Deep Barycentric Regression for Optimal Transport Map Estimation and Its Statistical Optimality

Kunwoong Kim
KAIST AI
kunwoong.kim@kaist.ac.kr

Insung Kong
UNIST
kong@unist.ac.kr

Yongdai Kim*
Seoul National University
ydkim0903@gmail.com

Abstract

The optimal transport (OT) map provides a geometric transformation for aligning probability distributions and has become a useful tool in machine learning. However, existing estimators of the OT map still exhibit a gap between sharp statistical guarantees and practical parametric estimation based on stable training objectives. Theoretical estimators achieve minimax optimal convergence rates, but they are typically nonparametric and can incur demanding implementation design or inference costs. Practical estimators are parametric and scalable, but their statistical guarantees remain underexplored, and their min-max, adversarial-like training objectives can be sensitive to optimization algorithms. We propose BROT (Barycentric Regression for Optimal Transport)$^{\diamond}$, a simple two-step method that first computes the unregularized OT plan and then fits a deep neural network (DNN) to the induced barycentric targets by least-squares regression. Under standard regularity conditions, we prove that the DNN estimator of BROT attains the minimax convergence rate, when the ground-truth OT map is Lipschitz. Numerical studies on synthetic datasets and an image dataset show that BROT provides accurate map estimates, strong target distribution matching, and competitive transport costs, compared to existing estimation methods. Experiments on two downstream tasks, single-cell perturbation prediction and unsupervised domain adaptation, further suggest that the accurate estimation of BROT can translate into stronger task performance.

1 Introduction

Optimal transport (OT) provides a geometric framework for comparing and aligning probability distributions. Given two distributions and a cost function, the OT map defines a way to transport masses from one distribution (called the source) to the other (called the target) with the minimum expected cost. As a result, the OT map not only quantifies a distributional discrepancy but also yields an explicit transportation, and has therefore been widely used in applications such as domain adaptation, self-supervised learning, generative modeling, algorithmic fairness, covariate balancing, point cloud matching, and image translation, to name just a few [1–10].

Related work. Recent theoretical works on estimation of the OT map have focused on recovering the ground-truth (population) OT map from i.i.d. data sampled from the source and target distributions. Representative examples of statistically minimax optimal estimators include the wavelet-based estimator of [11], which estimates the potential, whose gradient gives the OT map, and the 1-nearest neighbor (1NN) estimator of [12], which extends the OT plan (the optimal joint mass assignment between source and target data), to unseen data by nearest neighbor interpolation. More broadly,

*Corresponding author.
$^{\diamond}$Code at https://github.com/kwkimonline/BROT.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

[13] analyzed plug-in estimators based on smoothed densities, [14] obtained a minimax optimal estimator restricted to the semi-discrete setting (continuous source, discrete target), and [15] studied OT map estimation over general classes of convex functions without a practical training algorithm. [16] also proposed a near-optimal estimator for smooth OT maps, but with worst-case computational cost growing as the fifth power of the sample size. These works provide meaningful statistical guarantees, but the resulting estimators are mainly designed for theoretical analysis rather than practical use. In particular, they are highly nonparametric and typically require density smoothing, basis construction, nearest neighbor search, or regularization tuning, making their implementation demanding (often nearly infeasible) and their inference costs high. Appendix B.6 compares the (theoretical) computational complexity, while Table 5 reports the actual training and inference times measured in our experiments. Appendix C further covers more details about these methods.

On the algorithmic side, practical OT methods based on regularizations [17, 18] and DNN parameterizations [8, 9, 19–22] have provided several ways to estimate OT maps or plans. The Sinkhorn algorithm [17] computes an entropically regularized transport plan between two empirical distributions, but it does not by itself provide a parametric transport map. To obtain a map, [18] proposed a two-step procedure that first finds a regularized OT plan by stochastic optimization and then trains a DNN to fit the resulting regularized barycentric targets. More recent parametric (DNN-based) estimators [8, 9, 19–22] parameterize transport maps, but they typically rely on min-max, adversarial-like learning objectives, which can be sensitive to optimization algorithms. [23] derived a generalization bound for the min-max learning OT estimators, but their sharp finite-sample or minimax optimality is not known. Appendix C discusses more details about these methods.

Our goal and proposed approach. Bridging the gap between statistical optimality and practical feasibility raises a natural question: *Can we construct a parametric estimator of the OT map that is both statistically optimal and practically feasible?* In this paper, we aim to resolve this gap by proposing a learning algorithm of the OT map based on DNNs that attains statistical optimality (i.e., the minimax convergence rate) while remaining practically trainable by standard regression. The proposed algorithm, BROT (*Barycentric Regression for Optimal Transport*), proceeds in two steps. First, we solve the unregularized empirical Kantorovich problem by a standard linear program to obtain the OT plan between the source and target empirical distributions. We then use this plan to form barycentric targets for the source data. Second, we train a Lipschitz-constrained DNN to fit these targets by a standard least-squares regression. In other words, the barycentric targets serve as output variables for the source data, thereby reducing OT map estimation to a standard regression problem. Figure 1 depicts the overall framework of BROT.

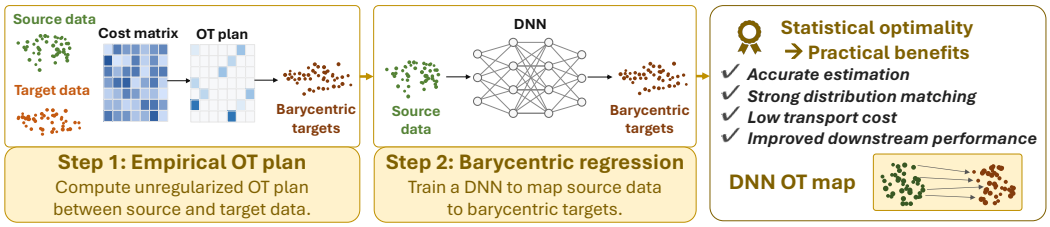


Figure 1: **Outline of our proposed framework BROT.** We first compute the unregularized OT plan, and train a DNN to fit source data to barycentric targets obtained from the OT plan.

Our main contributions are summarized as follows:

- ◇ **Statistically optimal and practically feasible OT map estimation:** We propose BROT, a barycentric regression algorithm for estimating the OT map using DNNs based on the unregularized OT plan. Under standard regularity conditions, we prove that the DNN estimator of BROT attains the minimax convergence rate when the ground-truth OT map is Lipschitz, while it remains trainable by a standard least-squares regression without min-max optimization. To our knowledge, the DNN estimator of BROT is the first parametric estimator that is simultaneously minimax optimal and practically feasible.
- ◇ **Empirical performance in statistical convergence, map quality, and downstream utility:** We numerically verify that the estimation error (i.e., the difference between the trained map and the ground-truth OT map) of BROT shows a near-minimax convergence trend, and is lower than the

estimation errors of the existing baseline methods. On synthetic and high-dimensional image datasets, BROT attains the lowest Wasserstein distance between the transported source and target distributions (i.e., produces a valid transport map), while maintaining a comparable transport cost compared with the baselines. On two downstream applications, single-cell perturbation prediction and unsupervised domain adaptation, BROT achieves the best or competitive performance compared to the baselines.

The remainder of this paper is organized as follows. Section 2 introduces preliminaries on the OT map and OT plan, and Section 3 presents our proposed algorithm, BROT. Section 4 states the main theoretical result, showing that under standard regularity conditions, the DNN estimator of BROT attains the minimax convergence rate for Lipschitz ground-truth OT map. Section 5 reports the results of three sets of experiments showing practical benefits of BROT: (i) superior finite-sample performance for estimating the OT map, (ii) validity of the trained map as a transport map (i.e., whether the transported source distribution is close to the target distribution) with low transport cost, and (iii) improved downstream task performance on two real-world applications. Section 6 concludes with a summary of our findings and directions for future work.

2 Preliminaries

Optimal transport map and its relaxation. Let P and Q denote the source and target probability distributions, respectively, supported on $\Omega \subseteq \mathbb{R}^d$. Suppose we observe data $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} P$ and $Y_1, \ldots, Y_m \overset{\text{i.i.d.}}{\sim} Q$. The corresponding empirical source and target distributions are defined as $P_n(\cdot) := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}(\cdot)$ and $Q_m(\cdot) := \frac{1}{m}\sum_{j=1}^{m} \delta_{Y_j}(\cdot)$, respectively. For a (measurable) map $\mathbf{T} : \Omega \to \Omega$, the push-forward measure of P induced by $\mathbf{T}$ is defined by $\mathbf{T}_{\#}P(\cdot) = P(\mathbf{T}^{-1}(\cdot))$. That is, $\mathbf{T}_{\#}P$ is the distribution of $\mathbf{T}(X)$ when $X \sim P$. We say $\mathbf{T}$ is a *transport map* from P to Q when it satisfies $\mathbf{T}_{\#}P = Q$. The *optimal transport (OT) map* $\mathbf{T}_0$ is a transport map that minimizes the expected transport cost[1] among all transport maps:

$$\mathbf{T}_0 := \underset{\mathbf{T}:\mathbf{T}_{\#}P=Q}{\arg\min} \int_{\Omega} \|x - \mathbf{T}(x)\|^2 dP(x). \tag{1}$$

Equation (1) is the so-called Monge problem [24]. Under mild regularity conditions, such as absolute continuity of P and finite second moments, Brenier's theorem guarantees the existence and uniqueness of $\mathbf{T}_0$ [11, 12, 25–27]. See Appendix A.2 for further theoretical backgrounds on the OT map.

Kantorovich relaxed the problem by replacing the deterministic map with a *joint distribution* on $\Omega \times \Omega$ whose marginals are P and Q [25, 26, 28]. The Kantorovich problem is formulated as

$$\inf_{\Gamma \in \Pi(P,Q)} \mathbb{E}_{(X,Y)\sim\Gamma} \|X - Y\|^2 = \inf_{\Gamma \in \Pi(P,Q)} \int_{\Omega\times\Omega} \|x - y\|^2 d\Gamma(x, y), \tag{2}$$

where $\Pi(P, Q)$ denotes the set of all joint distributions on $\Omega \times \Omega$ whose marginals are P and Q. Note that the resulting optimal transport cost defines the squared 2-Wasserstein distance, that is, $W_2(P, Q)^2 := \inf_{\Gamma\in\Pi(P,Q)} \int_{\Omega\times\Omega} \|x - y\|^2 \, d\Gamma(x, y)$.

Optimal transport plan. If $P(\cdot) = \delta_0(\cdot)$ and $Q(\cdot) = \frac{1}{2}\delta_{-1}(\cdot) + \frac{1}{2}\delta_1(\cdot)$, then the single point mass at $X = 0$ must be split between $Y = -1$ and $Y = 1$, so no deterministic map satisfies $\mathbf{T}_{\#}P = Q$, and the Monge problem in Equation (1) is infeasible. The Kantorovich problem in Equation (2), by contrast, has a unique optimal joint distribution that places mass $\frac{1}{2}$ on each of $(0, -1)$ and $(0, 1)$.

When P and Q are discrete, such as the empirical measures P_n and Q_m, the Kantorovich problem reduces to a linear program [1, 26, 29] (see Section 3.2 for its formulation). We refer to any element of $\Pi(P_n, Q_m)$ as a *transport plan*, and to the solution of the Kantorovich problem as an *optimal transport (OT) plan*. Unlike the OT plan, the OT map is generally not well-defined for discrete measures. Thus, estimating an OT map from empirical measures requires an out-of-sample extension, e.g., an approach for constructing a map from the OT plan and generalizing it to unseen data. In Section 3, we present our extension from the OT plan to a parametric map.

[1] We use the squared Euclidean cost to ensure the existence and uniqueness of the OT maps, as in [1, 9, 11, 12, 18].

3 BROT: Barycentric Regression for Optimal Transport

This section introduces our proposed OT map estimation algorithm, BROT (*Barycentric Regression for Optimal Transport*). The idea introduced in Section 3.1 is to reduce the OT map estimation to a regression problem where outputs are the *barycentric targets*, i.e., the weighted averages of target data induced by the (unregularized) OT plan. The detailed algorithm is formalized in Section 3.2, which consists of two steps: (i) computing the OT plan between the observed source and target data, and (ii) training a DNN by predicting the barycentric targets produced by the first step.

3.1 Motivation and Idea

The OT plan is more than just a joint distribution or coupling between two empirical distributions. It naturally defines barycentric targets for the source data, which in turn induce the empirical barycentric map [12–14, 18]. Here, the barycentric target of a source data point is the weighted average of the target data with weights determined by the OT plan, and the empirical barycentric map is the map that sends each source data point to its barycentric target (see Section 3.2 for the definitions).

The central question is how to use these barycentric targets when constructing a parametric estimator of the OT map. The barycentric targets and the associated empirical barycentric map provide useful information for the OT map estimation. For example, nearest neighbor interpolation of the targets is known to attain the minimax optimal rate when the ground-truth OT map is Lipschitz [12]. However, this nonparametric extension is inconvenient at inference time, since it must store the entire training dataset to perform a nearest neighbor search for each test point (see Table 5 in Appendix B.3 for empirical evidence). [18] also considered fitting a DNN to barycentric targets, but their targets are obtained from a regularized OT plan. Their main contribution is a scalable algorithm for computing the regularized OT plan between two empirical distributions, rather than a sharp statistical analysis of the resulting map estimator. In particular, it is not known whether the DNN estimator of [18] is minimax optimal. Further, [30] proposed to jointly estimate the coupling and the map by alternating optimization, but did not establish the minimax optimality. The differences from BROT are discussed in Remarks C.1 and C.2 in Appendix C.

Our goal is to estimate the OT map by a DNN with statistical optimality. For this purpose, we treat the unregularized barycentric targets as regression outputs for the source data, thereby reducing OT map estimation to a standard regression problem. We prove that the resulting least-squares estimator, implemented with carefully designed DNNs, attains the statistical minimax optimality (Section 4).

3.2 Proposed Estimation Method

Step 1: Compute the OT plan. Recall that we are given source data $\{X_i\}_{i=1}^n$ drawn from P and target data $\{Y_j\}_{j=1}^m$ drawn from Q. Let $c_{ij} := \|X_i - Y_j\|^2$ for $i \in [n]$ and $j \in [m]$. Following Equation (2), the OT plan is defined by

$$\widehat{\Gamma} := [\widehat{\gamma}_{ij}]_{i,j} = \underset{[\gamma_{ij}]_{i,j} \in \mathcal{M}_{nm}}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}\gamma_{ij}, \tag{3}$$

where $\mathcal{M}_{nm} := \{[\gamma_{ij}]_{i,j} \in \mathbb{R}_+^{n\times m} : \sum_{i=1}^n \gamma_{ij} = 1/m, \sum_{j=1}^m \gamma_{ij} = 1/n\}$ is the set of all transport plans whose marginals are P_n and Q_m. This OT plan $\widehat{\Gamma}$ derives the *barycentric targets*

$$\widetilde{Y}_i := \sum_{j=1}^{m} n\widehat{\gamma}_{ij} Y_j, \quad i \in [n],$$

which can be viewed as outputs (response variables) for the source data $\{X_i\}_{i=1}^n$.

Step 2: Fit a DNN by least-squares regression. Let Φ be a given function class. We define the estimator as a solution to the following least-squares regression problem:

$$\widehat{\mathbf{T}}_{nm}^{\Phi} := \underset{\mathbf{T}\in\Phi}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \|\widetilde{Y}_i - \mathbf{T}(X_i)\|^2. \tag{4}$$

That is, $\mathbf{T}$ is trained on the source data X_i, $i \in [n]$, to fit the barycentric targets $\widetilde{Y}_i$, $i \in [n]$.

Algorithm 1 BROT (Barycentric Regression for Optimal Transport)

Input: Source data $\{X_i\}_{i=1}^n$, Target data $\{Y_j\}_{j=1}^m$, DNN function class (architecture) Φ, Training hyperparameters (optimizer Opt, step size η, epochs T, mini-batch size B, Jacobian-penalty weight $\alpha > 0$).
(Step 1: Compute the empirical transport plan)
Build the cost matrix $\mathbf{C} = [c_{ij}]_{i\in[n],j\in[m]} \in \mathbb{R}^{n\times m}$ where $c_{ij} = \|X_i - Y_j\|^2$.
Compute $\widehat{\Gamma} = \underset{\Gamma\in\mathcal{M}_{nm}}{\arg\min} \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}\gamma_{ij}$.
Build the barycentric targets $\widetilde{Y}_i = \sum_{j=1}^m n\widehat{\gamma}_{ij}Y_j, i \in [n]$.
(Step 2: Fit a DNN by Jacobian-penalized least squares)
Initialize a model $\mathbf{T}_\theta \in \Phi$ and set $t = 0$.
while $t < T$ and not converged **do**
    Sample a mini-batch $\mathcal{B} \subset [n]$.
    Compute the regression loss $\mathcal{L}_{\text{reg}}(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \|\widetilde{Y}_i - \mathbf{T}_\theta(X_i)\|^2$.
    Sample $v_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$ for $i \in \mathcal{B}$ and compute $\widehat{\mathcal{R}}_J(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \left\|\nabla_x \langle v_i, \mathbf{T}_\theta(x)\rangle\big|_{x=X_i}\right\|^2$.
    Set $\mathcal{L}(\theta) = \mathcal{L}_{\text{reg}}(\theta) + \alpha\widehat{\mathcal{R}}_J(\theta)$.
    Update $\theta \leftarrow \text{Opt}\big(\theta, \nabla_\theta\mathcal{L}(\theta), \eta\big)$.
    $t \leftarrow t + 1$.
end while
Output: trained transport map $\mathbf{T}_\theta$.

We specify Φ in Equation (4) as a class of Lipschitz-constrained DNNs, motivated by both theory and practice. Theoretically, DNNs are known to approximate smooth functions effectively [31–33] and practically have been used for high-dimensional data and modern machine learning pipelines [34, 35], e.g., WGAN with Lipschitz constraint [36]. Let $\Phi^{\text{DNN}}(L, W; \sigma, d_{\text{out}})$ denote the class of DNNs with depth at most L, width at most W, activation function σ, and output dimension d_{out} (the full mathematical definition is deferred to Appendix A.3). For a constant $\lambda > 0$, we consider the Lipschitz-constrained DNN class $\Phi_\lambda^{\text{DNN}} := \left\{f \in \Phi^{\text{DNN}}(L, W; \sigma, d_{\text{out}}) : \text{Lip}(f) \le \lambda\right\}$, where $\text{Lip}(f) := \sup_{x\neq x'\in\Omega} \frac{\|f(x)-f(x')\|}{\|x-x'\|}$. We define the DNN estimator of BROT as $\widehat{\mathbf{T}}_{nm}^{\text{DNN}} := \widehat{\mathbf{T}}_{nm}^{\Phi_\lambda^{\text{DNN}}}$.

The Lipschitz constraint is introduced mainly for theoretical purposes, as it allows us to derive the minimax optimal convergence rate for the Lipschitz OT map in Section 4. In practice, the Lipschitz constraint can be approximately enforced in several ways: (i) by adding the Jacobian penalty $\frac{1}{n}\sum_{i=1}^n \left\|\nabla\mathbf{T}(X_i)\right\|_F^2$ to the objective in Equation (4), where $\|\cdot\|_F$ denotes the Frobenius norm [36], (ii) by constraining the spectral norms of the weight matrices [37], or (iii) by using architectures specially designed to be Lipschitz [38]. In our experiments in Section 5, we use the Jacobian penalty with a weight $\alpha > 0$, using Gaussian probes to estimate its squared Frobenius norm. We further examine the effect of the Jacobian penalty under input perturbations in Appendix B.7, showing the robustness of BROT. Algorithm 1 summarizes the practical algorithm for BROT.

Note that, for large-scale experiments, we use a chunked approximation of Step 1, where we partition the source and target data into paired chunks, compute an (unregularized) OT plan within each chunk pair, and collect the resulting barycentric targets (see Appendices B.1 and B.6 for details).

4 Theoretical Study: Minimax Optimality

An important statistical question is whether using BROT achieves the minimax optimal convergence rate for estimating the ground-truth OT map $\mathbf{T}_0$. As done in existing studies [11, 12, 15], we assume the following standard regularity conditions on the source and target densities and on the smoothness of the potential. Let $\mathcal{C}^2(\Omega)$ denote a class of functions with controlled derivatives up to order two. More precise definitions of the Hölder spaces $\mathcal{C}^\beta(\Omega), \beta > 0$ are deferred to Appendix A.1.

Assumption 4.1. *The distributions P and Q are absolutely continuous on Ω with density p and q, respectively. Moreover, there exists $\gamma > 0$ such that $\gamma^{-1} \le p(x), q(x) \le \gamma$ for all $x \in \Omega$.*

Assumption 4.2. *The potential function ϕ_0 is convex and belongs to $\mathcal{C}^2(\Omega)$. Furthermore, it satisfies $\lambda^{-1}I_d \preceq \nabla^2\phi_0(x) \preceq \lambda I_d, \forall x \in \Omega$ for some $\lambda \ge 1$. That is, $\mathbf{T}_0 = \nabla\phi_0$ is λ-Lipschitz.*

Under Assumptions 4.1 and 4.2, [11] proved the minimax lower bound $\widetilde{n}^{-2/d} \vee \widetilde{n}^{-1}$ for estimating a Lipschitz OT map, where $\widetilde{n} := n \wedge m$ and $\|f\|^2_{L^2(P)} := \int_\Omega \|f(x)\|^2 dP(x)$ for $f : \Omega \to \mathbb{R}^d$. Our theoretical goal is to show that our DNN estimator of BROT in Section 3 attains this rate, i.e., its risk is upper-bounded by the minimax rate.

For this purpose, we specially design the DNN architecture to depend on the sample sizes n and m in order to control the approximation error, following the standard convention in the literature on DNN-based nonparametric estimation [32, 33, 39]. Define $\Phi^{\mathrm{DNN}}_{nm} := \Phi^{\mathrm{DNN}}_{nm,\lambda} := \left\{ f \in \Phi^{\mathrm{DNN}}(L, W_{nm}; \sigma_{\mathrm{mix}}, d) : \mathrm{Lip}(f) \le \lambda + c' \right\}$, for a fixed $\lambda > 0$, where $c' = d\lambda + 1$, and σ_{mix} denotes a mixed ReLU/ReQU activation scheme (ReLU and ReQU applied at different nodes, for technical reasons). The depth is set as $L \asymp 1$, while the width W_{nm} grows polynomially in $\widetilde{n}$. The explicit formulation is given by Lemma A.12 in Appendix A.4.

Note that this sample size-dependent design as well as the Lipschitz constraint are mainly theoretical tools for proving Theorem 4.1. Accordingly, Theorem 4.1 provides a statistical guarantee for the exact empirical minimizer over this (theoretical) DNN class. In the experiments in Section 5, we instead use a fixed MLP regularized by a Jacobian penalty, and Appendix B.2 empirically suggests that BROT performs similarly across various architectures and Jacobian penalty weights.

Denote the resulting DNN estimator over this class by

$$\widehat{\mathbf{T}}^{\mathrm{DNN}}_{nm} := \operatorname*{arg\,min}_{\mathbf{T} \in \Phi^{\mathrm{DNN}}_{nm}} \sum_{i=1}^{n} \left\| \widetilde{Y}_i - \mathbf{T}(X_i) \right\|^2. \tag{5}$$

Theorem 4.1 below gives the convergence rate of this DNN estimator $\widehat{\mathbf{T}}^{\mathrm{DNN}}_{nm}$, whose full proof and technical conditions are deferred to Appendix A.4 and Appendix A.1, respectively.

Theorem 4.1 (Minimax convergence rate for the DNN estimator of BROT)**.** *Under Assumptions 4.1, 4.2, and regularity conditions described in Appendix A.1, with $\Omega = [0,1]^d$, we have that*

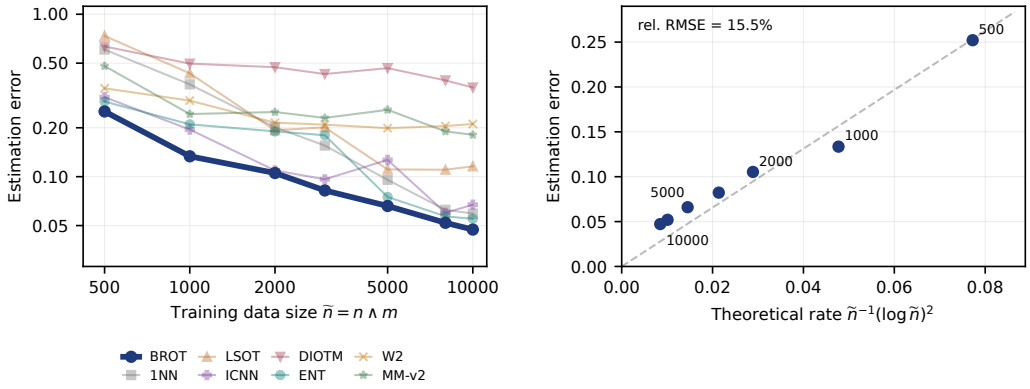
$$\mathbb{E} \left\| \widehat{\mathbf{T}}^{\mathrm{DNN}}_{nm} - \mathbf{T}_0 \right\|^2_{L^2(P)} \lesssim \begin{cases} \widetilde{n}^{-1} \log \widetilde{n}, & d = 1, \\ \widetilde{n}^{-1} (\log \widetilde{n})^2, & d = 2, \\ \widetilde{n}^{-2/d} \log \widetilde{n}, & d \ge 3. \end{cases}$$

Note that the upper bound on the risk in Theorem 4.1 matches the minimax lower bound $\widetilde{n}^{-2/d} \vee \widetilde{n}^{-1}$ for the ground-truth OT map that is Lipschitz, up to logarithmic factors. That is, the DNN estimator of BROT is (nearly) minimax optimal. It is also noteworthy that Theorem 4.1 does not trivially follow from standard DNN regression theory, since the barycentric targets are computed from the same empirical OT plan and hence depend jointly on all source and target data. Our proof handles this dependence, controls the population error by the empirical error using Lipschitz continuity similarly to [12], and constructs a DNN with the required approximation error. In particular, the Lipschitz condition is used to extend the estimation error at the observed source data to the population error.

Further, although several existing estimators of the OT map are also known to be minimax optimal, including [11, 12, 15], most of them are not practically applicable. Practically feasible learning algorithms for [11] and [15] are not available, and [12] requires unnecessarily large memory and computation at inference time. Moreover, our experiments in Section 5 empirically show that the DNN estimator of BROT outperforms the estimator of [12] even though both are minimax optimal. This empirical observation suggests that not only statistical optimality but also practical feasibility is important for the OT map estimation, and BROT is such an algorithm.

5 Experiments

In this section, we numerically assess the advantages of BROT from three perspectives. First, in Section 5.1, using a synthetic dataset with a known Lipschitz ground-truth OT map, we examine whether the estimation error of BROT follows the theoretical minimax rate in Theorem 4.1, and compare with existing estimation methods. Second, in Section 5.2, we evaluate the quality of the trained (estimated) OT map in terms of target distribution matching and transport cost, on two synthetic datasets and a high-dimensional image dataset. Third, in Sections 5.3 and 5.4, we investigate whether the improved quality of the trained OT map translates into better performance on two downstream tasks: single-cell drug-perturbation prediction and unsupervised domain adaptation.



(a) Training data size $\widetilde{n}$ vs. estimation error.

(b) Theoretical rate vs. estimation error.

Figure 2: **Statistical convergence to the ground-truth OT map.** (a) Training data size $\widetilde{n}$ vs. estimation error. Points show mean estimation errors; the corresponding standard-error bands are shown in Figure 3 of Appendix B.2. (b) Theoretical rate $\widetilde{n}^{-1}(\log \widetilde{n})^2$ vs. estimation error for BROT, with a linear regression line and its relative RMSE (rel. RMSE).

Baselines. Throughout Sections 5.1 to 5.4, we compare BROT with eight existing methods for estimating the OT map: 1NN [12], LSOT [18], ICNN [19], OTP [21], DIOTM [22], MM-v2 and W2 [40], and the entropic map estimator (ENT) [41]. For each method, we tune the hyperparameters by grid search and select the configuration that minimizes the validation metric specified in each experiment. More implementation details for each methods are provided in Appendix B.1 and Appendix C.

5.1 Statistical Convergence to the Ground-Truth Optimal Transport Map

Task setup. We study how the estimation error (roughly, $\mathbb{E}\|\widehat{\mathbf{T}}^{\mathrm{DNN}}_{nm} - \mathbf{T}_0\|^2_{L^2(P)}$ on unseen test data) changes as the training size n increases, in a setting where $\mathbf{T}_0$ is a known Lipschitz OT map. We construct a synthetic dataset on $\mathbb{R}^2$ whose source distribution $P = \mathcal{N}(0, I_2)|_R$ is the standard Gaussian truncated to the disk of radius $R = 3.5$. The ground-truth OT map is $\mathbf{T}_0(x) = (a_1 x_1 + b_1 x_1^3, a_2 x_2 + b_2 x_2^3)$ for $x = [x_1, x_2]^\top$, with $(a_1, a_2) = (1.5, 0.8)$ and $(b_1, b_2) = (0.3, 0.1)$. Its construction and derivation are given in Appendix B.2.

Implementation details. For $n = m \in \{500, 1{,}000, 2{,}000, 3{,}000, 5{,}000, 8{,}000, 10{,}000\}$, we split the source and target data into an 80:20 training/validation split. For each method, the training iteration and hyperparameters are selected using the 2-Wasserstein distance between the transported source and the target validation data. We then evaluate the trained maps on a test dataset of size $2{,}000$, and repeat the procedure with 10 random seeds for each method and sample size and report the mean.

Results. Figure 2(a) shows that the estimation error of BROT decreases monotonically with $\widetilde{n}$, and is lower than all baselines for all values of $\widetilde{n}$. Note that we omit OTP from the panel as its estimation error is markedly larger than the other methods (≈ 2.6 for all $\widetilde{n}$). Although BROT and 1NN have the same theoretical rate, BROT achieves smaller errors than 1NN at every value of $\widetilde{n}$, suggesting that BROT is empirically more accurate. Figure 2(b) compares the estimation error of BROT with the theoretical minimax rate $\widetilde{n}^{-1}(\log \widetilde{n})^2$. The relative RMSE of the fitted linear regression line is approximately 15.5%, indicating that the estimation error follows the theoretical minimax rate trend over the range of $\widetilde{n}$. Figure 5 of Appendix B.2 further shows that the estimation performance of BROT is not significantly sensitive to architectures and Jacobian penalty weights.

5.2 Quality of Maps: Distribution Matching and Transport Cost

Beyond the estimation error for a known Lipschitz ground-truth OT map studied in Section 5.1, we assess the quality of trained maps in settings where the ground-truth OT map is unknown. Specifically, we evaluate map quality in terms of (i) target distribution matching and (ii) transport cost: the former measures how well the learned maps match the target distribution (how much the learned maps are valid transport maps), while the latter measures the cost-efficiency of trained transport maps.

Table 1: **Comparison of map quality: distribution matching and transport cost.** Best values of `Wass` are **bolded** and '$*$' marks the failed cases due to algorithm instability. Results are averaged over five random trials, and standard deviations are reported in Table 4 of Appendix B.3.

	Gaussian $\to$ Rectangle		Gaussian $\to$ Ellipse		AFHQv2 wild $\to$ cat	
Method	`Wass`	`TC`	`Wass`	`TC`	`Wass`	`TC`
1NN [12]	0.126	1095.67	0.207	1122.30	54.819	3296.54
LSOT [18]	0.111	1095.69	0.172	1124.35	53.263	2627.29
ICNN [19]	0.211	1097.51	0.171	1122.54	$*$	$*$
OTP [21]	0.258	1099.17	0.596	1129.88	50.651	1408.23
DIOTM [22]	0.123	1098.21	0.228	1123.99	51.000	1453.99
MM-v2 [40]	0.120	1096.57	0.174	1125.20	205.078	40213.59
W2 [40]	0.191	1096.44	0.163	1124.44	51.811	2366.89
ENT [41]	0.209	1097.41	0.257	1125.98	45.708	1955.80
BROT	**0.110**	1095.65	**0.155**	1125.08	**45.484**	1993.85

Task setup. We consider two 2D synthetic datasets and one high-dimensional image dataset. The two synthetic cases are Gaussian $\to$ Uniform rectangle and Gaussian $\to$ Uniform ellipse (visualizations in Figure 6 in Appendix B.3). The image dataset is AFHQv2 wild $\to$ cat dataset [42] at 256×256 resolution. Rather than in raw image pixel space, where the Euclidean distance can be a poor similarity metric, we operate in the representation space of a pretrained visual encoder. We utilize a frozen Stable-Diffusion VAE [43] as the representation encoder, which maps each image to a $4{,}096$-dimensional representation.

For the synthetic cases, we draw $n = 3{,}000$ source and $m = 2{,}000$ target data and split them 8:1:1 into train/validation/test, and we use the validation data to select the best model with the lowest validation loss among all epochs. For AFHQv2, we use $n = 3{,}000$ wild and $m = 4{,}000$ cat images with separate 500 test source and 500 test target images, all encoded into representations using the VAE encoder. We run each method for five random trials and report the performance on the test data.

Implementation details. As the performance metrics, we consider target distribution matching by the 2-Wasserstein distance (`Wass`) and the displacement by the transport cost (`TC`). Lower values are better for both metrics. For BROT, LSOT, OTP, and DIOTM, we use 4-layer MLPs with 64 hidden units in the synthetic cases and wider MLPs, with hidden size up to $2{,}048$, on AFHQv2 dataset. ICNN uses the input-convex network architecture of [19], tuned separately for each setting. Full implementation details, including hyperparameter tuning and architectures, are provided in Appendix B.1.

Results. Table 1 compares the methods in terms of map quality on the three datasets, where we focus on `Wass` as the primary metric because a low `TC` alone does not imply a valid transport map, e.g., the trivial identity map attains the smallest `TC` but has $\mathbf{T}_{\#}P \neq Q$. BROT attains the lowest `Wass` on every dataset among all methods, while maintaining a comparable `TC`, suggesting that BROT provides a high-quality transport map. Experimental details and full results are provided in Appendix B.3.

In particular, BROT improves over 1NN in `Wass` while showing similar `TC`, showing the practical benefit of replacing the discontinuous nearest neighbor operation with DNN. ICNN also performs reasonably well on the two synthetic datasets, but diverges on AFHQv2 ($d = 4{,}096$). This is due to optimization instability in the ICNN min-max objective on high-dimensional data, despite extensive hyperparameter tuning (see Appendix B.1). LSOT, OTP, and DIOTM show competitive `Wass` on synthetic datasets but degrade on AFHQv2, indicating that BROT is a practically favorable DNN-based method. MM-v2 and W2 also perform reasonably well on the synthetic datasets but show larger `Wass` than BROT on AFHQv2. ENT shows larger `Wass` on synthetic datasets than BROT.

The runtime comparison in Table 5 of Appendix B.3 shows that BROT trains faster than the baseline DNN-based algorithms with min-max learning algorithms (ICNN, OTP, DIOTM) while keeping inference cost lower than 1NN. BROT also yields a smaller `Wass` than the regularized counterpart (i.e., LSOT) in Table 1, and in fact can train faster than LSOT. We further assess the transported representations in image space using decoded AFHQv2 images and FID. BROT also achieves the lowest FID, and the decoded image samples and full results are provided in Appendix B.3.

Table 2: **Comparison of single-cell perturbation prediction performance.** The results are averaged across the 35 drug perturbations from 4I dataset [47]. Among the methods (excluding Identity) that satisfy MMD ≤ 0.028, the best value is **bolded** and the second-best is underlined for each metric. Standard deviations are reported in Table 7 of Appendix B.4.

Method	MMD ↓	L_2 difference ↓	r-std ↑	TC ↓
Identity	0.028	1.213	0.917	0.000
1NN [12]	**0.002**	**0.189**	0.990	6.745
LSOT [18]	0.012	0.287	0.982	4.381
ICNN [19]	0.006	0.258	0.987	4.329
OTP [21]	0.035	1.435	0.917	0.514
DIOTM [22]	**0.002**	0.255	0.990	3.993
MM-v2 [40]	**0.002**	0.226	0.990	3.984
W2 [40]	**0.002**	0.199	0.990	3.934
ENT [41]	**0.002**	0.204	0.992	6.291
BROT	**0.002**	0.194	**0.993**	**3.928**

5.3 Application 1: Single-Cell Perturbation Prediction

Task setup. Predicting how an unperturbed cell would respond to a drug perturbation is a canonical task in modern computational biology [44–46]. Since cell measurements are destructive, we only observe two unpaired data distributions, a control source distribution ρ_c and a perturbed target distribution ρ_k, but never observe the same cell in both states. Following [44], we formulate this task as estimating the OT map $\mathbf{T}^{(k)} : \rho_c \to \rho_k$ for each perturbation k. The goal is to match the transported control distribution $(\mathbf{T}^{(k)})_{\#}\rho_c^{\text{test}}$ to the treated cell distribution, where ρ_c^{test} denotes the test distribution of control cells unseen during training. We compare BROT with several baselines, including CellOT [44], a state-of-the-art ICNN-based method for single-cell perturbation prediction. We use 4I benchmark dataset [47] with 35 drug perturbations, following the same setup as CellOT [44]. For each perturbation, we split the control and treated data into training and test data with an 80:20 ratio. Within the training split, we further hold out 20% as validation data for hyperparameter selection, train maps on the remaining 80%, and then evaluate them on the test data.

Implementation details. Following the evaluation protocol of [44], we compare the predicted (transported) control cell distribution with the ground-truth perturbed cell distribution. Each cell is represented by a 48-dimensional protein-marker feature vector. We use four performance metrics. First, MMD is the Gaussian-kernel maximum mean discrepancy (MMD) averaged over 50 bandwidths, measuring the distributional discrepancy between the transported control and treated distributions. Second, L_2 difference is the Euclidean distance between the feature means of the two distributions, measuring their feature-wise closeness. Third, r-std is the Pearson correlation between the marker standard deviations of the transported control and treated distributions, measuring whether the transported control captures the marker-wise spread of the treated cells. We additionally calculate the transport cost TC between each unperturbed cell and its prediction.

Results. Table 2 reports the average results over the 35 drug perturbations. It shows that BROT achieves the best MMD, r-std, and TC, while obtaining the second-best L_2 among the methods that satisfactorily match the target distribution (MMD ≤ 0.028). In contrast, OTP fails to reach MMD ≤ 0.028, which is the value of Identity (no transport). Although DIOTM also attains MMD ≤ 0.028, it is noticeably weaker in terms of the L_2 difference. Compared with 1NN, BROT achieves substantially lower TC with the same MMD level. MM-v2, W2, and ENT attain similar MMDs as BROT, but show lower r-stds and higher TCs.

5.4 Application 2: Unsupervised Domain Adaptation

Task setup. In unsupervised domain adaptation, a classifier trained using labeled source data and unlabeled target data is evaluated on the target domain, whose distribution differs from that of the source. Existing conventional OT-based approaches [18, 48] first transport the source data toward the unlabeled target data and then train the classifier on the transported source data. Although a more accurate OT map does not directly guarantee higher target-domain test accuracy, OT-based domain

Table 3: **Comparison of unsupervised domain adaptation performance.** The performance is evaluated by the per-class mean accuracy (%) on the target-domain test data. Best results are **bolded** and the second places are underlined, and the standard deviations are in Table 8 of Appendix B.5.

	USPS → MNIST		VISDA-17 (synthetic → real)	
Method	CLIP-ViT/B-32	DINOv2-ViT/B-14	CLIP-ViT/B-32	DINOv2-ViT/B-14
Identity	52.05	58.07	64.04	60.66
1NN [12]	80.34	75.25	74.17	74.39
LSOT [18]	30.12	52.20	76.87	81.57
ICNN [19]	75.76	54.70	74.12	77.77
OTP [21]	49.98	57.36	76.15	64.51
DIOTM [22]	82.71	77.46	76.45	27.05
MM-v2 [40]	84.00	**77.73**	76.95	74.81
W2 [40]	80.50	75.49	74.70	70.40
ENT [41]	79.76	76.21	73.88	74.28
BROT	**84.83**	77.60	**79.34**	**81.59**

adaptation is a standard downstream evaluation [18, 48]. Following [18], we run all methods on a fixed representation space using a pretrained visual encoder, where semantically similar images lie close and transport is more likely to preserve class information than in raw pixel space.

Implementation details. We evaluate on two benchmark datasets, USPS [49] → MNIST [50] and VISDA-17 synthetic → real [51], and with two pretrained encoders, CLIP-ViT/B-32 [52] and DINOv2-ViT/B-14 [53]. After training the OT map, we train a classifier on the transported source data. We report the per-class mean accuracy [54] averaged over three random trials. Further details such as classifier network selection and validation settings are given in Appendix B.5.

Results. Table 3 reports the per-class mean accuracy on the target domain. BROT performs best on VisDA with both encoders and on USPS → MNIST with CLIP, while MM-v2 is the strongest baseline and slightly outperforms BROT by 0.13pp on USPS → MNIST with the DINOv2 encoder. These results suggest that the accurate estimation of the OT maps using BROT translates into improved domain adaptation performance under various foundation visual encoders and datasets. Furthermore, we compare BROT with several end-to-end OT-based domain adaptation methods that jointly train the encoder, classifier, and transport plan. The results are reported in Table 9 of Appendix B.5, further showing that BROT with fixed representations is competitive with these end-to-end methods.

6 Conclusion and Discussion

We proposed BROT, a two-step algorithm for estimating the OT map that first computes the unregularized OT plan between the empirical source and target distributions and then fits a Lipschitz-constrained DNN to the induced barycentric targets by least-squares regression. On the theoretical side, we show that the DNN estimator of BROT attains the minimax convergence rate when the ground-truth OT map is Lipschitz. On the empirical side, BROT (i) attains the smallest estimation error, (ii) produces high-quality maps, and (iii) shows improved performance on two downstream applications.

We considered a Lipschitz-constrained DNN based on the empirical OT plan using full data, whereas our practical implementation uses an MLP with a Jacobian penalty and, for large-scale experiments, computes the empirical OT plan using chunked data. Analyzing the MLP with a Jacobian penalty and the statistical effect of the chunking are promising future directions.

Further, extending the analysis to smoother classes, such as Hölder classes, would provide a more general statistical guarantee. Several existing estimators [11, 15] are already known to be minimax optimal for estimating Hölder-smooth OT maps, but they are not accompanied by practical training algorithms. Designing a practically trainable DNN estimator that is minimax optimal for such smoother classes is therefore a promising direction for future work. The statistical optimality of existing neural estimators, including ICNN-based and semi-dual estimators [23], also remains open.

Acknowledgement

KK was partly supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Support Program (KAIST)), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00713, Meta-learning applicable to real-world problems), and the InnoCORE program of the Ministry of Science and ICT (No. N10250156).

YK was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)], the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00556079), and the Institute of Information & Communications Technology Planning & Evaluation (IITP) with a grant funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea in connection with the Development of Unified Reasoning Technology Mimicking Human Cognition for Hierarchical Understanding and Unbounded Problem Solving (No. RS-2026-25522672).

The authors declare no competing interests. The authors truly thank the anonymous reviewers and the area chair for their thoughtful comments, which helped improve this work. KK sincerely thanks Woncheol Jang, Johan Lim, Minwoo Chae, and Gunwoong Park for their insightful discussions and valuable feedback during the development of this work.

References

[1] Gabriel Peyré and Marco Cuturi. Computational optimal transport, 2020.

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 214–223. JMLR.org, 2017.

[4] Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 849–858. PMLR, 16–18 Apr 2019.

[5] Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2357–2365. PMLR, 09–15 Jun 2019.

[6] Taekyung Lee, Jaemoo Choi, Jaewoong Choi, and Myungjoo Kang. Unpaired point cloud completion via unbalanced optimal transport. In *Forty-second International Conference on Machine Learning*, 2025.

[7] Jose Blanchet, Jiajin Li, Markus Pelger, and Greg Zanotti. Automatic outlier rectification via optimal transport. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[8] Jiaojiao Fan, Shu Liu, Shaojun Ma, Hao-Min Zhou, and Yongxin Chen. Neural monge map estimation and its applications. *Transactions on Machine Learning Research*, 2023. Featured Certification.

[9] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023.

[10] Kunwoong Kim, Insung Kong, Jongjin Lee, Minwoo Chae, Sangchul Park, and Yongdai Kim. Fairness through matching. *Transactions on Machine Learning Research*, 2025.

[11] Jan-Christian Hütter and Philippe Rigollet. Minimax estimation of smooth optimal transport maps, 2020.

[12] Tudor Manole, Sivaraman Balakrishnan, Jonathan Niles-Weed, and Larry Wasserman. Plugin estimation of smooth optimal transport maps, 2024.

[13] Nabarun Deb, Promit Ghosal, and Bodhisattva Sen. Rates of estimation of optimal transport maps using plug-in estimators via barycentric projections. In *NeurIPS*, 2021.

[14] Aram-Alexandre Pooladian, Vincent Divol, and Jonathan Niles-Weed. Minimax estimation of discontinuous optimal transport maps: The semi-discrete case, 2023.

[15] Vincent Divol, Jonathan Niles-Weed, and Aram-Alexandre Pooladian. Optimal transport map estimation in general function spaces. *The Annals of Statistics*, 53(3):963 – 988, 2025.

[16] Adrien Vacher, Boris Muzellec, Francis Bach, François-Xavier Vialard, and Alessandro Rudi. Optimal estimation of smooth transport maps with kernel SoS. *SIAM Journal on Mathematics of Data Science*, 6(2):311–342, 2024.

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[18] Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large scale optimal transport and mapping estimation. In *International Conference on Learning Representations*, 2018.

[19] Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6672–6681. PMLR, 13–18 Jul 2020.

[20] Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative modeling with optimal transport maps. In *International Conference on Learning Representations*, 2022.

[21] Jaemoo Choi, Jaewoong Choi, and Dohyun Kwon. Overcoming spurious solutions in semi-dual neural optimal transport: A smoothing approach for learning the optimal transport plan. In *Forty-second International Conference on Machine Learning*, 2025.

[22] Jaemoo Choi, Yongxin Chen, and Jaewoong Choi. Improving neural optimal transport via displacement interpolation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[23] Roman Tarasov, Petr Mokrov, Milena Gazdieva, Evgeny Burnaev, and Alexander Korotin. A statistical learning perspective on semi-dual adversarial neural optimal transport solvers. In *International Conference on Learning Representations*, 2026.

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

[25] C. Villani and American Mathematical Society. *Topics in Optimal Transportation*. Graduate studies in mathematics. American Mathematical Society, 2003.

[26] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.

[27] Florian F. Gunsilius. On the convergence rate of potentials of brenier maps. *Econometric Theory*, 38(2):381–417, 2022.

[28] Lev Kantorovich. On the translocation of masses. *Journal of Mathematical Sciences*, 133:1381–1382, 2006.

[29] 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, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[30] Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation for discrete optimal transport. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[31] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[32] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[33] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48(4):1875 – 1897, 2020. Cited by: 128; All Open Access, Green Open Access.

[34] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[36] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[37] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[38] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 854–863. PMLR, 06–11 Aug 2017.

[39] Taiji Suzuki. Adaptivity of deep reLU network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

[40] Alexander Korotin, Lingxiao Li, Aude Genevay, Justin Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? a continuous Wasserstein-2 benchmark. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[41] Aram-Alexandre Pooladian and Jonathan Niles-Weed. Entropic estimation of optimal transport maps, 2021.

[42] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[44] Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768, 2023.

[45] Sisi Chen, Paul Rivaud, Jong H. Park, Tiffany Tsou, Emeric Charles, John R. Haliburton, Flavia Pichiorri, and Matt Thomson. Dissecting heterogeneous cell populations across drug and disease conditions with popalign. *Proceedings of the National Academy of Sciences of the United States of America*, 117(46):28784–28794, 11 2020.

[46] Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019.

[47] Gabriele Gut, Markus D. Herrmann, and Lucas Pelkmans. Multiplexed protein maps link subcellular organization to cellular states. *Science*, 361(6401), 2018.

[48] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

[49] J.J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[50] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[51] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[53] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. Featured Certification.

[54] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

[55] Denis Belomestny, Alexey Naumov, Nikita Puchkin, and Sergey Samsonov. Simultaneous approximation of a smooth function and its derivatives by deep neural networks with piecewise-polynomial activations. *Neural Networks*, 161:242–253, 2023.

[56] Guohao Shen, Yuling Jiao, Yuanyuan Lin, and Jian Huang. Differentiable neural networks with repu activation: with applications to score estimation and isotonic regression, 2024.

[57] Yuling Jiao, Guohao Shen, Yuanyuan Lin, and Jian Huang. Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics*, 51(2):691–716, 2023.

[58] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, 2018.

[59] Kilian Fatras, Thibault Séjourné, Nicolas Courty, and Rémi Flamary. Unbalanced minibatch optimal transport; applications to domain adaptation. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[60] Pei Yang, Yuhang Zhuang, and Qi Tan. Elastic optimal transport: Theory, application, and empirical evaluation. In *International Conference on Learning Representations*, 2026.

[61] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

A Theoretical Studies

This section introduces precise mathematical definitions, assumptions, and theoretical background on OT map to support Section 2, and provides proofs for our main theoretical result in Section 4.

A.1 Notations, Definitions, and Assumptions

Notations. For $\nu \in [1, \infty)$ and $v = (v_1, \ldots, v_d)^\top \in \mathbb{R}^d$, define $\|v\|_\nu = (\sum_{j=1}^d |v_j|^\nu)^{1/\nu}$. We write $\|v\| = \|v\|_2$ and $\|v\|_\infty = \max_j |v_j|$; the same conventions apply to vectors of other finite dimensions. For a matrix A, $\|A\|_{\text{op}}$ is its Euclidean operator norm and $\|A\|_F = (\sum_{i,j} A_{ij}^2)^{1/2}$ is its Frobenius norm. For a scalar function f on Ω, a subset $D \subseteq \Omega$, and a probability measure P, define

$$\|f\|_{L^\nu(\Omega)} = \left(\int_\Omega |f(x)|^\nu \, dx \right)^{1/\nu},$$
$$\|f\|_{L^\nu(P)} = \left(\int_\Omega |f(x)|^\nu \, dP(x) \right)^{1/\nu}, \quad \|f\|_{\infty, D} = \sup_{x \in D} |f(x)|.$$

The spaces $L^\nu(\Omega)$ and $L^\nu(P)$ consist of measurable functions with finite respective norms, and $\|f\|_\infty$ abbreviates $\|f\|_{\infty,\Omega}$. For vector-valued $f = (f_1, \ldots, f_{d_{\text{out}}})^\top$, the L^ν norms use $\|f(x)\|_2$ inside the integral, while $\|f\|_{\infty,D} := \max_j \sup_{x \in D} |f_j(x)|$. The notation $\|f\|_{L^\infty(D)}$ also denotes this uniform norm for the continuous functions used below.

For $k \in \mathbb{N}$, write $[k] = \{1, \ldots, k\}$. For any $a, b \in \mathbb{R}$, denote $a \vee b = \max\{a, b\}$, $a \wedge b = \min\{a, b\}$, and $a_+ = a \vee 0$. Let $\lfloor a \rfloor$ denote the greatest integer less than or equal to a. Given a set $\Omega \subset \mathbb{R}^d$, the diameter of Ω is defined as $\text{diam}(\Omega) = \sup\{\|x - y\| : x, y \in \Omega\}$. For all $x \in \mathbb{R}^d$ and $\varepsilon > 0$, let $\mathbb{B}(x, \varepsilon) = \{y \in \mathbb{R}^d : \|x - y\| \le \varepsilon\}$ be the ϵ-ball with center x. For two given sequences $(a_n)_{n \ge 1}$ and $(b_n)_{n \ge 1}$ depending on n, we write $a_n \lesssim b_n$ if there exists some $C > 0$ not depending on n such that $a_n \le C b_n$ for all $n \ge 1$. We write $a_n \asymp b_n$ if $b_n \lesssim a_n \lesssim b_n$.

Hölder spaces. Let $\Omega \subset \mathbb{R}^d$ be a support. Uniform norms on Ω and its subsets use the conventions above. We write $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Throughout this study, $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^d$.

Definition A.1 (Partial derivatives)**.** *Let $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_d]^\top \in \mathbb{N}_0^d$ with $|\boldsymbol{\alpha}| := \sum_{i=1}^d \alpha_i$. For $x = [x_1, \ldots, x_d]^\top \in \Omega$, we define the partial derivative of f of order $\boldsymbol{\alpha}$ as*

$$\partial^{\boldsymbol{\alpha}} f(x) := \frac{\partial^{|\boldsymbol{\alpha}|} f(x)}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}. \tag{6}$$

For $k \in \mathbb{N}_0$, let $\mathcal{C}^k(\Omega)$ denote the space of k-times continuously differentiable functions such that the partial derivatives $\partial^{\boldsymbol{\alpha}} f$ exist and are continuous for all $|\boldsymbol{\alpha}| \le k$.

For a scalar f and derivative order ℓ, write $\|\nabla^\ell f\|_{\infty,D} := \max_{|\boldsymbol{\alpha}|=\ell} \|\partial^{\boldsymbol{\alpha}} f\|_{\infty,D}$; vector-valued derivatives also take the maximum over output coordinates.

Definition A.2 (Hölder spaces $\mathcal{C}^\beta(\Omega)$ with order $\beta > 0$)**.** *Let $\beta > 0$ be noninteger. Define $s := \lfloor \beta \rfloor \in \mathbb{N}_0$ and $r := \beta - \lfloor \beta \rfloor \in (0, 1)$. For $f : \Omega \to \mathbb{R}$, define*

$$\|f\|_{\mathcal{C}^s(\Omega)} := \max_{|\boldsymbol{\alpha}| \le s} \|\partial^{\boldsymbol{\alpha}} f\|_\infty \text{ and } [f]_{\mathcal{C}^{s,r}(\Omega)} := \max_{|\boldsymbol{\alpha}|=s} \sup_{\substack{x,y \in \Omega \\ x \ne y}} \frac{|\partial^{\boldsymbol{\alpha}} f(x) - \partial^{\boldsymbol{\alpha}} f(y)|}{\|x - y\|^r}. \tag{7}$$

and define the Hölder norm of order β as $\|f\|_{\mathcal{C}^\beta(\Omega)} := \|f\|_{\mathcal{C}^s(\Omega)} + [f]_{\mathcal{C}^{s,r}(\Omega)}$. The Hölder space of order β is then defined as

$$\mathcal{C}^\beta(\Omega) := \left\{ f \in \mathcal{C}^s(\Omega) : \|f\|_{\mathcal{C}^\beta(\Omega)} < \infty \right\}. \tag{8}$$

Furthermore, for $H > 0$, we define the (closed) Hölder ball of radius H as

$$\mathcal{C}^\beta(\Omega, H) := \left\{ f \in \mathcal{C}^\beta(\Omega) : \|f\|_{\mathcal{C}^\beta(\Omega)} \le H \right\}. \tag{9}$$

For an integer $\beta = k$, we use the space $\mathcal{C}^k(\Omega)$ and its norm defined above.

Definition A.3 (Hölder spaces of vector-valued maps)**.** *For a vector-valued map* $\mathbf{T} = (T_1, \ldots, T_{d_{\text{out}}})^\top : \Omega \to \mathbb{R}^{d_{\text{out}}}$ *with* $d_{\text{out}} \in \mathbb{N}$*, set*

$$\|\mathbf{T}\|_{\mathcal{C}^\beta(\Omega)} := \max_{1 \le \ell \le d_{\text{out}}} \|T_\ell\|_{\mathcal{C}^\beta(\Omega)}. \tag{10}$$

We then write

$$\mathbf{T} \in \mathcal{C}^\beta(\Omega) \iff T_\ell \in \mathcal{C}^\beta(\Omega) \text{ for all } \ell \in [d_{\text{out}}]. \tag{11}$$

Regularity conditions. In addition, the standard regularity conditions on the support Ω are:

(S1) Ω is a compact, convex set with nonempty interior such that $\Omega \subseteq [0,1]^d$.

(S2) Ω is a standard set in the sense that there exist $\epsilon_0, \delta_0 > 0$ such that for all $x \in \Omega$ and $\epsilon \in (0, \epsilon_0)$, $\text{Leb}\big(\mathbb{B}(x,\epsilon) \cap \Omega\big) \ge \delta_0 \,\text{Leb}\big(\mathbb{B}(x,\epsilon)\big)$, where Leb is the Lebesgue measure on $\mathbb{R}^d$ and $\mathbb{B}(x,\epsilon)$ is the Euclidean ball of radius ϵ centered at x.

Remark on the regularity class of Assumption 4.2. The two-sided Hessian bound $\lambda^{-1} I_d \preceq \nabla^2 \phi_0(x) \preceq \lambda I_d$ in Assumption 4.2 makes ϕ_0 both λ-smooth and λ^{-1}-strongly convex on Ω, so $\mathbf{T}_0 = \nabla \phi_0$ is in fact λ-bi-Lipschitz: both $\mathbf{T}_0$ and $\mathbf{T}_0^{-1}$ are λ-Lipschitz. These assumptions follow the standard regularity regime used in minimax OT-map estimation [11, 12, 15], and we refer to it as 'Lipschitz OT map' throughout this paper, following the convention.

A.2 Existence, Uniqueness, and Duality of Optimal Transport Maps

Based on the Kantorovich relaxation in Equation (2), the 2-Wasserstein distance admits a convenient dual formulation.

Theorem A.1 (Kantorovich duality for quadratic cost)**.** *Assume that* P *and* Q *have finite second moments. Then*

$$\begin{aligned} W_2^2(P,Q) &= \inf_{\Gamma \in \Pi(P,Q)} \int \|x-y\|^2 d\Gamma(x,y) \\ &= \sup_{\substack{\varphi \in L^1(P), \psi \in L^1(Q) \\ \varphi(x)+\psi(y) \le \|x-y\|^2 \ \forall (x,y) \in \Omega \times \Omega}} \left\{ \int \varphi dP + \int \psi dQ \right\} \\ &= \sup_{u \in L^1(P)} \left\{ \int u dP + \int u^c dQ \right\}, \end{aligned} \tag{12}$$

where the c-transform is $u^c(y) := \inf_{x \in \Omega} \{\|x-y\|^2 - u(x)\}$*. Moreover, writing* ϕ *for a convex potential and* ϕ^* *for its Legendre-Fenchel conjugate, we obtain*

$$W_2^2(P,Q) = \int \|x\|^2 dP(x) + \int \|y\|^2 dQ(y) - 2 \inf_{\phi:\text{convex}} \left\{ \int \phi dP + \int \phi^* dQ \right\}. \tag{13}$$

Thus, computing $W_2^2(P,Q)$ reduces to solving

$$\inf_{\phi:\text{convex}} \left\{ \int \phi dP + \int \phi^* dQ \right\}.$$

More detailed derivations and proofs are given in previous literature, e.g., [1, 25, 26].

Along with this duality result, we recall the fundamental existence and uniqueness theorem for the OT map. Let $\mathcal{P}_{\text{ac}}(\Omega)$ denote the set of absolutely continuous probability distributions on Ω.

Theorem A.2 (Brenier's Theorem)**.** *Suppose that* $P \in \mathcal{P}_{\text{ac}}(\Omega)$ *and that* P, Q *have finite second moments. Then there exists a convex function* ϕ_0 *such that the unique (*P*-almost surely)* W_2*-optimal transport map is given by* $\mathbf{T}_0 = \nabla \phi_0$*. Moreover, an optimal dual pair for Equation (12) is* $\varphi_0(x) = \|x\|^2 - 2\phi_0(x), \psi_0(y) = \|y\|^2 - 2\phi_0^*(y)$*, and* (ϕ_0, ϕ_0^*) *is the pair (potential function and its conjugate) generating the map* $\mathbf{T}_0 = \nabla \phi_0$*.*

Corollary A.3 (Semi-dual maximizer). *Assume that $P \in \mathcal{P}_{\text{ac}}(\Omega)$ and Q have finite second moments. Let $u^\star$ be a c-concave maximizer of the semi-dual problem in Equation (12), and define $\phi_0(x) := \frac{1}{2}\big(\|x\|^2 - u^\star(x)\big), x \in \Omega$. Then ϕ_0 is convex, (ϕ_0, ϕ_0^*) attains Equation (13), and the unique optimal transport map satisfies*

$$\mathbf{T}_0(x) = \nabla\phi_0(x), \quad P\text{-a.s.}$$

Conversely, if ϕ_0 is any convex function such that $\mathbf{T}_0 = \nabla\phi_0$ is W_2-optimal, then

$$u^\star(x) = \|x\|^2 - 2\phi_0(x)$$

is a maximizer of the semi-dual problem in Equation (12).

The dual problem selects the highest pair of "roofs" (φ, ψ) that lie below the cost surface $c(x, y) = \|x - y\|^2$ everywhere. At optimality, mass is transported only along the contact set where the roofs touch the cost. For the quadratic cost, these roofs are generated by a single convex potential ϕ via conjugacy, and the optimal transport is given by the gradient field $x \mapsto \nabla\phi(x)$.

Examples of Lipschitz OT maps. Besides the example in Section 5.1, affine maps provide a simple class of Lipschitz OT maps. Let $A \in \mathbb{R}^{d\times d}$ be symmetric positive definite and $c \in \mathbb{R}^d$. Then $\mathbf{T}_0(x) = Ax + c$ is the gradient of $\phi_0(x) = x^\top Ax/2 + c^\top x$ and is therefore the OT map from P to $\mathbf{T}_0\#P$. It satisfies Assumption 4.2 with $\lambda = \lambda_{\max}(A) \vee \lambda_{\min}(A)^{-1}$.

The OT map between two Gaussian distributions is also affine with a symmetric positive definite linear coefficient matrix, and is therefore Lipschitz.

A.3 Deep Neural Network Function

Definition A.4 (DNN function). *A deep neural network (DNN) with $(L, \mathbf{d})$ architecture consists of*

- *$L \in \mathbb{N}$: the number of hidden layers;*
- *$\mathbf{d} = [d_0, \ldots, d_{L+1}]^\top \in \mathbb{N}^{L+2}$: the layer widths, where $d_0 = d$ and d_{L+1} are the input and output dimensions, respectively;*

is mathematically defined as

$$f^{\text{DNN}}_{\theta,\sigma}(x) := A_{L+1} \circ \sigma \circ A_L \circ \cdots \circ \sigma \circ A_1(x), \tag{14}$$

where

- *$A_l(x) := \mathbf{W}_l x + b_l$ with $\mathbf{W}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and $b_l \in \mathbb{R}^{d_l}$ for $l \in [L+1]$;*
- *σ is an element-wise activation function (e.g., ReLU: $\sigma(t) = \max\{t, 0\}$ or ReQU: $\sigma(t) = \max\{t, 0\}^2$);*
- *$\theta := [\text{vec}(\mathbf{W}_1)^\top, \ldots, \text{vec}(\mathbf{W}_{L+1})^\top, b_1^\top, \ldots, b_{L+1}^\top]^\top$ collects all parameters.*

Given $f^{\text{DNN}}_{\theta,\sigma}$ with $(L, \mathbf{d})$ in Equation (14), let $d_{\max} := \max\{d_1, \ldots, d_L\}$ be the width of $f^{\text{DNN}}_{\theta,\sigma}$.

Definition A.5 (Class of DNN functions). *Given an activation function σ, the class of DNN functions with at most L layers (depth), at most W neurons per layer (width), and output dimension $d_{\text{out}} \in \mathbb{N}$ is denoted as*

$$\begin{aligned}\Phi^{\text{DNN}}(L, W; \sigma, d_{\text{out}}) := \big\{ & f^{\text{DNN}}_{\theta,\sigma} \text{ with } (L', \mathbf{d}) \text{ as in Equation (14)} : \\ & L' \le L, \quad d_{L'+1} = d_{\text{out}}, \quad d_{\max} \le W, \|\theta\|_\infty \le 1 \big\}.\end{aligned} \tag{15}$$

We abbreviate $\Phi^{\text{DNN}}(L, W; \sigma, d_{\text{out}})$ by $\Phi^{\text{DNN}}(L, W; \sigma)$ when d_{out} is clear from context.

As we assume the true OT map is Lipschitz continuous, we also enforce the Lipschitz continuity to the DNN functions. That is, for a given Lipschitz constant $\lambda > 0$, we define

$$\Phi^{\text{DNN}}_\lambda := \big\{ f \in \Phi^{\text{DNN}}(L, W; \sigma) : \text{Lip}(f) \le \lambda \big\},$$

Here $\text{Lip}(f)$ is the difference-quotient Lipschitz constant on Ω defined in Section 3.2.

A.4 Proofs

In this section, we present the full proof of our main theoretical result (Theorem 4.1), with technical lemmas used to prove Theorem 4.1. Recall the assumptions used in the main body.

- (S1) Ω is a compact, convex set with nonempty interior such that $\Omega \subseteq [0,1]^d$.
- (S2) Ω is a standard set in the sense that there exist $\epsilon_0, \delta_0 > 0$ s.t. for all $x \in \Omega$ and $\epsilon \in (0, \epsilon_0)$, we have $\mathrm{Leb}(\mathbb{B}(x,\epsilon) \cap \Omega) \geq \delta_0 \,\mathrm{Leb}(\mathbb{B}(x,\epsilon))$, where Leb is the Lebesgue measure on $\mathbb{R}^d$.
- Assumption 4.1: P and Q are absolutely continuous on Ω with densities p and q satisfying $\gamma^{-1} \leq p(x), q(x) \leq \gamma$ for all $x \in \Omega$ and some $\gamma > 0$.
- Assumption 4.2: the potential ϕ_0 is convex, twice continuously differentiable on Ω, and satisfies $\lambda^{-1} I_d \preceq \nabla^2 \phi_0(x) \preceq \lambda I_d$ for all $x \in \Omega$, so that $\mathbf{T}_0 = \nabla \phi_0$ is λ-Lipschitz.

Technical Lemmas.

Lemma A.4 (Proposition 14 of [12]). *Let $\hat{\Gamma} = [\hat{\gamma}_{ij}]$ be an optimal transport plan between P_n and Q_m for the quadratic cost. Under Assumption 4.1 and Assumption 4.2, with $\Omega = [0,1]^d$, we have*

$$\mathbb{E}\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \hat{\gamma}_{ij} \|\mathbf{T}_0(X_i) - Y_j\|^2\right) \lesssim \bar{\kappa}_{\widetilde{n}} := \begin{cases} \widetilde{n}^{-1}, & d = 1, \\ \widetilde{n}^{-1}(\log \widetilde{n})^2, & d = 2, \\ \widetilde{n}^{-2/d}, & d \geq 3. \end{cases} \tag{16}$$

where $\widetilde{n} = n \wedge m$.

Proof of Lemma A.4. Under Assumption 4.2, we have

$$\frac{1}{2\lambda}\|x-y\|^2 \leq \phi_0(y) - \phi_0(x) - \langle \mathbf{T}_0(x), y - x\rangle \leq \frac{\lambda}{2}\|x-y\|^2, \quad \forall x, y \in \Omega. \tag{17}$$

Using these inequalities and the fact that

$$\mathbb{E}\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \hat{\gamma}_{ij} \|\mathbf{T}_0(X_i) - Y_j\|^2\right) \asymp \mathbb{E}\left(W_2^2(P_n, Q_m) - W_2^2(P, Q)\right) \tag{18}$$

concludes the proof. See [12] for the detailed proof. □

Lemma A.5 (Lemma 42 of [12]). *Suppose that Ω satisfies conditions* (S1)-(S2) *and assume Assumption 4.1. Let $V_i = \{x \in \Omega : \|x - X_i\| \leq \|x - X_j\|, \forall j \neq i\}$ for $i \in [n]$. Then, there exist constants C_1 and $C_2 > 0$, depending only on $d, \gamma, \epsilon_0, \delta_0$, such that*

1. *For all $\delta \in (0,1)$,*

$$\mathbb{P}\left(\max_{1\leq i\leq n} P(V_i) \;\geq\; \frac{C_1}{n}\Big[d\log n + \log(1/\delta)\Big]\right) \;\leq\; \delta. \tag{19}$$

2.

$$\mathbb{E}\left[\max_{1\leq i\leq n} \mathrm{diam}(V_i)^2\right] \leq C_2 \left(\frac{\log n}{n}\right)^{2/d}. \tag{20}$$

Proof of Lemma A.5. This is a direct corollary of the following lemma, as shown in [12]: There exists a universal constant $C > 0$ such that, for every Euclidean ball $\mathbb{B}$, $P(\mathbb{B}) \geq \frac{C}{n}(d\log n + \log(1/\delta))$ implies $P_n(\mathbb{B}) > 0$ with probability $1-\delta$ at least for any $\delta \in (0,1)$. □

Lemma A.6 (Lemma 44 of [12]). *Under Assumption 4.1, Assumption 4.2, (S1), and (S2), with $\Omega = [0,1]^d$, for any λ-Lipschitz map $f : \Omega \to \Omega$, we have*

$$\mathbb{E}\left(\sum_{i=1}^{n}\int_{V_i} \|f(X_i) - f(x)\|^2 dP(x)\right) \lesssim \left(\frac{\log n}{n}\right)^{2/d}, \tag{21}$$

and

$$\mathbb{E}\left(n\sum_{i=1}^{n}\sum_{j=1}^{m} \hat{\gamma}_{ij}\|\mathbf{T}_0(X_i) - Y_j\|^2 \left(\max_{1\leq i\leq n} P(V_i)\right)\right) \lesssim (\log n)\,\bar{\kappa}_{\widetilde{n}}, \tag{22}$$

where $\bar{\kappa}_{\widetilde{n}}$ is defined in Lemma A.4.

The next four lemmas construct a comparator from a smooth potential, which is used as an alternative comparator in the proof of Theorem 4.1.

Lemma A.7. *Fix $\beta > 2$ and $d_{\text{out}}, d \in \mathbb{N}$. For any $H > 0$, an integer $K \geq 2$, and any $f : [0,1]^d \to \mathbb{R}^{d_{\text{out}}}$ with $f \in \mathcal{C}^\beta([0,1]^d, H)$, there exists a DNN function $\mathbf{T}_f : [0,1]^d \to \mathbb{R}^{d_{\text{out}}}$ in $\Phi^{\text{DNN}}(L, W; \sigma)$ where $L \gtrsim \log d + \lfloor\beta\rfloor + \log\log H, W \gtrsim (d_{\text{out}} \vee d)(K + \lfloor\beta\rfloor)^d$ and $\sigma(t) = (t)_+^2$ (i.e., the ReQU activation), such that for all $\ell \in \{0, 1, \ldots, \lfloor\beta\rfloor\}$,*

$$\sup_{x\in[0,1]^d} \|\nabla^\ell f(x) - \nabla^\ell \mathbf{T}_f(x)\| = \|\nabla^\ell f - \nabla^\ell \mathbf{T}_f\|_{\infty,[0,1]^d} \leq C(\beta, d, H)\frac{1}{K^{\beta-\ell}} \tag{23}$$

where $C(\beta, d, H) > 0$ is a constant depending on β, d, and H. In particular, for $d_{\text{out}} = 1$,

$$\|\nabla f - \nabla \mathbf{T}_f\|_{\infty,[0,1]^d} \lesssim K^{-(\beta-1)}, \quad \|\nabla^2 f - \nabla^2 \mathbf{T}_f\|_{\infty,[0,1]^d} \lesssim K^{-(\beta-2)}.$$

Proof of Lemma A.7. This is a direct consequence of Theorem 1 in [55]. □

Lemma A.8. *Let $f \in \Phi^{\text{DNN}}(L, W; \text{ReQU}, 1)$ be any given ReQU DNN on $[0,1]^d$ such that there exist $M_f, M_{\nabla f} > 0$ satisfying*

$$\|f\|_{\infty,[0,1]^d} \leq M_f, \text{ and } \max_{1\leq j\leq d} \|\partial_{x_j} f\|_{\infty,[0,1]^d} \leq M_{\nabla f}.$$

Then, the gradient $\nabla f = (\partial_{x_1} f, \ldots, \partial_{x_d} f) : [0,1]^d \to \mathbb{R}^d$ is exactly implemented by a DNN with mixed activations (ReLU and ReQU), that is,

$$\nabla f \in \Phi^{\text{DNN}}\left(3L + 3, 6dW; \text{Mix of ReLU and ReQU}, d\right). \tag{24}$$

Proof of Lemma A.8. This is a consequence of Theorem 1 in [56]. We here provide a proof sketch, and more details can be found in [56].

The main idea is the following exact representations of polynomials, that are realizable by affine maps with ReLU or/and ReQU activations. Let $\sigma_1(t) = t_+$ and $\sigma_2(t) = (t_+)^2$ be the ReLU and ReQU activation functions, respectively. Then, we have $z = \sigma_1(z) - \sigma_1(-z)$ and $z^2 = \sigma_2(z) + \sigma_2(-z)$. Let the activation of layer ℓ be $z^\ell := A_\ell(h^{\ell-1}) = W_\ell h^{\ell-1} + b_\ell$ and $h^\ell := \sigma_2(z^\ell)$, with $h^0 := x$. By the chain rule, for any $u \in \mathbb{R}$, we have $\nabla_x(uf(x)) = \left(W_1^\top D^1(x) W_2^\top D^2(x) \cdots W_L^\top D^L(x) W_{L+1}^\top\right)u$, where $D^\ell(x) := \text{diag}(\sigma_2'(z^\ell(x)))$. Since $\sigma_2'(t) = 2\sigma_1(t)$, we have $D^\ell(x) = 2\text{diag}(\sigma_1(z^\ell(x)))$, i.e., a diagonal ReLU-gating matrix. Thus the differentiation circuit consists only of linear maps, ReLU, and element-wise products. Hence, a single backpropagation step is realized exactly by a combination of a linear map, ReLU/ReQU, and a linear map block. The construction of the remaining layers for the DNN with mixed activations follows [56], which yields the parameters $(3L + 3, 6dW)$. □

Fix $\beta > 2$ and $H > 0$. Let $\mathcal{J}_\lambda$ denote the class of twice continuously differentiable convex potentials ϕ on $[0,1]^d$ such that $\lambda^{-1} I_d \preceq \nabla^2\phi(x) \preceq \lambda I_d$ and $\nabla\phi([0,1]^d) \subseteq [0,1]^d$. Define the empirical OT-weighted least-squares comparator

$$\hat{\phi}^{\text{LS}} = \operatorname*{arg\,min}_{\phi\in\mathcal{J}_\lambda\cap\mathcal{C}^\beta([0,1]^d,H)} \sum_{i=1}^n \sum_{j=1}^m \hat{\gamma}_{ij}\|Y_j - \nabla\phi(X_i)\|^2,$$

and set

$$\widehat{\mathbf{T}}^{\text{LS}} := \nabla\hat{\phi}^{\text{LS}}.$$

Then $\hat{\phi}^{\text{LS}} \in \mathcal{C}^\beta([0,1]^d, H)$ holds by construction. Note that for continuously differentiable maps on the cube, we can rewrite the Lipschitz constant of f as $\text{Lip}(f) = \sup_{x\in[0,1]^d} \|Df(x)\|_{\text{op}}$.

Lemma A.9. *Let $K \geq 2$. Assume that there exists a ReQU DNN $h_{\text{LS}} \in \Phi^{\text{DNN}}(L, W; \text{ReQU}, 1)$ such that, for $\ell = 1, 2$,*

$$\|\nabla^\ell h_{\text{LS}} - \nabla^\ell\hat{\phi}^{\text{LS}}\|_{L^\infty([0,1]^d)} \leq C_0 K^{-(\beta-\ell)}, \tag{25}$$

where $C_0 > 0$ depends only on (β, H, d) and universal constants from Lemma A.7. Define $\nabla h_{\text{LS}} : [0,1]^d \to \mathbb{R}^d$. Then there exist constants $C_1, C_2, c > 0$, depending only on (β, H, d) and universal constants, such that

$$\begin{aligned} &\|\nabla h_{\text{LS}} - \widehat{\mathbf{T}}^{\text{LS}}\|_{L^\infty([0,1]^d)} \leq C_1 K^{-(\beta-1)}, \\ &\|\nabla^2 h_{\text{LS}} - \nabla\widehat{\mathbf{T}}^{\text{LS}}\|_{L^\infty([0,1]^d)} \leq C_2 K^{-(\beta-2)}, \end{aligned} \tag{26}$$

and hence

$$\mathrm{Lip}(\nabla h_{\mathrm{LS}}) \le \lambda + cK^{-(\beta-2)}. \tag{27}$$

Moreover, there exists a DNN with mixed activations (ReLU and ReQU), $\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}} : [0,1]^d \to \mathbb{R}^d$*, such that*

$$\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}} = \nabla h_{\mathrm{LS}} \text{ on } [0,1]^d, \tag{28}$$

with architecture $L' \le 3L+3, W' \le 6dW$ *and* $\|\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}}\|_{L^\infty([0,1]^d)} < \infty$*. In particular, we have*

$$\|\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}} - \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty \lesssim K^{-(\beta-1)}, \quad \mathrm{Lip}(\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}}) \le \lambda + cK^{-(\beta-2)}. \tag{29}$$

Proof of Lemma A.9. From Equation (25) with $\ell = 1$ and the definitions ∇h_{LS} and $\widehat{\mathbf{T}}^{\mathrm{LS}} = \nabla \hat{\phi}^{\mathrm{LS}}$,

$$\|\nabla h_{\mathrm{LS}} - \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty = \|\nabla h_{\mathrm{LS}} - \nabla \hat{\phi}^{\mathrm{LS}}\|_\infty \le C_0 K^{-(\beta-1)}. \tag{30}$$

Similarly, using Equation (25) with $\ell = 2$,

$$\|\nabla^2 h_{\mathrm{LS}} - \nabla \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty = \|\nabla^2 h_{\mathrm{LS}} - \nabla^2 \hat{\phi}^{\mathrm{LS}}\|_\infty \le C_0 K^{-(\beta-2)}. \tag{31}$$

Thus Equation (26) holds with $C_1 = C_2 = C_0$. For Lipschitz constants,

$$\begin{aligned} \big|\,\mathrm{Lip}(\nabla h_{\mathrm{LS}}) - \mathrm{Lip}(\widehat{\mathbf{T}}^{\mathrm{LS}})\big| &\le \sup_x \|\nabla^2 h_{\mathrm{LS}}(x) - \nabla \widehat{\mathbf{T}}^{\mathrm{LS}}(x)\|_{\mathrm{op}} \\ &\le d\|\nabla^2 h_{\mathrm{LS}} - \nabla \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty \le dC_2 K^{-(\beta-2)}. \end{aligned} \tag{32}$$

Since $\hat{\phi}^{\mathrm{LS}} \in \mathcal{J}_\lambda$, its gradient $\widehat{\mathbf{T}}^{\mathrm{LS}}$ is λ-Lipschitz. Combining this with the Hessian approximation bound above gives $\mathrm{Lip}(\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}}) \le \lambda + cK^{-(\beta-2)}$ with $c := dC_2$.

Finally, Lemma A.8 implies that there exists a DNN in which each hidden unit uses either the ReQU or a ReLU activation, $\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}} : [0,1]^d \to \mathbb{R}^d$, such that $\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}} = \nabla h_{\mathrm{LS}}$ on $[0,1]^d$, with architecture $L' \le 3L+3, W' \le 6dW$ and $\|\tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}}\|_{L^\infty([0,1]^d)} = \|\nabla h_{\mathrm{LS}}\|_{L^\infty([0,1]^d)} < \infty$. This concludes the proof. □

Lemma A.10. *Fix* $\beta > 2$*. Let* $\bar{\mathbf{T}}^{\mathrm{sm}} = \tilde{\mathbf{T}}_{\widehat{\phi}^{\mathrm{LS}}}$ *in Lemma A.9 with* $K = K_{\widetilde{n}}$*. Choose the DNN parameters as* $L \gtrsim \log d + \lfloor\beta\rfloor + \log\log H, W = W_{nm} \gtrsim (1 \vee d)(K_{\widetilde{n}} + \lfloor\beta\rfloor)^d$*. If* $K_{\widetilde{n}} \asymp \widetilde{n}^{\frac{1}{d(\beta-1)}}$*, then we have* $\|\bar{\mathbf{T}}^{\mathrm{sm}} - \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty^2 \lesssim \widetilde{n}^{-2/d}$ *and* $\mathrm{Lip}(\bar{\mathbf{T}}^{\mathrm{sm}}) \le \lambda + cK_{\widetilde{n}}^{-(\beta-2)}$*.*

Proof of Lemma A.10. By Lemma A.9 with $K = K_{\widetilde{n}}$, we have $\|\bar{\mathbf{T}}^{\mathrm{sm}} - \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty \lesssim K_{\widetilde{n}}^{-(\beta-1)}$. Choosing the parameter of the ReQU DNN h_{LS} in Lemma A.7 as:

$$\begin{aligned} &L \gtrsim \log d + \lfloor\beta\rfloor + \log\log H, \\ &W_{nm} \gtrsim (1 \vee d)\big(K_{\widetilde{n}} + \lfloor\beta\rfloor\big)^d. \end{aligned} \tag{33}$$

Let $(K_{\widetilde{n}})^d \asymp \widetilde{n}^{1/(\beta-1)}$. Then, we have

$$\|\bar{\mathbf{T}}^{\mathrm{sm}} - \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty^2 \lesssim K_{\widetilde{n}}^{-2(\beta-1)} = \Big(\widetilde{n}^{\frac{1}{d(\beta-1)}}\Big)^{-2(\beta-1)} = \widetilde{n}^{-2/d} \tag{34}$$

and

$$\mathrm{Lip}(\bar{\mathbf{T}}^{\mathrm{sm}}) \le \lambda + cK_{\widetilde{n}}^{-(\beta-2)}. \tag{35}$$

□

Lemma A.11 (Lipschitz-aware DNN comparator)**.** *Let* $K \ge 2$ *be an integer and let* $\mathbf{T} : \Omega \to \Omega$ *be* λ*-Lipschitz. There exists a DNN* $\bar{\mathbf{T}}_K$ *with mixed ReLU/ReQU activations, depth* L_d*, width at most* $C_d(K+1)^{d+1}$*, and all parameters in* $[-1,1]$*, such that*

$$\|\bar{\mathbf{T}}_K - \mathbf{T}\|_\infty \le d\lambda/K, \quad \mathrm{Lip}(\bar{\mathbf{T}}_K) \le d\lambda.$$

The constants depend only on d *and* λ*.*

Proof. Extend each coordinate of $\mathbf{T}$ to $[0,1]^d$ by

$$g_\ell(x) = \min\{1, \max\{0, \inf_{z\in\Omega}[T_\ell(z) + \lambda\|x - z\|]\}\}, \quad \ell \in [d].$$

Each g_ℓ is λ-Lipschitz and agrees with T_ℓ on Ω. For $j = 0, \ldots, K$, let $h_j(t) = (1 - K|t - j/K|)_+$ on $[0,1]$, and define

$$\bar{\mathbf{T}}_K(x) = \sum_{\mathbf{j}\in\{0,\ldots,K\}^d} \mathbf{g}(\mathbf{j}/K) \prod_{s=1}^{d} h_{j_s}(x_s).$$

The product weights are nonnegative, sum to one, and use grid points within distance $\sqrt{d}/K$ of x. Hence each coordinate has approximation error at most $\lambda\sqrt{d}/K$, giving the stated bound. Within each grid cell, each Jacobian entry is a convex combination of coordinate edge differences divided by $1/K$, and is bounded in absolute value by λ. Thus its operator norm is at most $d\lambda$, and continuity across the cells gives the Lipschitz bound on $[0,1]^d$.

The interpolant is exactly realizable by a bounded-parameter mixed ReLU/ReQU network. ReLU computes $|t - j/K|$; the factor K is obtained by summing K identical copies with unit weights. Products of nonnegative values are computed using

$$uv = \mathrm{ReQU}((u+v)/2) - \mathrm{ReQU}((u-v)/2) - \mathrm{ReQU}((v-u)/2), \quad u, v \in [0,1].$$

A binary tree of these multipliers has depth depending only on d, and parallelizing the grid terms gives width at most $C_d(K+1)^{d+1}$. The final coefficients $g_\ell(\mathbf{j}/K)$ lie in $[0,1]$, so all parameters can be taken in $[-1,1]$. Precomposing with the 1-Lipschitz clipping map $\pi(x)_s = \mathrm{ReLU}(x_s) - \mathrm{ReLU}(x_s - 1)$ gives the same global Lipschitz bound without changing the approximation on Ω; its constant additional depth and width are absorbed into L_d and C_d. □

Lemma A.12. *Let $\bar{\mathbf{T}} = \bar{\mathbf{T}}_{K_{\widetilde{n}}}$ be obtained by applying Lemma A.11 to $\mathbf{T} = \mathbf{T}_0$. Choose $K_{\widetilde{n}} = \max\{2, \lceil \widetilde{n}^{1/d} \rceil\}$, $L = L_d$, and $W_{nm} \geq C_d(K_{\widetilde{n}} + 1)^{d+1}$. Then, we have*

$$\|\bar{\mathbf{T}} - \mathbf{T}_0\|_\infty^2 \lesssim \widetilde{n}^{-2/d}, \quad \mathrm{Lip}(\bar{\mathbf{T}}) \leq d\lambda.$$

Proof. By Lemma A.11,

$$\|\bar{\mathbf{T}} - \mathbf{T}_0\|_\infty^2 \leq \frac{d^2\lambda^2}{K_{\widetilde{n}}^2} \lesssim \widetilde{n}^{-2/d}.$$

Moreover, $\mathrm{Lip}(\bar{\mathbf{T}}) \leq d\lambda < \lambda + c'$. □

Proof of Theorem 4.1.

Proof of Theorem 4.1. Write $\widehat{\mathbf{T}} = \widehat{\mathbf{T}}_{nm}^{\mathrm{DNN}}$ for brevity, and take expectation with respect to the data and to OT plan $\hat{\Gamma} = [\hat{\gamma}_{ij}]$. We also note that the OT plan naturally induces the weighted regression objective $\sum_{i,j} \hat{\gamma}_{ij}\|Y_j - \mathbf{T}(X_i)\|^2$, which is equivalent to the standard objective in Equation (4) up to a constant independent of $\mathbf{T}$. Thus, in this proof, we work with the weighted objective $\sum_{i,j} \hat{\gamma}_{ij}\|Y_j - \mathbf{T}(X_i)\|^2$. First, we decompose the error using the Voronoi cells. Let $V_i = \{x \in \Omega : \|x - X_i\| \leq \|x - X_k\| \ \forall k \neq i\}, i \in [n]$ be the Voronoi cells induced by $X_1, \ldots, X_n$. Then, we can decompose the error as

$$\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P)}^2 = \sum_{i=1}^{n} \int_{V_i} \|\widehat{\mathbf{T}}(x) - \mathbf{T}_0(x)\|^2 dP(x). \tag{36}$$

For every $x \in V_i$, by $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$ and the triangle inequality, we have $\|\widehat{\mathbf{T}}(x) - \mathbf{T}_0(x)\|^2 \leq 3\|\widehat{\mathbf{T}}(x) - \widehat{\mathbf{T}}(X_i)\|^2 + 3\|\widehat{\mathbf{T}}(X_i) - \mathbf{T}_0(X_i)\|^2 + 3\|\mathbf{T}_0(X_i) - \mathbf{T}_0(x)\|^2$. Integrating over V_i and summing over i yields

$$\begin{aligned}\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P)}^2 \leq\ & 3\sum_{i=1}^{n} \int_{V_i} \|\widehat{\mathbf{T}}(x) - \widehat{\mathbf{T}}(X_i)\|^2 dP(x) + 3\sum_{i=1}^{n} \int_{V_i} \|\mathbf{T}_0(x) - \mathbf{T}_0(X_i)\|^2 dP(x) \\ & + 3\sum_{i=1}^{n} P(V_i)\|\widehat{\mathbf{T}}(X_i) - \mathbf{T}_0(X_i)\|^2.\end{aligned} \tag{37}$$

Second, we derive upper bounds for the first and second terms of the right-hand-side in Equation (37). By construction $\widehat{\mathbf{T}} \in \Phi_{nm}^{\mathrm{DNN}}$ so that $\mathrm{Lip}(\widehat{\mathbf{T}}) \le \lambda + c'$, and by Assumption 4.2, $\mathrm{Lip}(\mathbf{T}_0) \le \lambda$. Applying Lemma A.6 with $f = \widehat{\mathbf{T}}$ and with $f = \mathbf{T}_0$, and then taking expectations in Equation (37), we obtain

$$\mathbb{E}\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P)}^2 \lesssim \left(\frac{\log n}{n}\right)^{2/d} + \mathbb{E}\left(n\Big(\max_{1\le i\le n} P(V_i)\Big) \cdot \|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P_n)}^2\right), \tag{38}$$

where $\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P_n)}^2 := \frac{1}{n}\sum_{i=1}^n \|\widehat{\mathbf{T}}(X_i) - \mathbf{T}_0(X_i)\|^2$. Third, we derive the bound for the third term. For any vectors a, b, c, it holds that $\|a-b\|^2 \le 2\|a-c\|^2 + 2\|c-b\|^2$. Hence, for all i, j,

$$\|\widehat{\mathbf{T}}(X_i) - \mathbf{T}_0(X_i)\|^2 \le 2\|\widehat{\mathbf{T}}(X_i) - Y_j\|^2 + 2\|Y_j - \mathbf{T}_0(X_i)\|^2. \tag{39}$$

Since each row sum of $\hat{\Gamma}$ is $1/n$, summing against $\hat{\gamma}_{ij}$ gives the following bound.

$$\begin{aligned}\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P_n)}^2 &= \sum_{i,j} \hat{\gamma}_{ij}\|\widehat{\mathbf{T}}(X_i) - \mathbf{T}_0(X_i)\|^2 \\ &\le 2\sum_{i,j} \hat{\gamma}_{ij}\|\widehat{\mathbf{T}}(X_i) - Y_j\|^2 + 2\sum_{i,j} \hat{\gamma}_{ij}\|Y_j - \mathbf{T}_0(X_i)\|^2.\end{aligned} \tag{40}$$

Let $\bar{\mathbf{T}}$ be the DNN approximator of $\mathbf{T}_0$ from Lemma A.12. It satisfies $\mathrm{Lip}(\bar{\mathbf{T}}) \le d\lambda < \lambda + c'$, so $\bar{\mathbf{T}} \in \Phi_{nm}^{\mathrm{DNN}}$. Hence, by optimality of $\widehat{\mathbf{T}}$ over Φ_{nm}^{DNN}, $\sum_{i,j}\hat{\gamma}_{ij}\|\widehat{\mathbf{T}}(X_i) - Y_j\|^2 \le \sum_{i,j}\hat{\gamma}_{ij}\|\bar{\mathbf{T}}(X_i) - Y_j\|^2$. Expanding $\bar{\mathbf{T}}$ around $\mathbf{T}_0$ gives, for each (i,j), $\|\bar{\mathbf{T}}(X_i) - Y_j\|^2 \le 2\|\bar{\mathbf{T}}(X_i) - \mathbf{T}_0(X_i)\|^2 + 2\|\mathbf{T}_0(X_i) - Y_j\|^2$, and summing over $\hat{\gamma}_{ij}$ and using the fact that the total mass is 1, we get

$$\sum_{i,j}\hat{\gamma}_{ij}\|\bar{\mathbf{T}}(X_i) - Y_j\|^2 \le 2d\|\bar{\mathbf{T}} - \mathbf{T}_0\|_\infty^2 + 2\sum_{i,j}\hat{\gamma}_{ij}\|\mathbf{T}_0(X_i) - Y_j\|^2. \tag{41}$$

Combining Equation (40)-Equation (41), we obtain

$$\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P_n)}^2 \lesssim \|\bar{\mathbf{T}} - \mathbf{T}_0\|_\infty^2 + \sum_{i,j}\hat{\gamma}_{ij}\|\mathbf{T}_0(X_i) - Y_j\|^2. \tag{42}$$

When, in addition, $\phi_0 \in \mathcal{C}^\beta([0,1]^d, H)$, the comparator can also be built from the least-squares map $\widehat{\mathbf{T}}^{\mathrm{LS}} = \nabla\hat{\phi}^{\mathrm{LS}}$ instead of $\mathbf{T}_0$. In this case, $\phi_0 \in \mathcal{J}_\lambda \cap \mathcal{C}^\beta([0,1]^d, H)$, so the optimality of $\hat{\phi}^{\mathrm{LS}}$ gives $\sum_{i,j}\hat{\gamma}_{ij}\|\widehat{\mathbf{T}}^{\mathrm{LS}}(X_i) - Y_j\|^2 \le \sum_{i,j}\hat{\gamma}_{ij}\|\mathbf{T}_0(X_i) - Y_j\|^2$. Moreover, $\bar{\mathbf{T}}^{\mathrm{sm}}$ in Lemma A.10 satisfies $\|\bar{\mathbf{T}}^{\mathrm{sm}} - \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty^2 \lesssim \widetilde{n}^{-2/d}$ and $\mathrm{Lip}(\bar{\mathbf{T}}^{\mathrm{sm}}) \le \lambda + c'$ for all sufficiently large $\widetilde{n}$, so it belongs to Φ_{nm}^{DNN} after enlarging its constant depth and its width, where each parameter of magnitude larger than one is split into parallel unit-weight copies as in the proof of Lemma A.11, if necessary. Replacing $\bar{\mathbf{T}}$ and $\mathbf{T}_0$ by $\bar{\mathbf{T}}^{\mathrm{sm}}$ and $\widehat{\mathbf{T}}^{\mathrm{LS}}$ in Equation (41) then gives Equation (42) with $\|\bar{\mathbf{T}}^{\mathrm{sm}} - \widehat{\mathbf{T}}^{\mathrm{LS}}\|_\infty^2$ in place of $\|\bar{\mathbf{T}} - \mathbf{T}_0\|_\infty^2$, which is of the same order.

Then, we aggregate the bounds as follows. Let

$$M_n := \max_{1\le i\le n} P(V_i), \quad Z_{nm} := \sum_{i,j}\hat{\gamma}_{ij}\|\mathbf{T}_0(X_i) - Y_j\|^2.$$

By Lemma A.5(i), taking $\delta = n^{-2}$ and using $M_n \le 1$, we have $\mathbb{E}(nM_n) \lesssim \log n + n\delta \lesssim \log n$, for all $n \ge 2$. Plugging Equation (42) into Equation (38) and taking expectations, we obtain

$$\mathbb{E}\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P)}^2 \lesssim \left(\frac{\log n}{n}\right)^{2/d} + \|\bar{\mathbf{T}} - \mathbf{T}_0\|_\infty^2\mathbb{E}[nM_n] + \mathbb{E}[nM_nZ_{nm}]. \tag{43}$$

Finally, we construct the DNN architecture to approximate $\mathbf{T}_0$. By Lemma A.12, with $K_{\widetilde{n}} = \max\{2, \lceil\widetilde{n}^{1/d}\rceil\}$, we have

$$\|\bar{\mathbf{T}} - \mathbf{T}_0\|_\infty^2 \lesssim \widetilde{n}^{-2/d}, \quad \mathrm{Lip}(\bar{\mathbf{T}}) \le d\lambda.$$

Hence $\bar{\mathbf{T}}$ belongs to the admissible Lipschitz class used to define $\widehat{\mathbf{T}}$. Combining Equations (22) and (43) with the bound $\mathbb{E}[nM_n] \lesssim \log n$ above and Lemma A.12, we obtain

$$\mathbb{E}\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P)}^2 \lesssim \left(\frac{\log n}{n}\right)^{2/d} + \widetilde{n}^{-2/d}\log n + \begin{cases}\widetilde{n}^{-1}\log n, & d = 1,\\ \widetilde{n}^{-1}(\log\widetilde{n})^2\log n, & d = 2,\\ \widetilde{n}^{-2/d}\log n, & d \ge 3,\end{cases} \tag{44}$$

achieving the minimax convergence rate up to logarithmic factors. This bound carries an additional $\log n$ factor for $d = 2$ compared with Theorem 4.1. To obtain the stated rates for every d, let π_n be an optimal coupling of P and P_n. Since $\widehat{\mathbf{T}} - \mathbf{T}_0$ is $(2\lambda + c')$-Lipschitz, integrating over π_n gives, for every realization of the data,

$$\|\widehat{\mathbf{T}} - \mathbf{T}_0\|^2_{L^2(P)} \leq 2\|\widehat{\mathbf{T}} - \mathbf{T}_0\|^2_{L^2(P_n)} + 2(2\lambda + c')^2 W_2^2(P, P_n). \tag{45}$$

Corollary 8 of [12], applied to the identity map transporting P to itself, gives $\mathbb{E}W_2^2(P, P_n) \lesssim \bar{\kappa}_n$. Combining this with Equation (42) and Lemmas A.4 and A.12, we obtain

$$\mathbb{E}\|\widehat{\mathbf{T}} - \mathbf{T}_0\|^2_{L^2(P)} \lesssim \widetilde{n}^{-2/d} + \bar{\kappa}_n + \bar{\kappa}_{\widetilde{n}} \lesssim \bar{\kappa}_{\widetilde{n}}, \tag{46}$$

which implies the stated bounds in Theorem 4.1. This completes the proof. □

B Experiments

B.1 Implementation Details

In this section, we describe the implementation details such as hyperparameter selection and model architectures used in our experiments. All experiments are run on several NVIDIA RTX 3090 GPUs.

Overall protocol. (i) Hyperparameter search spaces are documented below for each method and are chosen by combining each paper's reported settings and a two to three times expansion. (ii) Each method is trained with up to $10{,}000$ (outer) gradient steps, when applicable, to provide sufficient time for convergence. (iii) Data are scaled to have zero mean and unit variance, when applicable. All the methods are trained on the normalized inputs, and the transported inputs are denormalized back to the original target scale when we calculate the performance metrics. Note that this normalization particularly stabilizes the min-max learning methods such as ICNN.

BROT. We use the `emd` module of the POT library [29] to solve the unregularized empirical Kantorovich problem. Among the training iterations, we pick the hyperparameter with the lowest validation `Wass` (or task-appropriate validation `MMD` for the single-cell experiments). For large-scale experiments, we use a chunked version of Step 1. We randomly partition the source and target data into paired chunks, solve an unregularized OT problem within each chunk pair, and collect the resulting barycentric targets. For the domain adaptation experiments, we use a source chunk size of $2{,}048$, with target chunk sizes proportional to the target/source sample size ratio. The downstream classifier is trained for up to 80 epochs on the transported source data with an 80:20 training/validation split, and we select the model at the iteration with the highest validation accuracy. We then evaluate the model on the test data.

The search spaces are: the activation in $\{\mathrm{ReLU}, \mathrm{ReQU}, \mathrm{GELU}\}$, the hidden widths among a few MLP architectures from 4 to 6 layers wide between 64 and $2{,}048$ units, the optimizer (SGD with momentum or Adam), the learning rate in $\{10^{-3}, 5\cdot10^{-3}, 10^{-2}, 5\cdot10^{-2}\}$, the batch size in $\{256, 512, 1{,}024, 2{,}048\}$, the number of epochs in $\{100, 500, 1{,}000, 2{,}000\}$, and the gradient-penalty weight $\{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.

1NN [12]. 1NN solves the unregularized empirical Kantorovich problem and uses the resulting plan to assign each training data point X_i to its barycentric target. At test time, it extends the plan to unseen source data by the nearest neighbor interpolation using the training data.

LSOT [18]. LSOT solves a regularized dual Kantorovich problem (with either an L_2 or entropic regularizer) by stochastic optimization, and then trains a DNN by least-squares regression to fit the regularized barycentric targets. We reimplement the dual SGD procedure, Algorithm 1 of [18], and reuse the same MLP architecture as BROT on every dataset. For each experiment, we pick the hyperparameter with the lowest validation `Wass` (or task-appropriate validation `MMD` for the single-cell experiments) over a grid search. Because LSOT uses a regularized OT plan, a nonzero regularization strength may induce a bias relative to the unregularized OT map. We therefore search the regularization weight down to the smallest value for which the optimization remains numerically stable. The grid below starts at 0.001, and we kept the lowest value for which the optimization of LSOT remains numerically stable.

The search spaces are as follows: the regularizer type (either L_2 or entropic) and its strength in $\{0.001, 0.0025, 0.005, 0.01, 0.02, 0.05, 0.1\}$, the potential learning rate in $\{5\cdot10^{-4}, 10^{-3}, 5\cdot10^{-3}, 10^{-2}\}$, the potential optimizer (SGD or Adam), the potential batch size in $\{50, 100, 200\}$, the potential iteration number in $\{2{,}000, 5{,}000, 10{,}000\}$, the map learning rate in $\{10^{-4}, 5\cdot10^{-4}, 10^{-3}, 5\cdot10^{-3}\}$, the map batch size in $\{50, 100, 200\}$, and the map iteration number in $\{2{,}000, 5{,}000, 10{,}000\}$.

ICNN [19]. ICNN parameterizes the potential ϕ as an input-convex neural network (ICNN) and trains it together with its conjugate by the min-max dual objective. At inference time, we use the gradient of the trained potential as the estimate of the OT map. We follow the parameterization of the CellOT codebase [44] due to stability compared to the official codebase, with a linear output head and the uniform initialization on $[0, 0.1]$ for the convex weights. The min-max objective is optimized by alternating: 1 map update per 10 potential updates with a soft convexity penalty of weight λ_{cvx}

on the parameters. This penalty is the squared Frobenius norm of the negative parts of the convex weights summed over all layers and scaled by λ_{cvx}, which discourages negative weights and thus encourages the network to remain input-convex without requiring hard projection at every step.

For each experiment, we select the hyperparameter setting that minimizes the validation `Wass` (or the task-appropriate validation loss for the single-cell and domain adaptation experiments) over a random search. The search space follows the CellOT codebase [44] with the architecture (a 4-layer or 3-layer ICNN with widths in $\{64, 256, 512\}$), the leaky-ReLU activation, the convexity penalty weight $\lambda_{\text{cvx}} \in \{0.5, 1.0, 2.0, 5.0\}$, the map learning rate in $\{5 \cdot 10^{-5}, 10^{-4}, 10^{-3}\}$, the potential learning rate in $\{5 \cdot 10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}\}$, the potential updates per map update in $\{5, 10, 16\}$, the iteration budget in $\{2{,}000, 3{,}000, 5{,}000, 10{,}000\}$, the batch size in $\{256, 512\}$, the weight decay in $\{0, 10^{-5}, 10^{-4}\}$, and the convex-weight initialization in $\{\text{uniform-small}, \text{trunc-inv-sqrt}\}$. On AFHQv2, the training diverges across every configuration in this search space, marked $*$ in Table 1.

DIOTM [22]. DIOTM builds on the semi-dual neural OT (SNOT) problem $\inf_{\mathbf{T}} \sup_{\psi}(\mathbb{E}_P[\|x - \mathbf{T}(x)\|^2 - \psi(\mathbf{T}(x))] + \mathbb{E}_Q[\psi(y)])$ of [8, 19, 20]. It trains forward and backward transport maps jointly with a time-dependent potential along the displacement interpolation between the source and target distributions, and adds a Hamilton-Jacobi-Bellman (HJB) regularizer on the potential. Its hyperparameters include the divergence type in the potential loss, the cost coefficient τ, and the HJB regularization weight λ_{reg}. For each experiment we select the hyperparameter setting that minimizes the validation `Wass` (or task-appropriate validation `MMD` for the single-cell experiments).

The search spaces are: the divergence type in $\{\text{linear}, \text{KL}, \chi^2, \text{softplus}\}$, the cost coefficient $\tau \in \{0.01, 0.1, 1.0\}$, the HJB regularization weight $\lambda_{\text{reg}} \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, the dual learning rate in $\{10^{-5}, 10^{-4}, 5 \cdot 10^{-4}\}$, the map learning rate in $\{10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}\}$, the number of dual inner steps in $\{3, 5, 8, 10\}$ with map inner steps in $\{1, 2, 4\}$, the iteration number in $\{2{,}000, 5{,}000, 10{,}000, 20{,}000\}$, the hidden width in $\{64, 128, 256, 512\}$, and the batch size in $\{128, 200, 256, 512, 1{,}024\}$.

OTP [21]. OTP adds a Gaussian-smoothing schedule on top of the SNOT problem [8, 19, 20]. At each training step, each source data point is perturbed by Gaussian noise whose scale is annealed from $\xi_{\max}$ to $\xi_{\min}$ along a linear or cosine schedule, either additively or in a variance-preserving form. In the additive form the perturbed source is $\widetilde{X}_{i,t} = X_i + \xi_t \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I)$. In the variance-preserving form it is $\widetilde{X}_{i,t} = \sqrt{1 - \xi_t^2} X_i + \xi_t \boldsymbol{\epsilon}$, which keeps the marginal variance of $\widetilde{X}_{i,t}$ equal to that of X_i. The noise scale ξ_t decays from $\xi_{\max}$ at the first iteration to $\xi_{\min}$ at the last iteration along the chosen schedule. For each experiment we select the hyperparameter setting that minimizes the validation `Wass` (or the task-appropriate validation `MMD` for the single-cell experiments) over a grid search. We implement OTP by reusing the DIOTM codebase and adding the smoothing of the source data described above. The search space therefore inherits the DIOTM hyperparameters provided above. We additionally do the grid search: the noise mode in $\{\text{additive}, \text{variance-preserving}\}$, the noise schedule in $\{\text{linear}, \text{cosine}\}$, the maximum noise scale $\xi_{\max} \in \{0.1, 0.3, 0.5, 1.0\}$, and the minimum noise scale $\xi_{\min} \in \{0, 0.01\}$.

MM-v2 and W2 [40]. MM-v2 uses two input-convex neural networks as potentials with alternating maximin updates. W2 uses a non-maximin objective with a cycle-consistency penalty. We follow the implementations of [40] and select hyperparameters similarly to BROT. For the map quality experiment, both MM-v2 and W2 solvers use widths $(64, 64, 32)$, batch size $1{,}024$ and learning rate 10^{-3}. On AFHQ, we use widths $(256, 256)$, batch size 128 and learning rate 10^{-3}.

Entropic map estimator (ENT) [41]. ENT uses the dual potential of entropically regularized OT to extend its barycentric map to unseen source data. Its regularization strength is $\varepsilon = \rho \bar{c} > 0$, where ρ is a dimensionless scale and $\bar{c}$ is the mean pairwise quadratic cost on the training data. We select one from $\rho \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3\}$ using the same validation metric as BROT (and the neural estimators). In the experiment of Section 5.1, ε is instead selected as described in Appendix B.6.

B.2 Statistical Convergence to the Ground-Truth Optimal Transport Map

We provide Figure 3, a copy of Figure 2(a) with the standard error bands to show the statistical uncertainty of this experiment.

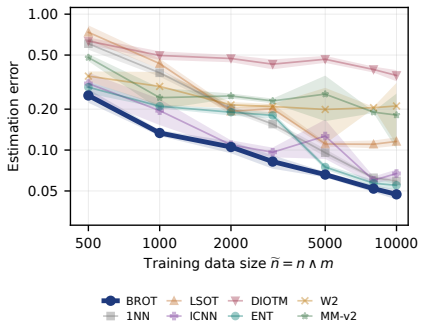


Figure 3: **Statistical convergence to the ground-truth OT map.** A copy of Figure 2(a) with the standard error bands.

We further give details of the synthetic dataset used in Section 5.1. Fix $R > 0$ and coefficients $a_s > 0$, $b_s \geq 0$ for $s = 1, \ldots, d$. Define the potential and candidate OT map for $x = [x_1, \ldots, x_d]^\top \in \mathbb{R}^d$ as:

$$\phi(x) := \frac{1}{2}\sum_{s=1}^{d} a_s x_s^2 + \frac{1}{4}\sum_{s=1}^{d} b_s x_s^4, \quad \mathbf{T}_0(x) := \nabla\phi(x) = \left(a_s x_s + b_s x_s^3\right)_{s=1}^{d}, \tag{47}$$

and let $P := \mathcal{N}(0, I_d)|_R$ be the standard Gaussian truncated to the ball of radius R. The target distribution is $Q := \mathbf{T}_{0\#}P$. In our experiments we use $d = 2, (a_1, a_2) = (1.5, 0.8), (b_1, b_2) = (0.3, 0.1)$, and $R = 3.5$. Figure 4 visualizes empirical distributions of P and Q used in this analysis, and Proposition B.1 shows that the construction in Equation (47) yields a Lipschitz OT map.

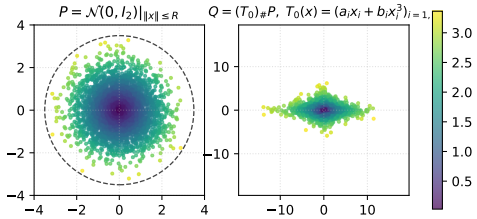


Figure 4: **Visualizations of the source P (left) and target $Q = \mathbf{T}_{0\#}P$ (right) distributions used in Section 5.1.** The color of each point indicates the radius of data $\|x\|$.

Proposition B.1 (Lipschitz OT map from P to Q)**.** *Let ϕ, $\mathbf{T}_0$, P, Q be as in Equation (47) with $a_s > 0$, $b_s \geq 0$, and $R < \infty$. Then, $\mathbf{T}_0$ is the unique OT map from P to Q, and is Lipschitz.*

Proof. Since ϕ is separable across coordinates, its Hessian is diagonal. That is, $\nabla^2\phi(x) = \mathrm{diag}\left(a_s + 3b_s x_s^2\right)_{s=1}^{d}$, which is strictly positive because $a_s > 0$ and $b_s \geq 0$. Hence ϕ is smooth and strictly convex, and lower-semicontinuous on $\mathbb{R}^d$. At the same time, P admits a density supported on the compact ball $\{\|x\| \leq R\}$ and hence has finite second moment. $\mathbf{T}_0$ is also smooth on this ball, since its outputs are in $\{|y_s| \leq a_s R + b_s R^3\}$, so $Q = \mathbf{T}_{0\#}P$ also has finite second moment.

By the Knott-Smith theorem (Theorem 2.12 of [25]), if $\phi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is convex and lower-semicontinuous with $\nabla\phi$, and if both P and $(\nabla\phi)_{\#}P$ have finite second moment, then $\nabla\phi$ is the unique OT map from P to $(\nabla\phi)_{\#}P$ for the quadratic cost. The above two steps satisfy these conditions, so $\mathbf{T}_0 = \nabla\phi$ is the optimal transport map from P to Q. On the support $\{\|x\| \leq R\}$, the diagonal Hessian satisfies $\nabla^2\phi(x) \preceq \mathrm{diag}\left(a_s + 3b_s R^2\right)_{s=1}^{d}$, so $\mathbf{T}_0$ is L_0-Lipschitz on this set with Lipschitz constant $L_0 = \max_{s\in[d]}\left(a_s + 3b_s R^2\right)$. For $d = 2$, our setting of $(a_1, a_2) = (1.5, 0.8)$, $(b_1, b_2) = (0.3, 0.1)$, $R = 3.5$ gives $L_0 \approx 12.52$. Hence, $\mathbf{T}_0$ is a Lipschitz OT map.

□

Sensitivity to DNN architecture and Jacobian penalty. The theoretically designed DNN architecture in Theorem 4.1 depends on the sample sizes n and m, whereas our experiments in Section 5 use a fixed MLP architecture with the Jacobian penalty. The sample size-dependent architecture is mainly a theoretical tool for controlling approximation error, as done in DNN-based nonparametric regression literature [32, 33, 39, 55, 56], so Theorem 4.1 can be interpreted as an existence result rather than as a practical architectural prescription. In particular, the theoretically required width can be large even at moderate dimensions, and for $d > 2$, the rate $\widetilde{n}^{-2/d}$ reflects the curse of dimensionality. Convergence rates depending on the intrinsic dimension may be possible under additional low-dimensional structure [57]. Furthermore, the Lipschitz constraint in Theorem 4.1 is practically enforced by a Jacobian penalty in our experiments, which encourages using the Lipschitz continuity.

Therefore, we empirically assess the numerical sensitivity of BROT to architecture choice and the Jacobian penalty weight on the synthetic dataset used in Section 5.1. For $n \in \{500, 1000, 2000, 3000, 5000, 8000, 10000\}$, we compare (i) three architectures with the Jacobian penalty weight fixed to 10^{-2} and (ii) three Jacobian penalty weights in $\{10^{-4}, 10^{-3}, 10^{-2}\}$ with the fixed architecture used for the main results in Theorem 4.1, under five random seeds. As shown in Figure 5, the estimation errors are broadly similar across the three architectures and the penalty weights, although the largest penalty weight (10^{-2}) leads to a moderately larger error at large n. Overall, BROT performs reasonably well across these choices, suggesting that the practical implementation of BROT (i.e., using a standard MLP with a moderate Jacobian penalty weight) can be easily done without extensive tuning.

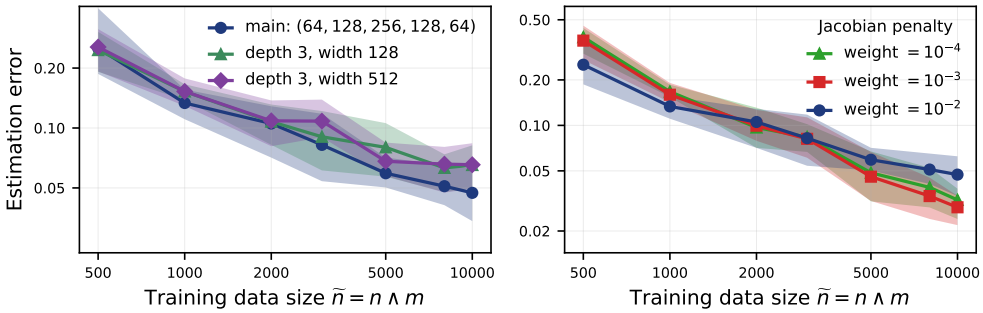


Figure 5: **Sensitivity of DNN architecture and Jacobian penalty to the statistical convergence.** (Left) Three MLP architectures (Jacobian penalty weight fixed to 10^{-2}). (Right) Three Jacobian penalty weights (architecture fixed).

B.3 Quality of Maps: Distribution Matching and Transport Cost (Section 5.2)

Data visualization and full results. Figure 6 visualizes the two synthetic datasets used in Section 5.2. Table 4 reports the full results, including the standard deviations across five random trials.

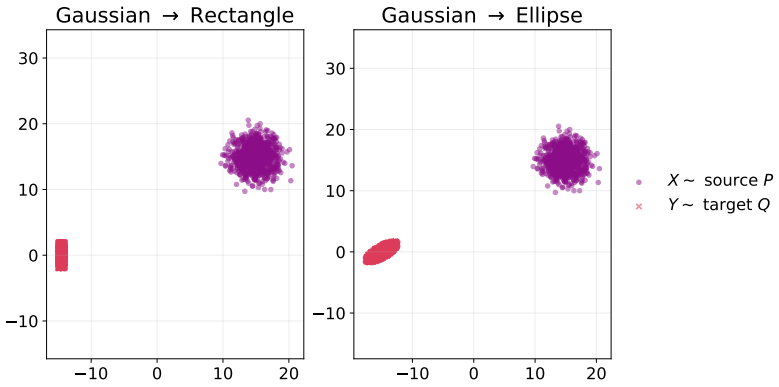


Figure 6: Source (purple) and target (red) data for the two synthetic cases used in Section 5.2.

Table 4: **Comparison of map quality: distribution matching and transport cost.** Results are reported as mean ± standard deviation across five random trials. Best values of `Wass` are **bolded**.

Source → Target	Method	`Wass` ↓	`TC` ↓
Gaussian → Rectangle	1NN	0.126 ± 0.008	1095.67 ± 0.61
	LSOT	0.111 ± 0.003	1095.69 ± 0.14
	ICNN	0.211 ± 0.005	1097.51 ± 0.09
	OTP	0.258 ± 0.008	1099.17 ± 2.39
	DIOTM	0.123 ± 0.007	1098.21 ± 2.35
	MM-v2 [40]	0.120 ± 0.012	1096.57 ± 1.90
	W2 [40]	0.191 ± 0.008	1096.44 ± 1.02
	ENT [41]	0.209 ± 0.017	1097.41 ± 1.84
	BROT	0.110 ± 0.003	1095.65 ± 0.05
Gaussian → Ellipse	1NN	0.207 ± 0.075	1122.30 ± 4.60
	LSOT	0.172 ± 0.036	1124.35 ± 0.59
	ICNN	0.171 ± 0.009	1122.54 ± 1.28
	OTP	0.596 ± 0.038	1129.88 ± 7.39
	DIOTM	0.228 ± 0.020	1123.99 ± 5.45
	MM-v2 [40]	0.174 ± 0.010	1125.20 ± 2.01
	W2 [40]	0.163 ± 0.013	1124.44 ± 1.61
	ENT [41]	0.257 ± 0.030	1125.98 ± 2.77
	BROT	0.155 ± 0.009	1125.08 ± 0.33

Training and inference time. Table 5 reports the training and inference times on a single RTX 3090, averaged over five random trials and normalized by BROT ($= 100\%$). Training time includes initialization, OT computation, DNN fitting, and validation, excluding data loading and hyperparameter search. For inference, we use batch size 128 and measure the elapsed time on $1{,}000$ synthetic or 500 AFHQv2 test points. 1NN is up to $4.8\times$ slower than BROT at inference because it performs a nearest neighbor search over the training data for each test point. LSOT is 1.5-$2.7\times$ slower than BROT at training, while the min-max learning methods (ICNN, OTP, and DIOTM) train 8-$50\times$ slower than BROT.

Table 5: **Comparison of computation time.** Relative training and inference times averaged over five random trials, normalized by BROT ($= 100\%$). The synthetic results are averaged over the two synthetic datasets.

Method	Synthetic cases		AFHQv2 wild → cat	
	Train (%)	Inference (%)	Train (%)	Inference (%)
1NN [12]	20%	339%	17%	486%
LSOT [18]	267%	95%	154%	108%
ICNN [19]	2129%	87%	4985%	229%
OTP [21]	1078%	105%	785%	286%
DIOTM [22]	1445%	81%	2958%	205%
MM-v2 [40]	469%	1658%	4422%	543%
W2 [40]	439%	1602%	4443%	559%
ENT [41]	15%	1211%	13%	1520%
BROT	100%	100%	100%	100%

Decoded images of transported source representation using BROT (Section 5.2). Figure 7 shows training source wild animal images and decoded output images using BROT, with unpaired real target images illustrating the target cat image distribution. We use a fixed AFHQ-fine-tuned SD-VAE decoder. We can see that the key semantics of the source images are preserved well in the decoded images of BROT (e.g., the background colors and the fur colors). For the quantitative comparison, we also compute FID against $2{,}000$ real target images. BROT achieves lower FID than all the baselines, as shown in Table 6.



(a) Source (b) BROT decoded (c) Target

Figure 7: **Decoded BROT outputs on AFHQv2 wild → cat.** (a) Source images. (b) Decoded BROT outputs corresponding by position to (a). (c) Unpaired target examples.

Table 6: **Comparison of FIDs of the decoded transported images on AFHQv2.**

Method	1NN [12]	LSOT [18]	ICNN [19]	OTP [21]	DIOTM [22]	MM-v2 [40]	W2 [40]	ENT [41]	**BROT**
FID ↓	22.41	33.01	58.73	59.03	32.38	40.89	78.11	31.40	**18.67**

B.4 Application 1: Single-Cell Perturbation Prediction (Section 5.3)

We follow the experimental setup described in Section 5.3. For MM-v2 and W2, we select the architecture and learning rate using the validation MMD for the ixazomib perturbation, from widths $(128, 128, 64)$ or $(256, 256, 128)$ and learning rates $3 \cdot 10^{-4}$ or 10^{-3}, with batch size 256. For ENT, we similarly select ρ using the validation split and use the selected value $\rho = 0.001$ for all drug perturbations. Table 7 reports the full results, including the standard deviation across the 35 drug perturbations.

Table 7: **Comparison of single-cell perturbation prediction performance.** The results are reported as mean ± standard deviation across the 35 drug perturbations. Among the transport estimators (excluding Identity) that satisfy $\texttt{MMD} \leq 0.028$, the best value is **bolded** and the second-best is underlined for each metric.

Method	MMD ↓	L_2 difference ↓	r-std ↑	TC ↓
Identity	0.028 ± 0.029	1.213 ± 0.899	0.917 ± 0.081	0.000 ± 0.000
1NN [12]	**0.002** ± 0.000	**0.189** ± 0.062	0.990 ± 0.009	6.745 ± 6.108
LSOT [18]	0.012 ± 0.002	0.287 ± 0.121	0.982 ± 0.013	4.381 ± 5.437
ICNN [19]	<u>0.006</u> ± 0.001	0.258 ± 0.096	0.987 ± 0.008	4.329 ± 5.751
OTP [21]	0.035 ± 0.029	1.435 ± 0.852	0.917 ± 0.081	0.514 ± 0.141
DIOTM [22]	**0.002** ± 0.001	0.255 ± 0.101	0.990 ± 0.010	3.993 ± 5.875
MM-v2 [40]	**0.002** ± 0.001	0.226 ± 0.078	0.990 ± 0.006	3.984 ± 5.821
W2 [40]	**0.002** ± 0.001	0.199 ± 0.086	0.990 ± 0.007	<u>3.934</u> ± 4.954
ENT [41]	**0.002** ± 0.000	0.204 ± 0.072	<u>0.992</u> ± 0.006	6.291 ± 6.397
BROT	**0.002** ± 0.000	<u>0.194</u> ± 0.076	**0.993** ± 0.005	**3.928** ± 5.356

B.5 Application 2: Unsupervised Domain Adaptation (Section 5.4)

We follow the experimental setup described in Section 5.4. After training each OT map, we hold out 20% of the transported source training data as validation data and select the classifier iteration by the source validation accuracy. For MM-v2 and W2, we select the network width from (d, d, d) and $(2d, 2d, d)$ and the learning rate from $3 \cdot 10^{-4}$ and 10^{-3}, with batch size 256. The downstream classifier is trained using Adam with learning rate 10^{-3} and batch size 512.

Table 8 reports the results as mean ± standard deviation over three random trials. USPS contains 7,291 source images, MNIST contains 60,000 target training and 10,000 test images, and VISDA-17 contains 152,397 synthetic source and 55,388 real target images.

Table 8: **Comparison of unsupervised domain adaptation performance.** Per-class mean accuracy (%) on the target-domain test data, reported as mean ± standard deviation over three random trials.

	USPS → MNIST		VISDA-17 (synthetic → real)	
Method	CLIP-ViT/B-32	DINOv2-ViT/B-14	CLIP-ViT/B-32	DINOv2-ViT/B-14
1NN [12]	80.34 ± 0.34	75.25 ± 0.77	74.17 ± 0.08	74.39 ± 0.01
LSOT [18]	30.12 ± 2.84	52.20 ± 0.88	76.87 ± 0.63	<u>81.57</u> ± 0.18
ICNN [19]	75.76 ± 4.63	54.70 ± 21.08	74.12 ± 1.07	77.77 ± 0.41
OTP [21]	49.98 ± 4.79	57.36 ± 5.18	76.15 ± 0.42	64.51 ± 0.37
DIOTM [22]	82.71 ± 1.00	77.46 ± 0.51	76.45 ± 0.55	27.05 ± 35.31
MM-v2 [40]	<u>84.00</u> ± 1.53	**77.73** ± 5.08	<u>76.95</u> ± 0.23	74.81 ± 0.71
W2 [40]	80.50 ± 3.34	75.49 ± 2.63	74.70 ± 0.81	70.40 ± 0.57
ENT [41]	79.76 ± 4.13	76.21 ± 0.52	73.88 ± 0.30	74.28 ± 0.30
BROT	**84.83** ± 1.13	<u>77.60</u> ± 0.24	**79.34** ± 0.07	**81.59** ± 0.30

Design of classifier head. We use a linear classifier for VISDA-17 and an MLP classifier for USPS→MNIST. This choice is based on the source-validation performance of the Identity baseline: the pretrained representations are already largely linearly separable on VISDA-17, whereas a nonlinear classifier performs substantially better on USPS→MNIST.

Comparison with end-to-end OT-based domain adaptation methods. End-to-end OT-based domain adaptation methods such as DeepJDOT [58], JUMBOT [59], and ELOT [60] are more tailored for domain adaptation than BROT. They jointly train the representation encoder, the classifier head, and the transport plan. In contrast, BROT estimates the OT map solely from the Euclidean distances between source and target data, and does not train the encoder. To see whether BROT can nevertheless achieve competitive performance, we compare it against these three end-to-end methods.

Table 9: **Comparison of domain adaptation performance for BROT and end-to-end OT-based domain adaptation methods.** The performance is evaluated by the per-class mean accuracy (%) on the target-domain test data, following [51]. The results of BROT are averaged over three random seeds, while baseline results are copied from the original papers.

Method	Train		Backbone	Accuracy
	Encoder	Head		
DeepJDOT [58]	✓	✓	ResNet-50	68.00
JUMBOT [59]	✓	✓	ResNet-50	72.50
ELOT [60]	✓	✓	ResNet-50	76.32
BROT	✗	✓	CLIP-ViT/B-32	79.34
	✗	✓	DINOv2-ViT/B-14	81.59
	✗	✓	ResNet-50	70.55

Table 9 summarizes the results. BROT attains a per-class mean accuracy of 79.34% with CLIP-ViT/B-32 and 81.59% with DINOv2-ViT/B-14, both higher than the end-to-end methods (DeepJDOT 68.0, JUMBOT 72.5, ELOT 76.32). The results suggest that, with strong frozen visual encoders, BROT can be competitive even without encoder fine-tuning or a task-specific training objective.

For a more fair comparison, we also report the performance of BROT using the fixed representations obtained from pretrained ResNet-50. BROT attains 70.55%, which exceeds DeepJDOT but falls below JUMBOT and ELOT, as expected since BROT does not jointly train the encoder. This gap suggests that the joint training can modestly improve domain adaptation performance, however, the relatively small gap indicates that BROT remains competitive even under the same encoder.

B.6 Computational Complexity

Using full data, we compute a single (full) empirical OT plan from all source and target data. For $n = m$, Step 1 of BROT using the full data has $O(n^2 d)$ complexity for building the cost matrix, $O(n^3)$ complexity for solving the linear program, and $O(n^2)$ peak memory [29].

The chunked procedure used in large-scale experiments, which is briefly introduced in Appendix B.1, uses $O(n/b)$ chunk pairs of size b when $n = m$, so the complexities of the cost matrix construction and the OT solver become $O(nbd)$ and $O(nb^2)$, respectively, and the peak memory becomes $O(b^2)$. Thus, a large ambient dimension d mainly affects the cost-matrix construction, whereas the chunk size b determines the OT solver time and peak memory of the chunked procedure. Step 2 costs $O(nN_{\text{param}})$ per epoch, where N_{param} is the number of DNN parameters, and inference costs $O(N_{\text{param}})$ per query.

Table 10 compares these computational costs with those of the existing statistically optimal estimators. It is also noteworthy that the cost of the exact OT computation is not specific to BROT, since the minimax optimal 1NN estimator computes the same OT plan. Further, the two methods differ at inference: 1NN stores the entire training data and requires an $O(nd)$ search per query, whereas BROT requires only a single forward pass, which makes 1NN up to 4.86 times slower than BROT at inference (see Table 5).

Table 10: **Computational complexity of OT map estimators (up to logarithmic factors).** N_{param} is the number of DNN parameters and b is the chunk size in chunked OT, with $n = m$.

Method	Computation
Wavelet [11]	Optimizes the semi-dual objective over a truncated wavelet expansion whose number of coefficients grows exponentially in d.
1NN [12]	OT plan: $O(n^3)$. Inference: $O(nd)$ per query.
Kernel [16]	Worst-case $O(n^5)$.
BROT (full)	OT plan: $O(n^3)$. DNN training: $O(nN_{\text{param}})$ per epoch. Inference: $O(N_{\text{param}})$ per query.
BROT (chunked)	OT plan: $O(nb^2)$. DNN training and inference: same as above.

Effect of the chunk size. We measure the computation time of the OT plan in the setting of Section 5.1 with $n = m$ and chunk size $b = 512$. As shown in Table 11, the computation time of full OT grows by a factor of about 187 as n grows from 1,000 to 10,000, whereas that of chunked OT grows only by a factor of about 10, following the linear $O(nb^2)$ scaling.

We vary the chunk size b in the Gaussian $\to$ rectangle experiment ($n = 3{,}000$) and the AFHQv2 experiment ($d = 4{,}096$) of Section 5.2, and compare the results with those of BROT using full OT. As shown in Table 12, even a moderate chunk size attains `Wass` and `TC` close to those obtained with full OT on both datasets. We expect that increasing the chunk size with n, e.g., $b = n^\kappa$ for $0 < \kappa < 1$, would further reduce the gap from full OT while keeping the OT computation at $O(n^{1+2\kappa})$. A theoretical analysis of this regime is left for future work.

Table 11: **OT computation time of Step 1.** Time (seconds) in the setting of Section 5.1, with chunk size $b = 512$ for chunked OT.

$n = m$	Full	Chunked	Speedup
1,000	0.184	0.086	2.1×
2,000	0.863	0.173	5.0×
5,000	6.698	0.425	15.8×
10,000	34.427	0.885	38.9×

Table 12: **Map quality vs. Chunk size.** The results from the experiments in Section 5.2 with various chunk sizes $b \in \{512, 1{,}024, 2{,}048\}$. The full results are in Table 1.

	Rectangle		AFHQv2	
b	`Wass`	`TC`	`Wass`	`TC`
512	0.110	1097.00	45.490	2000.92
1,024	0.107	1096.95	45.770	2009.81
2,048	0.106	1097.19	45.550	1993.18

Entropic regularization: computation and inference cost. We further compare BROT with the entropic map estimator (ENT) [41] in the ground-truth-map setting of Section 5.1. For ENT, we report the result with the lowest estimation error over the grid $\varepsilon \in \{0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 1.0\}$, which gives a favorable comparison to ENT.

As shown in Table 13, the OT computation times of BROT and ENT are comparable at $n = 5{,}000$, while BROT becomes faster as the sample size increases and achieves a lower estimation error at every sample size. BROT is also faster at inference, since ENT is nonparametric and its inference cost grows with the number of target samples, whereas BROT requires only a DNN forward pass.

Table 14 further shows that ENT is sensitive to the choice of ε. Reducing ε below approximately 0.1 does not improve the estimation error in this experiment, while substantially increasing the computation time. Moreover, the appropriate value of ε is not known in advance: evaluating the full grid takes 448 seconds, compared with 78.2 seconds for the entire BROT procedure. Thus, accounting for hyperparameter selection can further increase the practical computational cost of ENT.

Table 13: **Comparison between BROT and ENT [41] in terms of computation time.** (Upper) OT computation time in seconds, with estimation error in parentheses. (Lower) Elapsed to transport $20{,}000$ test data points, in milliseconds. For ENT, the best ε over the evaluated grid is reported.

$\widetilde{n} = m$	ENT	**BROT**
OT computation: seconds (estimation error)		
5,000	4.38 (0.075)	**5.19 (0.066)**
8,000	11.52 (0.057)	**9.06 (0.052)**
10,000	16.82 (0.055)	**10.99 (0.047)**
Inference: milliseconds		
5,000	9.83	**6.01**
8,000	15.19	**5.79**
10,000	18.75	**6.10**

Table 14: **Sensitivity of ENT to the regularization strength ε.** Computation time and estimation error at $\widetilde{n} = 10{,}000$. The total time and error of BROT are shown for reference.

ε	Time (s)	Error
1.0	4.68	0.348
0.5	6.23	0.113
0.3	8.26	0.078
0.2	10.59	0.064
0.1	16.82	0.055
0.05	28.71	0.056
0.02	372.75	0.059
Sweep over the grid	448.00	–
BROT (total)	**78.20**	**0.047**

B.7 Robustness to Input Perturbations

The DNN class in Theorem 4.1 is Lipschitz-constrained, so $\|\widehat{\mathbf{T}}(x+\delta) - \widehat{\mathbf{T}}(x)\| \le (\lambda + c')\|\delta\|$ for any input perturbation δ with $x, x+\delta \in \Omega$. That is, the output difference grows at most linearly with the perturbation size. We also examine this property empirically in the experiment of Section 5.1 with $n = 5{,}000$ and the Jacobian penalty weight 10^{-2}, applying random L_2 perturbations and a 20-step PGD attack [61]. PGD maximizes $\|\widehat{\mathbf{T}}(x+\delta) - \widehat{\mathbf{T}}(x)\|_2$ subject to $\|\delta\|_2 \le \varepsilon_{\text{pert}}$. Table 15 reports the mean output difference, which grows approximately linearly with $\varepsilon_{\text{pert}}$. The largest observed output-to-input displacement ratio decreases from 11.4-18.5 without the penalty to 5.7-5.8 with penalty weight 10^{-2}, showing that the Jacobian penalty makes the trained map less sensitive to both random and adversarial perturbations.

Table 15: **Average output difference under input perturbations.** Results in the setting of Section 5.1 with $n = 5{,}000$.

$\varepsilon_{\text{pert}}$	Random L_2	20-step PGD
0.05	0.089	0.118
0.10	0.179	0.237
0.20	0.357	0.471
0.50	0.897	1.180

C Background on Existing Estimators

Wavelet-based estimator. [11] proposed an estimator for smooth OT maps based on a wavelet approximation of the potential function. They considered the empirical dual objective

$$S_n(\phi) := \int \phi(x) dP_n(x) + \int \phi^*(y) dQ_n(y),$$

where $\phi^*(y) := \sup_{x\in\Omega}\{\langle x, y\rangle - \phi(x)\}$ denotes the convex conjugate of ϕ. They then restricted ϕ to a finite-dimensional wavelet class $\mathcal{F}_J^{\mathrm{W}} := \{\phi_w(x) = \sum_{j=1}^J w_j\psi_j(x) : w = [w_1, \ldots, w_J]^\top \in \mathbb{R}^J\}$, where $\psi_j, j \in [J]$ are suitable wavelet basis functions. The wavelet-based estimator is then defined as $\widehat{\phi}^{\mathrm{W}} \in \arg\min_{\phi\in\mathcal{F}_J^{\mathrm{W}}} S_n(\phi)$, $\widehat{\mathbf{T}}_{nm}^{\mathrm{W}} := \nabla\widehat{\phi}^{\mathrm{W}}$. As a result, $\widehat{\mathbf{T}}_{nm}^{\mathrm{W}}$ attains the minimax optimal convergence rate. However, selecting the basis functions $\psi_j, j \in [J]$, as well as the number of basis functions J, and computing the coefficients $w_j, j \in [J]$, requires solving a high-dimensional constrained optimization problem whose complexity grows quickly with the dimension, which severely limits its applicability, as discussed in [11] and [12].

Nearest neighbor estimator. As a complementary line of work, [12] introduced an estimator for Lipschitz OT maps that also attains the minimax optimal rate. Based on the barycentric map $\mathbf{T}_{\hat{\Gamma}}$, which is defined only on the observed source data $X_i, i \in [n]$, they extended it to all $x \in \Omega$ using Voronoi cells $V_i, i \in [n]$:

$$\widehat{\mathbf{T}}_{nm}^{\mathrm{1NN}}(x) = \sum_{i=1}^n \mathbf{T}_{\hat{\Gamma}}(X_i)\mathbb{I}\left(x \in V_i\right),$$

where $V_i := \left\{x \in \Omega : \ \|x - X_i\| \le \|x - X_j\|, \ \forall j \neq i\right\}$. As a result, $\widehat{\mathbf{T}}_{nm}^{\mathrm{1NN}}$ achieves the optimal convergence rate when $\mathbf{T}_0$ is Lipschitz. On the downside, inference at a new point x requires a nearest neighbor search among the n training data, so both memory and latency scale with n, which can be prohibitive for large-scale applications.

Estimators in general classes of functions. More recently, [15] proposed a general estimation framework for rich classes of functions of convex potentials. Similar to the wavelet-based estimator, they considered $\widehat{\phi}_{\mathcal{F}} \in \arg\min_{\phi\in\mathcal{F}} S_n(\phi)$ and define $\widehat{\mathbf{T}}_{\mathcal{F}} := \nabla\widehat{\phi}_{\mathcal{F}}$, where $\mathcal{F}$ is a given candidate class of potentials. They also derived oracle inequalities showing that the risk $\mathbb{E}\|\widehat{\mathbf{T}}_{\mathcal{F}} - \mathbf{T}_0\|_{L^2(P)}^2$ decomposes into an approximation term and a complexity term governed by the metric entropy of $\mathcal{F}$, and they showed that, for a suitably chosen class $\mathcal{F}$ such as ReQU DNNs, the resulting estimator attains the minimax optimal rate.

From a computational viewpoint, however, this plug-in strategy remains challenging. Evaluating the dual objective $S_n(\phi)$ requires computing the conjugate $\phi^*(y) = \sup_x\{\langle x, y\rangle - \phi(x)\}$ at many target points y. When ϕ is parameterized by a neural network, each conjugate evaluation amounts to solving a high-dimensional inner optimization problem over x, which is nonconvex in general and as hard as global optimization. Input convex neural networks (ICNNs) [19] offer a partial algorithmic remedy: by designing the architecture so that ϕ is convex, the inner problem defining $\phi^*(y)$ becomes a convex optimization and can be approximately solved by standard first-order methods. However, despite the practical implementation, the statistical optimality of ICNN-based OT estimators remains unclear, since the approximation capabilities of ICNNs (i.e., their ability to approximate smooth convex functions) are not yet well-investigated, which is a promising topic.

Neural OT estimators. [23] studied semi-dual min-max OT with separately parameterized potentials and maps. For the exact empirical saddle point $\widehat{\mathbf{T}}$, they derived a bound of the form

$$\mathbb{E}\|\widehat{\mathbf{T}} - \mathbf{T}_0\|_{L^2(P)}^2 \le \tau + O(n^{-1/2}) + O(m^{-1/2}),$$

where the network class is fixed in advance to achieve a prescribed approximation error $\tau > 0$. For a fixed class, however, τ does not decrease with the sample size, so this bound does not directly provide a vanishing minimax rate. Moreover, decreasing τ generally requires a richer network class, and the dependence of the statistical terms on this class must also be controlled.

To establish a minimax rate, one would therefore need to let the network class grow with the sample size, control the resulting approximation and statistical errors jointly, and compare the resulting

rate with the minimax lower bound. For BROT, we explicitly construct a sample size-dependent DNN class whose approximation error vanishes at the required rate, and Theorem 4.1 shows that the resulting estimation rate matches the minimax lower bound up to logarithmic factors. The minimax optimality of semi-dual neural OT estimators remains an interesting open problem.

Barycentric estimators. We here provide detailed comparison between BROT and existing barycentric estimators that use the barycentric targets when estimating the OT map. Remark C.1 compares BROT with LSOT from [18], while Remark C.2 compares BROT with a joint estimation of the coupling and map from [30].

Remark C.1 (BROT vs. LSOT)**.** *We use the unregularized OT plan in BROT despite the computational benefits often associated with regularized plans such as LSOT, for the following reasons. First, our minimax optimal analysis in Theorem 4.1 applies to the unregularized OT estimator, whereas the minimax optimality for the regularized DNN estimators is unknown. Second, regularization introduces bias relative to the unregularized ground-truth OT map, which is empirically shown in Table 1. That is, BROT achieves lower* `Wass` *than LSOT with a comparable* `TC`*. Third, the runtime efficiency of using the regularized OT plan is not substantial in our experiments. Table 5 shows that LSOT trains comparably to or sometimes slower than BROT in our experiments. This is because LSOT sometimes requires many iterations to learn the dual potentials by stochastic ascent until convergence, while BROT solves a single linear program for the first step. The training cost and inference cost of the second step are almost identical for the two methods, when they use the same architectures. Hence, the unregularized objective does not lead to a substantial runtime disadvantage.*

Remark C.2 (BROT vs. joint coupling–map estimation)**.** *[30] also considered learning an out-of-sample map from barycentric targets. Their method jointly estimates the coupling and a linear or kernel map by alternating optimization, so the regression objective also affects the coupling. In contrast, BROT first computes the unregularized empirical OT plan and then fits a DNN to its barycentric targets. Their analysis provides an error decomposition, but does not establish a sample-size-dependent convergence rate, hence its minimax optimality is unknown. In contrast, Theorem 4.1 shows that the DNN estimator of BROT attains the minimax rate up to logarithmic factors.*

Kernel-based estimator. [16] proposed a statistically optimal estimator based on a kernel sum-of-squares formulation. Their method estimates a scalar transport potential and obtains the OT map by differentiating the estimated potential, whereas BROT directly fits the transport map to empirical barycentric targets. However, their exact estimator is difficult to implement in practice, and also has a worst-case computational cost of $O(n^5)$ up to a logarithmic factor, while the empirical OT step of BROT costs $O(n^3)$ for full OT and $O(nb^2)$ with chunking.

D Broader Impacts

This work is a methodological contribution to optimal transport map estimation and does not propose a deployed system or release a generative model intended for direct societal use. The estimator can in principle be applied within downstream pipelines such as domain adaptation and biological perturbation prediction, where the broader impact would be inherited from the surrounding application rather than from the OT estimator itself. We therefore do not foresee specific negative societal effects beyond those already associated with the application areas in which OT methods are deployed.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction in Section 1 state our proposed framework, theoretical properties, and empirical results. The main claims are supported by Sections 4 and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion in Section 6 states limitations of this work and future directions. Appendix B.2 also clarifies the gap between the theoretical architecture and the practical implementation of DNN.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The assumptions are stated in Section 4 and Appendix A.1, and the proof of Theorem 4.1 is given in Appendix A.4.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main experimental setup is described in Section 5. Dataset construction, model architecture choices, and hyperparameter search grids are given in Appendix B. We also provide source code as supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: On the first page, we provide a link to the public repository containing our source code. The datasets used in the paper are public and are cited in Section 5. Implementation details are given in Appendix B.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training/test splits, model architectures, optimizers, hyperparameter grids, and validation criteria are described in Section 5 and Appendix B.1. Experiment-specific settings are given in Appendices B.2 to B.5; Algorithm 1 includes the practical Jacobian-penalized training loss.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Standard deviations for map quality, single-cell prediction, and domain adaptation are reported in Tables 4, 7 and 8. Figure 3 reports one-standard-error bands for the convergence experiment.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experiments use NVIDIA RTX 3090 GPUs, with one GPU per process for the timing comparisons. Common implementation and per-process timing settings are stated in Appendices B.1 and B.3; runtimes and complexity are reported in Table 5 and Appendix B.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work uses public datasets and standard benchmarks, and does not involve data collection from human subjects or release of high-risk models.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: A short discussion of broader impacts is given in Appendix D. This work is a methodological contribution to OT map estimation and does not deploy or release a generative model intended for direct societal use.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk pretrained model, image generator, or language model.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The datasets, software packages, and pretrained encoders are cited in Section 5 and Appendix B.1 and the experiment-specific appendices.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The code is publicly available at the link on the first page. Implementation and reproduction details are described in Appendix B.1 and Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This paper does not include crowdsourcing experiments or conducted human-subject studies.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not include crowdsourcing experiments or newly conducted human-subject studies, so we believe that IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development and experiments do not use LLMs as an important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.