Robust First and Second-Order Differentiation for Regularized Optimal Transport

Xingjie Li[1], Fei Lu[2], Molei Tao[3], Felix X.-F. Ye[4]

Abstract

Applications such as unbalanced and fully shuffled regression can be approached by optimizing regularized optimal transport (OT) distances, such as the entropic OT and Sinkhorn distances. A common approach for this optimization is to use a first-order optimizer, which requires the gradient of the OT distance. For faster convergence, one might also resort to a second-order optimizer, which additionally requires the Hessian. The computations of these derivatives are crucial for efficient and accurate optimization. However, they present significant challenges in terms of memory consumption and numerical instability, especially for large datasets and small regularization strengths. We circumvent these issues by analytically computing the gradients for OT distances and the Hessian for the entropic OT distance, which was not previously used due to intricate tensor-wise calculations and the complex dependency on parameters within the bi-level loss function. Through analytical derivation and spectral analysis, we identify and resolve the numerical instability caused by the singularity and ill-posedness of a key linear system. Consequently, we achieve scalable and stable computation of the Hessian, enabling the implementation of the stochastic gradient descent (SGD)-Newton methods. Tests on shuffled regression examples demonstrate that the second stage of the SGD-Newton method converges orders of magnitude faster than the gradient descent-only method while achieving significantly more accurate parameter estimations.

1 Introduction

Optimal transport (OT) provides a powerful tool for finding a map between source and target distributions, especially when they are represented by ensemble samples without correspondence. Examples include shuffled regression [1, 17, 23], unlabeled sensing [12, 33, 34, 37], homomorphic sensing [31, 32], regression with an unknown permutation [19], or more broadly, as regression without correspondence [3, 17, 22, 27, 36].

The task is to find a parameterized function $y = F(x;\theta)$ that maps ensemble of sources $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^M \in \mathbb{R}^{M\times D}$ to targets $\boldsymbol{Y}^* = \{\boldsymbol{y}_j^*\}_{j=1}^N \in \mathbb{R}^{N\times d}$ with probability weights $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. Here, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)^\top$ and $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_N)^\top$ satisfy $\boldsymbol{\mu}^\top \mathbb{1}_M = \sum_{i=1}^M \mu_i = 1, \boldsymbol{\nu}^\top \mathbb{1}_N = \sum_{j=1}^N \nu_j = 1$ and $0 < \mu_i, \nu_j < 1$. Note that M may not be equal to N, and the same applies to D and d. The absence of a one-to-one correspondence between the source and target data samples makes classical supervised regression methods inapplicable.

The OT solution finds an optimal θ by minimizing a loss function $\mathcal{L}(\theta)$ between the image of the source data $\boldsymbol{Y}_\theta = F(\boldsymbol{X};\theta)$ and the target data $\boldsymbol{Y}^*$, that is,

$$\min_\theta \mathcal{L}(C^{\boldsymbol{Y}_\theta \to \boldsymbol{Y}^*}, \boldsymbol{\mu}, \boldsymbol{\nu}), \tag{1}$$

where the cost matrix $C_{ij}^{\boldsymbol{Y}_\theta \to \boldsymbol{Y}^*} = c(\boldsymbol{y}_i(\theta), \boldsymbol{y}_j^*)$ and c is a function of cost between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j^*$. Throughout this study, we assume that $c(\boldsymbol{y}, \boldsymbol{y}^*)$ is twice-differentiable, for instance, the squared Euclidean distance $c(\boldsymbol{y}, \boldsymbol{y}^*) = \|\boldsymbol{y} - \boldsymbol{y}^*\|_2^2$. Several popular OT distances are candidates for the loss function, including the Wasserstein distance, the entropy-regularized OT (EOT) distance, and the Sinkhorn distance [24]; see Section 2.1 for a brief review. Each of them leads to a bi-level optimization problem; for example, the EOT distance $\mathrm{OT}_\epsilon(C^{\boldsymbol{Y}_\theta \to \boldsymbol{Y}^*}, \boldsymbol{\mu}, \boldsymbol{\nu})$ leads to

$$\min_\theta \min_{\substack{\Pi \in \mathbb{R}_{\geqslant 0}^{M\times N}, \\ \Pi \mathbb{1}_N = \boldsymbol{\mu}, \Pi^\top \mathbb{1}_M = \boldsymbol{\nu}}} \sum_{i=1}^M \sum_{j=1}^N C_{ij}^{\boldsymbol{Y}_\theta \to \boldsymbol{Y}^*} \Pi_{ij} + \epsilon \mathrm{KL}(\Pi, \boldsymbol{\mu} \otimes \boldsymbol{\nu}).$$

A crucial component of the optimization process is computing the derivatives of the loss function with respect to θ, and hence the derivatives of the OT distance with respect to data $\boldsymbol{Y}_\theta$. First-order optimization methods require the gradients of the OT distances. Danskin's theorem provides analytical gradients for the EOT distance [5,8,14,15], but it does not apply to the Sinkhorn distance. Hence, the generic implicit differentiation method [20] has been widely applied to OT distances [6,10,11,35,36]. However, first-order optimization methods often converge slowly.

[1]Department of Mathematics and Statistics, University of North Carolina, at Charlotte
[2]Department of Mathematics, Johns Hopkins University
[3]School of Mathematics, Georgia Institute of Technology
[4]Department of Mathematics & Statistics, University at Albany

To accelerate the convergence, a common strategy involves using stochastic gradient descent (SGD) initially, followed by Newton method's iterations, which necessitates computing the Hessian. Automatic differentiation and implicit differentiation are the two main methods for computing the Hessian [6, 9, 10]; see Section 3.1 for a detailed discussion. However, both methods encounter significant challenges, such as memory shortages when the dataset is large and numerical instability due to singularity or ill-posedness, particularly when the entropy regularization strength ϵ is small. These issues impede the success of the SGD-Newton strategy.

We solve these issues by introducing analytical gradients for the OT distances and an analytical Hessian for the EOT. In particular, we achieve scalable and stable computation of the Hessian by using the analytical expression to locate and resolve the singularity or ill-posedness through spectral analysis. Our Algorithm 1 significantly outperforms the automatic differentiation and implicit differentiation method in runtime and accuracy by orders of magnitudes; see Section 5. As a result, we enable the success of the SGD-Newton strategy for accelerating the bi-level optimization, as we demonstrate on parameter estimation for shuffled regression of mixed Gaussian and 3D Point Clouds Registration of MobilNet10 dataset [18, 25]; see Section 6.

The key in our derivation is the linear system for the optimal dual potentials, e.g., (17) or (19), which is inspired by the implicit differentiation in [6] and the second-order Fréchet derivative of the Sinkhorn divergence loss under the Wasserstein metric in [28]. Emerging from the implicit differentiation, this linear system facilitates efficient computation of the gradient of the OT distances as well as the Hessian of the EOT distance. In particular, when used together with Eq.(18) from the marginal constraints, it bridges implicit differentiation and Danskin's theorem in the context of EOT distance.

Furthermore, we provide a comprehensive spectral analysis for the linear system for the dual potentials through the matrix

$$H(\Pi) := \begin{bmatrix} \operatorname{diag}(\Pi \mathbb{1}_N) & \Pi \\ (\Pi)^\top & \operatorname{diag}(\Pi^\top \mathbb{1}_M) \end{bmatrix} \in \mathbb{R}^{(M+N)\times(M+N)}, \tag{2}$$

where Π is the coupling matrix. We show that when Π has positive entries, $H(\Pi)$ has zero as a simple eigenvalue, and its effective condition number (i.e., the ratio of the largest and smallest positive eigenvalues) has upper and lower bounds depending on the spectral gap of $\Pi^\top\Pi$. In particular, we construct an example showing that $H(\Pi)$ can be severely ill-conditioned with the smallest positive eigenvalue at the order of $O(e^{-\frac{1}{\epsilon}})$ when ϵ is small, or $O(\frac{1}{N})$ when N is large. Thus, when solving a linear system with H, proper regularization is crucial.

Our main contributions are threefold.

- *Analytical derivatives and spectral analysis.* We derive analytical gradients with respect to the data $\boldsymbol{Y}$ for EOT and Sinkhorn distances and Hessian for the EOT distance in Section 3.2–3.3. The spectral analysis in Section 4 helps us understand and resolve the numerical instability issue via a proper regularization in Section 3.4.

- *Fast stable computation of Hessian.* Our algorithm enables a stable, memory-efficient, and fast computation of the Hessian, significantly outperforming other state-of-the-art methods in runtime and accuracy by orders of magnitudes; see Section 5.

- *Enabling SGD-Newton for shuffled regression.* With the robust computation of the Hessian, we are able to apply the SGD-Newton method to shuffled regression problems in Section 6, significantly accelerating the optimization process.

1.1 Outline

This work is organized as follows. Section 2 reviews the various (OT) distances and the Sinkhorn algorithm. Section 3.2 is devoted to the analytical and numerical computation of the gradients and Hessian, leading to an algorithm with proper regularization. Then, we analyze the spectrum of the matrix $H(\Pi)$ in Section 4. In Section 5, we examine the efficiency and accuracy of Hessian computation using a benchmark example and compare the results with other approaches. Then we apply the proposed algorithm to applications in Section 6, including the parameter estimation for shuffled regression of mixed Gaussian and 3D Point Clouds Registration of MobilNet10 dataset.

2 Optimal Transport Loss and Sinkhorn Algorithm

Ideally, we could find θ^* by minimizing the optimal transport distance between the parameterized source data $\boldsymbol{Y}_\theta$ and the target data $\boldsymbol{Y}^*$. We will first review some classical results in computational optimal transport [24] in this section. In the section, we ignore θ in the notation and C is the abbreviation of $C^{\boldsymbol{Y}\to\boldsymbol{Y}^*}$, unless noted otherwise.

2.1 Optimal Transport Loss Functions

2.1.1 Wasserstein-2 Metric

One popular choice is to use the Wasserstein-2 metric as the optimal transport loss, equivalently, $\mathcal{L}(C, \boldsymbol{\mu}, \boldsymbol{\nu}) = W_2^2(C, \boldsymbol{\mu}, \boldsymbol{\nu})$. To calculate the Wasserstein-2 metric, one has to solve a constraint optimization problem,

$$W_2^2(C, \boldsymbol{\mu}, \boldsymbol{\nu}) := \min_{\substack{\Pi \in \mathbb{R}_{\geqslant 0}^{M \times N}, \\ \Pi \mathbb{1}_N = \boldsymbol{\mu}, \Pi^\top \mathbb{1}_M = \boldsymbol{\nu}}} \sum_{i=1}^{M} \sum_{j=1}^{N} C_{ij} \Pi_{ij}, \quad C_{ij} = c(\boldsymbol{y}_i, \boldsymbol{y}_j^*), \tag{3}$$

where $c(\boldsymbol{y}_i, \boldsymbol{y}_j^*) = \|\boldsymbol{y}_i - \boldsymbol{y}_j^*\|_2^2$ is the cost of transport, and the *coupling matrix* $\Pi \in [0,1]^{M \times N}$ is the transport plan from the parameterized source data $\boldsymbol{Y}$ to the target data $\boldsymbol{Y}^*$. To solve the constraint optimization via linear programming, the computational complexity is $O((N+M)NM \log(N+M))$ [24], which is very expensive when N, M are large. To overcome this issue, one often regularizes the objective function. Common regularization includes the EOT distance, the Sinkhorn distance, which we briefly review below.

2.1.2 Entropy-regularized OT (EOT) Distance

The EOT distance is the Wasserstein-2 loss plus the relative entropy between two measures:

$$\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu}) := \min_{\substack{\Pi \in \mathbb{R}_{\geqslant 0}^{M \times N}, \\ \Pi \mathbb{1}_N = \boldsymbol{\mu}, \Pi^\top \mathbb{1}_M = \boldsymbol{\nu}}} \sum_{i=1}^{M} \sum_{j=1}^{N} C_{ij} \Pi_{ij} + \epsilon \mathrm{KL}(\Pi, \boldsymbol{\mu} \otimes \boldsymbol{\nu}), \tag{4}$$

where the *relative entropy* between the coupling matrix Π and the outer product $\boldsymbol{\mu} \bigotimes \boldsymbol{\nu}$ is $\mathrm{KL}(\Pi, \boldsymbol{\mu} \otimes \boldsymbol{\nu}) := \sum_{i=1}^{M} \sum_{j=1}^{N} \Pi_{ij} \log \frac{\Pi_{ij}}{\mu_i \nu_j}$. This regularization drastically simplifies the study of the dual problem and further leads to the Sinkhorn algorithm for its unique numerical solution [24]. As ϵ goes to 0, EOT converges to the Wasserstein-2 distance at the rate of ϵ [20].

2.1.3 Sinkhorn Distance

Another candidate for the regularized OT loss is called the Sinkhorn distance, $\widetilde{\mathrm{OT}}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$,

$$\widetilde{\mathrm{OT}}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu}) := \sum_{ij} C_{ij} \Pi_{ij}^*, \text{ with } \Pi^* = \operatorname*{arg\,min}_{\substack{\Pi \in \mathbb{R}_{\geqslant 0}^{M \times N}, \\ \Pi \mathbb{1}_N = \boldsymbol{\mu}, \Pi^\top \mathbb{1}_M = \boldsymbol{\nu}}} \sum_{ij} C_{ij} \Pi_{ij} + \epsilon \mathrm{KL}(\Pi, \boldsymbol{\mu} \otimes \boldsymbol{\nu}). \tag{5}$$

Sinkhorn distance eliminates the contribution of the entropy regularization term from $\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$ to the total loss $\mathcal{L}$ after the transport plan Π^* has been obtained. It actually gives even better approximation results and converges to the Wasserstein-2 distance exponentially fast. More precisely, we have $\left|\widetilde{\mathrm{OT}}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu}) - W_2^2(C, \boldsymbol{\mu}, \boldsymbol{\nu})\right| \leqslant c \exp(-1/\epsilon)$ [20].

2.2 Sinkhorn Algorithm

From now on, we firstly choose the EOT cost as the loss function, i.e., $\mathcal{L}(C, \boldsymbol{\mu}, \boldsymbol{\nu}) = \mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$ though we will discuss the others later. The computation of this quantity, i.e. the constraint optimization (4) is solved by the well-known Sinkhorn algorithm. For the notation consistency and ease on the readers, we will review necessary details of this algorithm. One can account for the constraints by introducing two slack variables, known as the dual potentials, $\boldsymbol{f} \in \mathbb{R}^M, \boldsymbol{g} \in \mathbb{R}^N$, for each marginal constraint of (4). Because $\sum_i \sum_j \mu_i \nu_j = \sum_i \mu_i \sum_j \nu_j = 1$, the corresponding augmented Lagrangian is

$$\begin{aligned} \mathcal{L}(C, \Pi, \boldsymbol{f}, \boldsymbol{g}) = & \sum_{ij} C_{ij} \Pi_{ij} + \epsilon \sum_{ij} \Pi_{ij} \left(\log \frac{\Pi_{ij}}{\mu_i \nu_j} - 1 \right) \\ & + \epsilon - \sum_i f_i (\sum_j \Pi_{ij} - \mu_i) - \sum_j g_j (\sum_i \Pi_{ij} - \nu_j) \,. \end{aligned} \tag{6}$$

The first-order optimality condition $\frac{\partial \mathcal{L}(C, \Pi, \boldsymbol{f}, \boldsymbol{g})}{\partial \Pi_{ij}} = 0$ yields the expression of the optimal coupling Π^* is

$$\Pi_{ij}^* = \mu_i \nu_j \exp\left(\frac{-C_{ij} + f_i^* + g_j^*}{\epsilon} \right), \tag{7}$$

where f_i^* and g_j^* are the optimal dual potentials. It ensures that the optimal coupling matrix Π^* is entry-wise positive. An intuitive scheme is to alternatively rescale rows and columns of the Gibbs kernel to satisfy the marginal constraint, which is called *Sinkhorn algorithm* [7, 21, 29, 30]. Numerically, however, this computation becomes unstable when ϵ is small. The stable Sinkhorn iteration is thus performed in the log-domain [7, 24],

$$\boldsymbol{f}^{(l+1)} = \epsilon \log \boldsymbol{\mu} - \epsilon \log \left(K \exp(\boldsymbol{g}^{(l)}/\epsilon) \right), \ \boldsymbol{g}^{(l+1)} = \epsilon \log \boldsymbol{\nu} - \epsilon \log \left(K^\top \exp(\boldsymbol{f}^{(l+1)}/\epsilon) \right)$$

with the initial vector to be $\boldsymbol{g}^{(0)} = \mathbb{0}_N$ and the Gibbs kernel $K = \exp\left(-\frac{C}{\epsilon}\right)$. As l goes to $+\infty$, both converge to $\boldsymbol{f}^*$ and $\boldsymbol{g}^*$. In practice, the iteration stops when the 1-norm of marginal violation is within the threshold value.

Then we get the EOT distance $\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$ in terms of the dual problem,

$$\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu}) = \mathcal{L}(C, \Pi^*, \boldsymbol{f}^*, \boldsymbol{g}^*) = \begin{bmatrix} \boldsymbol{\mu}^\top & \boldsymbol{\nu}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{f}^* \\ \boldsymbol{g}^* \end{bmatrix}. \tag{8}$$

Overall the computational complexity of EOT to achieve τ-approximate of the unregularized OT problem is $O(N^2 \log(N) \tau^{-3})$ when $M = N$ [2,24], which is significant improvement to the linear programming of Wasserstein-2 metric.

Similarly, $\widetilde{\mathrm{OT}}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$ gives

$$\widetilde{\mathrm{OT}}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu}) = \sum_{ij} C_{ij} \Pi^*_{ij} = \sum_{ij} \mu_i \nu_j C_{ij} e^{\frac{-C_{ij} + f_i^* + g_j^*}{\epsilon}}. \tag{9}$$

3 Differentiation of Loss Functions

In this section, we introduce robust computations for the gradients of these regularized OT loss functions and for the Hessian of the EOT distance. Our computations show that costly backward propagation can be avoided even for the Hessian.

We recall the problem setup (1) and first study the analytic form of the gradient and hessian of EOT distance with respect to the parameter θ, which read

$$\frac{\partial \mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \theta_i} = \sum_{k=1}^{M} \frac{\partial \boldsymbol{y}_k}{\partial \theta_i} \frac{\partial \mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \boldsymbol{y}_k}, \tag{10}$$

$$\begin{aligned} \frac{\partial^2 \mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \theta_i \partial \theta_j} &= \sum_{k=1}^{M} \sum_{s=1}^{M} \left(\frac{\partial \boldsymbol{y}_s}{\partial \theta_i} \right) \frac{\partial^2 \mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \boldsymbol{y}_s \partial \boldsymbol{y}_k} \left(\frac{\partial \boldsymbol{y}_k}{\partial \theta_j} \right)^\top \\ &\quad + \sum_{k=1}^{M} \frac{\partial^2 \boldsymbol{y}_k}{\partial \theta_i \partial \theta_j} \frac{\partial \mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \boldsymbol{y}_k}. \end{aligned} \tag{11}$$

Here C_θ is the abbreviation of $C^{\boldsymbol{Y}_\theta \to \boldsymbol{Y}^*}$ emphasizing the dependence on $\boldsymbol{Y}_\theta$. The key step is to find the explicit expression for the first derivatives with respect to the source data $\frac{\partial \mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \boldsymbol{y}_k} \in \mathbb{R}^d$ and the second-order derivatives with respect to source data $\frac{\partial^2 \mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \boldsymbol{y}_s \partial \boldsymbol{y}_k} \in \mathbb{R}^{d \times d}$.

3.1 Previous methods computing first- and second-order derivatives

For the analytical expression of the first-order derivatives of regularized OT distances, there are two main approaches: the Danskin approach and the implicit differentiation approach. The Danskin approach computes the gradient of EOT, based on applying the Danskin's theorem to the dual function [5,8,14,15]. Recall the Lagrangian $\mathcal{L}(C, \Pi, \boldsymbol{f}, \boldsymbol{g})$ is a function of $C, \Pi, \boldsymbol{f}$ and $\boldsymbol{g}$ and $\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu}) = \max_{\Pi, \boldsymbol{f}, \boldsymbol{g}} \mathcal{L}(C, \Pi, \boldsymbol{f}, \boldsymbol{g})$. The Danskin's theorem states that given $\boldsymbol{f}^*, \boldsymbol{g}^*, \boldsymbol{\Pi}^*$, the gradient of $\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$ with respect to $\boldsymbol{y}_k$ is

$$\begin{aligned} \frac{\partial \mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \boldsymbol{y}_k} &= \sum_{ij} \frac{\partial C_{ij}}{\partial \boldsymbol{y}_k} \frac{\partial \mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial C_{ij}} = \sum_{ij} \frac{\partial C_{ij}}{\partial \boldsymbol{y}_k} \frac{\partial \mathcal{L}(C, \Pi^*, \boldsymbol{f}^*, \boldsymbol{g}^*)}{\partial C_{ij}} \\ &= \sum_{ij} \frac{\partial C_{ij}}{\partial \boldsymbol{y}_k} \Pi^*_{ij} = \sum_{j} \frac{\partial C_{kj}}{\partial \boldsymbol{y}_k} \Pi^*_{kj}. \end{aligned} \tag{12}$$

Once Π^* is obtained from Sinkhorn algorithm, the first-order derivative with respect to the source dataset $\boldsymbol{Y}$ immediately follows without additional computational cost. The Danskin's approach naturally extends to

the de-biased Sinkhorn divergence $S_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})$, but unfortunately this approach doesn't work on the Sinkhorn distances $\widetilde{\mathrm{OT}}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})$ because it is not of the form $\max_{\Pi, \boldsymbol{f}, \boldsymbol{g}} \phi(C, \Pi, \boldsymbol{f}, \boldsymbol{g})$. An implicit differentiation approach is thus introduced in [20]. It implicitly differentiates the associated marginal constraints to derive a large linear system, which is solved by the conjugate gradient solver in *lineax* [26]. It applies to different regularized OT problems [6, 10, 11, 35, 36].

For computing the Hessian of the OT distances, up to our knowledge, there is no direct analytical expression before the current work. There are two approaches suggested by *OTT* [9]. The first approach *unrolls* the Sinkhorn iterations and use the *JAX* in-build tools to handle autodiff via backward propagation and computational graph. The second approach *implicitly* differentiates the optimal solutions computed by *OTT*. The implicit differentiation approach involves differentiating the solution of an ill-conditioned linear system with the custom differentiation rules [9], hence, regularization techniques, such as preconditioning the marginal constraints [10] and the ridge regularization, have been introduced to try to resolve this issue. However, as we will demonstrate in Section 5, both approaches still encounter two major challenges: (i) memory shortages when the dataset is large, and (ii) numerical instability due to singularity and the ill-posed nature of the linear system, particularly when ϵ is small.

The key in our derivation is the linear system for the optimal dual potentials, e.g., (17) or (19), emerged from the application of the implicit differentiation. It facilitates efficient computation of the gradient of the Sinkhorn distance as well as the Hessian of the EOT distance. Additionally, together with (18), it bridges implicit differentiation and Danskin's theorem in the context of the EOT distance.

3.2 Analytical computation of the gradients

We first review the implicit differentiation approach to the gradient of the EOT distance $\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$ with respect to source data $\boldsymbol{Y}$ and re-derive the result of Danskin's approach in (12) through the key observation. We further provide a novel numerical method to efficiently compute the derivative of Sinkhorn distance $\widetilde{\mathrm{OT}}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$.

Gradient of $\mathbf{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$. We first consider the gradient of the EOT distance with respect to the source data $\boldsymbol{y}_k$

$$\frac{\partial \mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \boldsymbol{y}_k} = \sum_{i=1}^{M} \mu_i \frac{\partial f_i^*}{\partial \boldsymbol{y}_k} + \sum_{j=1}^{N} \nu_j \frac{\partial g_j^*}{\partial \boldsymbol{y}_k} = \begin{bmatrix} \boldsymbol{\mu}^\top & \boldsymbol{\nu}^\top \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{f}^*}{\partial \boldsymbol{y}_k} \\ \frac{\partial \boldsymbol{g}^*}{\partial \boldsymbol{y}_k} \end{bmatrix}, \tag{13a}$$

where $\frac{\partial \boldsymbol{f}^*}{\partial \boldsymbol{y}_k} = \left(\frac{\partial f_1^*}{\partial \boldsymbol{y}_k}, \ldots, \frac{\partial f_M^*}{\partial \boldsymbol{y}_k}\right)^\top \in \mathbb{R}^{M \times d}$, and similarly $\frac{\partial \boldsymbol{g}^*}{\partial \boldsymbol{y}_k} \in \mathbb{R}^{N \times d}$. To simplify the notation, we denote $\left(\frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}}\right)_{iks} = \frac{\partial f_i^*}{\partial (\boldsymbol{y}_k)_s}$ and $\left(\frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}}\right)_{jks} = \frac{\partial g_j^*}{\partial (\boldsymbol{y}_k)_s}$, so $\frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}} \in \mathbb{R}^{M \times M \times d}$ and $\frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}} \in \mathbb{R}^{N \times M \times d}$.

We write the gradient in the vector form

$$\frac{d\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\boldsymbol{Y}} = \boldsymbol{\mu}^\top \frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}} + \boldsymbol{\nu}^\top \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}} = \begin{bmatrix} \boldsymbol{\mu}^\top & \boldsymbol{\nu}^\top \end{bmatrix} \begin{bmatrix} \frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}} \\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}} \end{bmatrix} \in \mathbb{R}^{M \times d}. \tag{13b}$$

Theorem 1. *The derivative of EOT distance in* (8) *with respect to source data* $\boldsymbol{y}_k$*, as in* (13a)*, is given by*

$$\frac{\partial \mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})}{\partial \boldsymbol{y}_k} = \sum_{j=1}^{N} \frac{\partial C_{kj}}{\partial \boldsymbol{y}_k} \Pi_{kj}^*, \quad k = 1, \ldots, M. \tag{14}$$

In vector form, the derivative of $\mathrm{OT}_\epsilon(C, \mu, \nu)$ *with respect to whole source data* $\boldsymbol{Y}$ *is*

$$\frac{d\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\boldsymbol{Y}} = \mathcal{B} \cdot \mathbb{1}_N, \tag{15}$$

where $\mathcal{B} \in \mathbb{R}^{M \times d \times N}$ *is a tensor with entries* $\mathcal{B}_{ksj} = \frac{\partial C_{kj}}{\partial (\boldsymbol{y}_k)_s} \Pi_{kj}^*$*, and* $\mathcal{B} \cdot \mathbb{1}_N \in \mathbb{R}^{M \times d}$ *is the dot product which is the summation of the third index such that the k-th column is* $(\mathcal{B} \cdot \mathbb{1}_N)_k = \sum_j \frac{\partial C_{kj}}{\partial \boldsymbol{y}_k} \Pi_{kj}^*$ *for* $1 \leqslant k \leqslant M$*.*

Proof. The main task is to find $\frac{\partial \boldsymbol{f}^*}{\partial \boldsymbol{y}_k}$ and $\frac{\partial \boldsymbol{g}^*}{\partial \boldsymbol{y}_k}$ in (13a) using (7) and the marginal probability conditions. The partial derivative of entries of optimal coupling matrix Π^* with respect to $\boldsymbol{y}_k$ is

$$\frac{\partial \Pi_{ij}^*}{\partial \boldsymbol{y}_k} = \frac{\Pi_{ij}^*}{\epsilon} \left(-\frac{\partial C_{ij}}{\partial \boldsymbol{y}_k} + \frac{\partial f_i^*}{\partial \boldsymbol{y}_k} + \frac{\partial g_j^*}{\partial \boldsymbol{y}_k} \right) = \frac{\Pi_{ij}^*}{\epsilon} \left(-\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k} \delta_{ik} + \frac{\partial f_i^*}{\partial \boldsymbol{y}_k} + \frac{\partial g_j^*}{\partial \boldsymbol{y}_k} \right), \tag{16}$$

where δ_{ik} is a Kronecker delta function. We observe that with the marginal probability conditions $\sum_{j=1}^{N}\Pi^*_{ij}=\mu_i$ and $\sum_{i=1}^{M}\Pi^*_{ij}=\nu_j$, and by taking the partial derivative $\frac{\partial}{\partial \boldsymbol{y}_k}$ on both sides of these marginal constraints, we can get

$$0=\sum_{j=1}^{N}\frac{\partial \Pi^*_{ij}}{\partial \boldsymbol{y}_k}=\frac{\mu_i}{\epsilon}\frac{\partial f_i^*}{\partial \boldsymbol{y}_k}-\frac{1}{\epsilon}\left[\sum_{j=1}^{N}\left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\delta_{ik}-\frac{\partial g_j^*}{\partial \boldsymbol{y}_k}\right)\Pi^*_{ij}\right],$$
$$0=\sum_{i=1}^{M}\frac{\partial \Pi^*_{ij}}{\partial \boldsymbol{y}_k}=\frac{\nu_j}{\epsilon}\frac{\partial g_j^*}{\partial \boldsymbol{y}_k}-\frac{1}{\epsilon}\left[\sum_{i=1}^{M}\left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\delta_{ik}-\frac{\partial f_i^*}{\partial \boldsymbol{y}_k}\right)\Pi^*_{ij}\right].$$

Hence we have

$$\mu_i\frac{\partial f_i^*}{\partial \boldsymbol{y}_k}+\sum_{j=1}^{N}\frac{\partial g_j^*}{\partial \boldsymbol{y}_k}\Pi^*_{ij}=\sum_{j=1}^{N}\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\Pi^*_{ij}\delta_{ik}\in\mathbb{R}^d,\quad \forall 1\leqslant i\leqslant M,$$
$$\sum_{i=1}^{M}\frac{\partial f_i^*}{\partial \boldsymbol{y}_k}\Pi^*_{ij}+\nu_j\frac{\partial g_j^*}{\partial \boldsymbol{y}_k}=\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\Pi^*_{kj}\in\mathbb{R}^d,\quad \forall 1\leqslant j\leqslant N.$$

With $H(\Pi)$ defined in (2), the above linear system can be written in matrix form,

$$\underbrace{\begin{bmatrix}\operatorname{diag}(\boldsymbol{\mu}) & \Pi^*\\ (\Pi^*)^\top & \operatorname{diag}(\boldsymbol{\nu})\end{bmatrix}}_{=H(\Pi*)}\begin{bmatrix}\frac{\partial \boldsymbol{f}^*}{\partial \boldsymbol{y}_k}\\ \frac{\partial \boldsymbol{g}^*}{\partial \boldsymbol{y}_k}\end{bmatrix}=\begin{bmatrix}\mathbf{e}_k(B_k\mathbb{1}_N)^\top\\ B_k^\top\end{bmatrix},\tag{17}$$

where $\mathbf{e}_k$ is the k-th standard column basis of $\mathbb{R}^M$ and the matrix $B_k\in\mathbb{R}^{d\times N}$ is $(B_k)_{sj}=\frac{\partial C_{kj}}{\partial(\boldsymbol{y}_k)_s}\Pi^*_{kj}$. Instead of solving the above linear system to evaluate (13a), we observe from the marginal constraints that

$$\begin{bmatrix}\frac{1}{M}\mathbb{1}_M^\top & \frac{1}{N}\mathbb{1}_N^\top\end{bmatrix}H(\Pi^*)=\left(\frac{1}{M}+\frac{1}{N}\right)\begin{bmatrix}\boldsymbol{\mu}^\top & \boldsymbol{\nu}^\top\end{bmatrix}.\tag{18}$$

Then, multiplying both sides of (17) by $\begin{bmatrix}\frac{1}{M}\mathbb{1}_M^\top & \frac{1}{N}\mathbb{1}_N^\top\end{bmatrix}$, we obtain

$$\begin{bmatrix}\frac{1}{M}\mathbb{1}_M^\top & \frac{1}{N}\mathbb{1}_N^\top\end{bmatrix}H(\Pi^*)\begin{bmatrix}\frac{d\boldsymbol{f}^*}{d\boldsymbol{y}_k}\\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{y}_k}\end{bmatrix}\begin{bmatrix}\frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}}\\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}}\end{bmatrix}=\begin{bmatrix}\frac{1}{M}\mathbb{1}_M^\top & \frac{1}{N}\mathbb{1}_N^\top\end{bmatrix}\begin{bmatrix}\mathbf{e}_k(B_k\mathbb{1}_N)^\top\\ B_k^\top\end{bmatrix}.$$

Hence, the derivative of $\mathrm{OT}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})$ with respect to source data $\boldsymbol{y}_k$ is

$$\frac{d\mathrm{OT}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})}{d\boldsymbol{y}_k}=\begin{bmatrix}\boldsymbol{\mu}^\top & \boldsymbol{\nu}^\top\end{bmatrix}\begin{bmatrix}\frac{d\boldsymbol{f}^*}{d\boldsymbol{y}_k}\\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{y}_k}\end{bmatrix}$$
$$=\frac{1}{\frac{1}{M}+\frac{1}{N}}\begin{bmatrix}\frac{1}{M}\mathbb{1}_M^\top & \frac{1}{N}\mathbb{1}_N^\top\end{bmatrix}\begin{bmatrix}\mathbf{e}_k(B_k\mathbb{1}_N)^\top\\ B_k^\top\end{bmatrix}B_k\mathbb{1}_N=\sum_{j=1}^{N}\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\Pi^*_{kj}.$$

If we define the third-order tensor $\mathcal{B}$ by stacking the matrices $\{B_k\}_{k=1}^M$, that is the k-th component $\mathcal{B}_{k\cdot\cdot}=B_k$, then the linear system (17) can be further vectorized as

$$H(\Pi^*)\begin{bmatrix}\frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}}\\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}}\end{bmatrix}=\mathcal{R},\qquad\text{with}\quad \mathcal{R}:=\begin{bmatrix}\operatorname{diag}(\mathcal{B}\cdot\mathbb{1}_N)\\ \mathcal{B}^\top\end{bmatrix}\in\mathbb{R}^{(M+N)\times M\times d}\tag{19}$$

where $\operatorname{diag}(\mathcal{B}\cdot\mathbb{1}_N)$ is a third-order tensor, with $\operatorname{diag}(\mathcal{B}\cdot\mathbb{1}_N)_{kks}=(\mathcal{B}\cdot\mathbb{1}_N)_{ks}$ and zeros for the other entries, and $\mathcal{B}^\top$ is the transpose of permutation of indices defined by $\left(\mathcal{B}^\top\right)_{ijk}=\mathcal{B}_{kij}$. Hence, by plugging (19) back to (13b), the derivative of $\mathrm{OT}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})$ with respect to the source data $\boldsymbol{Y}$ in the tensor form is equal to $\frac{d\mathrm{OT}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})}{d\boldsymbol{Y}}=\mathcal{B}\cdot\mathbb{1}_N$. □

This theorem not only implies that automatic differentiation is unnecessary for computing the gradient of EOT distance but also that solving the large linear system (19) for $\frac{d\boldsymbol{f}^*}{d\boldsymbol{y}_k}$ and $\frac{d\boldsymbol{g}^*}{d\boldsymbol{y}_k}$ is not needed. The analytical expression

of the gradient follows directly from the cost matrix C_θ and optimal coupling matrix Π^* from the Sinkhorn algorithm, which can significantly speed up the computation. This theorem provides a different approach to derive the same analytical result other than the Danskin's theorem. We emphasize that the gradient formula (15) is generic and applies to any choice of cost matrix C without requiring $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ to be uniform. In particular, if the Sinkhorn iteration stops early and the coupling matrix $\widehat{\Pi}$ is suboptimal, the formula is still exact for $\frac{d\mathrm{OT}_\epsilon(C,\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\nu}})}{d\boldsymbol{Y}}$ given suboptimal coupling matrix $\widehat{\Pi}$ and associated marginals $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\nu}}$, where $\sum_j \widehat{\Pi}_{ij} = \tilde{\mu}_i$ and $\sum_i \widehat{\Pi}_{ij} = \tilde{\nu}_j$.

Gradient of $\widetilde{\mathrm{OT}}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})$. The above techniques also applies for computing the gradient of the Sinkhorn distance $\widetilde{\mathrm{OT}}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})$. Although it does not yield an analytical expression, it helps significantly reduce the computational cost by avoiding directly solving the linear system (19), which is costly since the right-hand-side is a large third-order tensor $\mathcal{R}$. Specifically, we have the following novel result.

Proposition 2. *The gradient of Sinkhorn distance in* (9) *with respect to the source data* $\boldsymbol{Y}$ *is*

$$\frac{d\widetilde{\mathrm{OT}}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})}{d\boldsymbol{Y}} = \left(\mathcal{B} - \frac{\mathcal{A}}{\epsilon}\right)\cdot \mathbb{1}_N + \frac{1}{\epsilon}\boldsymbol{r}^\top \mathcal{R}, \tag{20}$$

where $\mathcal{B}$ *and* $\mathcal{R}$ *are defined in* (19)*, the third-order tensor* $\mathcal{A}$ *is* $\mathcal{A}_{ksj} = \frac{\partial C_{kj}}{\partial (\boldsymbol{y}_k)_s} C_{kj}\Pi^*_{kj}$*. The vector* $\boldsymbol{r}$ *is the solution of the linear system,* $H(\Pi^*)\boldsymbol{r} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}$*, where vectors* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *have entries* $a_i = \sum_{j=1}^M C_{ij}\Pi^*_{ij}$ *and* $b_j = \sum_{i=1}^N C_{ij}\Pi^*_{ij}$*.*

Proof. From (9) and (16), the derivative of Sinkhorn distance with respect to the data y_k is

$$\frac{d\widetilde{\mathrm{OT}}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})}{d\boldsymbol{y}_k} = \sum_{j=1}^N \left(1 - \frac{C_{kj}}{\epsilon}\right)\Pi^*_{kj}\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k} + \frac{1}{\epsilon}\left(\sum_{i=1}^M a_i \frac{\partial f_i^*}{\partial \boldsymbol{y}_k} + \sum_{j=1}^N b_j \frac{\partial g_j^*}{\partial \boldsymbol{y}_k}\right),$$

where $a_i = \sum_{j=1}^N C_{ij}\Pi^*_{ij}$, and $b_j = \sum_{i=1}^M C_{ij}\Pi^*_{ij}$. The main computation burden is the second term. To compute $\left(\sum_i a_i \frac{\partial f_i^*}{\partial \boldsymbol{y}_k} + \sum_j b_j \frac{\partial g_j^*}{\partial \boldsymbol{y}_k}\right)$, a common practice is to solve the linear system (19) to get all $\frac{\partial f^*}{\partial \boldsymbol{y}_k}$ and $\frac{\partial g^*}{\partial \boldsymbol{y}_k}$, which is computationally expensive because the right-hand-side of (19) is a third-order tensor $\mathcal{R}$.
Although we do not have (18) as $\boldsymbol{a}$ and $\boldsymbol{b}$ are no longer the marginal constraints of Π^*, we can yet apply the same technique to reduce the computation costs by solving the following matrix-vector form of linear system only once $H(\Pi^*)\boldsymbol{r} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \in \mathbb{R}^{(M+N)}$.

That is, we only need to solve for one time the matrix-vector form of the linear system, for the column vector $\boldsymbol{r}$. This can be efficiently solved by conjugate gradient method with early stopping.
Notice that $\begin{bmatrix} \boldsymbol{a}^\top & \boldsymbol{b}^\top \end{bmatrix}\begin{bmatrix} \mathbb{1}_M \\ -\mathbb{1}_N \end{bmatrix} = \mathbb{0}_{M+N}$, and $\begin{bmatrix} \mathbb{1}_M \\ -\mathbb{1}_N \end{bmatrix}$ is in the kernel space of $H(\Pi^*)$ as proved in Lemma 6, so $\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}$ is in the span of $H(\Pi^*)$, therefore the linear system above always has a solution.
Notice that

$$\sum_i a_i \frac{\partial f_i^*}{\partial \boldsymbol{Y}} + \sum_j b_j \frac{\partial g_j^*}{\partial \boldsymbol{Y}} = \begin{bmatrix} \boldsymbol{a}^\top & \boldsymbol{b}^\top \end{bmatrix}\begin{bmatrix} \frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}} \\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}} \end{bmatrix} = \boldsymbol{r}^\top H(\Pi^*)\begin{bmatrix} \frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}} \\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}} \end{bmatrix} = \boldsymbol{r}^\top \mathcal{R}.$$

Hence, we prove (20). □

3.3 Computation of the Hessian

In this subsection, we *analytically* compute the Hessian of the loss function with respect to the source data $\mathbf{Y}$. We mainly focus on the EOT distance.

Theorem 3. *The second-order derivative of EOT distance* $\mathrm{OT}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})$ *with respect to the source data* $\boldsymbol{Y}$ *is given by the fourth order tensor* $\mathcal{T} \in \mathbb{R}^{M\times d\times M\times d}$

$$\mathcal{T}_{ktsl} = \frac{1}{\epsilon}\sum_{i,j=1}^{M+N} \mathcal{R}_{ikt}H^\dagger_{ij}\mathcal{R}_{jsl} + \mathcal{E}_{ktsl}, \text{ for } k,s = 1,\dots,M \text{ and } t,l = 1,\dots,d, \tag{21}$$

where $H^\dagger$ *is the Moore–Penrose inverse of matrix* $H(\Pi^*) \in \mathbb{R}^{(M+N)\times(M+N)}$ *defined in* (2)*, and* $\mathcal{R} = \begin{bmatrix} diag(\mathcal{B}\cdot \mathbb{1}_N) \\ \mathcal{B}^\top \end{bmatrix} \in \mathbb{R}^{(M+N)\times M\times d}$ *is the right-hand-side third-order tensor defined in* (19)*. The fourth-order tensor* $\mathcal{E} \in \mathbb{R}^{M\times d\times M\times d}$ *is defined as*

$$\mathcal{E}_{ktsl} = \begin{cases} \sum_{j=1}^N \Pi^*_{kj}\left(\left(\frac{\partial^2 C_{kj}}{\partial \boldsymbol{y}_k^2}\right)_{tl} - \frac{1}{\epsilon}\frac{\partial C_{kj}}{\partial(\boldsymbol{y}_k)_t}\cdot\frac{\partial C_{kj}}{\partial(\boldsymbol{y}_k)_l}\right), & \text{if } k = s \\ 0, & \text{Otherwise} \end{cases} \tag{22}$$

for $k, s = 1, \ldots, M$ *and* $t, l = 1, \ldots, d$*.*

Proof. With the help of the gradient given by (14), we know

$$\frac{d^2\mathrm{OT}_\epsilon(C,\mu,\nu)}{d\boldsymbol{y}_s d\boldsymbol{y}_k} = \begin{cases} \sum_{j=1}^N \left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)^\top \frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_s}, & \text{if } k \neq s, \\ \sum_{j=1}^N \frac{\partial^2 C_{kj}}{\partial \boldsymbol{y}_k^2}\Pi^*_{kj} + \sum_{j=1}^N \left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)^\top \frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_k}, & \text{if } k = s. \end{cases} \tag{23}$$

For $k \neq s$, we have $\delta_{ks} = 0$, so the term is expanded as

$$\begin{aligned} \sum_{j=1}^N \left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)^\top \frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_s} &= \sum_{j=1}^N \frac{\Pi^*_{kj}}{\epsilon}\left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)^\top \left(\frac{\partial f_k^*}{\partial \boldsymbol{y}_s} + \frac{\partial g_j^*}{\partial \boldsymbol{y}_s}\right) \\ &= \begin{bmatrix} \boldsymbol{e}_k(B_k\mathbb{1}_N)^\top \\ B_k^\top \end{bmatrix}^\top \frac{H^\dagger}{\epsilon} \begin{bmatrix} \boldsymbol{e}_s(B_s\mathbb{1}_N)^\top \\ B_s^\top \end{bmatrix}, \end{aligned}$$

where we use the fact that $\begin{bmatrix} \frac{\partial \boldsymbol{f}^*}{\partial \boldsymbol{y}_s} \\ \frac{\partial \boldsymbol{g}^*}{\partial \boldsymbol{y}_s} \end{bmatrix}$ is the solution of the linear system (17), so it can be expressed as $\begin{bmatrix} \frac{\partial \boldsymbol{f}^*}{\partial \boldsymbol{y}_s} \\ \frac{\partial \boldsymbol{g}^*}{\partial \boldsymbol{y}_s} \end{bmatrix} = H^\dagger \begin{bmatrix} \mathbf{e}_s(B_s\mathbb{1}_N)^\top \\ B_s^\top \end{bmatrix}$. For $k = s$, the term is expanded as follows

$$\begin{aligned} &\sum_{j=1}^N \frac{\partial^2 C_{kj}}{\partial \boldsymbol{y}_k^2}\Pi^*_{kj} + \sum_{j=1}^N \left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)^\top \frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_k} \\ &= \sum_{j=1}^N \frac{\Pi^*_{kj}}{\epsilon}\left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)^\top \left(\frac{\partial f_k^*}{\partial \boldsymbol{y}_k} + \frac{\partial g_j^*}{\partial \boldsymbol{y}_k}\right) + \sum_{j=1}^N \Pi^*_{kj}\left(\frac{\partial^2 C_{kj}}{\partial \boldsymbol{y}_k^2} - \frac{1}{\epsilon}\left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)^\top \left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)\right) \\ &= \begin{bmatrix} \boldsymbol{e}_k(B_k\mathbb{1}_N)^\top \\ B_k^\top \end{bmatrix}^\top \frac{H^\dagger}{\epsilon} \begin{bmatrix} \boldsymbol{e}_k(B_k\mathbb{1}_N)^\top \\ B_k^\top \end{bmatrix} + \sum_{j=1}^N \Pi^*_{kj}\left(\frac{\partial^2 C_{kj}}{\partial \boldsymbol{y}_k^2} - \frac{(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k})^\top(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k})}{\epsilon}\right). \end{aligned}$$

In vector form, we obtain the Hessian in (21). □

Given the cost matrix C and the optimal coupling matrix Π^* from Sinkhorn algorithm, the Hession tensor is analytically calculated once the large linear system (19) is numerically solved. Unlike the previous approaches, the solution of the linear system is no longer needed to be differentiated in order to obtain the second-order derivatives, so our direct analytical Hessian expression could significantly speed up the computation with less memory burden. Similar to the case of the first-order derivative, the analytical expression for the Hessian is generic and applicable to any choice of cost function. It does not require $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ to be uniform either. If the Sinkhorn iterations stops early with $\widehat{\Pi}$ being suboptimal, then the Hessian expression (21) is still exact for $\frac{d^2\mathrm{OT}_\epsilon(C,\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\nu}})}{d\boldsymbol{Y}^2}$ with suboptimal coupling matrix $\widehat{\Pi}$ and associated marginals $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\nu}}$, where $\sum_j \widehat{\Pi}_{ij} = \tilde{\mu}_i$ and $\sum_i \widehat{\Pi}_{ij} = \tilde{\nu}_j$.

With the analytical expression of Hessian, the following result show that the sum of the first index of the Hessian tensor is only dependent on the marginal probability vector $\boldsymbol{\mu}$. Later, it will be used as a marginal error to verify the accuracy of Hessian in the numerical implementation.

Proposition 4. *If* $C_{kj} = \|\boldsymbol{y}_k - \boldsymbol{y}_j^*\|_2^2$ *for each* k, j*, then* $\sum_{k=1}^M \mathcal{T}_{k\cdot s\cdot} = 2\mu_s \mathbb{I}_d$*.*

Proof. Since the cost is square distance $C_{kj} = \|\boldsymbol{y}_k - \boldsymbol{y}_j^*\|_2^2$, then $\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k} = 2(\boldsymbol{y}_k - \boldsymbol{y}_j^*)$ and $\frac{\partial^2 C_{kj}}{\partial \boldsymbol{y}_k^2} = 2\mathbb{I}_d$. The first term in the second-order derivative (23) expands as

$$\sum_{k=1}^M \sum_{j=1}^N \left(\frac{\partial C_{kj}}{\partial \boldsymbol{y}_k}\right)^\top \frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_s} = 2\sum_{k=1}^M \boldsymbol{y}_k^\top \sum_{j=1}^N \frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_s} - 2\sum_{j=1}^N (\boldsymbol{y}_j^*)^\top \sum_{k=1}^M \frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_s} = \mathbb{0}_{d\times d}.$$

The last equal sign is because that $\sum_j \Pi^*_{kj} = \mu_k$ and $\sum_k \Pi^*_{kj} = \nu_j$, as well as $\sum_j \left(\frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_s}\right) = \frac{\partial \sum_j \Pi^*_{kj}}{\partial \boldsymbol{y}_s}$ and $\sum_k \left(\frac{\partial \Pi^*_{kj}}{\partial \boldsymbol{y}_s}\right) = \frac{\partial \sum_k \Pi^*_{kj}}{\partial \boldsymbol{y}_s}$.

The remaining term is $\sum_{j=1}^N \frac{\partial^2 C_{sj}}{\partial \boldsymbol{y}_s^2} \Pi^*_{sj} = 2\mathbb{I}_d \sum_{j=1}^N \Pi^*_{sj} = 2\mu_s \mathbb{I}_d$. Thus we proved the statement. □

3.4 Solve the linear systems with truncated SVD

A major challenge in computing the gradient of Sinkhorn distance in (20) and the Hessians of Entroy-regularized distance in (21) is that pseudo-inverse $H(\Pi^*)^\dagger$ can severely amplify the rounding errors or early stopping errors when the matrix $H(\Pi^*)$ is ill-conditioned. The pseudo-inverse comes from the solutions to the linear systems (17) and (19) for computation of first- and second- order derivatives. Thus, instead of directly using $H(\Pi^*)^\dagger$, it is important to study the spectrum of the matrix $H(\Pi^*)$ and regularize properly when the linear systems are ill-posed.

Analytical and empirical results in the next section show that the H-matrix is often ill-conditioned when the Sinkhorn regularization parameter ϵ is small or when the optimal coupling matrix Π^* is close to a permutation. Such ill-conditioned H-matrices result in numerically unstable solutions when Π^* is slightly perturbed to $\widehat{\Pi}$ due to the early stopping of the Sinkhorn iterations. This is also the exact reason that the previous implicit differentiation approach fails due to numerically instability.

We tackle the potential ill-posedness by truncated Singular value decomposition (TSVD). We truncate the H-matrix's spectrum up to the K-th largest eigenvalue, with $K = \max\{j : \frac{\lambda_j}{\lambda_1} > \alpha\}$, where $\{\lambda_j\}_{j=1}^{M+N}$ are the eigenvalues of $H(\Pi^*)$ in descending order. In practice, we use the LAPACK's DGELSD algorithm building in the least-square solver [4] and set $\alpha = 10^{-10}$. The algorithm for gradient and Hessian computation is summarized in Algorithm 1.

Input: Optimized Π^*, cost matrix C, entropic regularization strength ϵ, source data $\boldsymbol{Y}$ and singular value threshold α.
Output: Gradient with respect to $\boldsymbol{Y}$: $\frac{d\mathrm{OT}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})}{d\boldsymbol{Y}} \in \mathbb{R}^{M\times d}$, Hessian with respect to $\boldsymbol{Y}$: $\mathcal{T} \in \mathbb{R}^{M\times d\times M\times d}$

1: Compute the marginal probability vector $\boldsymbol{\mu} \in \mathbb{R}^M$ and $\boldsymbol{\nu} \in \mathbb{R}^N$: $\boldsymbol{\mu} \leftarrow \Pi^* \mathbb{1}_N$ and $\boldsymbol{\nu} \leftarrow (\Pi^*)^\top \mathbb{1}_M$.
2: Compute matrix $H \in \mathbb{R}^{(M+N)\times(M+N)}$: $H \leftarrow \begin{bmatrix} \mathrm{diag}(\boldsymbol{\mu}) & \Pi^* \\ (\Pi^*)^\top & \mathrm{diag}(\boldsymbol{\nu}) \end{bmatrix}$.
3: Compute third-order tensor $\mathcal{B} \in \mathbb{R}^{M\times d\times N}$: $\mathcal{B}_{ksj} \leftarrow \frac{\partial C_{kj}}{\partial (\boldsymbol{y}_k)_s} \Pi^*_{kj}$, compute third order tensor $\mathcal{R} \in \mathbb{R}^{(M+N)\times M\times d}$: $\mathcal{R} \leftarrow \begin{bmatrix} \mathrm{diag}(\mathcal{B}\cdot \mathbb{1}_N) \\ \mathcal{B}^\top \end{bmatrix}$ and compute the fourth-order tensor $\mathcal{E}$ in (22).
4: Compute the gradient: $\frac{d\mathrm{OT}_\epsilon(C,\boldsymbol{\mu},\boldsymbol{\nu})}{d\boldsymbol{Y}} = \mathcal{B}\cdot \mathbb{1}_N$.
5: Compute the truncated singular value decomposition of H-matrix up to the K-th largest eigenvalue: $H \approx U_K \Lambda_K U_K^\top$ with $K = \max\{j : \frac{\lambda_1}{\lambda_j} > \alpha\}$.
6: Approximate $\begin{bmatrix} \frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}} \\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}} \end{bmatrix}$: $\begin{bmatrix} \frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}} \\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}} \end{bmatrix} \leftarrow U_K \Lambda_K^{-1} U_K^\top \mathcal{R}$.
7: Compute the Hessian $\mathcal{T}$: $\mathcal{T}_{ktsl} \leftarrow \frac{1}{\epsilon}\sum_{i=1}^{M+N} \mathcal{R}_{ikt} \begin{bmatrix} \frac{d\boldsymbol{f}^*}{d\boldsymbol{Y}} \\ \frac{d\boldsymbol{g}^*}{d\boldsymbol{Y}} \end{bmatrix}_{isl} + \mathcal{E}_{ktsl}$.

Algorithm 1: Computation of gradient and Hessian of EOT distance $\mathrm{OT}_\epsilon(C, \boldsymbol{\mu}, \boldsymbol{\nu})$ with respect to the source data $\boldsymbol{Y}$.

4 Spectral analysis of the H-matrix

This section analyzes the spectrum of the matrice $H(\Pi)$ in (2), which plays a crucial role in the computation of the derivative and the Hessian. Recall that $\Pi \in \mathbb{R}^{M\times N}$ is a *coupling matrix* if its entries are nonnegative and $\boldsymbol{\mu} = \Pi \mathbb{1}_N$ and $\boldsymbol{\nu} = \Pi^\top \mathbb{1}_M$ are marginal probability vectors. We say that it is a *positive coupling matrix* if its entries are all positive.

We consider two common types of coupling matrices Π: (i) positive coupling matrices, since the coupling matrix Π^* obtained from the Sinkhorn algorithm in practice is always positive due to the entropy regularization, [13,24]; and (ii) coupling matrix with uniform marginal distributions, i.e., $\boldsymbol{\mu} = \mathbb{1}_M/M$ and $\boldsymbol{\nu} = \mathbb{1}_N/N$, since they arise in most applications with randomly sampled data.

We show first that $H(\Pi)$ is singular with a simple zero eigenvalue for any positive coupling matrix. For coupling matrices with uniform marginal distributions (including permutation matrices), we analytically calculate the eigenvalues of $H(\Pi)$ in Section 4.2. In particular, we establish lower and upper bounds for the condition number of H-matrix in terms of the spectral gap of Π in Theorem 9. Section 4.3 extends the bounds for the condition number to the H-matrix computed by the Sinkhorn algorithm of EOT distance with early-stopping.

4.1 General positive coupling matrices

We show first that the H-matrix $H(\Pi)$ is singular with a simple zero eigenvalue for any positive coupling matrix Π. As a result, we call $\kappa(H) = \frac{\lambda_1(H)}{\lambda_{N+M-1}(H)}$, the *condition number* of the matrix $H = H(\Pi)$.

Theorem 5 (**Simple zero eigenvalue for the H-matrix**). *For any positive coupling matrix $\Pi \in \mathbb{R}^{M\times N}_{>0}$, the smallest eigenvalue of $H(\Pi)$ is zero and it is simple, with eigenvector $\mathbf{q}_0 = \begin{bmatrix} \mathbb{1}_M \\ -\mathbb{1}_N \end{bmatrix}$.*

Its proof relies on the next Perron-Frobenius type lemma, which shows that the largest eigenvalue of the matrix $\mathrm{diag}(\boldsymbol{\nu})^{-1}\Pi^\top \mathrm{diag}(\boldsymbol{\mu})^{-1}\Pi$ is 1 and is simple.

Lemma 6. *For any coupling matrix with strictly positive entries $\Pi \in \mathbb{R}^{M\times N}_{>0}$, the largest eigenvalue of matrix $diag(\boldsymbol{\nu})^{-1}\Pi^\top\, diag(\boldsymbol{\mu})^{-1}\Pi$ is $\lambda = 1$ and has multiplicity one, with eigenvector $\mathbb{1}_N$. Similarly, the largest eigenvalue of the matrix $diag(\boldsymbol{\mu})^{-1}\Pi\, diag(\boldsymbol{\nu})^{-1}\Pi^\top$ is $\lambda = 1$ and has multiplicity one, with eigenvector $\mathbb{1}_M$.*

Proof of Theorem 5. It is clear that 0 is an eigenvalue with eigenvector $\mathbf{q}_0$. We show next that any other eigenvector of 0 must be $\mathbf{q}_0$ up to a scalar factor. Note that the vector $\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{v} \end{bmatrix}$ is the eigenvector of 0 if and only if

$$\begin{bmatrix} \mathrm{diag}(\boldsymbol{\mu}) & \Pi \\ \Pi^\top & \mathrm{diag}(\boldsymbol{\nu}) \end{bmatrix}\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{v} \end{bmatrix} = \mathbb{0}_{M+N} \Leftrightarrow \begin{bmatrix} \mathrm{diag}(\boldsymbol{\mu})\boldsymbol{w} + \Pi\boldsymbol{v} = 0 \\ \mathrm{diag}(\boldsymbol{\nu})\boldsymbol{v} + \Pi^\top\boldsymbol{w} = 0 \end{bmatrix}, \tag{24}$$

which is equivalent to

$$\boldsymbol{v} = -\mathrm{diag}(\boldsymbol{\nu})^{-1}\Pi^\top\boldsymbol{w} = \mathrm{diag}(\boldsymbol{\nu})^{-1}\Pi^\top\mathrm{diag}(\boldsymbol{\mu})^{-1}\Pi\boldsymbol{v},$$
$$\boldsymbol{w} = -\mathrm{diag}(\boldsymbol{\mu})^{-1}\Pi\boldsymbol{v} = \mathrm{diag}(\boldsymbol{\mu})^{-1}\Pi\mathrm{diag}(\boldsymbol{\nu})^{-1}\Pi^\top\boldsymbol{w}.$$

By Lemma 6, $\boldsymbol{v} = a\mathbb{1}_N$ and $\boldsymbol{w} = b\mathbb{1}_M$ for some nonzero constant a, b for each equation to hold. Plugging back to (24), we have $a = -b$, and $\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{v} \end{bmatrix} = a\mathbf{q}_0$. Thus, the zero eigenvalue of H is simple. □

A quick corollary of the above lemma is that the smallest eigenvalue of $I - \mathrm{diag}(\boldsymbol{\nu})^{-1}\Pi^\top\mathrm{diag}(\boldsymbol{\mu})^{-1}\Pi$ is zero and is simple. As a result, we obtain an invertible matrix after dropping one of the rows.

Corollary 7. *For any coupling matrix Π, the matrix $diag(\overline{\boldsymbol{\nu}}) - \overline{\Pi}^\top\, diag(\boldsymbol{\mu})^{-1}\Pi$ is invertible, where $\overline{\boldsymbol{\nu}}$ and $\overline{\Pi}$ are the arrays after dropping the last rows.*

The invertibility of the above matrix has been used in [6,11,20,36] to remove the zero eigenvalues of $H(\Pi^*)$ in the computation of the gradient of Sinkhorn distance (5). However, the ill-posedness in solving the linear systems is not addressed as the other eigenvalues are still close to 0. Next, we analyze the spectrum of the H-matrix, which guides the treatment of the ill-posedness.

4.2 Coupling matrices with uniform marginal distributions

Coupling matrices with uniform marginal distributions are of particular interest as they arise in many applications where the data are randomly sampled. They lead to H-matrices of the form

$$H(\Pi) := \begin{bmatrix} \mathrm{diag}(\frac{1}{M}\mathbb{1}_M) & \Pi \\ (\Pi)^\top & \mathrm{diag}(\frac{1}{N}\mathbb{1}_N) \end{bmatrix} \in \mathbb{R}^{(M+N)\times(M+N)}. \tag{25}$$

In this section, we compute the eigenvalues of the H-matrices for coupling matrices with uniform marginal distributions (including the permutation matrices) in Proposition 8. These eigenvalues shed light on the root of the ill-conditionedness of the H-matrices. In particular, if the coupling matrix is entrywise positive, we provide upper and lower bounds for the condition number of the H-matrix in terms of the spectral gap of $\Pi^\top\Pi$ in Theorem 9.

Proposition 8 (**Eigenvalues of $H(\Pi)$ and singular values of Π**). *Let $\Pi \in [0,1]^{M\times N}$ be a (not necessarily positive) coupling matrix with uniform marginal distributions. Let $M \leqslant N$ and assume Π has rank M. Then, the eigenvalues (in descending order) and eigenvectors of H defined in (25) are, for $j = 1, \dots, M$,*

$$\lambda_j(H) = \frac{\left(\frac{M+N}{MN}\right) + \sqrt{\left(\frac{M-N}{MN}\right)^2 + 4\sigma_j(\Pi)^2}}{2}, \quad \mathbf{q}_j = \begin{bmatrix} \frac{\kappa_j}{\sqrt{1+\kappa_j^2}}\mathbf{u}_j \\ \frac{1}{\sqrt{1+\kappa_j^2}}\mathbf{v}_j \end{bmatrix},$$
$$\lambda_{M+1}(H) = \cdots = \lambda_N = \frac{1}{N}, \quad \mathbf{q}_i = \begin{bmatrix} \mathbb{0} \\ \boldsymbol{v}_i \end{bmatrix}, M < i \leqslant N, \tag{26}$$
$$\lambda_{N+M+1-j}(H) = \frac{\left(\frac{M+N}{MN}\right) - \sqrt{\left(\frac{M-N}{MN}\right)^2 + 4\sigma_j^2(\Pi)}}{2}, \quad \mathbf{q}_{N+M+1-j} = \begin{bmatrix} \frac{-1}{\sqrt{1+\kappa_j^2}}\mathbf{u}_j \\ \frac{\kappa_j}{\sqrt{1+\kappa_j^2}}\mathbf{v}_j \end{bmatrix},$$

where $\sigma_j(\Pi)$ is the j-th singular value of Π (in descending order) with $\mathbf{u}_j$ and $\mathbf{v}_j$ being the right and left singular vectors, $\{\mathbf{v}_i\}_{i=M+1}^N$ are the $N-M$ orthogonal vectors in the kernel of Π, and these scalars $\kappa_j = \left(\frac{\frac{1}{M}-\frac{1}{N}}{2\sigma_j(\Pi)}\right) + \sqrt{\left(\frac{\frac{1}{M}-\frac{1}{N}}{2\sigma_j(\Pi)}\right)^2 + 1}$.

In particular, when $M = N$ and Π is a permutation coupling matrix, i.e., its rows are a permutation of the rows of $\frac{1}{N}I_N$. Then, the eigenvalues of $H(\Pi)$ are $\lambda_1(H) = \cdots = \lambda_N(H) = \frac{2}{N}$ and $\lambda_{N+1}(H) = \cdots = \lambda_{2N}(H) = 0$.

Proof of Proposition 8. Denote an eigen-pair H by $\left(\lambda, \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}\right)$, where $\mathbf{u} \in \mathbb{R}^M$ and $\mathbf{v} \in \mathbb{R}^N$; that is, $\begin{bmatrix} \frac{1}{M}\mathbb{I}_M & \Pi \\ (\Pi)^\top & \frac{1}{N}\mathbb{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}$.

Then, $\Pi\mathbf{v} = (\lambda - \frac{1}{M})\mathbf{u}$ and $\Pi^\top\mathbf{u} = (\lambda - \frac{1}{N})\mathbf{v}$. This implies that

$$\left(\Pi(\Pi)^\top\right)\mathbf{u} = \left(\lambda - \frac{1}{M}\right)\left(\lambda - \frac{1}{N}\right)\mathbf{u}, \quad \left((\Pi)^\top\Pi\right)\mathbf{v} = \left(\lambda - \frac{1}{M}\right)\left(\lambda - \frac{1}{N}\right)\mathbf{v}. \tag{27}$$

That is, $\mathbf{u}$ and $\mathbf{v}$ are the right and left singular vectors of Π, corresponding to a singular value of Π satisfying $\sigma_j^2 = \left(\lambda - \frac{1}{M}\right)\left(\lambda - \frac{1}{N}\right)$, for some $1 \leqslant j \leqslant M$. Solving this equation, we obtain two eigenvalues $\lambda_\pm = \frac{\left(\frac{1}{M}+\frac{1}{N}\right) \pm \sqrt{\left(\frac{1}{M}-\frac{1}{N}\right)^2 + 4\sigma_j^2}}{2}$.

To compute the eigenvector of λ_+, let $\mathbf{v} = \mathbf{v}_j$. Then $\Pi\mathbf{v} = \sigma_j\mathbf{u}_j = (\lambda_+ - \frac{1}{M})\mathbf{u}$. So $\mathbf{u} = \frac{\sigma_j}{(\lambda_+ - \frac{1}{M})}\mathbf{u}_j$ (recall that $\sigma_j \neq 0$ by $\mathrm{rank}(\Pi) = m$). Note that $\frac{(\lambda_+ - \frac{1}{N})}{\sigma_j} = \frac{\left(\frac{1}{M}-\frac{1}{N}\right) + \sqrt{\left(\frac{1}{M}-\frac{1}{N}\right)^2 + 4\sigma_j^2}}{2\sigma_j}$; hence, we have

$$\frac{\sigma_j}{(\lambda_+ - \frac{1}{M})} = \frac{(\lambda_+ - \frac{1}{N})}{\sigma_j} = \frac{\frac{1}{M}-\frac{1}{N}}{2\sigma_j} + \sqrt{\left(\frac{\frac{1}{M}-\frac{1}{N}}{2\sigma_j}\right)^2 + 1} = \kappa_j$$

So an eigenvector for λ_+ is $\begin{bmatrix} \frac{\kappa_j}{\sqrt{1+\kappa_j^2}}\mathbf{u}_j \\ \frac{1}{\sqrt{1+\kappa_j^2}}\mathbf{v}_j \end{bmatrix}$. Similarly, λ_- has $\begin{bmatrix} \frac{-1}{\sqrt{1+\kappa_j^2}}\mathbf{u}_j \\ \frac{\kappa_j}{\sqrt{1+\kappa_j^2}}\mathbf{v}_j \end{bmatrix}$.

The above $\lambda_\pm$ account for $2M$ eigenvalues. The other $N-M$ eigenvalues correspond to $\sigma_i(\Pi) = 0$, which has singular vectors $\mathbf{v}_i$, for $i = M+1, \dots, N$. Thus, setting $\mathbf{u} = \mathbb{0}$ and $\mathbf{v} = \mathbf{v}_i$, we have $\Pi^\top\mathbf{u} = \mathbb{0} = (\lambda - \frac{1}{N})\mathbf{v}_i$. So $\lambda = \frac{1}{N}$. This eigenvalue has multiplicity of $N-M$, with eigenvectors $\begin{bmatrix} \mathbb{0} \\ \mathbf{v}_i \end{bmatrix}$ for $i = M+1, \dots, N$.

At last, when $M = N$ and Π is a permutation matrix, note that $\Pi^\top\Pi = \frac{1}{N^2}\mathbb{I}_N$. Thus, the singular values of Π are $\sigma_j = \frac{1}{N}$ with multiplicity N. Applying (26), we obtain that the eigenvalues of H are 0 and $\frac{2}{N}$, each with multiplicity N. □

Theorem 9 (**Condition number of $H(\Pi)$**). *Let Π be a positive coupling matrix with uniform marginal distributions and singular values $\{\sigma_k\}$ in descending order. Then the condition number $\kappa(H)$ of H in (25) is bounded by*

$$\frac{(M+N)^2}{2M^2N^2(\sigma_1^2 - \sigma_2^2)} \leqslant \kappa(H) \leqslant \frac{(M+N)^2}{M^2N^2(\sigma_1^2 - \sigma_2^2)}. \tag{28}$$

Proof of Theorem 9. The largest singular value of Π is $\sigma_1 = \frac{1}{\sqrt{NM}}$ and it is simple. Hence, by Prop.8, the largest and smallest eigenvalues of $H(\Pi)$ are $\lambda_1(H) = \frac{1}{M} + \frac{1}{N}$, $\lambda_{M+N}(H) = 0$, and both are simple.
The second smallest eigenvalue of H, denoted by $\lambda_{M+N-1}(H)$, can be obtained from $\left(\lambda - \frac{1}{M}\right)\left(\lambda - \frac{1}{N}\right) = \sigma_2^2$, i.e., $\lambda^2 - \lambda_1\lambda + \sigma_1^2 - \sigma_2^2 = 0$. With $\Delta := \frac{(\sigma_1^2 - \sigma_2^2)}{\lambda_1^2}$, this gives

$$\lambda_{M+N-1}(H) = \frac{1}{2}\lambda_1\left[1 - \sqrt{1-4\Delta}\right] = \frac{1}{2}\lambda_1\frac{4\Delta}{1+\sqrt{1-4\Delta}}. \tag{29}$$

Hence, using the fact that $1 \leqslant 1 + \sqrt{1-4\Delta} \leqslant 2$ we obtain $\frac{(\sigma_1^2-\sigma_2^2)}{\lambda_1} \leqslant \lambda_{M+N-1}(H) \leqslant \frac{2(\sigma_1^2-\sigma_2^2)}{\lambda_1}$. To obtain the bounds for the condition numbers, by (29), we have

$$\frac{(M+N)^2}{2M^2N^2(\sigma_1^2-\sigma_2^2)} = \frac{\lambda_1^2}{2(\sigma_1^2-\sigma_2^2)} \leqslant \kappa(H) \leqslant \frac{\lambda_1^2}{(\sigma_1^2-\sigma_2^2)} = \frac{(M+N)^2}{M^2N^2(\sigma_1^2-\sigma_2^2)}.$$

This gives the bounds in (28). □

The case $M = N$ is of particular interest, and we list the results as a corollary, which follows directly from Theorem 9.

Corollary 10. *Let $M = N$ and Π be a positive coupling matrix with uniform marginals. The eigenvalues of $H(\Pi)$ are*

$$\lambda_j(H) = \frac{1}{N} + \sigma_j, \quad \lambda_{2N+1-j}(H) = \frac{1}{N} - \sigma_j, \quad 1 \leqslant j \leqslant N, \tag{30}$$

where $\{\sigma_j\}$ are the singular values of Π in descending order. In particular, $\sigma_1 = \frac{1}{N}$, $\lambda_1(H) = \frac{2}{N}$ and $\lambda_{2N} = 0$. The condition number of $H(\Pi)$ is bounded by

$$\frac{2}{N^2(\sigma_1^2-\sigma_2^2)} \leqslant \kappa(H) \leqslant \frac{4}{N^2(\sigma_1^2-\sigma_2^2)}. \tag{31}$$

4.3 Condition number of H-matrices in entropy-regularized OT

We establish in this section lower and upper bounds for the condition number of the H-matrix when Π^* is approximated by the Sinkhorn algorithm for uniform marginal distributions.

The Sinkhorn algorithm alternatively re-scales the rows and columns of the coupling matrix to achieve the marginal constraints. They produce a sequence of coupling matrices $\{\Pi^{(l)}\}$ that converges to Π^* entry-wisely, i.e., $\lim_{l\to+\infty}\Pi_{ij}^{(l)} = \Pi_{ij}^*$ [21, 29, 30]. In practice, the Sinkhorn iteration stops when a criterion is met. One stopping criterion is that the marginal distributions of $\Pi^{(l)}$ are entry-wise δ away from the given $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. Thus, an important question is whether the condition number of $H(\Pi^{(l)})$ is controlled.

The next proposition shows that if $\delta = \|\Pi - \Pi^*\|_F$ is small with $\|\cdot\|_F$ denotes the Frobenius norm, the condition number of the H-matrix of $\Pi^{(l)}$ is almost as large as the condition number of $H(\Pi^*)$. The proof is based on Weyl's inequality, and we postpone it to Appendix A.

Proposition 11 (**Condition number of H-matrix in Sinkhorn**). *Let Π^*, with uniform marginal distributions, be the optimal coupling matrix minimizing the EOT distance. Assume that the coupling matrix $\widehat{\Pi}$ is computed by an early-stopped Sinkhorn algorithm that satisfies*

$$\max_{1\leqslant i\leqslant M}\left|\sum_{j=1}^{N}\widehat{\Pi}_{ij} - \frac{1}{M}\right| \leqslant \delta, \quad \max_{1\leqslant j\leqslant N}\left|\sum_{i=1}^{M}\widehat{\Pi}_{ij} - \frac{1}{N}\right| \leqslant \delta, \quad \sum_{i,j}|\widehat{\Pi}_{ij} - \Pi_{ij}^*|^2 \leqslant \delta_2^2. \tag{32}$$

Then, the eigenvalues of $H(\widehat{\Pi})$ satisfies

$$|\lambda_k(H(\widehat{\Pi})) - \lambda_k(H(\Pi^*))| \leqslant \delta + \delta_2, \quad 1 \leqslant k \leqslant N + M. \tag{33}$$

In particular, if $\delta + \delta_2 = t\frac{MN}{M+N}(\sigma_1^2 - \sigma_2^2)$ with $t \in [0, 1)$, where σ_1, σ_2 denoting the largest two singular values of Π^*, the condition number of $H(\widehat{\Pi})$ is bounded by

$$\frac{1-t\Delta}{(2+t)\Delta} \leqslant \kappa(H(\widehat{\Pi})) \leqslant \frac{1+t\Delta}{(1-t)\Delta},$$

where $\Delta = (\frac{MN}{M+N})^2(\sigma_1^2 - \sigma_2^2)$, while $\frac{1}{2\Delta} \leqslant \kappa(H(\Pi^*)) \leqslant \frac{1}{\Delta}$.

The above bounds apply to general H-matrices in entropy-regularized Sinkhorn algorithms. However, these bounds do not show explicit dependence on ϵ, the strength of regularization. In the next section shows, we study the dependence of the condition number on ϵ and N for point-clouds datasets.

4.4 Ill-conditioned H-matrices from data clouds

We investigate in this section the condition number of the H-matrix for a specific example of EOT that matches data clouds with N points. In this simple setting, $M = N$ and both marginals are uniform, so the condition number of the H-matrix is $\kappa = \frac{2}{N\lambda_{2N-1}}$ by By Corollary 10. Therefore, it suffices to investigate the dependence of smallest positive eigenvalue, λ_{2N-1}, on N and ϵ.

We show that the smallest positive eigenvalue of the H-matrix can decay at rate $O(e^{-\frac{1}{\epsilon}})$ for a fixed N and at $O(1/N)$ for a fixed ϵ. These asymptotic results are proved for equally-spaced points on the unit circle in Example 12, and are numerically demonstrated for random data clouds sampled from a uniform distribution.

Example 12 (Equally spaced points on the unit circle)**.** *Consider N equally spaced points on the unit circle $\{\boldsymbol{y}_i = \begin{bmatrix} \cos x_i & \sin x_i \end{bmatrix}\}_{i=0}^{N-1}$, where $x_i = \frac{2\pi i}{N}$. Let $\boldsymbol{\mu} = \frac{1}{N}\mathbb{1}_N$ be the uniform distribution, we are interested in the spectrum of H associate to the symmetric entropic regularized optimal transport loss $OT_\epsilon(C^{\boldsymbol{Y}\to\boldsymbol{Y}}, \boldsymbol{\mu}, \boldsymbol{\mu})$. The coupling matrix is*

$$\Pi^* = \underset{\Pi\in\mathbb{R}_{\geqslant 0}^{N\times N}:\Pi\mathbb{1}_N=\boldsymbol{\mu},\Pi^\top\mathbb{1}_N=\boldsymbol{\mu}}{\arg\min} \sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}\Pi_{ij} + \epsilon KL(\Pi, \boldsymbol{\mu}\otimes\boldsymbol{\mu})$$

with $\epsilon > 0$, where $C_{ij} = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2$. Also, let $H := H(\Pi^*)$ and denote its condition number by $\kappa(H) = \frac{\lambda_1(H)}{\lambda_{2N-1}(H)}$. Then, the following statements hold true.

(a) $\Pi^* = \frac{K}{\lambda_1(K)N}$, where the Gibbs kernel $K \in \mathbb{R}^{N\times N}$ is a symmetric matrix with entries $K_{ij} = \exp\left(-\frac{C_{ij}}{\epsilon}\right)$ and $\lambda_1(K)$ is the largest eigenvalue of K.

(b) The first two singular values of Π^* are $\sigma_1(\Pi^*) = \frac{1}{N}$, $\sigma_2(\Pi^*) = \frac{\lambda_2(K)}{\lambda_1(K)N}$, and the largest and smallest positive eigenvalues of H are $\lambda_1(H) = \frac{2}{N}$, and $\lambda_{2N-1}(H) = \frac{1}{N} - \sigma_2(\Pi^*)$, where $\lambda_2(K)$ is the second largest eigenvalue of K.

(c) The smallest positive eigenvalue of H and the condition number of H satisfies

$$\lim_{\epsilon\to 0^+}\lim_{N\to+\infty}\frac{N\cdot\lambda_{2N-1}(H)}{\epsilon} = \frac{1}{4}, \qquad \lim_{\epsilon\to 0^+}\lim_{N\to+\infty}\epsilon\cdot\kappa(H) = 8, \tag{34}$$

$$\lim_{N\to+\infty}\lim_{\epsilon\to 0^+}\frac{N\lambda_{2N-1}(H)}{r_{N,\epsilon}} = 4\pi^2, \qquad \lim_{N\to+\infty}\lim_{\epsilon\to 0^+} r_{N,\epsilon}\kappa(H) = \frac{1}{2\pi^2}, \tag{35}$$

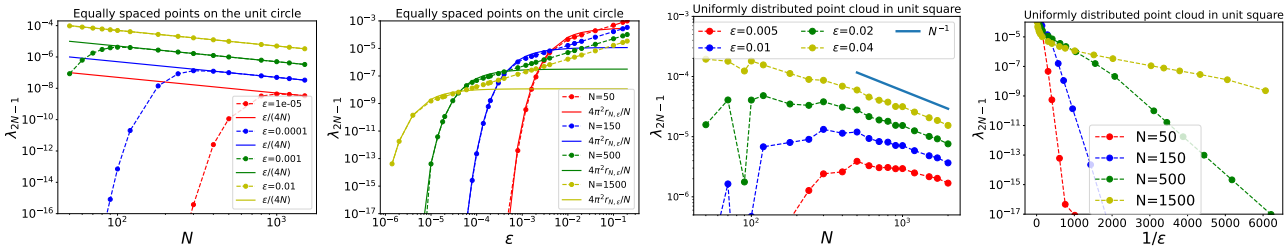
where $r_{N,\epsilon} = N^{-2}\exp\left(-\frac{4\sin^2(\pi/N)}{\epsilon}\right)$.

We postpone the proof to Appendix A.

For a fixed ϵ, when N is large enough, the smallest positive eigenvalue λ_{2N-1} scales as $\frac{\epsilon}{4N}$, which is numerically illustrated in Figure 1(a). Then the condition number scales as $\frac{8}{\epsilon}$. Meanwhile for fixed N, when ϵ is small enough (e.g., when $\epsilon < \frac{4\pi^2}{N^2}$), the smallest positive eigenvalue scales as $\frac{4\pi^2 r_{N,\epsilon}}{N}$, which is numerically illustrated in Figure 1(b). Then the condition number scales as $\frac{1}{2\pi^2 r_{N,\epsilon}}$, which grows exponentially at rate $O(e^{-\frac{1}{\epsilon}})$. For example, for $N = 50$ and $\epsilon = 0.0001$, the condition number of H is larger than 10^{70}; in this case, a truncated SVD for H is crucial in calculating the Hessian of EOT and the gradient of Sinkhorn distance. On the other hand, when $N = 1500$ and $\epsilon = 0.0001$, the condition number of H is only about 8×10^4. In addition, this scale phenomenon is also observed for some random datasets as well.

Next, we further numerically investigate the case of point-clouds datasets that are sampled from the uniform distribution in the unit square $[0,1]^2$. Similarly, we are interested in the spectrum of H associated with $\mathrm{OT}_\epsilon(C^{\boldsymbol{Y}\to\boldsymbol{Y}}, \boldsymbol{\mu}, \boldsymbol{\mu})$. In figure 1(d) show that $\lambda_{2N-1} = O(e^{-\frac{1}{\epsilon}})$ when ϵ is small enough for each fixed N, and Figure 1(c) show $\lambda_{2N-1} = O(\frac{1}{N})$ when N is large for each fixed ϵ. These asymptotic orders are the same as the analytical results proved in Example 12, but the exact limit depends on the distribution of data points, and it is beyond the scope of this study.

In summary, the H-matrix can be severely ill-conditioned with the smallest eigenvalue at the order of $O(e^{-\frac{1}{\epsilon}})$ when ϵ is small, or $O(\frac{1}{N})$ when N is large. Thus, when solving a linear system with H, it is important to properly regularize the ill-posed inverse problem.



(a) (b) (c) (d)

Figure 1: Decay of the smallest positive eigenvalue λ_{2N-1} in N and ϵ. Equally spaced points on the unique circle: **(a)** $\lambda_{2N-1} \approx \frac{\epsilon}{4N}$ when $N > \frac{2\pi}{\sqrt{\epsilon}}$; **(b)** $\lambda_{2N-1} \approx 4\pi^2 r_{N,\epsilon}$ when $\epsilon < \frac{4\pi^2}{N^2}$. Uniformly distributed point cloud in unit square: **(c)** $\lambda_{2N-1} = O(\frac{1}{N})$ when N is large; **(d)** $\lambda_{2N-1} = O(e^{-\frac{1}{\epsilon}})$ when ϵ is small.

5 Hessian computation: runtime, accuracy, and success rate

Our analytical approach enables efficient and accurate computation of the Hessian matrix. Here we compare it with the current two state-of-the-art approaches suggested by *OTT*: *unroll* and *implicit differentiation*. The details on these approaches are discussed in Section 3.1.

We use the point-cloud datasets sampled from the uniform distribution in unit square again. The task is to calculate the Hessian tensor $\mathcal{T}$ of $\mathrm{OT}_\epsilon(C^{\boldsymbol{Y}\to\boldsymbol{Y}}, \boldsymbol{\mu}, \boldsymbol{\mu})$ respect to the source data $\boldsymbol{Y}$, where $\boldsymbol{\mu} = \frac{1}{N}\mathbb{1}_N$. By proposition 4, the Hessian satisfies the marginal identity, i.e., $\sum_{k=1}^{M} \mathcal{T}_{k\cdot s\cdot} = 2\mu_s \mathbb{I}_d$. We evaluate the accuracy of the computed Hessian by the marginal error:

$$\text{error} = \sum_{t\neq l}(\sum_k \mathcal{T}_{ktsl})^2 + \sum_t(\sum_k \mathcal{T}_{ktst} - 2\mu_s)^2. \tag{36}$$

All simulations are performed on a single Nvidia A100 GPU using double-precision. The threshold α in truncated SVD is set to $\alpha = 10^{-10}$.

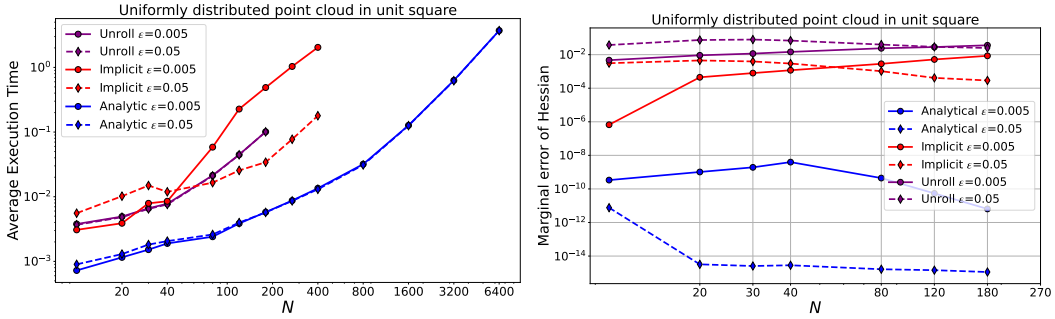


Figure 2: Comparison of runtime (in seconds) and marginal error for Hessian computing $\frac{d^2\mathrm{OT}_\epsilon(C^{\boldsymbol{Y}\to\boldsymbol{Y}}, \boldsymbol{\mu}, \boldsymbol{\mu})}{d\boldsymbol{Y}^2}$ among three approaches: unroll, implicit differentiation and analytic expression with regularization (ours).

Runtime. Figure 2(a) shows the average execution time in 10 independent tests for the three approaches with $N \in [10, 6400]$ and $\epsilon \in \{0.005, 0.05\}$, corresponding to low and median regularization regimes. The unrolling and implicit differentiation approaches fail in all 10 tests due to insufficient memory when $N > 180$ and $N > 400$, respectively. However, our analytical approach remains effective for all N, even beyond $N = 5000$. Additionally, when all three approaches work, our analytical approach is faster by at least one order of magnitude.

Accuracy. Figure 2(b) shows the average marginal error of the Hessian computed by the three approaches in 100 independent tests. Here we consider $\epsilon \in \{0.005, 0.05\}$ and $N \in [10, 180]$ where all three approaches work. Both implicit differentiation and unrolling approaches perform poorly across all parameter settings. In contrast, our analytical approach is significantly more accurate by 3-8 orders of magnitude.

Success rate. Table 1 further highlights the reliability of our analytical approach and the importance of regularization by reporting the success rate in 100 independent tests. A test is considered successful if the marginal error of the Hessian (36) is less than 0.1. In the most singular parameter setting, $N = 10$ and $\epsilon = 0.005$ (as discussed in section 4.4), the implicit differentiation approach fails 97% of the tests due to numerical instability. Importantly, if we do not regularize the problem using truncated SVD and instead apply the least square solver directly to solve the linear system, the analytical approach results in large errors ranging from 10^{-7} to 10^2 in 15% of the tests. Therefore, proper regularization is crucial when the problem is ill-posed.

To conclude, our analytical approach with regularization enables efficient and accurate computation of the

Method	Unroll	Implicit	Analytical(no reg)	Analytical(with reg)
$N = 10$	0.78	0.03	0.85	1.00
$N = 20$	0.68	0.18	0.99	1.00
$N = 120$	0.00	1.00	1.00	1.00
$N = 1600$	0.00	0.00	1.00	1.00

Table 1: Success rates of the three approaches for $N \in \{10, 20, 120, 1600\}$ and $\epsilon = 0.005$. A test is called successful if the marginal error of Hessian (36) is less than 0.1.

Hessian of EOT, significantly outperforming other current state-of-the-art approaches by a large margin in terms of runtime, accuracy and success rate.

6 Applications to Shuffled Regression

In this section, we apply our proposed algorithms to solve the shuffled regression problem introduced earlier. It is formulated as the multivariate regression model $\boldsymbol{y}^* = \boldsymbol{x}\theta + \xi$, where $\boldsymbol{x} \in \mathbb{R}^D, \boldsymbol{y}^* \in \mathbb{R}^d, \theta \in \mathbb{R}^{D\times d}$ and ξ is the Gaussian noise independent of x. The correspondence between $(\boldsymbol{X}, \boldsymbol{Y}^*)$ is missing. Our goal is to estimate the optimal θ^* using EOT distance as the loss function in the (1). This approach generalizes to unbalanced datasets without requiring $\boldsymbol{X}$ and $\boldsymbol{Y}^*$ to have the same number of rows. The gradient and the Hessian of the EOT distance with respect to the parameters θ are simplified as

$$\begin{aligned}
&\frac{d\mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\theta} = \boldsymbol{X}^\top \frac{d\mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\boldsymbol{Y}}, \\
&\left(\frac{d^2\mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\theta^2}\right)_{mtnl} = \sum_{k=1}^{M}\sum_{s=1}^{M} \boldsymbol{X}_{sm}\mathcal{T}_{ktsl}\boldsymbol{X}_{kn}
\end{aligned} \tag{37}$$

for $t, l = 1, \ldots, d$ and $m, n = 1, \ldots, D$. The EOT distance is generally non-convex with respect to θ, so optimization may not converge to the optimal θ^*. Our focus is on the convergence speed to a local minimum. First-order methods may converge to a local minimum but require many iterations due to the complicated landscape of the loss function. To accelerate optimization, we propose a two-stage approach. First, we use stochastic gradient descent (SGD) with a random subset of $\boldsymbol{X}$ and the full batch of $\boldsymbol{Y}^*$ to quickly approach the local minimum. Then, we switch to a relaxed Newton's method, using the updated parameter $\hat{\theta}$ as the initial condition. The relaxed-Newton's method uses step-size $\gamma < 1$. In practise, we switch from SGD to relaxed-Newton when the computed Hessian $\frac{d^2\mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\theta^2}$ is positive definite. The algorithm is summarized in Algorithm 2.

6.1 Shuffled Regression with Gaussian Mixtures

We first generate $N = 500$ data points $\boldsymbol{X} \in \mathbb{R}^5$ from a Gaussian mixture distribution with three clusters whose standard deviation is $[0.3, 0.05, 0.6]$. The parameter $\theta^* \in \mathbb{R}^{5\times 2}$ is generated with components $\theta^*_{mt} \sim \mathcal{N}(0, 1)$, and the Gaussian noise $\boldsymbol{\xi} \in \mathbb{R}^2$ follows $\boldsymbol{\xi} \sim \mathcal{N}(\mathbb{0}, 0.04\mathbb{1}_2)$. We then compute $\boldsymbol{y}^*_i = \boldsymbol{x}_i\theta^* + \boldsymbol{\xi}_i$, randomly and completely permute the order of $\boldsymbol{y}^*_i$, removing $\boldsymbol{X}$-to-$\boldsymbol{Y}^*$ correspondence.

Starting with an random initial condition $\theta^{(0)}$ from the standard normal distribution, the target data $\boldsymbol{Y}^*$ and the initial data $\boldsymbol{Y}(\theta^{(0)})$ are shown in Figure 3(a). We use the two-stage algorithm described in Algorithm 2. In the first stage, we perform 10 iterations of SGD on 100 random source data points with a learning rate of 0.001. In the second stage, we use a relaxed Newton's method with a learning rate of 0.5. We compare this to a gradient descent (GD) method with a learning rate of 0.001.

Figure 3(b) shows that both methods correctly map the data $\boldsymbol{X}$ to the target data $\boldsymbol{Y}^*$. Figure 3(c-d) shows that both methods converge to the optimal θ^*, but the relaxed Newton's method is faster and more accurate. The relaxed Newton's method converges in 12 iterations with a runtime of 2.35 seconds, while GD takes 2000 iterations and 64.77 seconds, which is 27 times longer. Additionally, the relaxed Newton's method achieves nearly one order of magnitude better accuracy in terms of the L_2 error $\|\theta - \theta^*\|_2$.

Further analysis shows that the eigenvalues of Hessian with respect to θ^* range from 10^{-2} to 10^2, indicating that the optimal parameter lies in a long, narrow, flat valley, causing the gradient descent method to converge slowly.

6.2 3D Point Cloud Registration

In this section, we extend our method to 3D point clouds registration, a critical task in computer vision. The goal is to find a spatial transformation that aligns two 3D data clouds without knowing the correspondence, known as simultaneous pose and correspondence registration [18, 25].

Input: Data $\boldsymbol{X}$, target data $\boldsymbol{Y}^*$, entropy regularization strength ϵ, truncated SVD threshold α; initial guess of $\theta^{(0)}$, SGD learning rate r_s, mini batch size n_s, maximum epochs T; Relaxed Newton learning rate r_n.
Output: Estimated optimal θ^*, regularized optimal transport loss $\mathrm{OT}_\epsilon(C_{\theta^*}, \boldsymbol{\mu}, \boldsymbol{\nu})$.

Set $\tilde{\boldsymbol{\mu}} \leftarrow \frac{1}{n_s}\mathbb{1}_{n_s}$, $\boldsymbol{\mu} \leftarrow \frac{1}{N}\mathbb{1}_N$ and $\boldsymbol{\nu} \leftarrow \frac{1}{N}\mathbb{1}_N$.
Stage 1 SGD :
for $t = 0, \ldots, T-1$ **do**,
  Randomly sample n_s rows of $\boldsymbol{X}$, denote as $\tilde{\boldsymbol{X}}$.
  Compute $\tilde{\boldsymbol{Y}} \leftarrow \tilde{\boldsymbol{X}} \cdot \theta^{(t)}$ and cost matrix $C_{ij}^{\tilde{\boldsymbol{Y}} \to \boldsymbol{Y}^*} \leftarrow \|\tilde{\boldsymbol{y}}_i - \boldsymbol{y}_j^*\|_2^2$.
  $\theta^{(t+1)} \leftarrow \theta^{(t)} - r_s \tilde{\boldsymbol{X}}^\top \frac{d\mathrm{OT}_\epsilon(C^{\tilde{\boldsymbol{Y}} \to \boldsymbol{Y}^*}, \tilde{\boldsymbol{\mu}}, \boldsymbol{\nu})}{d\boldsymbol{Y}}$
  if $\frac{d^2\mathrm{OT}_\epsilon(C_\theta, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\theta^2}|_{\theta=\theta^{(t+1)}}$ is positive definite **then**
    stop Stage 1 with the current $\hat{\theta} \leftarrow \theta^{(t+1)}$.
Stage 2 Relaxed Newton's method :
Set $\theta^{(0)} \leftarrow \hat{\theta}$.
for $t = 0, \ldots, T-1$ **do**
  Compute $\boldsymbol{Y} \leftarrow \boldsymbol{X} \cdot \theta^{(t)}$ and cost matrix $C_{ij} \leftarrow \|\boldsymbol{y}_i - \boldsymbol{y}_j^*\|_2^2$
  $\theta^{(t+1)} \leftarrow \theta^{(t)} - r_n \left(\frac{d^2\mathrm{OT}_\epsilon(C_{\theta^{(t)}}, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\theta^2}\right)^{-1} \left(\tilde{\boldsymbol{X}}^\top \frac{d\mathrm{OT}_\epsilon(C_{\theta^{(t)}}, \boldsymbol{\mu}, \boldsymbol{\nu})}{d\boldsymbol{Y}}\right)$
  if $\mathrm{OT}_\epsilon(C_{\theta^{(t+1)}}, \boldsymbol{\mu}, \boldsymbol{\nu})$ doesn't improve **then**
    Quit Stage 2 with $\theta^* \leftarrow \theta^{(t+1)}$.

Algorithm 2: Two-stage algorithm to estimate optimal θ^* of EOT distance (4).

Using the MobilNet10 dataset [25], we create a study room with a chair (500 points), a desk (1500 points), and a sofa (1500 points), denoted as $\boldsymbol{X}$. We apply a linear transformation including random rotation and scaling, and add Gaussian noise: $\boldsymbol{Y}^* = \boldsymbol{X}\theta^* + \boldsymbol{\xi}$ with $\boldsymbol{\xi} \sim \mathcal{N}(0, 4 \times 10^{-4}\mathbb{I}_3)$. The rows of $\boldsymbol{Y}^*$ are randomly permuted to remove correspondence.

We use the algorithm from Algorithm 2. The initial parameter $\theta^{(0)}$ is a standard Gaussian perturbation of the optimal parameter θ^*. In the first stage, we perform 5 iterations of SGD on 500 random data points with a learning rate of 0.1. In the second stage, we use a relaxed Newton's method with a learning rate of 0.5. For comparison, the GD method uses a learning rate of 0.1.

As shown in Figure 4, both methods converge, but at different speeds. The relaxed Newton's method converges in 9 iterations with a runtime of 17.20 seconds, while the GD-only method takes 922 iterations (runtime 314.55 seconds) to reach a comparable loss. Additionally, the relaxed Newton's method achieves about 0.6 orders of magnitude improvement in accuracy in terms of L_2 error.

7 Conclusion

In this work, we computed first-order and second-order derivatives for the parameterized regularized optimal transport (OT) distance. Specifically, we derived explicit analytical expressions for the gradient of the Sinkhorn distance and the Hessian of the entropy-regularized OT (EOT) distance with respect to the source data $\boldsymbol{Y}$.

To address the numerical instability and high memory consumption typically associated with Hessian computation in large-scale, multi-dimensional problems, we developed a fast, stable, and memory-efficient algorithm using spectral analysis of the ill-posed linear system. Our algorithm demonstrated significant improvements in both efficiency and accuracy on various benchmark datasets.

These results highlight the potential of our proposed algorithm to enhance the performance and reliability of optimization tasks in complex, high-dimensional spaces, particularly in regression without correspondence.

Future work may explore further refinements of our stabilization thresholds by studying the limiting behavior of condition numbers on other random datasets, as well as the computation of robust second-order differentiation for more general regularized and constrained optimal transport problems.

Data Availability

The software package implementing the proposed algorithms can be found on:
https://github.com/yexf308/OTT-Hessian.

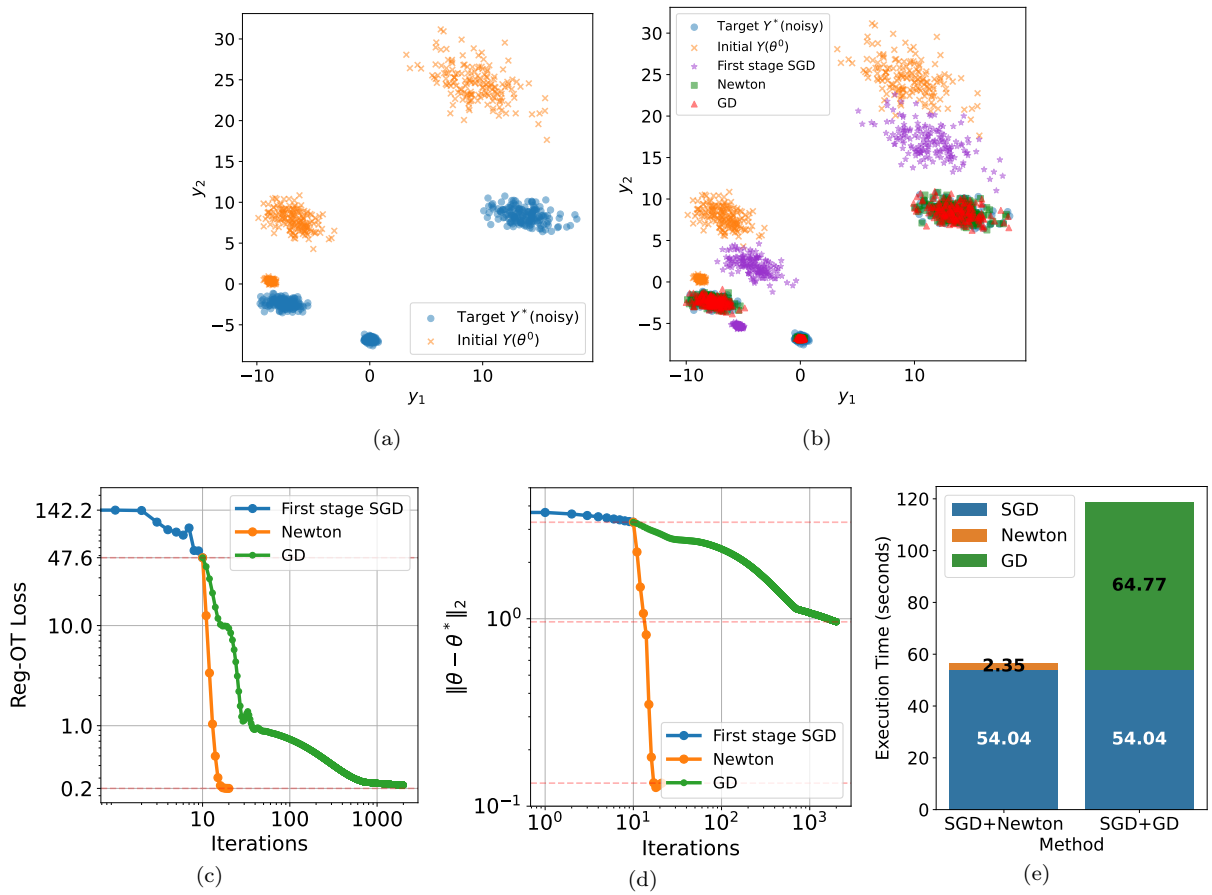


Figure 3: Shuffled Regression with Gaussian Mixtures.

Acknowledgements

X. Li is grateful for partial support by the NSF Award DMS-1847770 and the 2023 UNC Charlotte faculty research grant. F. Lu is grateful for partial support by NSF DMS-2238486. M. Tao is grateful for partial support by the NSF Award DMS-1847802, the Cullen-Peck Scholar Award and the Emory-GT AI Humanity Award. F. Ye is grateful for partial support by Simons Foundation Award MPS-TSM-00002666.

A Proofs for spectral analysis

The proof of Proposition 11 is based on an application of Weyl's inequality to study the eigenvalues of the H-matrix under perturbation.

Lemma 13 (Eigenvalues under perturbation)**.** *Let Π, Π^* be two positive coupling matrices with $A = \Pi - \Pi^*$ satisfying*

$$\max_i |\sum_j A_{ij}| \leqslant \delta_1, \quad \max_j |\sum_i A_{ij}| \leqslant \delta_1, \quad \sum_{i,j} A_{ij}^2 \leqslant \delta_2^2. \tag{38}$$

Then, the eigenvalues of their H-matrices are close:

$$|\lambda_k(H(\Pi)) - \lambda_k(H(\Pi^*))| \leqslant \delta_1 + \delta_2, \quad 1 \leqslant k \leqslant N + M.$$

Proof of Lemma 13. Note that we can write the

$$H(\Pi) - H(\Pi^*) = \begin{bmatrix} \mathrm{diag}(\sum_{j=1}^N A_{i,j}) & A \\ A^\top & \mathrm{diag}(\sum_{i=1}^N A_{i,j}) \end{bmatrix} =: E. \tag{39}$$

By Weyl's inequality, we have $|\lambda_k(H^*) - \lambda_k(H)| \leqslant \|E\|_{op}$. Thus, it suffices to estimate $\|E\|_{op}$. Note that first that using $|A_{i,j}| \leqslant 1$ and (38), we have $\sum_i |\sum_j A_{i,j} u_i|^2 = \sum_i |\sum_j A_{i,j}|^2 u_i^2 \leqslant \sum_i |\sum_j A_{i,j}| u_i^2 \leqslant \delta_1^2 \|u\|^2$, and similarly, $\sum_j |\sum_i A_{i,j} v_j|^2 \leqslant \delta_1^2 \|v\|^2$; also, $\|Av\|^2 = \sum_{i=1} |\sum_j A_{i,j} v_j|^2 \leqslant \sum_{i=1} [\sum_j A_{i,j}^2 \sum_j |v_j|^2] \leqslant \delta_2^2 \|v\|^2$, and similarly, $\|A^\top u\|^2 \leqslant \delta_2^2 \|u\|^2$.

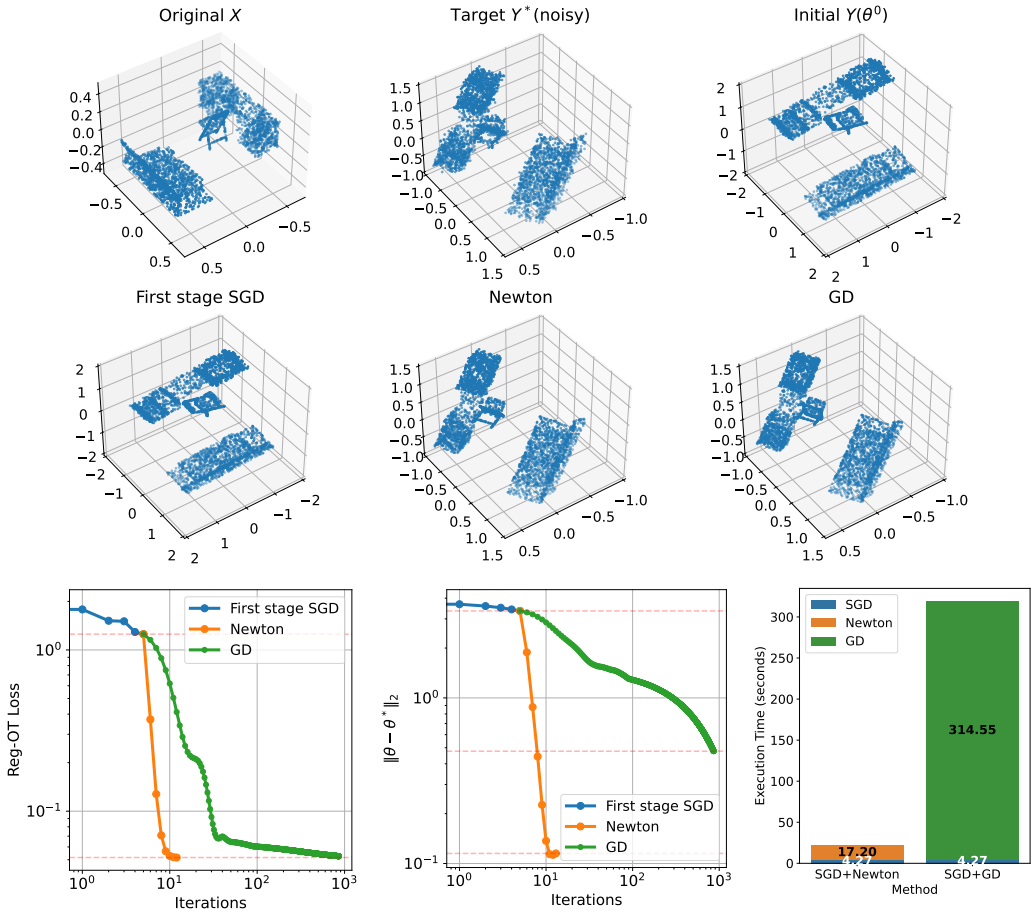


Figure 4: 3D Point Cloud Registration.

Using these four bounds, we have $\|E\|_{op}^2 = \sup_{\boldsymbol{u}\in\mathbb{R}^M,\boldsymbol{v}\in\mathbb{R}^N,\|\boldsymbol{u}\|^2+\|\boldsymbol{v}\|^2=1} \|E\begin{bmatrix}\boldsymbol{u}\\ \boldsymbol{v}\end{bmatrix}\|^2$, then $\|E\|_{op}^2 = \sum_i |\sum_j A_{i,j}u_i|^2 + \|Av\|^2 + \sum_j |\sum_i A_{i,j}v_j|^2 + \|A^\top u\|^2 \leqslant \delta_1^2+\delta_2^2 \leqslant \delta_1+\delta_2$. Combining with Weyl's inequality, we conclude the proof. □

Proof of Proposition 11. The bound for the eigenvalues in (33) follows from (32) and Lemma 13. To prove the bounds for the condition number, recall that in the proof of Theorem 9, we have shown that $\lambda_1 := \lambda_1(H(\Pi^*)) = \frac{1}{N}+\frac{1}{M}$ and $\lambda_{N+M-1} := \lambda_{N+M-1}(H(\Pi^*)) = \frac{1}{2}\lambda_1\left[1-\sqrt{1-4\Delta}\right] \in [\lambda_1\Delta, 2\lambda_1\Delta]$ with $\Delta = \lambda_1^{-2}(\sigma_1^2-\sigma_2^2)$. For $\delta+\delta_2 = t\lambda_1\Delta$ with $t\in[0,1)$, Eq.(33) implies that

$$\begin{aligned}(1-t)\lambda_1\Delta \leqslant \lambda_{N+M-1}-\delta-\delta_2 \leqslant \lambda_{N+M-1}(H(\Pi)) &\leqslant \lambda_{N+M-1}+\delta+\delta_2\\ &\leqslant (2+t)\lambda_1\Delta\\ \lambda_1(1-t\Delta) \leqslant \lambda_1-(\delta+\delta_2) \leqslant \lambda_1(H(\Pi)) &\leqslant \lambda_1+\delta+\delta_2 \leqslant \lambda_1(1+t\Delta).\end{aligned}$$

Consequently, we obtain the bounds by noting that

$$\frac{1+t\Delta}{(2+t)\Delta} \leqslant \frac{\lambda_1-(\delta+\delta_2)}{\lambda_{N+M-1}+(\delta+\delta_2)} \leqslant \kappa(H(\Pi)) \leqslant \frac{\lambda_1+(\delta+\delta_2)}{\lambda_{N+M-1}-(\delta+\delta_2)} \leqslant \frac{1+t\Delta}{(1-t)\Delta}.$$

□

Proof of Example 12. **Part (a)**: Note that the cost matrix C is $C_{ij} = \|y_i-y_j\|_2^2 = 4\sin^2\left(x_{|j-i|}/2\right)$ and the Gibbs kernel $K_{ij} = \exp\left(-\frac{4\sin^2(|i-j|\pi/N)}{\epsilon}\right)$ is circulant, whose rows and columns sum is $\lambda_1(K) = \sum_{j=0}^{N-1}\exp\left(-\frac{4\sin^2(j\pi/N)}{\epsilon}\right)$. Then the matrix $\Pi^* = \frac{K}{\lambda_1(K)N}$ satisfies the uniform marginal constraints on Π^*, hence, it is the optimal coupling matrix due to uniqueness of the solution of the contraint optimization (4).

Part (b) follows directly from Corollary 10.

Part (c), We first compute the largest two eigenvalues of K. Recall that the matrix K is symmetric and positive-definite, so its singular values are the same as its eigenvalues. Since K is circulant, the first two eigenvalues of K are (see e.g., [16]), we have

$$\lambda_1(K) = \sum_{j=0}^{N-1}\exp\left(-\frac{4\sin^2(\frac{j\pi}{N})}{\epsilon}\right), \lambda_2(K) = \sum_{j=0}^{N-1}\exp\left(-\frac{4\sin^2(\frac{j\pi}{N})}{\epsilon}\right)\cos\left(\frac{2j\pi}{N}\right)$$

Meanwhile, combining Part (a) and Part (b), we have $\lambda_{2N-1}(H) = \frac{1}{N}(1-\frac{\lambda_2(K)}{\lambda_1(K)})$. Thus, to study the limits, we first study the limit of $\lambda_1(K)$ and $\lambda_2(K)$.

As $N\to+\infty$, the Riemann summations in λ_1 and λ_2 approaches the integrals

$$\begin{aligned}\lim_{N\to+\infty}\frac{\lambda_1(K)}{N} &= \frac{1}{\pi}\int_0^\pi \exp\left(-\frac{4\sin^2(x)}{\epsilon}\right)dx = \exp\left(-\frac{2}{\epsilon}\right)I_0\left(\frac{2}{\epsilon}\right)\\ \lim_{N\to+\infty}\frac{\lambda_2(K)}{N} &= \frac{1}{\pi}\int_0^\pi \exp\left(-\frac{4\sin^2(x)}{\epsilon}\right)\cos(2x)dx = \exp\left(-\frac{2}{\epsilon}\right)I_1\left(\frac{2}{\epsilon}\right),\end{aligned}$$

where $I_1(x)$ and $I_2(x)$ are the modified Bessel functions of first kind. Then, the limit of the second smallest eigenvalue of H is

$$\lim_{N\to+\infty}(N\cdot\lambda_{2N-1}(H)) = 1-\lim_{N\to+\infty}\frac{\lambda_2(K)}{\lambda_1(K)} = 1-\frac{I_1(2/\epsilon)}{I_0(2/\epsilon)}.$$

When ϵ is small, we can expand $I_1(x)$ and $I_0(x)$ around $x=+\infty$,

$$\lim_{N\to+\infty}(N\cdot\lambda_{2N-1}(H)) = 1-\frac{\frac{(\epsilon/2)^{1/2}}{\sqrt{2\pi}}-\frac{3(\epsilon/2)^{3/2}}{8\sqrt{2\pi}}+O(\epsilon/2)^{5/2}}{\frac{(\epsilon/2)^{1/2}}{\sqrt{2\pi}}+\frac{(\epsilon/2)^{3/2}}{8\sqrt{2\pi}}+O(\epsilon/2)^{5/2}} = \frac{\epsilon}{4}+O(\epsilon^2).$$

Then, we have $\lim_{\epsilon\to0^+}\lim_{N\to+\infty}\frac{N\cdot\lambda_{2N-1}(H)}{\epsilon} = \frac{1}{4}$, and $\lim_{\epsilon\to0^+}\lim_{N\to+\infty}\epsilon\frac{\lambda_1(H)}{\lambda_{2N-1}(H)} = 8$, which gives (34).

To prove (35), note first that when N is fixed and $\epsilon\to0^+$, $\lambda_1(K)$ and $\lambda_2(K)$ are approximated by the three largest terms,

$$\lambda_1(K) = 1+2\exp(-\frac{4}{\epsilon}\sin^2(\frac{\pi}{N}))+O_\epsilon, \lambda_2(K) = 1+2\exp(-\frac{4}{\epsilon}\sin^2(\frac{\pi}{N}))\cos(\frac{2\pi}{N})+O_\epsilon$$

where $O_\epsilon := O\left(\exp\left(-\frac{4\sin^2(2\pi/N)}{\epsilon}\right)\right)$. Consequently,

$$\exp\left(\frac{4}{\epsilon}\sin^2(\frac{\pi}{N})\right) N\lambda_{2N-1}(H) = \frac{2(1-\cos(\frac{2\pi}{N}))+O_\epsilon}{1+2\exp\left(-\frac{4}{\epsilon}\sin^2(\frac{\pi}{N})\right)+O_\epsilon}.$$

Taking the limits with $\lim_{N\to+\infty}\lim_{\epsilon\to 0^+}$, we obtain

$$\lim_{N\to+\infty}\lim_{\epsilon\to 0^+} N^3\exp\left(\frac{4\sin^2(\pi/N)}{\epsilon}\right)\lambda_{2N-1}(H) = 4\pi^2.$$

□

References

[1] Abubakar Abid, Ada Poon, and James Zou. Linear regression with shuffled labels. *arXiv preprint arXiv:1705.01342*, 2017.

[2] Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems*, 30, 2017.

[3] Qingci An, Yannis Kevrekidis, Fei Lu, and Mauro Maggioni. Unsupervised learning of observation functions in state space models by nonparametric moment methods. *Foundations of Data Science*, 5(3), 2023.

[4] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK users' guide*. SIAM, 1999.

[5] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[6] Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in neural information processing systems*, 35:5230–5242, 2022.

[7] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[8] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014.

[9] Marco Cuturi, Laetitia Meng-Papaxanthos, Yingtao Tian, Charlotte Bunne, Geoff Davis, and Olivier Teboul. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*, 2022.

[10] Marco Cuturi, Olivier Teboul, Jonathan Niles-Weed, and Jean-Philippe Vert. Supervised quantile normalization for low rank matrix factorization. In *International Conference on Machine Learning*, pages 2269–2279. PMLR, 2020.

[11] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, Florian Bernard, and Daniel Cremers. A unified framework for implicit sinkhorn differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 509–518, 2022.

[12] Golnoosh Elhami, Adam Scholefield, Benjamin Bejar Haro, and Martin Vetterli. Unlabeled sensing: Reconstruction algorithm and theoretical guarantees. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4566–4570. Ieee, 2017.

[13] Jean Feydy. Geometric data analysis, beyond convolutions. *Applied Mathematics*, 2020.

[14] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR, 2019.

[15] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018.

[16] Robert M Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006.

[17] Daniel J Hsu, Kevin Shi, and Xiaorui Sun. Linear regression without correspondence. *Advances in Neural Information Processing Systems*, 30, 2017.

[18] Siddharth Katageri, Srinjay Sarkar, and Charu Sharma. Metric learning for 3d point clouds using optimal transport. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 561–569, 2024.

[19] Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita. Generalized shuffled linear regression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6474–6483, 2021.

[20] Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of sinkhorn approximation for learning with wasserstein distance. *Advances in Neural Information Processing Systems*, 31, 2018.

[21] Albert W Marshall and Ingram Olkin. Scaling of matrices to achieve specified row and column sums. *Numerische Mathematik*, 12:83–90, 1968.

[22] Amin Nejatbakhsh and Erdem Varol. Robust approximate linear regression without correspondence. *arXiv preprint arXiv:1906.00273*, 2019.

[23] Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Linear regression with shuffled data: Statistical and computational limits of permutation recovery. *IEEE Transactions on Information Theory*, 64(5):3286–3300, 2017.

[24] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning. arXiv:1803.00567*, 11(5-6):355–607, 2019.

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[26] Jason Michael Rader, Terry Lyons, and Patrick Kidger. Lineax: unified linear solves and linear least-squares in jax and equinox. In *NeurIPS 2023 AI for Science Workshop*, 2023.

[27] Ali Rahimi and Ben Recht. Unsupervised regression with applications to nonlinear system identification. In *Advances in Neural Information Processing Systems*, pages 1113–1120, 2007.

[28] Zebang Shen, Zhenfu Wang, Alejandro Ribeiro, and Hamed Hassani. Sinkhorn natural gradient for generative models. *Advances in Neural Information Processing Systems*, 33:1646–1656, 2020.

[29] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

[30] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[31] Manolis Tsakiris and Liangzu Peng. Homomorphic sensing. In *International Conference on Machine Learning*, pages 6335–6344. PMLR, 2019.

[32] Manolis C Tsakiris, Liangzu Peng, Aldo Conca, Laurent Kneip, Yuanming Shi, Hayoung Choi, et al. An algebraic-geometric approach to shuffled linear regression. *arXiv preprint arXiv:1810.05440*, 2018.

[33] Jayakrishnan Unnikrishnan, Saeid Haghighatshoar, and Martin Vetterli. Unlabeled sensing: Solving a linear system with unordered measurements. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 786–793. IEEE, 2015.

[34] Jayakrishnan Unnikrishnan, Saeid Haghighatshoar, and Martin Vetterli. Unlabeled sensing with random linear measurements. *IEEE Transactions on Information Theory*, 64(5):3237–3253, 2018.

[35] Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. *Advances in Neural Information Processing Systems*, 33:20520–20531, 2020.

[36] Yujia Xie, Yixiu Mao, Simiao Zuo, Hongteng Xu, Xiaojing Ye, Tuo Zhao, and Hongyuan Zha. A hypergradient approach to robust regression without correspondence. In *International Conference on Learning Representations*, 2021.

[37] Hang Zhang and Ping Li. Optimal estimator for unlabeled linear regression. In *International Conference on Machine Learning*, pages 11153–11162. PMLR, 2020.