2013. (Cited on pages 32 and 55.)

[Mirza 2014] Mehdi Mirza et Simon Osindero. *Conditional Generative Adversarial Nets*, 2014. (Cited on page 39.)

[Misra 2016] I. Misra, C.L. Zitnick, M. Mitchell et R. Girshick. *Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels*. In The IEEE Conference on Computer Vision and Pattern Recognition, 2016. (Cited on page 52.)

[Miyato 2018a] T. Miyato, S. Maeda, S. Ishii et M. Koyama. *Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. (Cited on pages 6, 4, 48, 49, 53, 58 and 60.)

[Miyato 2018b] Takeru Miyato, Toshiki Kataoka, Masanori Koyama et Yuichi Yoshida. *Spectral Normalization for Generative Adversarial Networks*. In International Conference on Learning Representations, 2018. (Cited on page 39.)

[Mohajerin Esfahani 2018] Peyman Mohajerin Esfahani et Daniel Kuhn. *Data-Driven Distributionally Robust Optimization Using the Wasserstein Metric: Performance Guarantees and Tractable Reformulations*. Math. Program., vol. 171, no. 1–2, page 115–166, Septembre 2018. (Cited on page 138.)

[Monge 1781] Gaspard Monge. *Mémoire sur la théorie des déblais et des remblais*. Histoire de l'Académie Royale des Sciences, pages 666–704, 1781. (Cited on pages 5 and 3.)

[Mroueh 2018] Y. Mroueh, C.-L. Li, T. Sercu, A. Raj et Y. Cheng. *Sobolev GAN*. In International Conference on Learning Representations, 2018. (Cited on page 57.)

[Mukherjee 2020] Debarghya Mukherjee, Aritra Guha, Justin Solomon, Yuekai Sun et Mikhail Yurochkin. *Outlier-Robust Optimal Transport*. CoRR, 2020. (Cited on page 138.)

[Müller 2019] Rafael Müller, Simon Kornblith et Geoffrey E Hinton. *When does label smoothing help?* In Advances in Neural Information Processing Systems, 2019. (Cited on pages 54 and 147.)

[Murez 2018] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi et Kyungnam Kim. *Image to Image Translation for Domain Adaptation*. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4500–4509, 2018. (Cited on pages 34 and 41.)

[Muzellec 2020] Boris Muzellec, Julie Josse, Claire Boyer et Marco Cuturi. *Missing Data Imputation using Optimal Transport*. In Hal Daumé III et Aarti Singh, editeurs, Proceedings of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 7130–7140. PMLR, 13–18 Jul 2020. (Cited on pages 7, 5 and 92.)

[Natarajan 2013] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar et Ambuj Tewari. *Learning with Noisy Labels*. In Advances in Neural Information Processing Systems 26. 2013. (Cited on page 52.)

[Nath 2020] J. Saketha Nath. *Unbalanced Optimal Transport using Integral Probability Metric Regularization*. CoRR, 2020. (Cited on pages 27 and 138.)

[Netzer 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu et Andrew Y. Ng. *Reading Digits in Natural Images with Unsupervised Feature Learning*. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, 2011. (Cited on page 144.)

[Nguyen 2017] Lam M. Nguyen, Jie Liu, Katya Scheinberg et Martin Takáve. *SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient*. In Doina Precup et Yee Whye Teh, editeurs, Proceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, pages 2613–2621. PMLR, 06–11 Aug 2017. (Cited on page 21.)

[Nguyen 2021] Khai Nguyen, Quoc Nguyen, Nhat Ho, Tung Pham, Hung Bui, Dinh Phung et Trung Le. *BoMb-OT: On Batch of Mini-batches Optimal Transport*, 2021. (Cited on page 153.)

[Nowozin 2016] Sebastian Nowozin, Botond Cseke et Ryota Tomioka. *f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization*. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon et R. Garnett, editeurs, Advances in Neural Information Processing Systems 29, pages 271–279. Curran Associates, Inc., 2016. (Cited on page 39.)

[Odena 2017] Augustus Odena, Christopher Olah et Jonathon Shlens. *Conditional Image Synthesis with Auxiliary Classifier GANs*. In Doina Precup et Yee Whye Teh, editeurs, Proceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, pages 2642–2651, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. (Cited on pages 39 and 47.)

[Onken 2021] Derek Onken, Samy Wu Fung, Xingjian Li et Lars Ruthotto. *OT-Flow: Fast and Accurate Continuous Normalizing Flows via Optimal Transport*. In AAAI Conference on Artificial Intelligence, volume 35, pages 9223–9232, May 2021. (Cited on page 153.)

[Oord 2016] Aaron Van Oord, Nal Kalchbrenner et Koray Kavukcuoglu. *Pixel Recurrent Neural Networks*. In Maria Florina Balcan et Kilian Q. Weinberger, editeurs, Proceedings of The 33rd International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, pages 1747–1756, New York, New York, USA, 20–22 Jun 2016. PMLR. (Cited on page 65.)

[Pan 2010] Sinno Jialin Pan et Qiang Yang. *A Survey on Transfer Learning*. IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pages 1345–1359, 2010. (Cited on page 33.)

[Pang 2018] Tianyu Pang, Chao Du, Yinpeng Dong et Jun Zhu. *Towards robust detection of adversarial examples*. In Advances in Neural Information Processing Systems, pages 4579–4589, 2018. (Cited on page 46.)

[Papa 2015] Guillaume Papa, Stéphan Clémençon et Aurélien Bellet. *SGD Algorithms based on Incomplete U-statistics: Large-Scale Minimization of Empirical Risk*. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama et R. Garnett, editeurs, Advances in Neural Information Processing Systems 28, pages 1027–1035. Curran Associates, Inc., 2015. (Cited on page 124.)

[Papernot 2016] N. Papernot, P. McDaniel, X. Wu, S. Jha et A. Swami. *Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks*. pages 582–597, 2016. cited By 507. (Cited on page 45.)

[Park 2019] Taesung Park, Ming-Yu Liu, Ting-Chun Wang et Jun-Yan Zhu. *Semantic Image Synthesis with Spatially-Adaptive Normalization*, 2019. (Cited on page 40.)

[Paszke 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga et Adam Lerer. *Automatic differentiation in PyTorch*. 2017. (Cited on pages 4, 2 and 142.)

[Patrini 2017] G. Patrini, A. Rozza, Aditya Krishna M., R. Nock et L. Qu. *Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach*. In The IEEE Conference on Computer Vision and Pattern Recognition, 2017. (Cited on pages 52, 59, 60 and 63.)

[Patrini 2019] Giorgio Patrini, Marcello Carioni, Patrick Forré, Samarth Bhargav, Max Welling, Rianne van den Berg, Tim Genewein et Frank Nielsen. *Sinkhorn AutoEncoders*. In UAI, 2019. (Cited on page 38.)

[Paty 2019] François-Pierre Paty et Marco Cuturi. *Subspace Robust Wasserstein Distances*. In International Conference on Machine Learning, pages 5072–5081, 2019. (Cited on page 138.)

[Pedregosa 2011a] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot et E. Duchesnay. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, vol. 12, 2011. (Cited on page 57.)

[Pedregosa 2011b] Fabian Pedregosa*et al.* *Scikit-learn: Machine learning in Python*. Journal of machine learning research, vol. 12, no. Oct, pages 2825–2830, 2011. (Cited on pages 72 and 132.)

[Pedregosa 2019] Fabian Pedregosa, Kilian Fatras et Mattia Casotto. *Proximal Splitting Meets Variance Reduction*. In Kamalika Chaudhuri et Masashi Sugiyama, editeurs, Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, volume 89 of *Proceedings of Machine Learning Research*, pages 1–10. PMLR, 16–18 Apr 2019. (Cited on pages 8 and 6.)

[Peng 2017] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang et Kate Saenko. *VisDA: The Visual Domain Adaptation Challenge*. CoRR, vol. abs/1710.06924, 2017. (Cited on page 144.)

[Peyré 2015] G. Peyré. *Entropic Approximation of Wasserstein Gradient Flows*. SIAM Journal on Imaging Sciences, 2015. (Cited on pages 125 and 142.)

[Peyré 2016] Gabriel Peyré, Marco Cuturi et Justin Solomon. *Gromov-Wasserstein Averaging of Kernel and Distance Matrices*. In Maria Florina Balcan et Kilian Q. Weinberger, editeurs, Proceedings of The 33rd International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, pages 2664–2672, New York, New York, USA, 20–22 Jun 2016. PMLR. (Cited on pages 26 and 133.)

[Peyré 2019] Gabriel Peyré et Marco Cuturi. *Computational Optimal Transport*. Foundations and Trends® in Machine Learning, vol. 11, no. 5-6, pages 355–607, 2019. (Cited on pages 5, 3, 15, 16, 17, 18, 19, 20, 21, 23, 25, 55, 93, 111, 123 and 153.)

[Pham 2020] Khiem Pham, Khang Le, Nhat Ho, Tung Pham et Hung Bui. *On Unbalanced Optimal Transport: An Analysis of Sinkhorn Algorithm*. In Hal Daumé III et Aarti Singh, editeurs,

Proceedings of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 7673–7682. PMLR, 13–18 Jul 2020. (Cited on pages 21 and 28.)

[Piccoli 2014] Benedetto Piccoli et Francesco Rossi. *On properties of the Generalized Wasserstein distance*, 2014. (Cited on page 27.)

[pot] *ISPRS Potsdam 2D Semantic Labeling Dataset - Potsdam.* http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html. Accessed: 2020-11-10. (Cited on pages 81, 84, 86 and 189.)

[Rabin 2012] Julien Rabin, Gabriel Peyré, Julie Delon et Marc Bernot. *Wasserstein Barycenter and Its Application to Texture Mixing*. In Alfred M. Bruckstein, Bart M. ter Haar Romeny, Alexander M. Bronstein et Michael M. Bronstein, editeurs, Scale Space and Variational Methods in Computer Vision, pages 435–446, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. (Cited on pages 17 and 18.)

[Ramdas 2017] Aaditya Ramdas, Nicolás García Trillos et Marco Cuturi. *On Wasserstein Two-Sample Testing and Related Families of Nonparametric Tests*. Entropy, vol. 19, no. 2, 2017. (Cited on page 24.)

[Redko 2017] Ievgen Redko, Amaury Habrard et Marc Sebban. *Theoretical Analysis of Domain Adaptation with Optimal Transport*. In Michelangelo Ceci, Jaakko Hollmén, Ljup**v**co Todorovski, Celine Vens et Sa**v**so D**v**zeroski, editeurs, Machine Learning and Knowledge Discovery in Databases, pages 737–753, Cham, 2017. Springer International Publishing. (Cited on page 34.)

[Redmon 2018] Joseph Redmon et Ali Farhadi. *YOLOv3: An Incremental Improvement*, 2018. (Cited on page 85.)

[Reed 2015] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan et A. Rabinovich. *Training Deep Neural Networks on Noisy Labels with Bootstrapping*. In The International Conference on Learning Representations (Workshop), 2015. (Cited on pages 52 and 59.)

[Ren 2018] M. Ren, W. Zeng, B. Yang et R. Urtasun. *Learning to Reweight Examples for Robust Deep Learning*. In The International Conference on Machine Learning, 2018. (Cited on page 52.)

[Robbins 1951a] Herbert Robbins et Sutton Monro. *A Stochastic Approximation Method*. The Annals of Mathematical Statistics, vol. 22, no. 3, pages 400 – 407, 1951. (Cited on pages 2 and 21.)

[Robbins 1951b] Herbert Robbins et Sutton Monro. *A Stochastic Approximation Method*. The Annals of Mathematical Statistics, vol. 22, no. 3, pages 400 – 407, 1951. (Cited on page 95.)

[Ronneberger 2015] Olaf Ronneberger, Philipp Fischer et Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, page 234–241, 2015. (Cited on pages 4, 2, 65 and 81.)

[Rooyen 2015] B. Rooyen, A.K. Menon et R.C. Williamson. *Learning with Symmetric Label Noise: The Importance of Being Unhinged*. In Advances in Neural Information Processing Systems, 2015. (Cited on pages 59 and 63.)

[Rosenblatt 1958] F. Rosenblatt. *The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain*. Psychological Review, pages 65–386, 1958. (Cited on pages 3 and 2.)

[Rumelhart 1986] D. E. Rumelhart, G. E. Hinton et R. J. Williams. Learning internal representations by error propagation, page 318–362. MIT Press, Cambridge, MA, USA, 1986. (Cited on pages 3, 2 and 36.)

[Salimans 2018] Tim Salimans, Han Zhang, Alec Radford et Dimitris Metaxas. *Improving GANs Using Optimal Transport*. In International Conference on Learning Representations, 2018. (Cited on pages 5, 3, 40, 92, 117, 123 and 129.)

[Samangouei 2018] Pouya Samangouei, Maya Kabkab et Rama Chellappa. *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*. In International Conference on Learning Representations, 2018. (Cited on page 46.)

[Sankaranarayanan 2018] Swami Sankaranarayanan, Yogesh Balaji, Carlos D. Castillo et Rama Chellappa. *Generate to Adapt: Aligning Domains Using Generative Adversarial Networks*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. (Cited on page 34.)

[Santambrogio 2015] F. Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. 2015. (Cited on pages 5, 3, 14, 17 and 96.)

[Scetbon 2020] Meyer Scetbon et Marco Cuturi. *Linear Time Sinkhorn Divergences using Positive Features*. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan et H. Lin, editeurs, Advances in Neural Information Processing Systems, volume 33, pages 13468–13480. Curran Associates, Inc., 2020. (Cited on page 91.)

[Scetbon 2021] Meyer Scetbon, Marco Cuturi et Gabriel Peyré. *Low-Rank Sinkhorn Factorization*. In Marina Meila et Tong Zhang, editeurs, Proceedings of the 38th International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pages 9344–9354. PMLR, 18–24 Jul 2021. (Cited on page 91.)

[Schiebinger 2019] Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev et Eric S. Lander. *Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming*. Cell, vol. 176, no. 4, pages 928–943.e22, 2019. (Cited on pages 5 and 3.)

[Schmidt 2017a] Mark Schmidt, Nicolas Le Roux et Francis Bach. *Minimizing Finite Sums with the Stochastic Average Gradient*. Math. Program., vol. 162, no. 1–2, page 83–112, Mars 2017. (Cited on pages 21 and 22.)

[Schmidt 2017b] Mark Schmidt, Nicolas Le Roux et Francis Bach. *Minimizing finite sums with the stochastic average gradient*. Mathematical Programming, vol. 162, no. 1, pages 83–112, 2017. (Cited on page 124.)

[Schmitzer 2017] Bernhard Schmitzer et B. Wirth. *A Framework for Wasserstein-1-Type Metrics*. ArXiv, vol. abs/1701.01945, 2017. (Cited on page 27.)

[Schrodinger 1931] E. Schrodinger. *Uber die umkehrung der naturgesetze*. Sitzungsberichte Preuss. Akad. Wiss. Berlin. Phys. Math., 144, pages 876—879, 1931. (Cited on page 19.)

[Schroff 2011] F. Schroff, A. Criminisi et A. Zisserman. *Harvesting Image Databases from the Web*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011. (Cited on page 52.)

[Schütt 2017] Kristof T. Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R. Müller et Alexandre Tkatchenko. *Quantum-chemical insights from deep tensor neural networks*. Nature Communications, vol. 8, no. 1, Jan 2017. (Cited on pages 4 and 2.)

[Seguy 2018] Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet et Mathieu Blondel. *Large Scale Optimal Transport and Mapping Estimation*. In International Conference on Learning Representations, 2018. (Cited on pages 4, 22 and 123.)

[Séjourné 2019] Thibault Séjourné, Jean Feydy, François-Xavier Vialard, Alain Trouvé et Gabriel Peyré. *Sinkhorn Divergences for Unbalanced Optimal Transport*. arXiv preprint arXiv:1910.12958, 2019. (Cited on pages 28, 93 and 141.)

[Shafahi 2018] A. Shafahi, A. Ghiasi, F. Huang et T. Goldstein. *Label Smoothing and Logit Squeezing: A Replacement for Adversarial Training?* In The International Conference on Learning Representations, 2018. (Cited on page 54.)

[Shaham 2015] U. Shaham, Y. Yamada et S. Negahban. *Understanding Adversarial Training: Increasing Local Stability of Neural Nets through Robust Optimization*. Neurocomputing, 2015. (Cited on page 53.)

[Shen 2018] Jian Shen, Yanru Qu, Weinan Zhang et Yong Yu. *Wasserstein Distance Guided Representation Learning for Domain Adaptation*. In Sheila A. McIlraith et Kilian Q. Weinberger, editeurs, Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 4058–4065. AAAI Press, 2018. (Cited on page 35.)

[Si 2010] Si Si, Dacheng Tao et Bo Geng. *Bregman Divergence-Based Regularization for Transfer Subspace Learning*. IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 7, pages 929–942, 2010. (Cited on page 34.)

[Sinkhorn 1967] Richard Sinkhorn et Paul Knopp. *Concerning nonnegative matrices and doubly stochastic matrices*. Pacific J. Math., vol. 21, no. 2, pages 343–348, 1967. (Cited on page 21.)

[Solomon 2015] Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du et Leonidas Guibas. *Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains*. ACM Trans. Graph., vol. 34, no. 4, Juillet 2015. (Cited on page 133.)

[Solomon 2016] Justin Solomon, Gabriel Peyré, Vladimir G. Kim et Suvrit Sra. *Entropic Metric Alignment for Correspondence Problems*. ACM Trans. Graph., vol. 35, no. 4, Juillet 2016. (Cited on page 132.)

[Sommerfeld 2019] Max Sommerfeld, Jörn Schrieber, Yoav Zemel et Axel Munk. *Optimal Transport: Fast Probabilistic Approximation with Exact Solvers*. Journal of Machine Learning Research, vol. 20, no. 105, pages 1–23, 2019. (Cited on page 103.)

[Song 2018] Yang Song, Rui Shu, Nate Kushman et Stefano Ermon. *Constructing unrestricted adversarial examples with generative models*. In Advances in Neural Information Processing Systems, 2018. (Cited on pages 45, 46, 47, 74, 76, 78, 79, 84, 87 and 188.)

[Song 2019] H. Song, M Kim et J.-G. Lee. *SELFIE: Refurbishing Unclean Samples for Robust Deep Learning*. In The International Conference on Machine Learning, 2019. (Cited on page 52.)

[Sturm 2012] Karl-Theodor Sturm. *The space of spaces: curvature bounds and gradient flows on the space of metric measure spaces*. arXiv e-prints, page arXiv:1208.0434, 2012. (Cited on page 27.)

[Sugiyama 2008] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Buenau et Motoaki Kawanabe. *Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation*. In J. Platt, D. Koller, Y. Singer et S. Roweis, editeurs, Advances in Neural Information Processing Systems, volume 20. Curran Associates, Inc., 2008. (Cited on page 33.)

[Sukhbaatar 2014] S. Sukhbaatar et R. Fergus. *Learning from Noisy Labels with Deep Neural Networks*. In The International Conference on Learning Representations (Workshop), 2014. (Cited on page 52.)

[Sun 2016] Baochen Sun et Kate Saenko. *Deep CORAL: Correlation Alignment for Deep Domain Adaptation*. In Gang Hua et Hervé Jégou, editeurs, Computer Vision – ECCV 2016 Workshops, pages 443–450, Cham, 2016. Springer International Publishing. (Cited on page 34.)

[Szegedy 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow et Rob Fergus. *Intriguing properties of neural networks*, 2013. (Cited on pages 45 and 46.)

[Szegedy 2014] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke et Andrew Rabinovich. *Going Deeper with Convolutions*, 2014. (Cited on pages 46 and 188.)

[Séjourné 2020] Thibault Séjourné, François-Xavier Vialard et Gabriel Peyré. *The Unbalanced Gromov Wasserstein Distance: Conic Formulation and Relaxation*, 2020. (Not cited)

[Talagrand 1995] Michel Talagrand. *Concentration of measure and isoperimetric inequalities in product spaces*. Publications Mathématiques de l'Institut des Hautes Études Scientifiques, vol. 81, no. 1, pages 73–205, Dec 1995. (Cited on page 153.)

[Tanaka 2018a] D. Tanaka, D. Ikami, T. Yamasaki et K. Aizawa. *Joint Optimization Framework for Learning with Noisy Labels*. In The IEEE Conference on Computer Vision and Pattern Recognition, 2018. (Cited on page 52.)

[Tanaka 2018b] D. Tanaka, D. Ikami, T. Yamasaki et K. Aizawa. *Joint Optimization Framework for Learning with Noisy Labels*. In The IEEE Conference on Computer Vision and Pattern Recognition, 2018. (Cited on page 63.)

[Thibault 2017] Alexis Thibault, Lenaic Chizat, Charles H Dossal et Nicolas Papadakis. *Overrelaxed Sinkhorn-Knopp Algorithm for Regularized Optimal Transport*. In NIPS'17 Workshop on Optimal Transport and Machine Learning, Long Beach, United States, Décembre 2017. (Cited on page 21.)

[Tieleman 2012a] Tijmen Tieleman et Geoffrey Hinton. *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*. COURSERA: Neural networks for machine learning, vol. 4, no. 2, pages 26–31, 2012. (Cited on pages 4, 2 and 129.)

[Tieleman 2012b] Tijmen Tieleman et Geoffrey Hinton. *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*. COURSERA: Neural networks for machine learning, vol. 4, no. 2, pages 26–31, 2012. (Cited on page 72.)

[Tolstikhin 2018] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly et Bernhard Schoelkopf. *Wasserstein Auto-Encoders*. In International Conference on Learning Representations, 2018. (Cited on pages 3, 37, 38, 187 and 188.)

[Tzeng 2015] Eric Tzeng, Judy Hoffman, Trevor Darrell et Kate Saenko. *Simultaneous Deep Transfer Across Domains and Tasks*. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 4068–4076, 2015. (Cited on page 34.)

[Vahdat 2017] A. Vahdat. *Toward Robustness against Label Noise in Training Deep Discriminative Neural Networks*. In Advances in Neural Information Processing Systems, 2017. (Cited on page 52.)

[vai] *ISPRS Potsdam 2D Semantic Labeling Dataset - Vaihingen*. https://www2.isprs.org/commissions/comm2/wg4/benchmark/2d-sem-label-vaihingen/. Accessed: 2020-11-10. (Cited on page 81.)

[Vapnik 1998] Vladimir N. Vapnik. Statistical learning theory. Wiley-Interscience, 1998. (Cited on page 32.)

[Vayer 2019a] Titouan Vayer, Nicolas Courty, Romain Tavenard, Chapel Laetitia et Rémi Flamary. *Optimal Transport for structured data with application on graphs*. In Kamalika Chaudhuri et Ruslan Salakhutdinov, editeurs, Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 6275–6284. PMLR, 09–15 Jun 2019. (Cited on pages 5 and 3.)

[Vayer 2019b] Titouan Vayer, Rémi Flamary, Nicolas Courty, Romain Tavenard et Laetitia Chapel. *Sliced Gromov-Wasserstein*. In Advances in Neural Information Processing Systems 32, pages 14753–14763. Curran Associates, Inc., 2019. (Cited on pages 26, 132 and 133.)

[Venkateswara 2017] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty et Sethuraman Panchanathan. *Deep Hashing Network for Unsupervised Domain Adaptation*. In (IEEE) Conference on Computer Vision and Pattern Recognition (CVPR), 2017. (Cited on pages 33 and 144.)

[Villani 2009] Cédric Villani. Optimal transport : old and new / cédric villani. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, right 2009. (Cited on pages 5 and 3.)

[Voreiter 2020] Claire Voreiter, Jean-Christophe Burnel, Pierre Lassalle, Marc Spigai, Romain Hugues et Nicolas Courty. *A cycle gan approach for heterogeneous domain adaptation in land use classification*, 2020. (Cited on page 70.)

[Wang 2018a] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian et C. Change Loy. *The Devil of Face Recognition is in the Noise*. In The European Conference on Computer Vision, 2018. (Cited on page 52.)

[Wang 2018b] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz et Bryan Catanzaro. *Video-to-Video Synthesis*. In Conference on Neural Information Processing Systems (NeurIPS), 2018. (Cited on page 41.)

[Wang 2018c] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song et S. Xia. *Iterative Learning with Open-set Noisy Labels*. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8688–8696, 2018. (Cited on pages 65 and 66.)

[Wang 2019] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi et J. Bailey. *Symmetric Cross Entropy for Robust Learning with Noisy Labels*. In The International Conference on Computer Vision, 2019. (Cited on pages 52, 59 and 63.)

[Weed 2019] Jonathan Weed et Francis Bach. *Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance*. Bernoulli, 2019. (Cited on pages 5, 3, 93, 94 and 120.)

[Wei 2020] Hongxin Wei, Lei Feng, Xiangyu Chen et Bo An. *Combating noisy labels by agreement: A joint training method with co-regularization*. CVPR, 2020. (Cited on pages 59 and 63.)

[Wilson 1969] A. G. Wilson. *The Use of Entropy Maximising Models, in the Theory of Trip Distribution, Mode Split and Route Split*. Journal of Transport Economics and Policy, vol. 3, no. 1, pages 108–126, 1969. (Cited on page 19.)

[Xiao 2015] T. Xiao, T. Xia, Y. Yang, C. Huang et X. Wang. *Learning from Massive Noisy Labeled Data for Image Classification*. In The IEEE Conference on Computer Vision and Pattern Recognition, 2015. (Cited on pages 52 and 63.)

[Xiao 2017] H. Xiao, K. Rasul et R. Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv, vol. 1708.07747, 2017. (Cited on page 59.)

[Xu 2018] Yongyang Xu, Liang Wu, Zhong Xie et Zhanlong Chen. *Building extraction in very high resolution remote sensing imagery using deep learning and guided filters*. Remote Sensing, vol. 10, no. 1, page 144, 2018. (Cited on page 65.)

[Xu 2019] Yonghao Xu, Bo Du, Liangpei Zhang, Daniele Cerra, Miguel Pato, Emiliano Carmona, Saurabh Prasad, Naoto Yokoya, Ronny Hänsch et Bertrand Le Saux. *Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 IEEE GRSS Data Fusion Contest*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 6, pages 1709–1724, 2019. (Cited on page 75.)

[Xu 2020a] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen et Jindong Wang. *Reliable Weighted Optimal Transport for Unsupervised Domain Adaptation*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. (Cited on pages 35 and 92.)

[Xu 2020b] Renjun Xu, Pelen Liu, Yin Zhang, Fang Cai, Jindong Wang, Shuoying Liang, Heting Ying et Jianwei Yin. *Joint Partial Optimal Transport for Open Set Domain Adaptation*. In Christian Bessiere, editeur, Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pages 2540–2546. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track. (Cited on page 35.)

[Yu 2016] Fisher Yu et Vladlen Koltun. *Multi-Scale Context Aggregation by Dilated Convolutions*. In International Conference on Learning Representations, 2016. (Cited on page 82.)

[Yu 2019] X. Yu, B. Han, J. Yao, G. Niu, I.W. Tsang et M. Sugiyama. *How does Disagreement Help Generalization against Label Corruption?* In The International Conference on Machine Learning, 2019. (Cited on pages 52, 59, 61, 65 and 66.)

[Zhang 2013] Kai Zhang, Vincent Zheng, Qiaojun Wang, James Kwok, Qiang Yang et Ivan Marsic. *Covariate Shift in Hilbert Space: A Solution via Sorrogate Kernels*. In Sanjoy Dasgupta et David McAllester, editeurs, Proceedings of the 30th International Conference on Machine Learning, volume 28 of *Proceedings of Machine Learning Research*, pages 388–395, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. (Cited on pages 33 and 34.)

[Zhang 2017] C. Zhang, S. Bengio, M. Hardt, B. Recht et O. Vinyals. *Understanding deep learning requires rethinking generalization*. In The International Conference on Learning Representations, 2017. (Cited on pages 6, 4, 48, 52 and 66.)

[Zhang 2018a] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin et David Lopez-Paz. *mixup: Beyond Empirical Risk Minimization*. In International Conference on Learning Representations, 2018. (Cited on pages 45 and 48.)

[Zhang 2018b] Z. Zhang et M.R. Sabuncu. *Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels*. In Advances in Neural Information Processing Systems, 2018. (Cited on pages 52 and 63.)

[Zhang 2019] Yuchen Zhang, Tianle Liu, Mingsheng Long et Michael Jordan. *Bridging Theory and Algorithm for Domain Adaptation*. In Kamalika Chaudhuri et Ruslan Salakhutdinov, editeurs, Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 7404–7413. PMLR, 09–15 Jun 2019. (Cited on page 34.)

[Zhao 2018] Zhengli Zhao, Dheeru Dua et Sameer Singh. *Generating Natural Adversarial Examples*. In International Conference on Learning Representations, 2018. (Cited on pages 46 and 47.)

[Zhu 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola et Alexei A Efros. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. In Computer Vision (ICCV), 2017 IEEE International Conference on, 2017. (Cited on page 41.)

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Université Bretagne Sud

Titre : Transport optimal et apprentissage profond : apprendre l'un de l'autre

Mot clés : Transport Optimal, Sous-lots, labels corrompus, Réseaux adversaires génératifs

Résumé : Les modèles d'apprentissage profond sont des réseaux de neurones artificiels et sont très compétitifs dans le cadre de problèmes décisionnels. En classification, ces réseaux permettent par exemple une représentation plus complexe des données et donc des décisions plus pertinentes elles aussi. Cependant, l'essor de ces réseaux est aussi lié au développement de plusieurs champs des mathématiques : cette thèse porte sur l'interaction des réseaux de neurones avec l'un de ces champs appelé Transport Optimal (TO).

Le TO est en particulier adapté à la mesure de distance entre distributions de probabilité, en calculant le coût minimal pour déplacer une distribution vers une autre. Sa force consiste en l'utilisation de la géométrie des espaces de probabilité grâce aux différents coûts.

Cette thèse explore les deux faces de l'interaction entre transport optimal et apprentissage profond. D'abord, nous travaillerons sur comment le TO peut définir des fonctions de coût pertinentes pour les réseaux de neurones. Nous nous concentrerons sur la définition d'une régularisation basée sur le TO pour l'apprentissage en présence de labels corrompus et sur l'utilisation du TO pour la génération de données mal classifiées pour un réseau pré-entrainé. Nous étudierons ensuite le TO au travers de son utilisation dans l'apprentissage profond avec un focus particulier sur l'utilisation de sous lots. Nous analyserons alors les gains et pertes tant théoriques que pratiques de cette méthode.

Title: Optimal Transport and Deep Learning : Learning from one another

Keywords: Optimal Transport, Minibatch, Noisy Labels, Generative Adversarial Network

Abstract: Deep learning (DL) models are artificial neural networks and they have arisen as the current most competitive method to make data-driven decisions. In classification, these networks have a more complex representation of data and thus they make more complex predictions. However, DL's recent successes are also due to the development of some mathematical fields : this thesis is about studying the different interactions of DL with one of these fields called Optimal Transport (OT).

To measure the distance between probability distributions, one can rely on the OT theory. It defines a measure through the minimal displacement cost of a distribution to another. Its strength is to use the space geometry with a given ground cost on the data space. Several DL methods are built upon this theory.

This thesis proposes to explore two faces of the interaction between OT and DL. We will first focus on how OT can define meaningful cost functions for neural networks. We will focus on how to define an OT based regularization for learning with noisy labels and how we can use OT to generate misclassified data for a pre-trained classifier. We will then explore what can be learnt about OT from DL applications, with a focus on the minibatch approximation of OT. We will answer what are the gains and downsides of the minibatch formulation both in theory and practice.