Outlier-Robust Optimal Transport

Debarghya Mukherjee [1 2] **Aritra Guha** [3] **Justin Solomon** [4 2] **Yuekai Sun** [1] **Mikhail Yurochkin** [5 2]

Abstract

Optimal transport (OT) measures distances between distributions in a way that depends on the geometry of the sample space. In light of recent advances in computational OT, OT distances are widely used as loss functions in machine learning. Despite their prevalence and advantages, OT loss functions can be extremely sensitive to outliers. In fact, a single adversarially-picked outlier can increase the standard W_2-distance arbitrarily. To address this issue, we propose an outlier-robust formulation of OT. Our formulation is convex but challenging to scale at a first glance. Our main contribution is deriving an *equivalent* formulation based on cost truncation that is easy to incorporate into modern algorithms for computational OT. We demonstrate the benefits of our formulation in mean estimation problems under the Huber contamination model in simulations and outlier detection tasks on real data.

1. Introduction

Optimal transport (OT) is a fundamental problem in applied mathematics. In its original form (Monge, 1781), the problem seeks the minimum-cost way to transport mass from a probability distribution μ on $\mathcal{X}$ to another distribution ν on $\mathcal{X}$. In its original form, Monge's problem proved hard to study, and Kantorovich (1942) relaxed Monge's formulation of the optimal transport problem to

$$\mathrm{OT}(\mu, \nu) \triangleq \min_{\Pi \in \mathcal{F}(\mu,\nu)} \mathbb{E}_{(X_1,X_2)\sim\Pi}\big[c(X_1, X_2)\big], \qquad (1.1)$$

where $\mathcal{F}(\mu, \nu)$ is the set of couplings between μ and ν (probability distributions on $\mathcal{X} \times \mathcal{X}$ whose marginals are μ and ν) and c is a cost function. In this paper, we assume $c(x, y) \geq 0$ and $c(x, x) = 0$. Compared to other common measure of distance between probability distributions (*e.g.*d-divergences), optimal transport uniquely depends on the geometry of the sample space (through the cost function).

Recent advancements in optimization for optimal transport (Cuturi, 2013; Solomon et al., 2015; Genevay et al., 2016; Seguy et al., 2018) enable its broad adaptation in machine learning applications where geometry of the data is important; see (Peyré & Cuturi, 2018) for a survey. Optimal transport has found applications in natural language processing (Kusner et al., 2015; Huang et al., 2016; Alvarez-Melis & Jaakkola, 2018; Yurochkin et al., 2019), generative modeling (Arjovsky et al., 2017), clustering (Ho et al., 2017), domain adaptation (Courty et al., 2014; 2017), large-scale Bayesian modeling (Srivastava et al., 2018), anomaly detection (Tong et al., 2020), and many other domains.

Many applications use OT as a loss in an optimization problem of the form:

$$\theta \in \arg\min\nolimits_{\theta\in\Theta} \mathrm{OT}(\mu_n, \nu_\theta), \qquad (1.2)$$

where $\{\nu_\theta\}_{\theta\in\Theta}$ is a collection of parametric models and μ_n is the empirical distribution of the samples. Such estimators are called *minimum Kantorovich estimators (MKE)* (Bassetti et al., 2006). They are popular alternatives to likelihood-based estimators, especially in generative modeling. For example, when $\mathrm{OT}(\cdot, \cdot)$ is the Wasserstein-1 distance and ν_θ is a generator parameterized by a neural network with weights θ, equation 1.2 corresponds to the Wasserstein GAN (Arjovsky et al., 2017).

One drawback of optimal transport is its sensitivity to outliers. Because *all* the mass in μ must be transported to ν, a small fraction of outliers can have an outsized impact. For statistics and machine learning applications in which the data is corrupted or noisy, this is a major issue. For example, the poor performance of Wasserstein GANs in the presence of outliers was noted in the recent works on outlier-robust generative learning with f-divergence GANs (Chao et al., 2018; Wu et al., 2020). The problem of outlier-robustness in MKE has not been studied except in two recent works proposing changes to the OT formulation that are challenging to handle computationally (Staerman et al., 2020; Balaji et al., 2020). Our goal is to derive an outlier-robust OT formulation compatible with existing efficient

[1] Department of Statistics, University of Michigan [2] MIT-IBM Watson AI Lab [3] Department of Statistical Science, Duke University [4] MIT CSAIL [5] IBM Research. Correspondence to: Debarghya Mukherjee <mdeb@umich.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

computational OT methods (Peyré & Cuturi, 2018).

In this paper, we propose a modification of OT to address its sensitivity to outliers. Our formulation can be used as a loss in equation 1.2, so that it is robust to a small fraction of outliers in the data. For simplicity, we consider the ϵ-contamination model (Huber & Ronchetti, 2009). Let ν_{θ_0} be a member of a parametric family $\{\nu_\theta : \theta \in \Theta\}$ and let

$$\mu = (1-\epsilon)\nu_{\theta_0} + \epsilon\tilde{\nu},$$

where μ is the data-generating distribution, $\epsilon > 0$ is the fraction of outliers, and $\tilde{\nu}$ is the distribution of the outliers. Although the fraction of outliers is capped at ϵ, the value of the outliers can be arbitrary, so their effect on the optimal transport problem can be arbitrarily large. We modify the problem so that it is more robust to such outliers, targeting the downstream application of learning θ_0 from (samples from) μ in the ϵ-contamination model.

Our main contributions are as follows:

- We propose a robust OT formulation suitable for statistical estimation in the ϵ-contamination model using MKE.
- We show that our formulation is equivalent to the original OT problem with a clipped transport cost. This connection enables us to leverage the voluminous literature on computational optimal transport to develop efficient algorithm to perform MKE robust to outliers.
- Our formulation enables a new application of optimal transport: outlier detection in data.

2. Problem Formulation

2.1. Robust OT for MKE

To promote outlier-robustness in MKE, we need to allow the corresponding OT problem to ignore outliers in the data distribution μ. The ϵ-contamination model imposes a cap on the fraction of outliers, so it is not hard to see that $\|\mu - \nu_{\theta_0}\|_{\text{TV}} \le \epsilon$, where $\|\cdot\|_{\text{TV}}$ is the total-variation norm defined as $\|\mu\|_{\text{TV}} = \int \frac{1}{2}|\mu(\mathrm{d}x)|$. This suggests we solve a TV-constrained/regularized version of equation 1.2:

$$\begin{aligned} \min_{\theta\in\Theta,\tilde{\mu}} \quad & \text{OT}(\tilde{\mu}, \nu_\theta) \\ \text{subject to} \quad & \|\mu - \tilde{\mu}\|_{\text{TV}} \le \epsilon; \;\; \tilde{\mu} \in \mathbf{P}, \end{aligned} \tag{2.1}$$

where $\mathbf{P}$ is the set of all probability measures. The constrained version, however, suffers from identification issues. It cannot distinguish between "clean" distributions within TV distance ϵ of ν_{θ_0}. To see this, fix $\theta \in \Theta$, and note that the optimal value of equation 2.1 is zero whenever μ is within ϵ-TV distance of ν_θ. Thus equation 2.1 cannot distinguish between two parameter values θ_1 and θ_2 such that $\|\nu_{\theta_1} - \nu_{\theta_2}\|_{\text{TV}} \le \epsilon$. This makes it unsuitable as a loss function for statistical estimation, because it cannot lead to a consistent estimator. As an alternative, its regularized counterpart does not suffer from this issue:

$$\min_{\substack{\theta\in\Theta \\ s:\mu+s\in\mathbf{P}}} \text{OT}(\mu+s, \nu_\theta) + \lambda\|s\|_{\text{TV}}, \tag{2.2}$$

where $\lambda > 0$ is a regularization parameter. Note that, the constrained and Lagrangian formulations are equivalent, but the equivalence depends on the two distributions in the arguments of the Wasserstein distance. In other words, as we vary the distributions (keeping ϵ fixed), the equivalent λ will change. In our formulation, we are fixing λ and varying the distributions, so the solution paths are different, making the parameter identifiable. In the rest of this paper, we work with the TV-regularized formulation equation 2.2.

The main idea of our formulation is to allow for modifications of μ, while penalizing their magnitude and ensuring that the modified μ is still a probability measure. Below we formulate this intuition in an optimization problem titled ROBOT (ROBust Optimal Transport):

Formulation 1:

$$\begin{aligned} \min_{\pi,s} \quad & \iint c(x,y)\,\pi(x,y)\,dx\,dy + \lambda\|s\|_{\text{TV}} \\ \text{subject to} \quad & \int \pi(x,y)\,dy = \mu(x) + s(x) \ge 0 \\ & \int \pi(x,y)\,dy = \nu(y) \\ & \int s(\mathrm{d}x) = 0. \end{aligned} \tag{2.3}$$

where π is a density function on $\mathcal{X} \times \mathcal{X}$. The first and the last constraints ensure that $\mu + s$ is a valid probability measure, while $\lambda\|s\|_{\text{TV}}$ penalizes the amount of modifications in μ. We can identify exact locations of outliers in μ by inspecting $\mu + s$, i.e. if $\mu(x) + s(x) = 0$, then x got eliminated and is an outlier. We will use this property to propose an outlier detection method.

ROBOT, unlike classical OT, guarantees that an adversarially-picked outliers cannot increase the (robust) transport distance arbitrarily. Let $\tilde{\mu} = (1-\epsilon)\mu + \epsilon\mu_c$, i.e., $\tilde{\mu}$ is μ contaminated with outliers from μ_c, and let ν be an arbitrary measure; in MKE, $\tilde{\mu}$ is the contaminated data and ν is the model we learn. The adversary can arbitrarily increase $\text{OT}(\tilde{\mu}, \nu)$ by manipulating the outlier distribution μ_c. For ROBOT, we have the following bound:

Theorem 2.1. *Let $\tilde{\mu} = (1-\epsilon)\mu + \epsilon\mu_c$ for some $\epsilon \in [0,1)$. Then,*

$$\begin{aligned} \text{ROBOT}(\tilde{\mu}, \nu) \le \min\{&\text{OT}(\mu,\nu) + \lambda\epsilon\|\mu - \mu_c\|_{\text{TV}}, \\ &\lambda\|\tilde{\mu} - \nu\|_{\text{TV}}, \text{OT}(\tilde{\mu},\nu)\}, \end{aligned} \tag{2.4}$$

This bound has two key takeaways: since TV norm of distributions is bounded by 1, the adversary can not increase ROBOT$(\tilde{\mu}, \nu)$ arbitrarily; in the absence of outliers, ROBOT is bounded by classical OT. See Appendix B for the proof.

Related work. We note a connection between equation 2.3 and unbalanced OT (UOT) (Chizat., 2017; Chizat et al., 2018). UOT is typically formulated by replacing the TV norm with $\mathrm{KL}(\mu + s|\mu)$ and adding an analogous term for ν. Chizat et al. (2018) studied entropy regularized UOT with various divergences penalizing marginal violations. Optimization problems similar to equation 2.3 have also been considered outside of ML (Piccoli & Rossi, 2014; Liero et al., 2018). Balaji et al. (2020) use UOT with χ^2-divergence penalty on marginal violations to achieve outlier-robustness in generative modeling. Another relevant variation of OT is partial OT (Figalli, 2010; Caffarelli & McCann, 2010). It may also be considered for outlier-robustness but has a drawback of forcing mass destruction rather than adjusting marginals to ignore outliers when they are present. Staerman et al. (2020) take a different path: they replace the expectation in the Wasserstein-1 dual with a median-of-means to promote robustness. It is unclear what is the corresponding primal problem, making their formulation hard to interpret as an optimal transport problem.

A major challenge with the aforementioned methods, including our Formulation 1, is the large scale implementation of the optimization problem. Chizat et al. (2018) propose a Sinkhorn-like algorithm for entropy regularized UOT, but it is not amenable to stochastic optimization. Balaji et al. (2020) propose a stochastic optimization algorithm based on the UOT dual, but it requires two additional neural networks (total of four including dual potentials) to parameterize modified marginal distributions (i.e., $\mu + s$ and analogous one for ν). Optimizing with a median-of-means in the objective function (Staerman et al., 2020) is also challenging. The key contribution of our work is a formulation *equivalent* to equation 2.3, which is *easily compatible* with the large body of classical OT optimization techniques (Cuturi, 2013; Solomon et al., 2015; Genevay et al., 2016; Seguy et al., 2018).

More efficient equivalent formulation. At a first glance, there are two issues with equation 2.3: it appears asymmetric, and it is unclear if it can be optimized efficiently. Below we present an *equivalent* formulation that is free of these issues:

Formulation 2:

$$\begin{aligned} \min_{\Pi \in \mathcal{F}^+(\mathbb{R}^d \times \mathbb{R}^d)} \quad & \mathbb{E}_{(X,Y)\sim\Pi}\left[C_\lambda(X,Y)\right] \\ \text{subject to} \quad & X \sim \mu, \;\; Y \sim \nu\,. \end{aligned} \tag{2.5}$$

where C_λ is the *truncated cost* function defined as $C_\lambda(x, y) = \min\{c(x, y), 2\lambda\}$. Looking at equation 2.5, it is not apparent that it adds robustness to MKE, but it is symmetric, easy to combine with entropic regularization by simply truncating the cost, and benefits from stochastic optimization algorithms (Genevay et al., 2016; Seguy et al., 2018).

This formulation also has a distant relation to the idea of loss truncation for achieving robustness (Shen & Sanghavi, 2019). Pele & Werman (2009) consider the Earth Mover Distance (discrete OT) with truncated cost to achieve computational improvements; they also mentioned its potential to promote robustness against outlier noise but did not explore this direction.

In Section 3, we establish an *equivalence* between the two ROBOT formulations, equation 2.3 and equation 2.5. This equivalence allows us to obtain an efficient algorithm based on equation 2.5 for robust MKE. We also provide a simple procedure for computing the optimal s in equation 2.3 from the solution of equation 2.5, enabling a new OT application: outlier detection. We verify the effectiveness of robust MKE and outlier detection in our experiments in Section 4. Before presenting the equivalence proof, however, we formulate the discrete analogs of the two ROBOT formulations for their practical value.

2.2. Discrete ROBOT formulations

In practice, we typically encounter samples from distributions, rather then the distributions themselves. Sampling is also built into stochastic optimization. In this subsection, we present the discrete versions of the ROBOT formulations. The key detail is that, in equation 2.3, μ, ν and s are all supported on $\mathbb{R}^d$, while in the discrete case the empirical measures $\mu_n \in \Delta^{n-1}$ and $\nu_m \in \Delta^{m-1}$ are supported on a set of points (Δ^r is the unit probability simplex in $\mathbb{R}^r$). As a result, to formulate a discrete version of equation 2.3, we need to augment μ_n and ν_m with each others' supports. To be precise, let $\mathrm{supp}(\mu_n) = \{X_1, \ldots, X_n\}$ and $\mathrm{supp}(\nu_m) = \{Y_1, \ldots, Y_m\}$. Define $\mathcal{C} = \{Z_1, Z_2, \ldots, Z_{m+n}\} = \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$. Then discrete analog of equation 2.3 is

Formulation 1 (discrete):

$$\begin{aligned} \min_{\Pi, \mathbf{s}} \quad & \langle C_{aug}, \Pi \rangle + \lambda\left[\|s_1\|_1 + \|t_1\|_1\right] \\ \text{subject to} \quad & \Pi 1_{m+n} = \begin{bmatrix} \mu_n + s_1 \\ t_1 \end{bmatrix}, \quad \Pi^\top 1_{m+n} = \begin{bmatrix} 0 \\ \nu_m \end{bmatrix} \\ & \Pi \succeq 0, \quad 1_{m+n}^\top \mathbf{s} = 0, \end{aligned} \tag{2.6}$$

where $C_{aug} \in \mathbb{R}^{(m+n)\times(m+n)}$ is the augmented cost function $C_{aug,i,j} = c(Z_i, Z_j)$ (c is the ground cost, e.g., squared Euclidean distance), $\mathbf{s} = (s_1, t_1)$ and 1_r is the vector all ones in $\mathbb{R}^r$. The TV norm got replaced with its discrete analog, the L_1 norm. Similarly to its continuous counterpart, the optimization problem is harder than typical OT due to additional constraint optimization variable $\mathbf{s}$ and increased cost matrix size.

The discrete analog of equation 2.5 is straightforward:

Formulation 2 (discrete):

$$\begin{aligned} &\min_{\Pi \in \mathbb{R}^{n \times m}} && \langle C_\lambda, \Pi \rangle \\ &\text{subject to} && \Pi 1_n = \mu_n, \quad \Pi^\top 1_m = \nu_m, \quad \Pi \succeq 0, \end{aligned} \tag{2.7}$$

where $C_{\lambda,i,j} = \min\{c(X_i, Y_j), 2\lambda\}$. As in the continuous case, it is easy to adapt modern (regularized) OT solvers without any computational overhead, and formulations of equation 2.6 and equation 2.7 are equivalent. It is also possible to recover $\mathbf{s}$ of equation 2.6 from the solution of equation 2.7 to perform outlier detection.

Two-sided formulation. So far we have assumed that one of the input distributions does not have outliers, which is the setting of MKE, where the clean distribution corresponds to the model we learn. In some applications, both distributions may be corrupted. To address this case, we provide an *equivalent* two-sided formulation, analogous to UOT with TV norm:

Formulation 3 (two-sided):

$$\begin{aligned} &\min_{\Pi, \mathbf{s}_1, \mathbf{s}_2} && \langle C_{aug}, \Pi \rangle + \lambda \left[\|s_1\|_1 + \|t_1\|_1 + \|s_2\|_1 + \|t_2\|_1 \right] \\ &\text{subject to} && \Pi 1_{m+n} = \begin{bmatrix} \mu_n + s_1 \\ t_1 \end{bmatrix} \\ &&& \Pi^\top 1_{m+n} = \begin{bmatrix} s_2 \\ \nu_m + t_2 \end{bmatrix} \\ &&& \Pi \succeq 0, \quad 1_{m+n}^\top \mathbf{s}_1 = 0, \quad 1_{m+n}^\top \mathbf{s}_2 = 0 \end{aligned} \tag{2.8}$$

where $\mathbf{s}_1 = (s_1^\top, t_1^\top)^\top$ and $\mathbf{s}_2 = (s_2^\top, t_2^\top)^\top$.

3. Equivalence of the ROBOT formulations

In this section, we present our main theorem, which demonstrates the equivalence between two formulations of the robust optimal transport:

Theorem 3.1. *For any two measures μ and ν, ROBOT(μ, ν) has same value for both the formulations, i.e., Formulation 1 is equivalent to Formulation 2 for the discrete case. Additionally, if the (non-truncated) cost function $c(\cdot, \cdot)$ is a metric, then the equivalence of the two formulations also holds for the continuous case. Moreover, we can recover optimal coupling of one formulation from the other.*

Below we sketch the proof of this theorem and highlight some important techniques used in the proof. We focus on the discrete case as it is more intuitive and has concrete practical implications in our experiments. A complete proof can be found in Appendix A. Please also see Appendix A.2 for the proof of equivalence between Formulations 1, 2 and 3 in the discrete case.

3.1. Proof sketch

In the remainder of this section, we consider the discrete case, i.e., equation 2.6 for Formulation 1 (F1) and equation 2.7 for Formulation 2 (F2). Suppose Π_2^* is an optimal solution of F2. Then we construct a feasible solution $\Pi_1^*, \mathbf{s}_1^* = (s_1^*, t_1^*)$ of F1 based on Π_2^* with the same value of the objective function as F2 and claim that $(\Pi_1^*, \mathbf{s}_1^*)$ is an optimal solution. We prove the claim by contradiction: if $(\Pi_1^*, \mathbf{s}_1^*)$ is not optimal, then there exists another pair $(\tilde{\Pi}_1, \tilde{\mathbf{s}}_1)$ which is optimal for F1 with strictly less objective value. We then construct another feasible solution $\Pi_{2,new}^*$ of Formulation 2 which has the same objective value as of $(\tilde{\Pi}_1, \tilde{\mathbf{s}}_1)$ for F1. This implies $\Pi_{2,new}^*$ has strictly less objective value for F2 than Π_2^*, which is a contradiction.

The two main steps of this proof are (1) constructing a feasible solution of F1 starting from a feasible solution of F2 and (2) showing that the solution constructed is indeed optimal for F1. Hence step (1) gives a recipe to construct an optimal solution of F1 starting from an optimal solution of F2. We elaborate the first point in the next subsection, which has practical implications for outlier detection. The other point is more technical; interested readers may go through the proof in Appendix A.1.

Algorithm 1 Generating optimal solution of F1 from F2

```
Start with $\Pi_2^* \in \mathbb{R}^{n \times m}$, an optimal solution of Formulation 2.
Create an augmented matrix $\Pi \in \mathbb{R}^{m+n \times m+n}$ with all 0. Divide $\Pi$ into four blocks:
    $$\Pi = \begin{bmatrix} \underbrace{\Pi_{11}}_{n \times n} & \underbrace{\Pi_{12}}_{n \times m} \\ \underbrace{\Pi_{21}}_{m \times n} & \underbrace{\Pi_{22}}_{m \times m} \end{bmatrix}$$
Set $\Pi_{12} \leftarrow \Pi_2^*$ and collect all the indices $\mathcal{I} = \{(i, j) : C_{i,j} > 2\lambda\}$.
Set $\Pi_{12}(i, j) \leftarrow 0$ for $(i, j) \in \mathcal{I}$.
Set $\Pi_{22}(j, j) \leftarrow \sum_{i=1}^n \Pi_2^*(i, j) \mathbb{1}_{(i,j) \in \mathcal{I}}$ for all $1 \le j \le m$ and set $\Pi_1^* \leftarrow \Pi$.
Set $s_1^*(i) = -\sum_{j=1}^m \Pi_2^*(i, j) \mathbb{1}_{(i,j) \in \mathcal{I}}$ for all $1 \le i \le n$.
Set $t_1^*(j) = \Pi_{22}(j, j)$ for all $1 \le j \le m$.
return $\Pi_1^*, s_1^*, t_1^*$.
```

3.2. Going from Formulation 2 to Formulation 1

Let Π_2^* (respectively Π_1^*) be an optimal solution of F2 (respectively F1). Recall that Π_1^* has dimension $(m+n)\times(m+n)$. From the column sum constraint in F1, we need to take the first n columns of Π_1^* to be exactly 0, whereas the last m columns must sum up to ν_m. For any matrix A, we denote by $A[(a:b)\times(c:d)]$ the submatrix consisting of rows from a to b and columns from c to d. Our main idea is to put a modified version of Π_2^* in $\Pi_1^*[(1:n)\times(n+1:m+n)]$ and make $\Pi_1^*[(n+1:m+n)\times(n+1:m+n)]$ diagonal. First we describe how to modify Π_2^*. Observe that, if for some (i,j) $C_{i,j}>2\lambda$, we expect $X_i\in\mathrm{supp}(\mu_n)$ to be an outlier resulting in high transportation cost, which is why we truncate the cost in F2. Therefore, to get an optimal solution of F1, we make the corresponding value of optimal plan 0 and dump the mass into the corresponding slack variable t_1^* in the diagonal of the bottom right submatrix. This changes the row sum, which is taken care of by s_1^*. But, as we are not moving this mass outside the corresponding column, the column sum of $\Pi_1^*[(1:(m+n)):((n+1):(m+n))]$ remains same as column sum of Π_2^*, which is ν_n. We summarize this procedure in Algorithm 1.

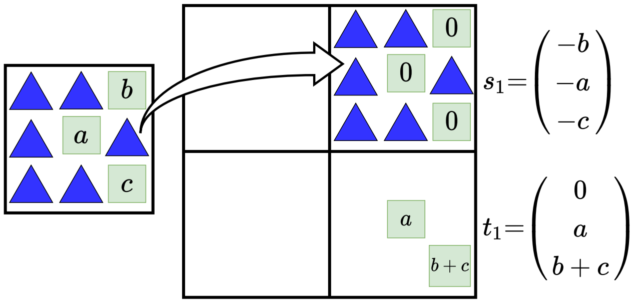

b
a
c
0
0
0
a
b + c
$s_1 = \begin{pmatrix} -b \\ -a \\ -c \end{pmatrix}$
$t_1 = \begin{pmatrix} 0 \\ a \\ b+c \end{pmatrix}$

Figure 1: Constructing optimal solution of Formulation 1 from optimal solution of Formulation 2.

Example. In Figure 1, we provide an example to visualize the construction. On the left, we have Π_2^*, an optimal solution of Formulation 2. The blue triangles denote the positions where the corresponding cost value is $\leq 2\lambda$, and light-green squares denote the positions where the corresponding value of the cost matrix is $>2\lambda$. To construct an optimal solution Π_1^* of Formulation 1 from this Π_2^*, we first create an augmented matrix of size 6×6. We keep all the entries of the left 6×3 sub-matrix as 0 (in this picture blank elements indicate 0). On the right submatrix, we put Π_2^* into the top-right block, but remove the masses from light-green squares, i.e. where cost value is $>2\lambda$, and put it in the diagonal entries of the bottom right block as shown in Figure 1. This mass contributes to the slack variables s_1 and t_1, and this augmented matrix along with s_1, t_1 give us an optimal solution of Formulation 1.

3.3. Outlier detection with ROBOT

Our construction algorithm has practical consequences for outlier detection. Suppose we have two datasets, a clean dataset ν_m (i.e., has no outliers) and an outlier-contaminated dataset μ_n. We can detect the outliers in μ_n without directly solving costly Formulation 1 by following Algorithm 2. In this algorithm, λ is a regularization parameter that can be chosen via cross-validation or heuristically (see Section 4.2 for an example). In Section 4.2, we use this algorithm to perform outlier detection on image data.

Outlier detection with entropic regularization. Algorithm 1 allows us to recover solution of Formulation 1, which ultimately is used for outlier detection in Algorithm 2, by solving the simpler truncated cost Formulation 2 problem in equation 2.7. Similarly to the regular OT, it can be solved exactly with a linear program solver—Pele & Werman (2009) propose a faster exact solution based on min-cost-flow solvers benefiting from the cost truncation—or approximately using entropic regularization techniques, e.g. Sinkhorn algorithm (Cuturi, 2013). In the following lemma, we show that Algorithm 1 recovers a meaningful approximate solution of Formulation 1 from an approximate solution of Formulation 2 obtained with entropy-regularized OT solvers.

Lemma 3.2. *Let $\Pi_{2,\alpha}^*$ be a solution of the entropy regularized version of equation 2.7:*

$$\begin{aligned} &\underset{\Pi\in\mathbb{R}^{n\times m}}{\arg\min} && \langle C_\lambda,\Pi\rangle+\alpha H(\Pi) \\ &\text{subject to} && \Pi 1_n=\mu_n,\quad \Pi^\top 1_m=\nu_m,\quad \Pi\succeq 0, \end{aligned} \tag{3.1}$$

and let $\left(\Pi_{1,\alpha}^*,\mathbf{s}_{1,\alpha}^*\right)$ be the corresponding approximate solution of Formulation 1 recovered from $\Pi_{2,\alpha}^*$ by Algorithm 1. Then

$$\|\Pi_{1,\alpha}^*-\Pi_1^*\|_F+\|\mathbf{s}_{1,\alpha}^*-\mathbf{s}_1^*\|_2\to 0$$

as $\alpha\to 0$, where $(\Pi_1^*,\mathbf{s}_1^*)$ is the exact solution of Formulation 1 in equation 2.6.

When the solution of equation 2.7 is non-unique, then the solution of equation 3.1 converges to the solution of equation 2.7 with maximum entropy (see Proposition 4.1 of Peyré & Cuturi (2018)) and consequently recovers the corresponding maximum entropy solution $(\Pi_1^*,\mathbf{s}_1^*)$ of equation 2.6 by Algorithm 1 in the limit. Please find the proof in Appendix C.

4. Empirical studies

To evaluate effectiveness of ROBOT, we consider the task of robust mean estimation under the Huber contamination model. The data is generated from $(1-\epsilon)\mathcal{N}(\eta_0,I_d)+$

Algorithm 2 Outlier detection in contaminated data

```
Start with $\mu_n$ (contaminted data) and $\nu_m$ (clean data).
Solve Formulation 2 and obtain $\Pi_2^*$ using a suitable value of $\lambda$.
Use Algorithm 1 to obtain $\Pi_1^*, s_1^*, t_1^*$ from $\Pi_2^*$.
Find $\mathcal{I}$, the set of all the indices where $\mu_n + s_1^* = 0$.
Return $\mathcal{I}$ as the indices of outliers in $\mu_n$.
```

$\epsilon\mathcal{N}(\eta_1, I_d)$ and the goal is to estimate η_0. Prior work has advocated for using f-divergence GANs (Chao et al., 2018; Wu et al., 2020) for this problem and pointed out inefficiencies of Wasserstein GAN in the presence of outliers. We show that our robust OT formulation allows to estimate the uncontaminated mean η_0 comparably or better than a variety of f-divergence GANs. We also use this simulated setup to study sensitivity to the cost truncation hyperparameter λ.

In our second experiment, we present a new application of optimal transport enabled by ROBOT. Suppose we have collected a curated dataset ν_m (i.e., we know that it has no outliers)—such data collection is expensive, and we want to benefit from it to automate subsequent data collection. Let μ_n be a second dataset collected "in the wild," i.e., it may or may not have outliers. We demonstrate how ROBOT can be used to identify outliers in μ_n using the curated dataset ν_m.

4.1. Robust mean estimation

Following Wu et al. (2020), we consider a simple generator of the form $g_\theta(x) = x + \theta$, $x \sim \mathcal{N}(0, I_d)$, d is the data dimension. The basic idea of robust mean estimation with GANs is to minimize various distributional divergences between samples from g_θ and observed data simulated from $(1-\epsilon)\mathcal{N}(\eta_0, I_d) + \epsilon\mathcal{N}(\eta_1, I_d)$. The goal is to estimate η_0 with θ.

To efficiently implement ROBOT GAN, we use a standard min-max optimization approach: solve the inner max (ROBOT) and use gradient descent for the outer min parameter. To solve ROBOT, it is straightforward to adopt any of the prior stochastic regularized OT solvers: the only modification is the truncation of the cost entries as in equation 2.7. We use the stochastic algorithm for semi-discrete regularized OT (Genevay et al., 2016, Algorithm 2). We summarize ROBOT GAN in Algorithm 3. Line 5 - Line 11 perform the inner optimization where we solve the entropy regularized OT dual with truncated cost and Line 12 - Line 14 perform gradient update of θ.

For the f-divergence GANs (Nowozin et al., 2016), we use the code of Wu et al. (2020) for GANs with Jensen-Shannon (JS) loss, squared Hellinger (SH) loss, and Reverse Kullback-Leibler (RKL) loss. For the exact expressions of these divergences, see Table 1 of Wu et al. (2020). We report estimation error measured by the Euclidean distance between true uncontaminated mean η_0 and estimated mean θ for various contamination distributions in Table 1. ROBOT GAN performs well across all considered contamination distributions. As the difference between true mean η_0 and contamination mean η_1 increases, the estimation error of all methods tends to increase. However, when it becomes easier to distinguish outliers from clean samples, i.e., $\eta_1 = 2 \cdot \mathbf{1_5}$, performance of ROBOT noticeably improves. We present an analogous study with data simulated from a mixture of Cauchy distributions in Appendix F.

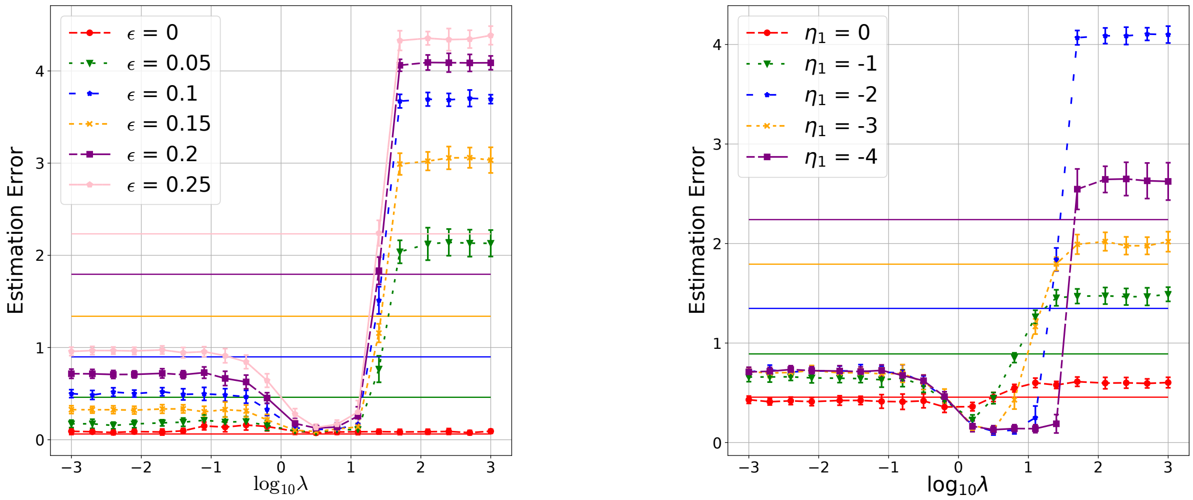
Algorithm 3 ROBOT GAN

```
1: **Input:** robustness regularizion $\lambda$, entropic regularization $\alpha$, data distribution $\mu_n \in \Delta^{n-1}$, $supp(\mu_n) = \mathcal{X} = [X_1, \dots, X_n]$, steps sizes $\tau$ and $\gamma$
2: **Initialize:** Initialize $\theta = \theta_{init}$, set number of iterations $M$ and $L$, $i = 0$, $\mathbf{v} = \tilde{\mathbf{v}} = \mathbf{0}$.
3: **for** $j = 1, \dots, M$ **do**
4:     Generate $\tilde{z} \sim \mathcal{N}(0, I_d)$ and set $z = \tilde{z} + \theta$.
5:     Set the cost vector $\mathbf{c} \in \mathbb{R}^n$ as $\mathbf{c}(k) = \min\{c(X_k, z), 2\lambda\}$ for $k = 1, \dots, n$.
6:     **for** $i = 1, \dots, L$ **do**
7:         Set $\mathbf{h} \leftarrow \frac{\tilde{\mathbf{v}} - \mathbf{c}}{\alpha}$ and do the normalized exponential transformation $\mathbf{u} \leftarrow \frac{e^{\mathbf{h}}}{\langle \mathbf{1}, e^{\mathbf{h}} \rangle}$.
8:         Calculate the gradient $\nabla\tilde{\mathbf{v}} \leftarrow \mu_n - \mathbf{u}$.
9:         Update $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + \gamma\nabla\tilde{\mathbf{v}}$ and $\mathbf{v} \leftarrow (1/(j+i))\tilde{\mathbf{v}} + (j+i-1/(j+i))\mathbf{v}$.
10:    **end for**
11:    Do the same transformation of $\mathbf{v}$ as in Step 7, i.e. set $\mathbf{h} \leftarrow \frac{\mathbf{v} - \mathbf{c}}{\alpha}$ and set $\Pi \leftarrow \frac{e^{\mathbf{h}}}{\langle \mathbf{1}, e^{\mathbf{h}} \rangle}$.
12:    Set $\Pi(k) = 0$ for $k$ such that $c(X_k, z) > 2\lambda$ for $k = 1, \dots, n$.
13:    Calculate gradient with respect to $\theta$ as $\nabla\theta = 2\left[z \sum_k \Pi(k) - \mathcal{X}^\top \Pi\right]$
14:    Update $\theta \leftarrow \theta - \tau\nabla\theta$.
15: **end for**
16: **Ouput:** $\theta$
```

We also compared to the Sinkhorn-based UOT algorithm (Chizat et al., 2018) available in the Python Optimal Transport (POT) library (Flamary & Courty, 2017); to obtain a UOT GAN, we modified steps 5-12 of Algorithm 3 for computing Π. Unsurprisingly, both ROBOT and UOT perform similarly: recall equivalence to Formulation 3, which is similar to UOT with TV norm. The key insight of our work is the equivalence to classical OT with truncated cost, that greatly simplifies optimization and allows to use existing stochastic OT algorithms. In this experiment, the sample size $n = 1000$ is sufficiently small for the Sinkhorn-based UOT POT implementation to be effective, but it

Table 1: Robust mean estimation with GANs using different distribution divergences. True mean is $\eta_0 = \mathbf{0}_5$; sample size $n = 1000$; contamination proportion $\epsilon = 0.2$. We report results over 30 experiment restarts.

Contamination	JS Loss	SH Loss	RKL Loss	ROBOT	UOT
$\mathcal{N}(0.1 \cdot \mathbf{1_5}, I_5)$	**0.09** ± 0.03	0.11 ± 0.03	0.115 ± 0.03	0.1 ± 0.03	0.1 ± 0.04
$\mathcal{N}(0.5 \cdot \mathbf{1_5}, I_5)$	0.23 ± 0.04	0.24 ± 0.05	0.24 ± 0.05	**0.117** ± 0.03	0.2 ± 0.04
$\mathcal{N}(1 \cdot \mathbf{1_5}, I_5)$	0.43 ± 0.05	0.43 ± 0.06	0.43 ± 0.06	0.261 ± 0.06	**0.25** ± 0.05
$\mathcal{N}(2 \cdot \mathbf{1_5}, I_5)$	0.67 ± 0.07	0.67 ± 0.08	0.67 ± 0.08	0.106 ± 0.03	**0.1** ± 0.03



(a) Varying proportion of contamination

(b) Varying outlier distribution mean

Figure 2: Empirical study of the cost truncation hyperparameter λ sensitivity.

breaks in the experiment we present in Section 4.2. We also tried the code of Balaji et al. (2020) based on CVXPY (Diamond & Boyd, 2016), but it is too slow even for the $n = 1000$ sample size. In Subsection 4.3 we present a comparison to Balaji et al. (2020) on a smaller sample size.

Hyperparameter sensitivity study. In the previous experiment, we set $\lambda = 0.5$. Now we demonstrate empirically that there is a broad range of λ values performing well. In Figure 2(a), we study sensitivity of λ under various contamination proportions ϵ holding $\eta_0 = \mathbf{1}_5$ and $\eta_1 = 5 \cdot \mathbf{1}_5$ fixed. Horizontal lines correspond to $\lambda = \infty$, i.e., vanilla OT. The key observations are: there is a wide range of λ efficient at all contamination proportions (note the $\log_{10}$ x-axis scale), and ROBOT is always at least as good as vanilla OT (even when there is no contamination $\epsilon = 0$). In Figure 2(b), we present a similar study varying the mean of the contamination distribution and holding $\epsilon = 0.2$ fixed. We see that as the contamination distribution gets closer to the true distribution, it becomes harder to pick a good λ, but the performance is always at least as good as the vanilla OT (horizontal lines).

4.2. Outlier detection for data collection

Our robust OT formulation equation 2.6 enables outlier identification. Let ν_m be a clean dataset and μ_n potentially contaminated with outliers. Recall that ROBOT allows modification of one of the input distributions to eliminate potential outliers. We can identify outliers in μ_n as follows: if $\mu_n(i) + s_1^*(i) = 0$, then X_i, the ith point in μ_n, is an outlier. Instead of directly solving equation 2.6, which may be inefficient, we use our equivalence results and solve an easier optimization problem equation 2.7, followed by recovering $\mathbf{s}$ to find outliers via Algorithm 2. When using entropy-regularized approximate solutions to detect outliers with Algorithm 2, in step 4, $\mu_n + s_1^*$ will not be exactly 0 for the outliers, so a small threshold should be used instead. We modify step 4 to "Find $\mathcal{I}$, the set of all the indices where $\mu_n + s_1^* < 1/n^2$" when using entropy

Table 2: Outlier detection on MNIST.

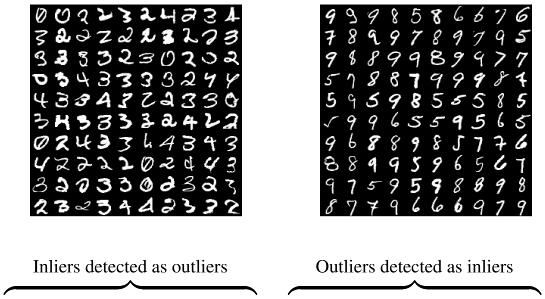
Methods	Accuracy
One class SVM	0.496 ± 0.003
Local outlier factor	0.791 ± 0.001
Isolation forest	0.636 ± 0.010
Elliptical envelope	0.739 ± 0.002
Orthonormal Certificates	0.819 ± 0.008
ROBOT	0.859 ± 0.002
ROBOT-Sinkhorn	$\mathbf{0.897} \pm 0.004$

regularization.

To test our outlier-identification methodology we follow the experimental setup of Tagasovska & Lopez-Paz (2019). Specifically, let ν_m be a clean dataset consisting of 10k MNIST digits from 0 to 4 and μ_n be a dataset collected "in the wild" consisting of (different) 8k MNIST digits from 0 to 4 and 2k outlier MNIST images of digits from 5 to 9. We compute ROBOT(μ_n, ν_m) to identify outlier digit images in μ_n. For each point in μ_n we obtain a prediction, outlier or clean, which allows us to evaluate accuracy. Tagasovska & Lopez-Paz (2019) use last-layer features of a neural network pre-trained on the clean data ν_m—it is straightforward to combine ROBOT and other baselines we consider with any feature extractor, but in this experiment we simply use the raw data.

We compare to the Orthonormal Certificates (OC) method of Tagasovska & Lopez-Paz (2019) and to a variety of standard outlier detection algorithms available in Scikit-learn (Pedregosa et al., 2011): one class SVM (Schölkopf et al., 1999), local outlier factor (Breunig et al., 2000), isolation forest (Liu et al., 2008) and elliptical envelope (Rousseeuw & Driessen, 1999). All baselines except one class SVM and local outlier factor use clean data for training as does our method.

Results of 30 experiment repetitions are summarized in Table 2. ROBOT, i.e. Algorithm 2 where equation 2.7 is solved exactly with a linear program solver, and ROBOT-Sinkhorn, i.e. Algorithm 2 where equation 2.7 is solved approximately using Sinkhorn (Cuturi, 2013), produce the best results. In this experiment ROBOT-Sinkhorn outperforms ROBOT, but it is not necessarily to be expected in general. We conclude that our method is effective in assisting data collection once an initial set of clean data has been acquired and is compatible with entropy-regularized OT solvers ensuring scalability.



digit	count	digit	count
0	10	5	191
1	1	6	94
2	53	7	175
3	32	8	181
4	22	9	309

Figure 3: Insights into ROBOT-Sinkhorn performance: the top left figure is a collection of random inlier digits miss-classified as outliers; the top right picture represents random outlier digits miss-classified as inliers and the table illustrates frequency distribution of miss-classified images. The majority of the errors are on digit 9, possibly due to its similarity to 4.

Elliptical envelope assumes that clean data is Gaussian, while one class SVM, isolation forest and local outlier factor correspondingly attempt to fit SVM, random forest and k-nearest neighbors classifiers to distinguish clean and outlier samples. All these baselines work best when the clean data is unimodal, which is not the case for the MNIST 0 to 4 digits considered in our experiment. The orthonormal certificates method is rooted in PCA and assumes that clean and outlier data live in different subspaces. This assumption is reasonable for the MNIST data, but it might not hold broadly in practice. We believe that empirical success of our method is due to its optimal transport nature. OT is a geometry-sensitive metric on distributions that can distinguish multi-modal distributions and distributions supported on the same subspace. We provide additional insights into the ROBOT performance in Figure 3. A theoretical investigation of ROBOT outlier-detection guarantees is an interesting future work direction.

Hyperparameter selection. To select the cost truncation hyperparameter λ, we propose the following heuristic: since we know that ν_m is clean, we can subsample two datasets from it, compute vanilla OT to obtain transportation plan Π and set λ to be half the maximum distance between matched elements, i.e. $2\lambda = \max_{i,j}\{C_{ij} : \Pi_{ij} > 0\}$, where C is the cost matrix for the two subsampled datasets. This procedure is essentially estimating maximum distance between matched clean samples. To avoid subsampling noise we use 99th percentile instead of the maximum. Our experiments also revealed that increasing λ increases the set of outliers progressively, i.e. if a sample is detected as outlier for some value of λ, then it will be detected as outlier for all higher values of λ. A rigorous theoretical analysis for this observation is a potential future

Table 4: Robust mean estimation for $n = 200$.

Method	Estimation error	Run-time
ROBOT-Sinkhorn	0.196 ± 0.1	$35s \pm 0.6s$
Balaji et al. (2020)	0.212 ± 0.074	$7745s \pm 1670s$

Table 5: Outlier detection for $n = 1000$.

Method	Estimation error	Run-time
ROBOT-Sinkhorn	0.86 ± 0.015	$6 \pm 2s$
Balaji et al. (2020)	0.8 ± 0.0005	$3343 \pm 960s$

work.

The Orthonormal Certificates method (Tagasovska & Lopez-Paz, 2019) also requires setting a threshold. It computes the null-space of the clean train data and uses the norm of the projection into that space as a score to distinguish outliers. Similarly to ROBOT, and as the authors do in their code, we use 99th percentile of those scores computed on the clean train data as the threshold. For other baselines, we use the default hyperparameters.

4.3. Comparison with Balaji et al. (2020)

We conduct additional experiments comparing to the recent robust optimal transport method of Balaji et al. (2020). Their method relies on CVXPY and does not scale to the sample sizes considered in our previous experiments. We compare on smaller data sizes.

Robust mean estimation. We set $n = 200$ samples with contamination distribution mean equal to 2 and true mean equal to 0 (same configuration as in the last row in Table 1). The runtime and the estimation error is reported in Table 4.

Outlier detection experiment. In this experiment, we consider the size of the entire dataset (inliers + outliers) to be $n = 1000$ (with 800 inliers and 200 outliers). The results (accruacy and run-time) are provided in Table 5.

The method of Balaji et al. (2020) is significantly slower as expected. Comparing the performance, we think that ROBOT performs better because it is based on TV norm, while the method of Balaji et al. (2020) uses a chi-squared constraints on the marginal perturbations. A TV constraint on the marginal perturbations is more closely related to the ϵ-contamination model for outlier detection, suggesting that a TV-based constraint/regularizer could be a better choice for the outlier detection applications. We also note that in the outlier detection experiment, using chi-square divergence results in a non-sparse solution and requires tuning a threshold parameter to perform outlier detection, in addition to the chi-square distance hyperparameter. We tuned those parameters, but were not able to achieve significant performance improvements for the method of Balaji et al. (2020).

5. Summary and discussion

We propose and study ROBOT, a robust formulation of optimal transport. Although the problem is seemingly asymmetric and challenging to optimize, there is an equivalent formulation based on cost truncation that is symmetric and compatible with modern stochastic optimization methods for OT.

ROBOT closely resembles unbalanced optimal transport (UOT). In our formulation, we added a TV regularizer to the vanilla optimal transport problem. This is motivated by the ϵ-contamination model. In UOT, the TV regularizer is typically replaced with a KL divergence. Other choices of the regularizer may lead to new properties and applications. Studying equivalent, simpler formulations of UOT with different divergences may be a fruitful future work direction.

From the practical perspective, in our experiments we observed no degradation of ROBOT GAN in comparison to OT GAN, even when there were no outliers. It is possible that replacing OT with ROBOT may be beneficial for various machine learning applications of OT. Data encountered in practice may not be explicitly contaminated with outliers, but it often has errors and other deficiencies, suggesting that a "no-harm" robustness is desirable.

Acknowledgements

This paper is based upon work supported by the National Science Foundation (NSF) under grants no. 1830247 and 1916271. J. Solomon acknowledges the generous support of Army Research Office grants W911NF1710068 and W911NF2010168, of Air Force Office of Scientific Research award FA9550-19-1-031, of National Science Foundation grant IIS-1838071, from the CSAIL Systems that Learn program, from the MIT–IBM Watson AI Laboratory, from the Toyota–CSAIL Joint Research Center, from a gift from Adobe Systems, and from the Skoltech–MIT Next Generation Program. We also thank anonymous reviewers, whose comments were extremely helpful for further improvement of our paper.

References

David Alvarez-Melis and Tommi S Jaakkola. Gromov-Wasserstein alignment of word embedding spaces. *arXiv:1809.00013*, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*, January 2017.

Yogesh Balaji, Rama Chellappa, and Soheil Feizi. Robust optimal transport with applications in generative modeling and domain adaptation. *Advances in Neural Information Processing Systems*, 33, 2020.

Federico Bassetti, Antonella Bodini, and Eugenio Regazzini. On minimum Kantorovich distance estimators. *Statistics & Probability Letters*, 76(12):1298–1302, 2006.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Luis A Caffarelli and Robert J McCann. Free boundaries in optimal transport and Monge-Ampere obstacle problems. *Annals of Mathematics*, pp. 673–730, 2010.

Gao Chao, Yao Yuan, and Zhu Weizhi. Robust estimation via generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

Lenaïc Chizat. Unbalanced optimal transport: Models, numerical methods, applications. *Numerical Analysis [math.NA]. Université Paris sciences et lettres*, 2017.

Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 274–289. Springer, 2014.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 3730–3739, 2017.

Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2292–2300. Curran Associates, Inc., 2013.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Alessio Figalli. The optimal partial transport problem. *Archive for Rational Mechanics and Analysis*, 195(2): 533–560, 2010.

Rémi Flamary and Nicolas Courty. POT Python optimal transport library, 2017. URL `https://pythonot.github.io/`.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, pp. 3440–3448, 2016.

Nhat Ho, XuanLong Nguyen, Mikhail Yurochkin, Hung Hai Bui, Viet Huynh, and Dinh Phung. Multilevel clustering via Wasserstein means. In *International Conference on Machine Learning*, pp. 1501–1509, 2017.

Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. Supervised word mover's distance. In *Advances in Neural Information Processing Systems*, pp. 4862–4870, 2016.

Peter J. Huber and Elvezio Ronchetti. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley, Hoboken, N.J, 2nd ed edition, 2009. ISBN 978-0-470-12990-6.

Leonid Vitalievich Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pp. 199–201, 1942.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From Word Embeddings To Document Distances. In *International Conference on Machine Learning*, pp. 957–966, June 2015.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones Mathematicae*, 211(3):969–1117, 2018.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE, 2008.

Gaspard Monge. *Mémoire sur la théorie des déblais et des remblais*. De l'Imprimerie Royale, 1781.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.

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, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pp. 460–467. IEEE, 2009.

Gabriel Peyré and Marco Cuturi. Computational Optimal Transport. *arXiv:1803.00567 [stat]*, March 2018.

Benedetto Piccoli and Francesco Rossi. Generalized Wasserstein distance and its application to transport equations with source. *Archive for Rational Mechanics and Analysis*, 211(1):335–358, 2014.

Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pp. 582–588. Citeseer, 1999.

Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-Scale Optimal Transport and Mapping Estimation. *arXiv:1711.02283 [stat]*, February 2018.

Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pp. 5739–5748. PMLR, 2019.

Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015.

Sanvesh Srivastava, Cheng Li, and David B. Dunson. Scalable Bayes via Barycenter in Wasserstein Space. *arXiv:1508.05880 [stat]*, June 2018.

Guillaume Staerman, Pierre Laforgue, Pavlo Mozharovskyi, and Florence d'Alché Buc. When OT meets MOM: Robust estimation of Wasserstein distance. *arXiv:2006.10325*, 2020.

Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems*, 32:6417–6428, 2019.

Alexander Tong, Guy Wolf, and Smita Krishnaswamyt. Fixing bias in reconstruction-based anomaly detection with lipschitz discriminators. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2020.

Kaiwen Wu, Gavin Weiguang Ding, Ruitong Huang, and Yaoliang Yu. On Minimax Optimality of GANs for Robust Mean Estimation. In *International Conference on Artificial Intelligence and Statistics*, pp. 4541–4551, June 2020.

Mikhail Yurochkin, Sebastian Claici, Edward Chien, Farzaneh Mirzazadeh, and Justin Solomon. Hierarchical Optimal Transport for Document Representation. *arXiv:1906.10827 [cs, stat]*, June 2019.