Wiley Interdisciplinary Reviews: Computational Statistics

WILEY

WIREs COMPUTATIONAL STATISTICS

ADVANCED REVIEW

Sparsification Techniques for Large-Scale Optimal Transport Problems

Xiaxue Ouyang[1] | Hao Zheng[1] | Haoxian Liang[2] | Jingyi Zhang[3] | Yixuan Qiu[4,5] | Cheng Meng[1] | Mengyu Li[6]

[1]Institute of Statistics and Big Data, Renmin University of China, Beijing, China | [2]School of Mathematical Sciences, Beijing Normal University, Beijing, China | [3]School of Science, Beijing University of Posts and Telecommunications, Beijing, China | [4]School of Statistics and Data Science, Shanghai University of Finance and Economics, Shanghai, China | [5]Institute of Big Data Research, Shanghai University of Finance and Economics, Shanghai, China | [6]Department of Statistics and Data Science, Tsinghua University, Beijing, China

Correspondence: Mengyu Li (mengyuli@tsinghua.edu.cn)

Received: 28 September 2025 | **Revised:** 9 December 2025 | **Accepted:** 19 December 2025

Commissioning Editor: James E. Gentle | **Review Editor:** David W. Scott

Keywords: entropic regularization | matrix approximation | Sinkhorn-scaling algorithm | subsampling

ABSTRACT

Optimal transport (OT) methods and their variants have become increasingly prominent tools in computer science and machine learning, owing to their appealing geometric properties and powerful potency. Despite broad applications, OT methods suffer from prohibitively high computational cost, limiting the scalability even for moderately sized datasets. To address this challenge, regularized OT formulations and the corresponding Sinkhorn algorithm have emerged as standard alternatives to improve efficiency. However, these methods still face the high per-iteration cost and slow convergence rate drawbacks. Sparsification techniques have emerged as an effective and practically valuable class of methods for mitigating these computational bottlenecks by leveraging inherent or induced sparsity in the matrices involved in OT optimization. Broadly, sparsification methods can be grouped into two main categories: (1) kernel-based sparsification building on the primal regularized OT formulation, and (2) Hessian-based sparsification, derived from the dual formulation. In this survey, we provide an extensive and comprehensive review of sparsification techniques developed for OT problems, highlighting their underlying motivations, algorithmic distinctions, and theoretical guarantees.

This article is categorized under:

Statistical and Graphical Methods of Data Analysis > Sampling

Algorithms and Computational Methods > Computational Complexity

1 | Introduction

In the 18th century, the French mathematician Gaspard Monge formulated a fundamental transportation problem involving a pile of sand (Monge 1781): given a distribution of sand (referred to as the déblai) and a collection of holes or pits (the remblai) to be filled, the goal was to move the sand in such a way that the piles exactly fill the holes. There are many possible ways to transport the sand, each associated with a global transportation cost, which aggregates the local effort to move each individual grain from its original location to a destination. The central question is how to find an efficient transport plan that minimizes the total cost of moving the sand from the source to the target. To generalize the problem, both the sand and the holes can be modeled as mass distributions over a spatial domain, mathematically represented by two probability measures, denoted by μ and ν, respectively. The central objective is to determine the most efficient way to transport μ

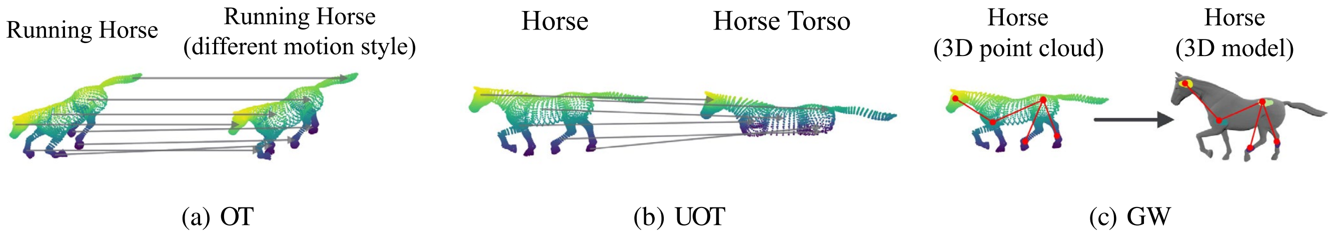
https://doi.org/10.1002/wics.70056

to ν. This leads to the optimal transport (OT) problem, a powerful and widely used framework for comparing probability distributions (Villani 2008).

More specifically, the OT problem can be interpreted from a resource allocation perspective (Peyré and Cuturi 2019; Zhang et al. 2021; Zhang, Ma, et al. 2023). Consider a scenario in which an operator manages n warehouses and m factories. Each factory has a specified demand for raw materials stored in the warehouses. It is assumed that the total supply exactly matches the total demand, and that all available resources must be transported from the warehouses to fulfill the factories' needs. The objective of the OT problem in this context is to determine a transportation plan that allocates materials from warehouses to factories in a way that minimizes the total logistic cost, while fully satisfying the demand at each factory. The cost is typically modeled as the total amount of material transported multiplied by the distance over which it is shipped, aggregated over all warehouse-factory pairs.

Due to the natural formulation across diverse contexts and the ability to capture the underlying geometry of data, OT methods have shown remarkable adaptability in modern data science applications. Numerous tasks in statistics and machine learning can be fundamentally reduced to comparing probability distributions. For example, generative adversarial networks (GANs) seek to align the distribution of generated samples with that of real data (Goodfellow et al. 2014; Arjovsky et al. 2017; Gulrajani et al. 2017). In semantic matching, word embeddings can be interpreted as distributions, and OT provides a means to quantify their structural divergence (Werner and Laber 2020; Yurochkin et al. 2019). In domain adaptation, the goal is to adapt a model trained on a source domain to perform effectively on a different target domain, often requiring alignment between their respective data distributions (Courty et al. 2016; Muzellec and Cuturi 2019). Thus, OT methods have found widespread applications across computer science and machine learning, including computer vision (e.g., image registration, style transfer) (Petric Maretic et al. 2019; Wang et al. 2021; Luo, Xu, and Carin 2022; Vincent-Cuaz et al. 2022; Wang et al. 2023, 2025), generative modeling (such as GANs and variational models) (Tolstikhin et al. 2018; Deshpande et al. 2019; Lei et al. 2019), natural language processing (e.g., word embedding alignment, semantic similarity) (Xu et al. 2018; Grave et al. 2019; Wang et al. 2020; Yu et al. 2022; Fang et al. 2025), domain adaptation (Courty et al. 2014; Flamary et al. 2016), and knowledge distillation (Nguyen and Luu 2022; Yang et al. 2023). In statistics, OT has been widely used for tasks such as two-sample testing (Ramdas et al. 2017), Wasserstein barycenter computation (Cuturi and Doucet 2014; Claici et al. 2018; Xu et al. 2021), estimation and statistical inference (Blanchet et al. 2021; Tameling and Munk 2018; Meng et al. 2020; Zhang, Meng, et al. 2023; Kroshnin et al. 2021; Zemel and Panaretos 2019; Klatt et al. 2020), and empirical process theory (Fournier and Guillin 2015; Weed and Bach 2019; Horowitz and Karandikar 1994; Si et al. 2020). In addition to these established areas, OT has also shown promise in emerging fields such as algorithmic fairness (Zehlike et al. 2020) and distributional clustering (Farnia et al. 2022; Li et al. 2024), further demonstrating its versatility in handling distributional and geometric tasks.

Beyond the original formulation of OT, several mathematical extensions have been developed, such as the unbalanced OT (UOT, Chizat et al. 2018; Pham et al. 2020) and the Gromov–Wasserstein (GW) distance (Mémoli 2011). The UOT problem relaxes the strict mass conservation constraint by allowing the total mass to differ between the source and target distributions. In this setting, the goal is to seek an optimal transport plan between two measures that may not exactly match the original distributions μ and ν, but are instead close to them in some divergence sense. This generalization enables UOT to be well suited for real-world scenarios where exact mass preservation does not hold, such as in the presence of noise, corrupted data, or outliers. Another important extension of the OT framework is the GW distance, which enables the comparison of the internal geometric structures of two probability measures, even when they are supported on different metric spaces. Unlike the classical OT formulation, which seeks to align individual points across two distributions defined on a common ground space, the GW distance aligns the pairwise distance relationships within each distribution. In other words, GW compares how well the relational structure (e.g., distance matrices) of one space can be mapped onto that of another, rather than matching points based on their absolute positions. Figure 1 presents schematic examples of horse registration under different OT formulations, with the horse data taken from Sumner and Popović (2004). In Figure 1a, the classical OT formulation aligns two 3D horse point clouds by seeking the most efficient transport plan from one running horse to another exhibiting a different motion style but preserving an overall similar structure. Figure 1b illustrates the UOT formulation, which applies to a more challenging scenario where the masses differ between two distributions, for example, transporting a full horse point cloud to a partial horse torso. In contrast to OT and UOT, the GW formulation in Figure 1c enables registration across different metric spaces, such as mapping a horse point cloud to a 3D mesh model.



(a) OT (b) UOT (c) GW

FIGURE 1 | Comparison of different OT formulations. (a) The classical OT aligns two 3D horse point clouds that share a similar overall structure, transporting a running horse to another of a different motion style. (b) The UOT formulation accommodates unequal total masses, enabling the registration of a full horse point cloud to a partial torso. (c) The GW formulation establishes correspondences across different metric spaces, exemplified by mapping a horse point cloud to a 3D mesh model.

Although OT has found widespread applications across various domains, its original formulation as a linear program incurs a prohibitively high computational cost (Rubner et al. 1997; Pele and Werman 2009), typically with super-cubic complexity in the number of data points. To alleviate this issue, Cuturi (2013) introduced a regularized OT formulation by adding an entropic penalty to the objective. This modification transforms the original problem into an optimization task that can be solved efficiently via iterative matrix–vector operations, reducing computational burden from super-cubic to approximately quadratic. Due to its favorable properties (such as parallelizability, smoothness, and differentiability), the resulting Sinkhorn algorithm has become a standard and widely adopted method for solving OT problems, particularly in large-scale and differentiable learning settings (Montavon et al. 2016; Eisenberger et al. 2022). Moreover, similar regularization techniques have also been extended to the computation of the GW distance. The original GW formulation poses significantly greater computational challenges compared to the classical OT. It has higher computational complexity involving fourth-order tensor product and is known to be NP-hard as it corresponds to solving a nonconvex, non-smooth optimization problem (Peyré et al. 2016; Solomon et al. 2016). To address this issue, various forms of regularization (such as the Bregman proximal term or entropic regularizer) have been introduced into the GW objective (Peyré et al. 2016; Xu et al. 2019). These relaxations enable the use of Sinkhorn-like iterative algorithms, which significantly improve computational efficiency and make the problem more tractable in large-scale settings.

Despite the computational improvements brought about by entropic regularization, the Sinkhorn algorithm still suffers from several practical limitations. For example, each iteration of the Sinkhorn algorithm requires dense matrix operations with quadratic memory and time complexity, and the algorithm typically exhibits sublinear convergence in common settings (Altschuler et al. 2017; Peyré and Cuturi 2019; Carlier 2022). Sparsification techniques have emerged as an effective and practically important class of methods to alleviate these bottlenecks. Taking advantage of the inherent or induced sparsity in the matrices involved in OT optimization, these methods aim to reduce computational overhead while preserving the accuracy of the solution. In the past few years, substantial efforts have been devoted to developing sparsity-aware variants of Sinkhorn-based algorithms to enhance their scalability (Lin et al. 2022; Li, Yu, Li, and Meng 2023; Gasteiger et al. 2021; Nguyen et al. 2023; Tang et al. 2024; Tang and Qiu 2024; Wang and Qiu 2025). In this survey, we provide a comprehensive review of such techniques, with a focus on how sparsification contributes to (i) reducing per-iteration computational cost and (ii) accelerating convergence. We will introduce two representative kinds of sparsification-based OT methods, detailing their underlying motivations, algorithmic distinctions, and practical computational complexities.

The remainder of this article is organized as follows: In Section 2, we provide a brief overview of the OT problem and several important variants of its original formulation. We also outline the role of sparsification in optimizing Sinkhorn-based algorithms from two perspectives: reducing per-iteration computational cost and improving convergence behavior. Sections 3 and 4 present detailed discussions of these two aspects, respectively. Section 5 introduces additional related works, such as partial-update OT and mini-batch OT, and distinguishes among these methods. Finally, Section 6 concludes the survey and highlights potential future directions.

2 | Problem Formulation

In this section, we begin by providing a mathematical overview of the classical OT problem, including both the Monge formulation and the Kantorovich formulation. We then introduce several widely used extensions of OT and present their corresponding mathematical formulations in detail, including entropic-regularized OT, unbalanced OT, and the GW distance. Finally, we discuss the computational bottlenecks that arise in large-scale OT problems and introduce the motivation for sparsification-based methods. We conclude this section with a review of recent advances in sparsification techniques that have been proposed to improve the scalability and efficiency of OT solvers.

2.1 | Notations

Throughout this survey, we adopt the following notational conventions. We adopt the standard convention of using uppercase boldface letters for matrices, lowercase boldface letters for vectors, and regular font for scalars. We denote by $\mathbf{1}_n \in \mathbb{R}^n$ the vector of all ones. The exponential and division operators in expression $\exp\{-\boldsymbol{A}/\lambda\}$ are applied element-wise, with a scalar $\lambda > 0$. For two probability mass vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}_+^n$, the Kullback–Leibler divergence between them is defined as $\mathrm{KL}(\boldsymbol{a} \| \boldsymbol{b}) = \sum_{i=1}^n a_i \log\left(a_i / b_i\right) - a_i + b_i$ with the standard convention that $0 \log(0) = 0$. The division operator $\oslash$ and multiplication operator $\odot$ between two vectors are also applied element-wise. For matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same dimension, the Frobenius inner product is denoted by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{i,j} A_{ij} B_{ij}$. Given a coupling matrix $\boldsymbol{P} \in \mathbb{R}_+^{n \times n}$, we denote $H(\boldsymbol{P}) = \sum_{i,j=1}^n P_{ij}\left(1 - \log P_{ij}\right)$ as the Shannon entropy of $\boldsymbol{P}$, and we adopt the standard convention that $0 \log(1/0) = 0$. We use $\| \cdot \|$ to represent the Euclidean norm for vectors and the operator norm for matrices. The ℓ_1-norm and ℓ_0-norm are denote by $\|\cdot\|_1$ and $\|\cdot\|_0$, respectively, and may be applied to both vectors and matrices. The general ℓ_p-norm for vectors is denoted by $\|\cdot\|_p$. Finally, for two non-negative sequences $\{x_n\}$ and $\{y_n\}$, we write $x_n = \tilde{\mathcal{O}}(y_n)$ if there exist constants $c, c' > 0$ such that $x_n \leq c' y_n (\log(n))^c$ for all sufficiently large n.

2.2 | The Monge Formulation

Consider the illustrative example presented in Section 1: the task of transporting sand into a collection of holes with minimal total effort can be mathematically formulated as an OT problem, where the goal is to find the most efficient way to move a probability measure μ onto another probability measure ν. The solution that minimizes the total transport cost is called the OT map if a deterministic mapping is sought (as in the Monge formulation). Mathematically, let μ and ν be two probability measures supported on $\mathbb{R}^d$. For a measurable function $T: \mathbb{R}^d \to \mathbb{R}^d$, the push-forward of μ by T, denoted by $T_\# \mu$, is defined by

$$T_{\#}\mu(\Omega) = \mu\left(T^{-1}(\Omega)\right), \forall \text{Borel sets } \Omega \subset \mathbb{R}^d.$$

Intuitively, T describes how individual points in the domain are transported, and $T_{\#}$ is the induced transformation of μ. Let $\Pi_1(\mu, \nu) = \left\{T: \mathbb{R}^d \to \mathbb{R}^d \mid T \text{ is measurable and } T_{\#}\mu = \nu\right\}$ denote the set of all plausible transport maps that push μ forward ν. The Monge formulation of OT problem is then defined as

$$\mathrm{OT}_1(\mu, \nu) = \inf_{T \in \Pi_1(\mu,\nu)} \int c(\boldsymbol{x}, T(\boldsymbol{x}))\mathrm{d}\mu(\boldsymbol{x}), \quad T^{\star} = \underset{T \in \Pi_1(\mu,\nu)}{\arg\inf} \int c(\boldsymbol{x}, T(\boldsymbol{x}))\mathrm{d}\mu(\boldsymbol{x}), \tag{1}$$

where $c(\cdot, \cdot): \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the ground cost function. A widely used choice is the pth power of the Euclidean distance, that is, $c(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^p$ with $p \geq 1$. The solution of $\mathrm{OT}_1(\mu, \nu)$ is the OT map, denoted by $T^{\star}: \mathbb{R}^d \to \mathbb{R}^d$.

However, the Monge formulation is not always well defined, as the set of admissible transport maps may be empty; that is, there may exist no measurable map T such that $T_{\#}\mu = \nu$. A classical counterexample arises when μ is a Dirac measure concentrated at a single point, while ν is a continuous probability measure or places mass on multiple disjoint points. In such cases, no deterministic map can push forward all mass from a single location to a distribution that spreads mass across multiple target points. This limitation stems from the fact that the Monge formulation does not allow mass splitting, only restricting transport to pointwise (non-branching) mappings.

2.3 | The Kantorovich Formulation

The Kantorovich formulation (Kantorovich 1942) addressed this limitation by allowing mass splitting from a source point to multiple target locations, relaxing the deterministic map to a probabilistic transportation. Specifically, it replaces the transport map T with a joint probability measure π on the product space $\mathbb{R}^d \times \mathbb{R}^d$, referred to as a coupling between μ and ν. In this more general setting, the mass conservation condition is relaxed to marginal distribution constraints on the coupling π, which must satisfy:

$$\Pi(\mu, \nu) = \left\{\pi \in \mathcal{P}\left(\mathbb{R}^d \times \mathbb{R}^d\right) \mid \pi\left(A \times \mathbb{R}^d\right) = \mu(A), \pi\left(\mathbb{R}^d \times B\right) = \nu(B), \forall \text{Borel sets} A, B \subset \mathbb{R}^d\right\}.$$

The Kantorovich formulation of the OT problem is then written as follows:

$$\mathrm{OT}_2(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int c(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\pi(\boldsymbol{x}, \boldsymbol{y}), \quad \pi^{\star} = \underset{\pi \in \Pi(\mu,\nu)}{\arg\inf} \int c(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\pi(\boldsymbol{x}, \boldsymbol{y}), \tag{2}$$

where $c(\cdot, \cdot)$ is the cost function defined earlier. The solution $\pi^{\star}$ that achieves the infimum in $\mathrm{OT}_2(\mu, \nu)$ is called the OT plan.

For the discrete case, the probability measures μ and ν are approximated by two empirical distributions supported on the bound subsets $\left\{\boldsymbol{x}_i\right\}_{i=1}^n$ and $\left\{\boldsymbol{y}_j\right\}_{j=1}^m \subset \mathbb{R}^d$, respectively. These distributions are associated with the probability mass vectors $\boldsymbol{a} = \left(a_1, \ldots, a_n\right)^{\top} \in \mathbb{R}^n$ and $\boldsymbol{b} = \left(b_1, \ldots, b_m\right)^{\top} \in \mathbb{R}^m$, where the entries satisfy $a_i \geq 0, b_j \geq 0$, and $\sum_{i=1}^n a_i = \sum_{j=1}^m b_j = 1$. Let $\boldsymbol{C} = \left(C_{ij}\right) \in \mathbb{R}_+^{n \times m}$ denote the cost matrix, where each entry is given by $C_{ij} = c\left(\boldsymbol{x}_i, \boldsymbol{y}_j\right)$. A common choice for the cost function is the squared Euclidean distance, that is, $C_{ij} = \left\|\boldsymbol{x}_i - \boldsymbol{y}_j\right\|^2$. Let $\boldsymbol{P} = \left(P_{ij}\right) \in \mathbb{R}_+^{n \times m}$ represent a transport plan, where P_{ij} denotes the amount of mass to be transported from $\boldsymbol{x}_j$ to $\boldsymbol{y}_j$. Under the marginal constraints ensuring mass conservation, the set of feasible transport plans takes the form

$$\Pi(\boldsymbol{a}, \boldsymbol{b}) = \left\{\boldsymbol{P} \in \mathbb{R}_+^{n \times m} | \boldsymbol{P}\boldsymbol{1}_m = \boldsymbol{a}, \boldsymbol{P}^{\top}\boldsymbol{1}_n = \boldsymbol{b}\right\}. \tag{3}$$

Under this formulation, the Kantorovich problem defined in Equation (2) reduces to the following finite-dimensional linear program:

$$\mathrm{OT}_2(\boldsymbol{a}, \boldsymbol{b}) = \inf_{\boldsymbol{P} \in \Pi(\boldsymbol{a},\boldsymbol{b})} \langle \boldsymbol{C}, \boldsymbol{P} \rangle, \quad \boldsymbol{P}^{\star} = \underset{\boldsymbol{P} \in \Pi(\boldsymbol{a},\boldsymbol{b})}{\arg\inf} \langle \boldsymbol{C}, \boldsymbol{P} \rangle, \tag{4}$$

where the corresponding OT plan $\boldsymbol{P}^{\star}$ is the solution that attains the minimum of $\mathrm{OT}_2(\boldsymbol{a}, \boldsymbol{b})$.

Building upon the resource allocation perspective introduced in Section 1, we consider a scenario involving n warehouses located at positions $\left\{\boldsymbol{x}_i\right\}_{i=1}^n$, each storing an amount a_i of raw materials, and m factories located at positions $\left\{\boldsymbol{y}_j\right\}_{j=1}^m$, each requiring a demand b_j of raw materials. The total supply and demand are assumed to be balanced, that is, $\sum_{i=1}^n a_i = \sum_{j=1}^m b_j = 1$, and no material is lost or created during transportation. A transport plan P_{ij} represents the amount of material transported from warehouse $\boldsymbol{x}_i$ to factory $\boldsymbol{y}_j$, and the associated transport cost C_{ij} typically reflects the distance between them, often modeled as the squared Euclidean distance, that is, $C_{ij} = \left\|\boldsymbol{x}_i - \boldsymbol{y}_j\right\|^2$. Under this setting, the feasible transport plans are subject to marginal constraints ensuring that the mass dispatched from each warehouse and received by each factory matches the prescribed supply and demand, respectively (as defined in Equation 3). The total cost incurred by the operator is calculated as the sum of transported mass multiplied by the respective cost, aggregated over all warehouse-factory pairs. Consequently, the OT problem in this context amounts to solving the linear program formalized in Equation (4), to determine an OT plan $\boldsymbol{P}^{\star}$.

Compared to the Monge formulation, the Kantorovich formulation offers several advantages. First, the Kantorovich formulation guarantees the existence of a feasible solution. The set of admissible couplings $\Pi(\boldsymbol{a}, \boldsymbol{b})$, which consists of all joint distributions with marginals $\boldsymbol{a}$ and $\boldsymbol{b}$, is always non-empty, ensuring the existence of a solution to Equation (4). Second, the Kantorovich problem can be cast as a linear program, specifically a form of the classical minimum-cost network flow problem. This convex structure allows for the use of efficient and well-established linear programming techniques and solvers. Third, the Kantorovich formulation is inherently more flexible and applicable in practical scenarios due to its allowance for mass splitting. This is especially relevant in applications such as resource allocation, where, for example, a single warehouse may need to supply materials to multiple factories. In addition, Brenier's theorem (Brenier 1991) establishes a fundamental connection between the two formulations. Specifically, when the cost function is given by $c(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2$, and at least one of the measures μ

or ν has a density, then the OT plan in the Kantorovich formulation corresponds to a deterministic transport map, as required by the Monge formulation. Throughout this survey, we focus exclusively on the Kantorovich formulation.

2.4 | Wasserstein Distance

An important feature of OT is its ability to define a meaningful distance between probability measures. Specifically, when the cost function $c(\boldsymbol{x}, \boldsymbol{y})$ is a ground metric on $\mathbb{R}^d$, then the OT problem gives rise to the Wasserstein distance, which quantifies the dissimilarity between probability distributions in a geometry-aware manner. Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be two discrete probability vectors. When the cost matrix is chosen as the pth power of the Euclidean distance, that is, $C_{ij} = \left\| \boldsymbol{x}_i - \boldsymbol{y}_j \right\|^p, p \geq 1$, the associated p-Wasserstein distance in Equation (4) is defined as

$$W_p(\boldsymbol{a}, \boldsymbol{b}) = \left(\inf_{\boldsymbol{P} \in \Pi(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{C}, \boldsymbol{P} \rangle \right)^{\frac{1}{p}}.$$

Intuitively, the Wasserstein distance captures the minimal total "transport cost" required to morph one distribution into another, effectively lifting the ground metric on the sample space to a metric on the space of probability measures. This enables a geometry-aware comparison of distributions, even when their supports are disjoint. The Wasserstein distance possesses several desirable properties that distinguish it from traditional divergence measures such as the Kullback–Leibler divergence, Jensen–Shannon divergence, or total variation distance. In particular, it provides a meaningful notion of dissimilarity even when the distributions have mismatched supports, explicitly capturing the spatial displacement between probability masses. Moreover, its strong geometric interpretability makes it especially suited for tasks that require an understanding of the structural relationship between distributions. Thus, the Wasserstein distance has found exceptionally broad application in modern machine learning, notably in deep generative modeling (Tolstikhin et al. 2018; Deshpande et al. 2019; Lei et al. 2019), and in the analysis and processing of natural language and visual data (Rolet et al. 2016; Balikas et al. 2018; Xu et al. 2018).

2.5 | Entropic-Regularized OT

Without loss of generality, we assume that m and n are of the same magnitude, that is, set $m = n$ throughout the remainder of this survey unless otherwise specified. The calculation of the original OT problem, as formulated in Equation (4), involves solving a large-scale linear programming problem. The corresponding computational complexity is typically at the order of $\mathcal{O}\left(n^3 \log(n)\right)$ (Pele and Werman 2009), which becomes prohibitive even for moderately sized datasets. This issue was bypassed by Cuturi (2013), which proposed approximating the solution by adding an entropic penalty term to the objective function, resulting in the entropic-regularized OT problem. This work demonstrated the computational advantages of the entropic formulation and its compatibility with loss functions in modern machine learning pipelines, including support for parallel computation and automatic differentiation. In practice, the entropic-regularized OT problem corresponding to Equation (4) is defined as

$$\mathrm{OT}_\lambda(\boldsymbol{a}, \boldsymbol{b}) = \inf_{\boldsymbol{P} \in \Pi(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{C}, \boldsymbol{P} \rangle - \lambda H(\boldsymbol{P}), \tag{5}$$

where $\lambda > 0$ is the regularization parameter.

ALGORITHM 1 | Computation of entropic-regularized OT.

```
Input: Kernel matrix $\boldsymbol{K}$, probability mass vectors $\boldsymbol{a}, \boldsymbol{b}$
Initialize: $t \leftarrow 0$, $\boldsymbol{v}^{(0)} \leftarrow \mathbf{1}_n$
repeat
    $t \leftarrow t + 1$
    $\boldsymbol{u}^{(t)} \leftarrow \boldsymbol{a} \oslash \boldsymbol{K}\boldsymbol{v}^{(t-1)}$, $\boldsymbol{v}^{(t)} \leftarrow \boldsymbol{b} \oslash \boldsymbol{K}^\top \boldsymbol{u}^{(t)}$
until convergence
Output: $\boldsymbol{P}_\lambda^\star = \mathrm{diag}\left(\boldsymbol{u}^{(t)}\right) \boldsymbol{K} \,\mathrm{diag}\left(\boldsymbol{v}^{(t)}\right)$
```

The entropic-regularized OT problem can be efficiently approximated using an iterative matrix scaling procedure known as the Sinkhorn algorithm (Sinkhorn 1964; Sinkhorn and Knopp 1967). Given the kernel matrix $\boldsymbol{K} = \exp\{-\boldsymbol{C} / \lambda\}$, it can be shown (Cuturi 2013; Peyré and Cuturi 2019) that the OT plan $\boldsymbol{P}_\lambda^\star$ corresponding to Equation (5) can be expressed as a projection of $\boldsymbol{K}$ onto the set of couplings $\Pi(\boldsymbol{a}, \boldsymbol{b})$. Specifically, it takes the form:

$$\boldsymbol{P}_\lambda^\star = \mathrm{diag}\left(\boldsymbol{u}^\star\right) \boldsymbol{K} \,\mathrm{diag}\left(\boldsymbol{v}^\star\right), \tag{6}$$

where $\boldsymbol{u}^\star$ and $\boldsymbol{v}^\star$ are scaling vectors that can be computed iteratively. Sinkhorn algorithm is summarized in Algorithm 1.

It can be noted that Sinkhorn algorithm only involves matrix–vector multiplication operations, whose computation is parallel and GPU friendly, effectively improving the efficiency. Franklin and Lorenz (1989) established the linear convergence of Sinkhorn's iterations with respect to (w.r.t.) the Hilbert projective metric by demonstrating that each iteration constitutes a contraction mapping under this metric. However, in practical applications, the Sinkhorn algorithm typically incurs a near-linear convergence with a computational cost of order $\widetilde{\mathcal{O}}\left(n^2\right)$, where $\widetilde{\mathcal{O}}(\cdot)$ suppresses logarithmic factors. More precisely, the complexity is $\mathcal{O}\left(Ln^2\right)$, where L denotes the total number of iterations. This iteration count L depends on the convergence accuracy and the total mass of the kernel matrix, which is often bounded above by a quantity smaller than $\log(n)$. We refer the reader to Altschuler et al. (2017), Peyré and Cuturi (2019), and Carlier (2022) for a detailed review of the convergence of the Sinkhorn algorithm.

2.6 | Unbalanced OT

Classical OT relies on the restrictive assumption that the total mass of the two marginal measures must be equal. This limitation can be problematic in applications where one needs to handle arbitrary (unnormalized) positive measures or allow for partial mass transport. Unbalanced OT (UOT) addresses this issue by relaxing the hard marginal constraints into soft ones, introducing a penalty for mass variation instead of enforcing exact conservation. A common formulation of UOT incorporates a penalty on marginal deviation using the Kullback–Leibler divergence and is expressed as:

$$\mathrm{UOT}(\boldsymbol{a},\boldsymbol{b}) = \inf_{\boldsymbol{P}\in\mathbb{R}_+^{n\times n}} \langle \boldsymbol{C},\boldsymbol{P}\rangle + \tau\mathrm{KL}\left(\boldsymbol{P}\mathbf{1}_n \,\|\, \boldsymbol{a}\right) + \tau\mathrm{KL}\left(\boldsymbol{P}^\top\mathbf{1}_n \,\|\, \boldsymbol{b}\right),$$

where the KL divergence terms softly enforce marginal alignment by penalizing discrepancies between the marginals of the transport plan $\left(\boldsymbol{P}\mathbf{1}_n, \boldsymbol{P}^\top\mathbf{1}_n\right)$ and the source/target mass $(\boldsymbol{a},\boldsymbol{b})$. The regularization parameter $\tau > 0$ controls this relaxation, balancing the trade-off between transport effort and fidelity to the input measures. As $\tau \to \infty$, the soft constraints become the original hard ones, and the UOT formulation reduces to the classical OT problem described in Equation (4).

Similarly, the regularization techniques can be applied to UOT problems to mitigate high computational costs and improve scalability. Consider the following entropic-regularized formulation of UOT:

$$\mathrm{UOT}_{\tau,\lambda}(\boldsymbol{a},\boldsymbol{b}) = \inf_{\boldsymbol{P}\in\mathbb{R}_+^{n\times n}} \langle \boldsymbol{C},\boldsymbol{P}\rangle + \tau\mathrm{KL}\left(\boldsymbol{P}\mathbf{1}_n \,\|\, \boldsymbol{a}\right) + \tau\mathrm{KL}\left(\boldsymbol{P}^\top\mathbf{1}_n \,\|\, \boldsymbol{b}\right) - \lambda H(\boldsymbol{P}), \tag{7}$$

where $\tau > 0$ and $\lambda > 0$ are regularization parameters. The objective in Equation (7) is strictly convex w.r.t. $\boldsymbol{P}$ over $\mathbb{R}_+^{n\times n}$, and therefore admits a unique solution (Chizat et al. 2018; Pham et al. 2020). It can be efficiently solved via iterative matrix scaling with kernel matrix $\boldsymbol{K} = \exp\{-\boldsymbol{C}/\lambda\}$. Chizat et al. (2018) proposed a generalized Sinkhorn algorithm to address this problem, as outlined in Algorithm 2. Notably, as $\tau \to \infty$, we have $\tau/(\tau+\lambda) \to 1$, causing the update steps for $\boldsymbol{u}$ and $\boldsymbol{v}$ in Algorithm 2 recover those in the classical Sinkhorn algorithm, given in Algorithm 1. Furthermore, Pham et al. (2020) showed that the computational complexity of the unbalanced Sinkhorn algorithm is of order $\widetilde{\mathcal{O}}\left(n^2\right)$.

ALGORITHM 2 | Computation of entropic-regularized UOT.

```
Input: Kernel matrix $\boldsymbol{K}$, probability mass vectors $\boldsymbol{a}$, $\boldsymbol{b}$,
    regularization parameters $\tau$, $\lambda$
Initialize: $t \leftarrow 0$, $\boldsymbol{u}^{(0)} \leftarrow \mathbf{1}_n$, $\boldsymbol{v}^{(0)} \leftarrow \mathbf{1}_n$
repeat
    $t \leftarrow t+1$
    $\boldsymbol{u}^{(t)} \leftarrow \left(\boldsymbol{a} \oslash \boldsymbol{K}\boldsymbol{v}^{(t-1)}\right)^{\frac{\tau}{\tau+\lambda}}$, $\boldsymbol{v}^{(t)} \leftarrow \left(\boldsymbol{b} \oslash \boldsymbol{K}^\top\boldsymbol{u}^{(t)}\right)^{\frac{\tau}{\tau+\lambda}}$
until convergence
Output: $\boldsymbol{P}^\star_{\tau,\lambda} = \mathrm{diag}\left(\boldsymbol{u}^{(t)}\right)\boldsymbol{K}\,\mathrm{diag}\left(\boldsymbol{v}^{(t)}\right)$
```

2.7 | The Gromov–Wasserstein Distance

The classical OT formulation (or Wasserstein distance) considers that the input probability measures are supported on the same underlying metric space. The Gromov–Wasserstein (GW) distance generalizes this framework to handle probability distributions supported on different metric spaces, making it well-suited for structural matching tasks. Specifically, the GW problem measures the minimal distortion required to align the intrinsic distance structures of two metric measure spaces via a joint coupling of their distributions. Given two metric measure spaces $\left(\mathcal{X}, d_{\mathcal{X}}, \mu\right)$ and $\left(\mathcal{Y}, d_{\mathcal{Y}}, \nu\right)$, where $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ are respective distances and μ, ν are probability measures, the squared GW distance is defined as

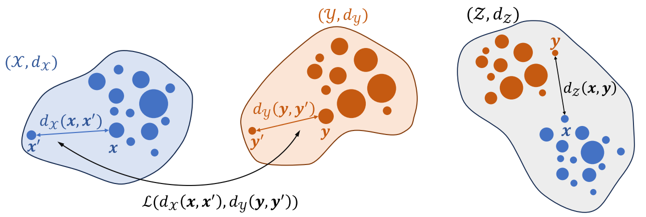
$$\begin{aligned}&\mathrm{GW}\left(\left(d_{\mathcal{X}},\mu\right),\left(d_{\mathcal{Y}},\nu\right)\right)\\ &= \inf_{\pi\in\Pi(\mu,\nu)} \iint_{\mathcal{X}\times\mathcal{Y}} \mathcal{L}\left(d_{\mathcal{X}}\left(\boldsymbol{x},\boldsymbol{x}'\right), d_{\mathcal{Y}}\left(\boldsymbol{y},\boldsymbol{y}'\right)\right)\mathrm{d}\pi(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\pi\left(\boldsymbol{x}',\boldsymbol{y}'\right),\end{aligned}$$

where $\mathcal{L}\left(d_{\mathcal{X}}\left(x,x'\right), d_{\mathcal{Y}}\left(y,y'\right)\right)$ is the ground cost function. Typical choices include the ℓ_2 loss (i.e., $\mathcal{L}\left(x_1,x_2\right) = \left|x_1 - x_2\right|^2$) and the KL divergence (i.e., $\mathcal{L}\left(x_1,x_2\right) = x_1\log\left(x_1/x_2\right) - x_1 + x_2$). Intuitively, the construction of the GW is under the assumption that if a point $\boldsymbol{x}\in\mathcal{X}$ is matched to $\boldsymbol{y}\in\mathcal{Y}$, and $\boldsymbol{x}'$ to $\boldsymbol{y}'$, then the distance $d_{\mathcal{X}}\left(\boldsymbol{x},\boldsymbol{x}'\right)$ should closely match $d_{\mathcal{Y}}\left(\boldsymbol{y},\boldsymbol{y}'\right)$. This alignment of pairwise distances enables GW to compare structural information across different domains. Figure 2 provides this geometric illustration and highlights the differences between the GW distance and the Wasserstein distance (i.e., the classical OT formulation). The GW distance compares distributions defined on different metric spaces by aligning their internal pairwise distances, whereas the Wasserstein distance compares distributions defined on a common metric space by aligning individual points directly.

To broaden the applicability of GW, Peyré et al. (2016) relaxed the requirement of $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ by allowing similarity matrices as inputs. Given two such similarity matrices $\boldsymbol{C}^{\mathcal{X}} = \left(C^{\mathcal{X}}_{ii'}\right) \in \mathbb{R}^{n\times n}$ and $\boldsymbol{C}^{\mathcal{Y}} = \left(C^{\mathcal{Y}}_{jj'}\right) \in \mathbb{R}^{n\times n}$, which encode pairwise relations (e.g., the kernel matrix and the adjacency matrix of a graph), the GW problem becomes

$$\begin{aligned}\mathrm{GW}\left(\left(\boldsymbol{C}^{\mathcal{X}},\boldsymbol{a}\right),\left(\boldsymbol{C}^{\mathcal{Y}},\boldsymbol{b}\right)\right) &= \inf_{\boldsymbol{P}\in\Pi(\boldsymbol{a},\boldsymbol{b})} \sum_{i,i',j,j'} \mathcal{L}\left(C^{\mathcal{X}}_{ii'}, C^{\mathcal{Y}}_{jj'}\right)P_{ij}P_{i'j'}\\ &= \inf_{\boldsymbol{P}\in\Pi(\boldsymbol{a},\boldsymbol{b})} \left\langle \mathcal{L}\left(\boldsymbol{C}^{\mathcal{X}},\boldsymbol{C}^{\mathcal{Y}}\right)\otimes\boldsymbol{P},\boldsymbol{P}\right\rangle \qquad (8)\\ &= \inf_{\boldsymbol{P}\in\Pi(\boldsymbol{a},\boldsymbol{b})} \langle \mathcal{C}(\boldsymbol{P}),\boldsymbol{P}\rangle,\end{aligned}$$

where the term $\mathcal{L}\left(C^{\mathcal{X}}_{ii'}, C^{\mathcal{Y}}_{jj'}\right)P_{ij}P_{i'j'}$ can be interpreted as the cost of jointly transporting the pair $\left(i,i'\right)$ to $\left(j,j'\right)$, and $\mathcal{C}(\boldsymbol{P}) = \mathcal{L}\left(\boldsymbol{C}^{\mathcal{X}},\boldsymbol{C}^{\mathcal{Y}}\right)\otimes\boldsymbol{P}$. The second line rewrites the objective using a tensorized matrix form (Peyré et al. 2016), where $\mathcal{L}\left(\boldsymbol{C}^{\mathcal{X}},\boldsymbol{C}^{\mathcal{Y}}\right)$ is a 4th-order cost tensor with entries $\mathcal{L}\left(C^{\mathcal{X}}_{ii'}, C^{\mathcal{Y}}_{jj'}\right)$, and the contraction $\mathcal{C}(\boldsymbol{P}) \in \mathbb{R}^{n\times n}$ denotes the tensor-matrix multiplication defined by


$(\mathcal{X}, d_{\mathcal{X}})$
$d_{\mathcal{X}}(\boldsymbol{x},\boldsymbol{x}')$
$\boldsymbol{x}'$
$\boldsymbol{x}$
$(\mathcal{Y}, d_{\mathcal{Y}})$
$d_{\mathcal{Y}}(\boldsymbol{y},\boldsymbol{y}')$
$\boldsymbol{y}'$
$\boldsymbol{y}$
$\mathcal{L}(d_{\mathcal{X}}(\boldsymbol{x},\boldsymbol{x}'), d_{\mathcal{Y}}(\boldsymbol{y},\boldsymbol{y}'))$
$(\mathcal{Z}, d_{\mathcal{Z}})$
$\boldsymbol{y}$
$d_{\mathcal{Z}}(\boldsymbol{x},\boldsymbol{y})$
$\boldsymbol{x}$

(a) The GW distance.

(b) The Wasserstein distance.

FIGURE 2 | Geometric interpretation of the GW and Wasserstein distances. (a) The GW distance compares two distributions supported on different metric spaces, $\left(\mathcal{X}, d_{\mathcal{X}}\right)$ and $\left(\mathcal{Y}, d_{\mathcal{Y}}\right)$. (b) The Wasserstein distance to compare two distributions supported on the same metric space, $\left(\mathcal{Z}, d_{\mathcal{Z}}\right)$, where $c(\boldsymbol{x},\boldsymbol{y}) = d_{\mathcal{Z}}(\boldsymbol{x},\boldsymbol{y})$.

ALGORITHM 3 | Computation of entropic-regularized GW.

```
1: Input: Similarity matrices $\boldsymbol{C}^{\mathcal{X}}, \boldsymbol{C}^{\mathcal{Y}}$, probability mass vectors $\boldsymbol{a}, \boldsymbol{b}$, ground cost function $\mathcal{L}$, regularization parameter $\lambda$, number
   of outer/inner iterations $R, H$
2: Initialize: $\boldsymbol{P}^{(0)} \leftarrow \boldsymbol{a}\boldsymbol{b}^\top$
3: for $r = 0$ to $R-1$ do
4:     Compute the cost matrix $\mathcal{C}\left(\boldsymbol{P}^{(r)}\right) = \mathcal{L}\left(\boldsymbol{C}^{\mathcal{X}}, \boldsymbol{C}^{\mathcal{Y}}\right) \otimes \boldsymbol{P}^{(r)}$
5:     Compute the kernel matrix $\boldsymbol{K}^{(r)} = \exp\left\{ -\mathcal{C}\left(\boldsymbol{P}^{(r)}\right) / \lambda \right\}$
6:     Update the coupling matrix $\boldsymbol{P}^{(r+1)}$ using Sinkhorn scaling in Algorithm 1
        (a) Initialize: $\boldsymbol{u}^{(0)} \leftarrow \boldsymbol{1}_n, \boldsymbol{v}^{(0)} \leftarrow \boldsymbol{1}_n$
        (b) for $h = 0$ to $H-1$ do
                $\boldsymbol{u}^{(h+1)} \leftarrow \boldsymbol{a} \oslash \boldsymbol{K}^{(r)}\boldsymbol{v}^{(h)}$, $\boldsymbol{v}^{(h)} \leftarrow \boldsymbol{b} \oslash \boldsymbol{K}^{(r)\top}\boldsymbol{u}^{(h+1)}$
            end for
        (c) $\boldsymbol{P}^{(r+1)} \leftarrow \mathrm{diag}\left(\boldsymbol{u}^{(H)}\right)\boldsymbol{K}^{(r)}\mathrm{diag}\left(\boldsymbol{v}^{(H)}\right)$
7: end for
8: Output: $\mathrm{GW}_\lambda = \left\langle \mathcal{C}\left(\boldsymbol{P}^{(R)}\right), \boldsymbol{P}^{(R)} \right\rangle - \lambda H\left(\boldsymbol{P}^{(R)}\right)$
```

$$(\mathcal{C}(\boldsymbol{P}))_{ij} = \left(\mathcal{L}\left(\boldsymbol{C}^{\mathcal{X}}, \boldsymbol{C}^{\mathcal{Y}}\right) \otimes \boldsymbol{P}\right)_{ij} = \sum_{i',j'} \mathcal{L}\left(C^{\mathcal{X}}_{ii'}, C^{\mathcal{Y}}_{jj'}\right) P_{i'j'}. \tag{9}$$

Solving the GW formulation in Equation (8) leads to a non-convex quadratic optimization problem. Similarly, to improve tractability, the entropic-regularized variant is often considered:

$$\mathrm{GW}_\lambda\left(\left(\boldsymbol{C}^{\mathcal{X}}, \boldsymbol{a}\right), \left(\boldsymbol{C}^{\mathcal{Y}}, \boldsymbol{b}\right)\right) = \inf_{\boldsymbol{P} \in \Pi(\boldsymbol{a},\boldsymbol{b})} \langle \mathcal{C}(\boldsymbol{P}), \boldsymbol{P} \rangle - \lambda H(\boldsymbol{P}), \tag{10}$$

where $\lambda > 0$ denotes the entropic regularization strength. This objective can be minimized using an iterative scheme based on projected gradient descent (Peyré et al. 2016; Solomon et al. 2016), where each iteration updates the coupling by solving a regularized OT problem. Specifically, the update at iteration $t+1$ solves

$$\boldsymbol{P}^{(t+1)} = \underset{\boldsymbol{P} \in \Pi(\boldsymbol{a},\boldsymbol{b})}{\arg\inf} \left\langle \mathcal{C}\left(\boldsymbol{P}^{(t)}\right), \boldsymbol{P} \right\rangle - \lambda H(\boldsymbol{P}), \tag{11}$$

with $\mathcal{C}\left(\boldsymbol{P}^{(t)}\right) = \mathcal{L}\left(\boldsymbol{C}^{\mathcal{X}}, \boldsymbol{C}^{\mathcal{Y}}\right) \otimes \boldsymbol{P}^{(t)}$ serving as the cost matrix based on the current iteration. This subproblem (11) coincides with the entropic-regularized OT problem in Equation (5) using $\boldsymbol{C} = \mathcal{C}\left(\boldsymbol{P}^{(t)}\right)$, and can thus be efficiently solved via the Sinkhorn algorithm. The complete procedure is summarized in Algorithm 3. The computation complexity of Algorithm 3 is $\mathcal{O}\left(n^4\right)$ per iteration in general scenarios, primarily due to the repeated tensor-matrix operations. This high cost significantly limits the scalability of GW in large-scale applications.

2.8 | Sparsification Techniques for OT Problems

Recall that to achieve a given approximation accuracy, the Sinkhorn algorithm for OT typically incurs a computational cost of order $\widetilde{\mathcal{O}}\left(n^2\right)$. This cost arises from two main sources: (1) each iteration requires matrix–vector multiplications with complexity $\mathcal{O}\left(n^2\right)$, and (2) the algorithm exhibits relatively slow, near-linear convergence. These limitations persist in the entropic-regularized variants of UOT and GW problems. In particular, the GW optimization suffers from even higher complexity, $\mathcal{O}\left(n^4\right)$ per iteration, due to the costly construction of the kernel matrix involving tensor-matrix contractions.

Numerous studies have proposed techniques to accelerate the entropic-regularized versions, including partial updates of the selected rows or columns of the transport plan (Genevay et al. 2016; Altschuler et al. 2017; Alaya et al. 2019; Lin et al. 2022), first-order acceleration schemes (Dvurechensky et al. 2018; Guminov et al. 2021; Thibault et al. 2021; Lin et al. 2022), and the incorporation of structural priors on the coupling matrix or the ground cost function (Xu et al. 2019; Chowdhury et al. 2021; Scetbon et al. 2022). In this report, we focus on sparse subsampling methods, which aim to reduce the computational burden of entropic-regularized OT and its variants (e.g., UOT and GW) by addressing the key sources of inefficiency: high per-iteration cost and slow convergence. A summary of these methods is provided below.

- Reducing the per-iteration computational cost. These methods apply element-wise subsampling and construct sparse approximations of the kernel matrix based on different selection criteria (Gasteiger et al. 2021; Li, Yu, Li, and Meng 2023; Li, Yu, Xu, and Meng 2023). Using such a surrogate for the original matrix, they achieve substantial savings in runtime without severely compromising accuracy with existing sparse matrix multiplication techniques (Drineas et al. 2006; Mahoney 2011; Gupta and Sidford 2018).
- Accelerating convergence through second-order methods. When viewed from the dual perspective, OT problems can benefit from second-order optimization techniques (Brauer et al. 2017; Tang et al. 2024; Tang and Qiu 2024; Wang and Qiu 2025). In particular, sparsified Newton-type methods approximate the Hessian and solve sparse linear systems to compute search directions efficiently, thereby reducing the total number of iterations required.

3 | Sparsification for Kernel Matrix

In this section, we focus on methods that sparsely subsample the kernel matrix based on specific selection criteria, such as importance sampling (Li, Yu, Li, and Meng 2023; Li, Yu, Xu, and Meng 2023) and locality-sensitive hashing (Gasteiger et al. 2021). These methods significantly reduce the matrix operations per iteration, for instance, line 5 in Algorithms 1 and 2, and lines 4–6 in Algorithm 3. By constructing efficient approximations of the original kernel matrix, these methods enable near log-linear computational complexity $\widetilde{\mathcal{O}}(n)$ for problems (5) and (7), and near-quadratic complexity $\mathcal{O}\left(n^{2+\delta}\right)$ ($\delta > 0$ is an arbitrary small number) for problem (10), while maintaining a fixed approximation accuracy.

3.1 | Importance Sparse Sinkhorn Method

Li, Yu, Li, and Meng (2023) adopted the Poisson sampling framework following the recent work of Braverman et al. (2021) to construct a sparse sketch $\widetilde{\boldsymbol{K}}$ that approximates the original kernel matrix $\boldsymbol{K}$. The central objective is to construct an asymptotically unbiased approximation $\widetilde{\boldsymbol{K}}$ with low variance. To this end, Li, Yu, Li, and Meng (2023) leveraged importance sampling (Liu and Liu 2001), which assigns higher sampling probabilities to entries with larger magnitudes to improve estimation efficiency and reduce variance when estimating a summation. When the exact values are not available, appropriate upper bounds can be used to approximate the sampling probabilities (Kahn and Marshall 1953; Owen 2013; Zhao and Zhang 2015).

Specifically, for the entropic-regularized OT problem in Equation (5), a sparse approximation $\widetilde{\boldsymbol{K}}$ of $\boldsymbol{K}$ is constructed via Poisson sampling. In this approach, a small fraction of elements from $\boldsymbol{K}$ are selected and rescaled, while the remaining entries are set to zero. Given a subsampling budget $s < n^2$, and a set of sampling probabilities $\{p_{ij}\}_{i,j=1}^{n}$ satisfying $\sum_{i,j} p_{ij} = 1$, the sparse matrix $\widetilde{\boldsymbol{K}}$ is defined as

$$\widetilde{K}_{ij} = \begin{cases} K_{ij}/p_{ij}^{*} & \text{with prob. } p_{ij}^{*} = \min\left(1, sp_{ij}\right) \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where the rescaling ensures that $\widetilde{\boldsymbol{K}}$ is an unbiased estimator of $\boldsymbol{K}$, and the expected number of non-zero entries satisfies $\mathbb{E}\left\{\mathrm{nnz}\left(\widetilde{\boldsymbol{K}}\right)\right\} = \sum_{i,j} p_{ij}^{*} \leq s$. Li, Yu, Li, and Meng (2023) proposed an importance sampling scheme to determine p_{ij}:

$$p_{ij} = \frac{\sqrt{a_i b_j}}{\sum_{i,j=1}^{n} \sqrt{a_i b_j}}, \qquad 1 \leq i,j \leq n. \tag{13}$$

This is motivated by three key observations. First, the OT plan $\boldsymbol{P}_{\lambda}^{\star}$ shares the same sparsity structure as the kernel $\boldsymbol{K}$ due to Equation (6). Second, the transport loss can be expressed as $\left\langle \boldsymbol{C}, \boldsymbol{P}_{\lambda}^{\star} \right\rangle = \sum_{i,j} C_{ij} \left(P_{\lambda}^{\star}\right)_{ij}$, so sparsifying $\boldsymbol{K}$ can be interpreted as selecting terms from this summation. Third, from a variance reduction perspective, the optimal sampling probabilities for estimating this sum should be proportional to $C_{ij}\left(P_{\lambda}^{\star}\right)_{ij}$. Suppose that C_{ij} is upper bounded by a constant c_0, then due to the marginal constraints $\left(P_{\lambda}^{\star}\right)_{ij} \leq \min\left\{a_i, b_j\right\}$, there exits an upper bound $C_{ij}\left(P_{\lambda}^{\star}\right)_{ij} \leq c_0 \sqrt{a_i b_j}$, which leads to the probability formulation in Equation (13).

The corresponding procedure is provided in Algorithm 4, denoted as Spar-Sink algorithm. Figure 3 provides a visual comparison between the Spar-Sink and Sinkhorn algorithm.

ALGORITHM 4 | Spar-Sink algorithm for entropic-regularized OT.

```
Input: Kernel matrix $\boldsymbol{K}$, probability mass vectors $\boldsymbol{a}, \boldsymbol{b}$
Construct Spar-Sink according to (12) and (13)
Compute $\widetilde{\boldsymbol{P}}_{\lambda}^{\star}$ with $\widetilde{\boldsymbol{K}}$ using Sinkhorn scaling in
  Algorithm 1
  (a) Initialize: $t \leftarrow 0, \boldsymbol{u}^{(0)} \leftarrow \mathbf{1}_n, \boldsymbol{v}^{(0)} \leftarrow \mathbf{1}_n$
  (b) repeat
          $t \leftarrow t + 1$
          $\boldsymbol{u}^{(t)} \leftarrow \boldsymbol{a} \oslash \widetilde{\boldsymbol{K}} \boldsymbol{v}^{(t-1)}, \boldsymbol{v}^{(t)} \leftarrow \boldsymbol{b} \oslash \widetilde{\boldsymbol{K}}^{\top} \boldsymbol{u}^{(t)}$
      until convergence
  (c) Compute $\widetilde{\boldsymbol{P}}_{\lambda}^{\star} = \mathrm{diag}\left(\boldsymbol{u}^{(t)}\right) \widetilde{\boldsymbol{K}} \,\mathrm{diag}\left(\boldsymbol{v}^{(t)}\right)$
Output: $\widetilde{\mathrm{OT}}_{\lambda}(\boldsymbol{a}, \boldsymbol{b}) = \left\langle \boldsymbol{C}, \widetilde{\boldsymbol{P}}_{\lambda}^{\star} \right\rangle - \lambda H\left(\widetilde{\boldsymbol{P}}_{\lambda}^{\star}\right)$
```

Specifically, Spar-Sink applies importance sampling to the kernel matrix, yielding a sparse structure $\widetilde{\boldsymbol{K}}$ and the resulting OT plan $\widetilde{\boldsymbol{P}}_{\lambda}^{\star}$. Both theoretical analysis and empirical evidence suggest that the subsampling budget s should be at least of order $\widetilde{\mathcal{O}}(n)$ to ensure a reliable approximation.

Following the same line of thinking, the sampling probabilities for the entropic-regularized UOT problem in Equation (7) are similarly defined as follows:

$$p_{ij} = \frac{\left(a_i b_j\right)^{\frac{\tau}{2\tau+\lambda}} K_{ij}^{\frac{\tau}{2\tau+\lambda}}}{\sum_{i,j}\left(a_i b_j\right)^{\frac{\tau}{2\tau+\lambda}} K_{ij}^{\frac{\tau}{2\tau+\lambda}}}, \qquad 1 \leq i,j \leq n. \tag{14}$$

The Spar-Sink algorithm for entropic-regularized UOT is obtained by replacing the full kernel matrix in Algorithm 2 with its sparse counterpart, computed using Equations (12) and (14). The resulting estimator is denoted as $\widetilde{\mathrm{UOT}}_{\tau,\lambda}(\boldsymbol{a}, \boldsymbol{b})$. For further theoretical justification and algorithmic details, we refer the reader to Li, Yu, Li, and Meng (2023).

Theoretical analysis demonstrates that, under mild regularized conditions, the approximation error between the sparse estimators and theirs exact counterparts remains sufficiently small, provided that the subsampling budget s and the sample size n satisfy a suitable scaling relation. That is, $\widetilde{\mathrm{OT}}_{\lambda}(\boldsymbol{a}, \boldsymbol{b})$ and $\widetilde{\mathrm{UOT}}_{\tau,\lambda}(\boldsymbol{a}, \boldsymbol{b})$ are statistically consistent w.r.t. the entropic-regularized OT and UOT distance $\mathrm{OT}_{\lambda}(\boldsymbol{a}, \boldsymbol{b})$ and $\mathrm{UOT}_{\tau,\lambda}(\boldsymbol{a}, \boldsymbol{b})$, respectively. The proposed sparse algorithms reduce the per-iteration complexity to $\mathcal{O}(s) = \widetilde{\mathcal{O}}(n)$ while maintaining the same convergence rate in terms of the number of iterations.

The underlying motivation of this sparsification strategy is rooted in the fact that, when the regularization parameter λ is small, the OT plan tends to be sparse; see Peyré and Cuturi (2019) for a detailed discussion. Importance-based sparsification effectively exploits this structure by constructing a sparse approximation of the kernel matrix $\boldsymbol{K}$, which mirrors the sparsity pattern of the transport plan. It is worth noting that the sampling probabilities for entropic-regularized OT, defined in Equation (13), depend solely on the marginal distributions a_i and b_j, and are independent of the corresponding cost values C_{ij}. From a different perspective,

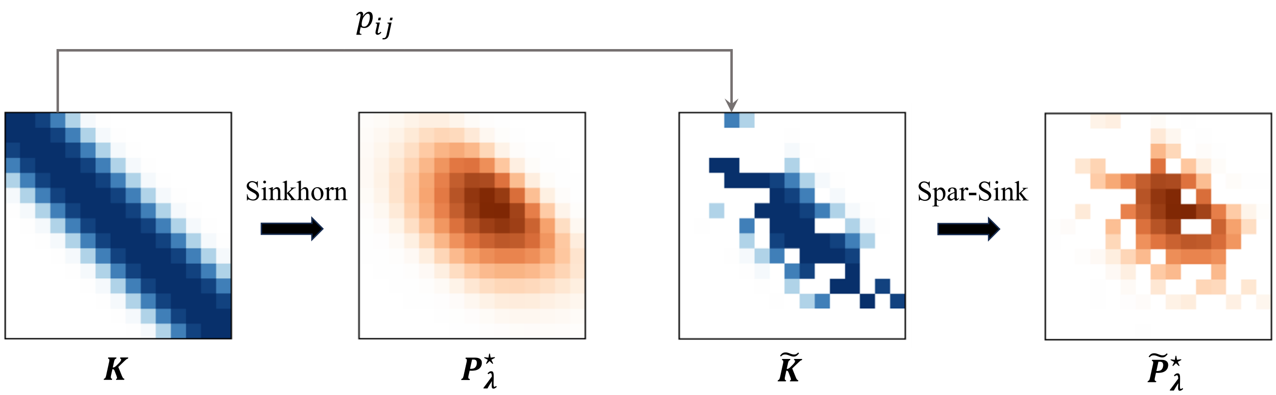

p_{ij}
Sinkhorn
Spar-Sink
$\boldsymbol{K}$
$\boldsymbol{P}_{\lambda}^{\star}$
$\widetilde{\boldsymbol{K}}$
$\widetilde{\boldsymbol{P}}_{\lambda}^{\star}$

FIGURE 3 | Illustration of the Sinkhorn algorithm (left) and the Spar-Sink algorithm (right). In Spar-Sink, each entry of $\boldsymbol{K}$ is independently sampled according to $\{p_{ij}\}$ defined in Equation (12). The non-zero entries are highlighted in color.

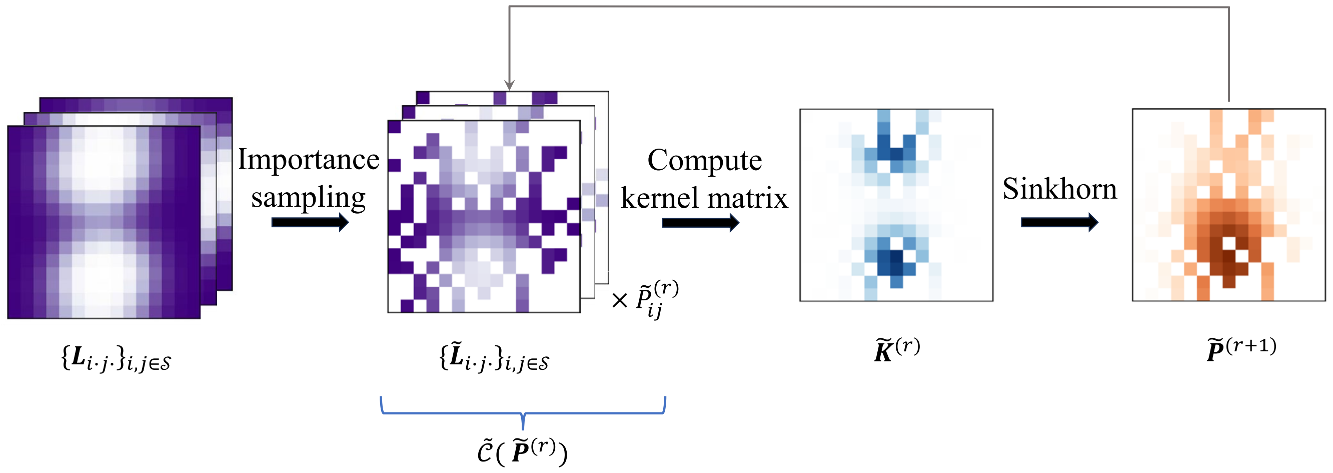

Importance sampling
Compute kernel matrix
Sinkhorn
$\times \widetilde{P}_{ij}^{(r)}$
$\{\boldsymbol{L}_{i\cdot j\cdot}\}_{i,j\in\mathcal{S}}$
$\{\widetilde{\boldsymbol{L}}_{i\cdot j\cdot}\}_{i,j\in\mathcal{S}}$
$\widetilde{\boldsymbol{K}}^{(r)}$
$\widetilde{\boldsymbol{P}}^{(r+1)}$
$\tilde{\mathcal{C}}(\widetilde{\boldsymbol{P}}^{(r)})$

FIGURE 4 | Illustration of the Spar-GW method. The sparse matrices $\left\{\widetilde{\boldsymbol{L}}_{i\cdot j\cdot}\right\}_{(i,j)\in\mathcal{S}}$ are sampled from $\{\boldsymbol{L}_{i\cdot j\cdot}\}_{(i,j)\in\mathcal{S}}$ respectively according to the probabilities $\{p_{ij}\}$ defined in Equation (13), yielding the sparse kernel and transport plan matrices $\widetilde{\boldsymbol{K}}^{(r)}$ and $\widetilde{\boldsymbol{P}}^{(r)}$.

alternative sparsification scheme (Gasteiger et al. 2021) has also been proposed that relies directly on the cost structure itself.

3.2 | Importance Sparsification for GW Distance

Following a similar line of reasoning, Li, Yu, Xu, and Meng (2023) proposed a randomized sparsification method called Spar-GW to calculate the entropic-regularized GW distance (10), using importance sampling principles. Instead of directly sparsifying the kernel matrix, they construct a sparse surrogate $\widetilde{\mathcal{C}}(\boldsymbol{P}^{(r)})$ to approximate the full cost matrix $\mathcal{C}(\boldsymbol{P}^{(r)})$ in the rth iteration. This strategy serves a dual purpose: it accelerates the Sinkhorn scaling steps and substantially reduces the computational cost of evaluating the tensor-matrix contractions involved in $\mathcal{C}(\boldsymbol{P}^{(r)})$.

To construct $\widetilde{\mathcal{C}}\left(\widetilde{\boldsymbol{P}}^{(r)}\right)$, consider the definition of $\mathcal{C}(\boldsymbol{P}^{(r)})$ in Equation (9). The idea is to build a collection of s sparse matrices $\left\{\widetilde{\boldsymbol{L}}_{i\cdot j\cdot}\right\}_{(i,j)\in\mathcal{S}}$ extracted from the 4th-order tensor $\mathcal{L}(\boldsymbol{C}^{\mathcal{X}}, \boldsymbol{C}^{\mathcal{Y}})$, where $\mathcal{S} = \{(i_l, j_l)\}_{l=1}^{s}$ is a set of index pairs sampled according to a given sampling budget s. For each $(i,j) \in \mathcal{S}$, the matrix $\widetilde{\boldsymbol{L}}_{i\cdot j\cdot}$ contains only a small number of meaningful entries:

$$\widetilde{L}_{ii'jj'} = \begin{cases} \mathcal{L}\left(C_{ii'}^{\mathcal{X}}, C_{jj'}^{\mathcal{Y}}\right) & \text{if } (i', j') \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } (i,j) \in \mathcal{S}. \tag{15}$$

The final sparse cost approximation is then computed as $\widetilde{\mathcal{C}}\left(\widetilde{\boldsymbol{P}}^{(r)}\right) = \sum_{(i,j)\in\mathcal{S}} \widetilde{\boldsymbol{L}}_{i\cdot j\cdot} \widetilde{P}_{ij}^{(r)}$, where $\widetilde{P}_{ij}^{(r)}$ corresponds to the (i,j)th entry of the transport matrix in the iteration $\boldsymbol{r}$. Setting the zero entries to ∞ in Equation (15) includes a sparse structure in the corresponding kernel $\widetilde{\boldsymbol{K}}^{(r)}$, which is inherited by the transport plan matrix $\widetilde{\boldsymbol{P}}^{(r)}$. This sparsification scheme simultaneously reduces the cost of the tensor-matrix computation and accelerates the Sinkhorn scaling steps. The sparsification scheme $\mathcal{S}$ is constructed following a similar importance sampling principle as in Equation (13), motivated by the same reasons discussed in Section 3.1. The overall procedure is illustrated in Figure 4; see Li, Yu, Xu, and Meng (2023) for further algorithmic and theoretical details.

Compared to other accelerated methods for computing the entropic-regularized GW distance, such as scalable GW learning (Xu et al. 2019), low-rank GW (Scetbon et al. 2022), sliced GW (Titouan, Flamary, et al. 2019), and linear-time GW (Scetbon et al. 2022), which typically rely on restrictive assumptions on the ground cost function $\mathcal{L}$ (e.g., requiring a ℓ_2

loss or decomposability), on the coupling matrix $\boldsymbol{P}$ (e.g., low-rank or tree-structured) or on the input data type (e.g., requiring point clouds or graphs), this method achieves a substantial computational speed up without imposing such constraints. Specifically, it reduces the overall complexity from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^2 + s^2) = \mathcal{O}(n^{2+\delta})$, where the subsampling budget satisfies $s = \mathcal{O}(n^{1+\delta})$ for any arbitrary small $\delta > 0$. This improvement implies that only approximately n^2 elements need to be accessed from a full 4D cost tensor containing n^4 elements, while still maintaining a provable approximation guarantee. Moreover, this importance sparsification mechanism can be further applied to other related problems, including approximation of the original GW distance (8), the unbalanced GW distance (Séjourné et al. 2021; Kawano and Mason 2021; Luo, Wang, et al. 2022), and the fused GW distance (Titouan, Courty, et al. 2019; Vayer et al. 2020).

3.3 | Sparse LSH and Locally Corrected Nyström Method

Unlike prior methods that focus solely on kernel matrix sparsification for entropic-regularized OT, Gasteiger et al. (2021) proposed a two-stage approximation method, called locally corrected Nyström Sinkhorn (LCN-Sinkhorn) method. In the first stage, rather than applying random sparsification, LCN-Sinkhorn performs hard-thresholding based on locality-sensitive hashing (LSH) (Shrivastava and Li 2014) to generate a sparse kernel structure that preserves local interactions. In the second stage, the method applies a local correction to the classical Nyström approximation, using the LSH-induced sparsity pattern to better correct global geometry. By fusing both local (sparse) and global (low-rank) approximations, LCN-Sinkhorn is able to effectively model interactions between both nearby and distant points.

In the context of the entropic-regularized OT problem in Equation (5), LCN-Sinkhorn focuses exclusively on interactions between spatially proximate points based. In the first stage, it approximates the full cost matrix $\boldsymbol{C}$ with a sparse matrix $\boldsymbol{C}^{\mathrm{sp}}$, where

$$C_{ij}^{\mathrm{sp}} = \begin{cases} C_{ij} & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{y}_j \text{ are ``near''} \\ \infty & \text{otherwise.} \end{cases}$$

This sparsified cost matrix then induces the corresponding kernel matrix $\boldsymbol{K}^{\mathrm{sp}}$ and transport plan $\boldsymbol{P}^{\mathrm{sp}}$ following from their definitions, whose non-zero entries are concentrated almost entirely around each point's nearest neighbors, effectively capturing only the most relevant local interactions. To efficiently identify such "near" point pairs, LCN-Sinkhorn employs LSH, a randomized technique that maps similar points to the same hash bucket with high probability, while ensuring dissimilar points are likely to fall into different buckets. In this work, the authors specifically consider cross-polytope LSH (Andoni et al. 2015) and k-means LSH (Paulevé et al. 2010), offering a distinct balance between locality preservation and computational efficiency.

The second stage leverages the locally sparse structure induced by LSH to refine the global low-rank approximation obtained via the Nyström method (Williams and Seeger 2000; Musco and Musco 2017). The Nyström method approximates the positive semi-definite kernel matrix $\boldsymbol{K}$ by selecting a small subset of l representative points (landmarks) and constructing a low-rank factorization of the form $\boldsymbol{K} \approx \boldsymbol{K}_{\mathrm{Nsy}} = \boldsymbol{U}\boldsymbol{A}^{-1}\boldsymbol{V}$, where $\boldsymbol{A}$ is the kernel matrix evaluated on the landmarks, and $\boldsymbol{U}, \boldsymbol{V}$ encode cross-similarities between the input points and the landmarks. The authors adopted k-means Nyström (Oglic and Gärtner 2017) to choose the landmarks from $\{\boldsymbol{x}_i\} \cup \{\boldsymbol{y}_j\}$. The final approximation is then obtained by the correction of the Nyström approximation by the exact values of LSH sparsification elements: $\boldsymbol{K}_{\mathrm{LCN}} = \boldsymbol{K}_{\mathrm{Nsy}} - \boldsymbol{K}_{\mathrm{Nsy}}^{\mathrm{sp}} + \boldsymbol{K}^{\mathrm{sp}}$, where $\boldsymbol{K}_{\mathrm{Nsy}}^{\mathrm{sp}}$ contains the entries of the Nyström approximation $\boldsymbol{K}_{\mathrm{Nsy}}$ corresponding to the non-zero elements of $\boldsymbol{K}^{\mathrm{sp}}$. Figure 5 provides a direct illustration of LCN-Sinkhorn approximation $\boldsymbol{K}_{\mathrm{LCN}}$ w.r.t. the kernel matrix $\boldsymbol{K}$. The intermediate approximated kernel matrices at each step are

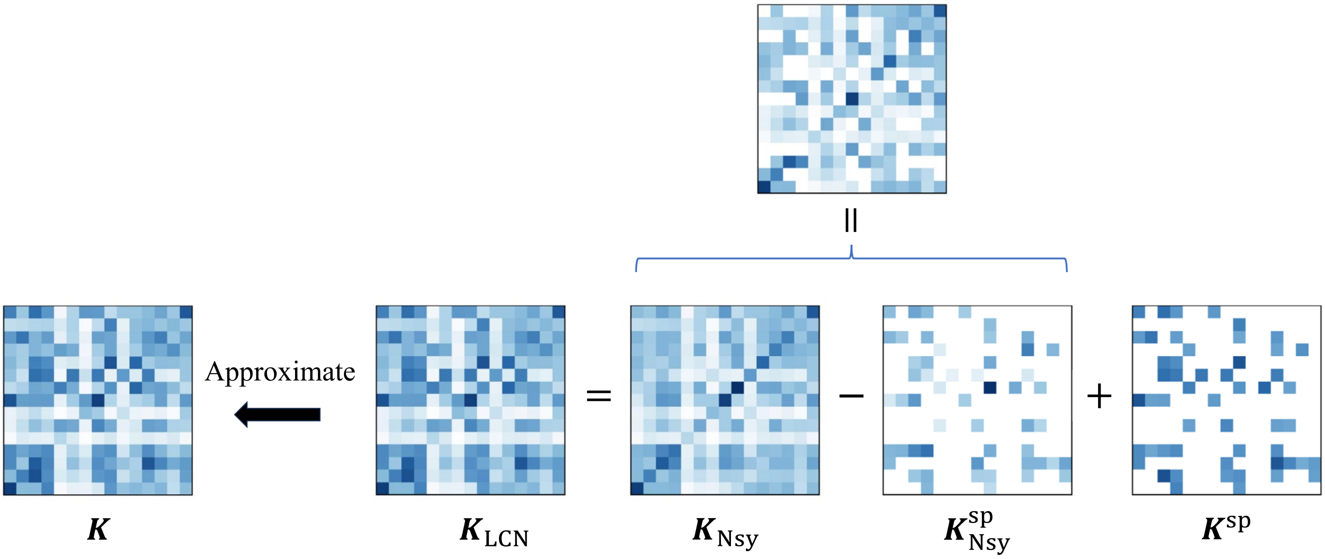

Approximate
=
−
+
$\boldsymbol{K}$
$\boldsymbol{K}_{\mathrm{LCN}}$
$\boldsymbol{K}_{\mathrm{Nsy}}$
$\boldsymbol{K}_{\mathrm{Nsy}}^{\mathrm{sp}}$
$\boldsymbol{K}^{\mathrm{sp}}$

FIGURE 5 | Schematic illustration of the LCN-Sinkhorn approximation. The matrix $\boldsymbol{K}_{\mathrm{Nsy}}$ provides a global low-rank approximation of the kernel matrix $\boldsymbol{K}$, which is subsequently refined by incorporating the locally sparse correction $\boldsymbol{K}^{\mathrm{sp}}$. The fusion of these two components yields the locally corrected Nyström approximation $\boldsymbol{K}_{\mathrm{LCN}}$.

also shown. This approach improves upon prior work Nyström–Sinkhorn (Altschuler et al. 2019), which applied the Nyström approximation directly to the entropic-regularized OT problem without local correction. Further technical details can be found in Altschuler et al. (2019) and Gasteiger et al. (2021).

Gasteiger et al. (2021) demonstrates that the two stages (e.g., local LSH sparsification and global low rank approximation) are complementary rather than competing. LSH-based sparsification may become ineffective in scenarios where pairwise costs are very similar, the regularization is very strong, or points have a large number of close neighbors. Conversely, the Nyström approximation can be unstable under weak regularization or when the cost matrix exhibits high variability. By adjusting the number of neighbors and landmarks, the method enables a smooth interpolation between LSH-based sparsification and Nyström–Sinkhorn, providing a flexible trade-off between local and global approximation. Combining both strategies, the method offers great effectiveness and flexibility across a wider range of problem settings. Moreover, while the Spar-Sink method focuses on independent importance sampling based on the marginal distributions (i.e., the pointwise masses), LCN-Sinkhorn captures both local and global structural interactions in the data. It first achieves log-linear time complexity and is stable enough to substitute full entropic-regularized OT problem. Specifically, the overall computational complexity is $\mathcal{O}\left(n\log(n)+nl^2\right)$, where l is the number of landmarks.

4 | Sparsification for Hessian Matrix

In this section, we explore the sparsified second-order methods based on Newton-type framework to accelerate the overall convergence of entropic-regularized OT problem. It is well known that, under some smoothness assumptions on the objective function, the Newton method enjoys a fast quadratic local convergence rate, typically requiring significantly fewer iterations than the Sinkhorn algorithm. However, for entropic-regularized OT, each Newton iteration involves solving a dense linear system with a computational cost of $\mathcal{O}\left(n^3\right)$, which undermines the efficiency advantage of Sinkhorn. To address this challenge, recent works have proposed sparsified second-order methods that leverage the dual formulation of the entropic-regularized OT problem. We emphasize that the previously discussed kernel-based methods are directly applied to the primal OT formulation by substituting the original kernel matrix with a sparser one. In contrast, the Hessian-based sparsification methods, typically framed within a Newton-type framework, are primarily based on the dual formulation of the entropic-regularized OT problem. These methods aim to reduce the computational burden by sparsifying the Hessian matrix and further offer some valuable insights into efficient Hessian approximation. Notable methods include hard-thresholding-based sparsification (Tang et al. 2024), off-diagonal sparsification scheme that guarantees strict approximation error control (Tang and Qiu 2024), and hybrid techniques combining low-rank structure with top-entry selection for efficient Hessian approximation (Wang and Qiu 2025).

4.1 | Dual Formulation for Entropic-Regularized OT

By introducing the dual variables $\boldsymbol{\alpha}\in\mathbb{R}^n, \boldsymbol{\beta}\in\mathbb{R}^n$, and applying the minimax theorem, the entropic-regularized OT problem in Equation (5) admits the following dual formulation (Peyré and Cuturi 2019):

$$\begin{aligned}\max_{\alpha,\beta} F(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \max_{\alpha,\beta}\min_{\boldsymbol{P}}\langle \boldsymbol{C},\boldsymbol{P}\rangle - \lambda H(\boldsymbol{P}) - \alpha^\top\left(\boldsymbol{P}\mathbf{1}_n-\boldsymbol{a}\right)-\beta^\top\left(\boldsymbol{P}^\top\mathbf{1}_n-\boldsymbol{b}\right)\\ &= \max_{\alpha,\beta}\boldsymbol{\alpha}^\top\boldsymbol{a}+\boldsymbol{\beta}^\top\boldsymbol{b}-\lambda\sum_{i,j=1}^{n}\exp\left\{\left(\alpha_i+\beta_j-C_{ij}\right)/\lambda\right\},\end{aligned} \tag{16}$$

where $F(\boldsymbol{\alpha},\boldsymbol{\beta})$ is referred to as the Lyapunov potential function. Solving the original entropic-regularized formulation is thus equivalent to maximizing this dual potential. Within this framework, the matrix scaling steps in the Sinkhorn algorithm can be interpreted as performing alternating maximization over $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, also known as applying the block coordinate ascent (Sinkhorn 1964; Cuturi 2013; Yule 1912); a detailed algorithm is provided in the following subsection. Let $\boldsymbol{\alpha}^\star$ and $\boldsymbol{\beta}^\star$ denote an optimal solution to Equation (16). Then, the corresponding OT plan $\boldsymbol{P}^\star$ can be recovered element-wise via

$$P_{ij}^\star = \exp\left\{\left(\alpha_i^\star+\beta_j^\star-C_{ij}\right)/\lambda\right\}. \tag{17}$$

Obviously, optimizing Equation (16) is equivalently to solving the following problem:

$$\begin{aligned}\min_{\alpha,\beta} f(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \min_{\alpha,\beta} -F(\boldsymbol{\alpha},\boldsymbol{\beta})\\ &= \min_{\alpha,\beta}\lambda\sum_{i,j=1}^{n}\exp\left\{\left(\alpha_i+\beta_j-C_{ij}\right)/\lambda\right\}-\boldsymbol{\alpha}^\top\boldsymbol{a}-\boldsymbol{\beta}^\top\boldsymbol{b}.\end{aligned} \tag{18}$$

It is worth noting that the solution to Equation (18), or equivalently to (16), is not unique: if $\left(\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star\right)$ is one solution, then so is $\left(\boldsymbol{\alpha}^\star+c\mathbf{1}_n,\boldsymbol{\beta}^\star-c\mathbf{1}_n\right)$ for any constant c. This redundant degree of freedom makes the Hessian matrix of $f(\boldsymbol{\alpha},\boldsymbol{\beta})$ singular, but one can simply remove it by forcing $\boldsymbol{\alpha}^\top\mathbf{1}_n+\boldsymbol{\beta}^\top\mathbf{1}_n=0$ or $\boldsymbol{\beta}_n=0$. In what follows, we omit this subtlety for the brevity of presentation, and assume that the Hessian matrix is nonsingular. Importantly, Equation (18) defines a smooth, unconstrained, and strictly convex optimization problem after resolving the redundant degree of freedom in the variables $(\boldsymbol{\alpha},\boldsymbol{\beta})$, which makes it amenable to a wide range of optimization techniques (Brauer et al. 2017; Dvurechensky et al. 2018; Guminov et al. 2021; Thibault et al. 2021; Lin et al. 2022). First-order methods such as gradient descent, as well as second-order approaches including Newton (Dembo et al. 1982; Li et al. 2004) and quasi-Newton (Dennis and Moré 1977; Liu and Nocedal 1989) methods, can be effectively applied to solve it. The gradient and Hessian matrix of $f(\boldsymbol{\alpha},\boldsymbol{\beta})$ have the following closed-form expressions:

$$\begin{aligned}\nabla f(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \begin{bmatrix}\boldsymbol{P}\mathbf{1}_n-\boldsymbol{a}\\ \boldsymbol{P}^\top\mathbf{1}_n-\boldsymbol{b}\end{bmatrix},\quad \boldsymbol{H}(\boldsymbol{\alpha},\boldsymbol{\beta})=\nabla^2 f(\boldsymbol{\alpha},\boldsymbol{\beta})\\ &= \lambda^{-1}\begin{bmatrix}\operatorname{diag}\left(\boldsymbol{P}\mathbf{1}_n\right) & \boldsymbol{P}\\ \boldsymbol{P}^\top & \operatorname{diag}\left(\boldsymbol{P}^\top\mathbf{1}_n\right)\end{bmatrix}.\end{aligned} \tag{19}$$

In particular, the matrix $\boldsymbol{P}$ is a function of the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, as defined in Equation (17). There has been extensive research devoted to developing efficient sparsification strategies for the Hessian matrix $\boldsymbol{H}(\boldsymbol{\alpha}, \boldsymbol{\beta})$, motivated by various perspectives (Tang et al. 2024; Tang and Qiu 2024; Wang and Qiu 2025).

4.2 | Hard-Thresholding Rule for Hessian Matrix

Tang et al. (2024) proposed a simple yet effective sparsification strategy for the Hessian matrix based on a hard-thresholding rule, aimed at accelerating the Newton method in the context of Sinkhorn scaling. This approach enables efficient approximate solutions to the Newton system $\boldsymbol{H}\Delta\boldsymbol{z} = -\nabla f$ by leveraging the sparsity of the Hessian, significantly reducing the per-iteration computational cost from the original $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. This method, called the Sinkhorn–Newton–Sparse (SNS) algorithm, maintains the fast convergence guarantees of the Newton framework while substantially alleviating the computational burden.

The design of SNS is motivated by the observation that the Hessian of the Lyapunov potential becomes approximately sparse when the number of Sinkhorn iterations is sufficiently large and the regularization parameter λ is sufficiently small. More precisely, if the number of iterations t and the inverse regularization parameter $1/\lambda$ are large enough, then after t Sinkhorn updates, the Hessian matrix becomes $(\rho, \widetilde{\rho}) = (3/(2n), \widetilde{\rho})$-sparse for some parameter ρ, which depends on the problem size n, the iteration count t, and λ. Here, the $(\rho, \widetilde{\rho})$-sparsity is defined in the sense that there exists a matrix $\boldsymbol{M}$ with at most a proportion ρ of non-zero entries (i.e., $\|\boldsymbol{M}\|_0/n^2 \le \rho$) such that $\|\boldsymbol{H} - \boldsymbol{M}\|_1 \le \widetilde{\rho}$. Based on this analysis, one can safely approximate $\boldsymbol{M}$ with a relaxed target sparsity $\rho = \mathcal{O}(1/n) > 3/(2n)$ after a moderate number of Sinkhorn iterations.

The SNS algorithm consists of two stages. In the first stage, it performs N_1 iterations of the Sinkhorn algorithm to update the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, providing a warm start and promoting approximate sparsity in the Hessian matrix $\boldsymbol{H}$. The number of iterations N_1 can be either fixed in advance or determined adaptively. In the second stage, SNS switches to the Newton method for further updating the dual variables, while applying a hard-thresholding rule to sparsify the Hessian. Specifically, all elements in $\boldsymbol{H}$ smaller than a user-specified threshold η (equals to the $\lfloor \rho n^2 \rfloor$ th largest element in $\boldsymbol{H}$) are set to zero. This truncation preserves both the symmetry and diagonal dominance of the original matrix, yielding a sparse approximation $\widetilde{\boldsymbol{H}}$ with a desired sparsity level ρ. Based on $\widetilde{\boldsymbol{H}}$, an approximate Newton search direction $\widetilde{\Delta\boldsymbol{z}}$ is computed as a surrogate for the exact solution $\Delta\boldsymbol{z}$ to the Newton system $\boldsymbol{H}\Delta\boldsymbol{z} = -\nabla f$. This system can be efficiently solved using the conjugate gradient method for linear systems (Golub and Van Loan 2013), leading to a per-iteration complexity of $\mathcal{O}(\rho n^3) = \mathcal{O}(n^2)$. It is worth noting that SNS differs from the method in Brauer et al. (2017) that directly applies Newton's method to minimize the dual objective in Equation (18), which incur a much higher cost of $\mathcal{O}(n^3)$ per iteration. The complete procedure is summarized in Algorithm 5.

4.3 | Off-Diagonal Sparsification for Hessian Matrix

However, as pointed out by Tang and Qiu (2024), due to the sparsification strategy adopted in SNS, the approximated Hessian matrix $\widetilde{\boldsymbol{H}}$ may not be positive definite. As a result, there is no strong guarantee of invertibility when solving the Newton system using the conjugate gradient method. Furthermore, while Tang et al. (2024) claimed that SNS achieves a faster convergence rate, the justification is purely empirical; no rigorous theoretical analysis is provided to support the convergence behavior of the method. Acknowledging these limitations, Tang and Qiu (2024) introduced a safe Newton-type algorithm, referred to as the Safe and Sparse Newton method for Sinkhorn (SSNS), which incorporates a novel sparsification scheme designed to have a

ALGORITHM 5 | Sinkhorn-Newton-Sparse (SNS) algorithm.

```
Input: Cost matrix $\boldsymbol{C}$, probability mass vectors $\boldsymbol{a}, \boldsymbol{b}$, initial dual variables $\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0$, number of iterations $N_1, N_2$, threshold $\eta$
Initialize: $t \leftarrow 0, (\boldsymbol{\alpha}, \boldsymbol{\beta}) \leftarrow (\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0)$,
# Sinkhorn stage
while $t < N_1$ do
    $\boldsymbol{P} \leftarrow \exp\left\{\frac{\boldsymbol{\alpha}\mathbf{1}_n^\top + \mathbf{1}_n\boldsymbol{\beta}^\top - \boldsymbol{C}}{\lambda}\right\}, \boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \lambda\left(\log(\boldsymbol{a}) - \log\left(\boldsymbol{P}\mathbf{1}_n\right)\right)$
    $\boldsymbol{P} \leftarrow \exp\left\{\frac{\boldsymbol{\alpha}\mathbf{1}_n^\top + \mathbf{1}_n\boldsymbol{\beta}^\top - \boldsymbol{C}}{\lambda}\right\}, \boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \lambda\left(\log(\boldsymbol{b}) - \log\left(\boldsymbol{P}^\top\mathbf{1}_n\right)\right)$
    $t \leftarrow t + 1$
end while
# Newton stage
$\boldsymbol{z} \leftarrow (\boldsymbol{\alpha}, \boldsymbol{\beta})$
while $t < N_1 + N_2$ do
    Construct sparse approximation $\widetilde{\boldsymbol{H}}$ by thresholding Hessian $\boldsymbol{H}$ with parameter $\eta$
    Compute Newton direction $\widetilde{\Delta\boldsymbol{z}}$ by solving $\widetilde{\boldsymbol{H}}\widetilde{\Delta\boldsymbol{z}} = -\nabla f$ using the conjugate gradient method
    Perform line search to determine step size $r$
    $\boldsymbol{z} \leftarrow \boldsymbol{z} + r \cdot \widetilde{\Delta\boldsymbol{z}}$
    $t \leftarrow t + 1$
end while
Output: $\boldsymbol{P}^\star = \exp\left\{\frac{\boldsymbol{\alpha}\mathbf{1}_n^\top + \mathbf{1}_n\boldsymbol{\beta}^\top - \boldsymbol{C}}{\lambda}\right\}$
```

ALGORITHM 6 | Sparsifying the Hessian matrix in SSNS.

```
Input: Cost matrix $\boldsymbol{C}$, probability mass vectors $\boldsymbol{a}, \boldsymbol{b}$, dual variables $\boldsymbol{\alpha}, \boldsymbol{\beta}$, threshold $\gamma$
Initialize: $\Delta \leftarrow$ zero matrix in $\mathbb{R}^{n \times n}$, $\boldsymbol{P} \leftarrow \exp\left\{\frac{\boldsymbol{\alpha}\boldsymbol{1}_n^\top + \boldsymbol{1}_n\boldsymbol{\beta}^\top - \boldsymbol{C}}{\lambda}\right\}$
for $j = 1, 2, \ldots, n$ do
    Identify the smallest-magnitude entries in $\boldsymbol{P}_{\cdot j}$ whose cumulative sum is below $\gamma$ but would exceed $\gamma$ if one more entry
    were included; copy them to the corresponding positions in the zero vector $\Delta_{\cdot j} \in \mathbb{R}^n$
end for
for $i = 1, 2, \ldots, n$ do
    Identify the smallest-magnitude entries in $\Delta_{i\cdot}$ whose cumulative sum is below $\gamma$ but would exceed $\gamma$ if one more entry
    were included; set all remaining entries in $\Delta_{i\cdot}$ to zero
end for
$\widetilde{\boldsymbol{P}} \leftarrow \boldsymbol{P} - \Delta$
Output: $\widetilde{\boldsymbol{H}} \leftarrow \lambda^{-1}\begin{bmatrix}\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_n) & \widetilde{\boldsymbol{P}} \\ \widetilde{\boldsymbol{P}}^\top & \mathrm{diag}(\boldsymbol{P}^\top\boldsymbol{1}_n)\end{bmatrix}$
```

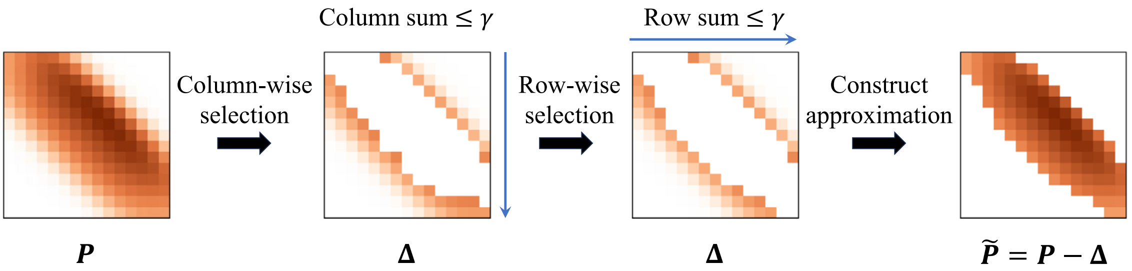


FIGURE 6 | Illustration of the sparsification procedure for matrix $\boldsymbol{P}$ in SSNS. In the first step, the smallest entries in each column of $\boldsymbol{P}$ are selected such that their cumulative sum remains below the threshold γ, while the inclusion of the next entry would make it exceed γ. These elements are stored in the auxiliary matrix $\boldsymbol{\Delta}$. The same operation is then applied row-wise to Δ. The final sparse approximation is obtained as $\widetilde{\boldsymbol{P}} = \boldsymbol{P} - \Delta$.

well-controlled approximation error and to ensure the positive definiteness of the resulting Hessian approximation. Theoretical analyses of both global and local convergence rates are also presented.

The sparsification scheme in SSNS is constructed to satisfy two essential criteria. First, the approximated matrix $\widetilde{\boldsymbol{H}}$ should remain sufficiently close to the original Hessian $\boldsymbol{H}$, with a tunable approximation error γ, so that essential information is retained. Second, the positive definiteness of $\widetilde{\boldsymbol{H}}$ should be preserved, as the Newton system $\boldsymbol{H}\Delta\boldsymbol{z} = -\nabla f$ must remain solvable. To this end, Tang and Qiu (2024) proposed an adaptive off-diagonal sparsification algorithm tailored to these two goals, which retains the diagonal blocks $\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_n)$ and $\mathrm{diag}(\boldsymbol{P}^\top\boldsymbol{1}_n)$ unchanged, while applying sparsification only to the off-diagonal components $\boldsymbol{P}$. Given a threshold γ, SSNS proceeds by identifying the smallest-magnitude entries in each column whose cumulative sum is below γ. In particular, it retains the largest set of entries such that their cumulative sum is $\leq \gamma$, while the inclusion of the next smallest entry would make the sum strictly greater than γ. These identified entries are then removed, resulting in a sparsified version of the Hessian, followed by the same procedure across each row. This off-diagonal truncation strategy always avoids singularity in $\widetilde{\boldsymbol{H}}$ regardless of sparsification parameter during the computation of Newton search directions. Moreover, the scheme guarantees that the residual error, measured in the elementwise ℓ_1-norm, is bounded by the specified threshold γ (up to a scaling factor λ). That is, $\left\|\boldsymbol{H} - \widetilde{\boldsymbol{H}}\right\|_1 \leq \gamma / \lambda$. The complete sparsification procedure is summarized in Algorithm 6 and illustrated in Figure 6. The left of SSNS is to choose moderate value of step size r and threshold γ; see more details in Tang and Qiu (2024).

Theoretically, SSNS is demonstrated to achieve global convergence to the unique optimal solution from any arbitrary initialization, without requiring a warm start via the Sinkhorn algorithm. Furthermore, SSNS attains a quadratic local convergence rate, matching that of the classical Newton method based on the full (dense) Hessian matrix. The algorithm is robust with respect to the choice of initial value, step sizes, sparsification strength, and hyper-parameters. However, this work does not provide a detailed analysis of the per-iteration computational complexity of each Sinkhorn–Newton step, leaving open the question of its precise computational guarantees.

4.4 | Low-Rank and Sparsification for Hessian Matrix

The SSNS method faces a key limitation: The sparsity level of the Hessian matrix after sparsification is not known in advance, making it challenging to predict the actual computational cost in practice. To address this issue, Wang and Qiu (2025) conducted a more thorough investigation of the effects of Hessian matrix sparsification, which motivates a novel scheme that enables explicit control over the sparsity of the Hessian approximation. Similar to the idea in Gasteiger et al. (2021), Wang and Qiu (2025) investigate scenarios where the transport plan is dense, in which case existing sparsification strategies tend to perform poorly. To overcome this, they introduce an additional low-rank correction term to the sparsified Hessian, aiming to recover the essential curvature information lost during sparsification. This leads to the Sparse-Plus-Low-Rank (SPLR) method, which combines a hard-thresholded sparse approximation with carefully constructed low-rank components. The resulting quasi-Newton algorithm achieves both fast convergence and improved computational efficiency.

Wang and Qiu (2025) conducted a comprehensive analysis of the eigenvalue structure of the sparsified Hessian matrix, offering new theoretical insights into the effects of sparsification and establishing a rigorous foundation for designing flexible sparsification strategies. Within the off-diagonal sparsification framework adopted in SSNS, the sparsification is applied exclusively to the off-diagonal component $\boldsymbol{P}$. A sparsified version of $\boldsymbol{P}$, denoted by $\widetilde{\boldsymbol{P}}_{\mathcal{S}}$, is defined over an index set $\mathcal{S} = \left\{ \left(i_l, j_l \right) \right\}_{l=1}^{s} \subset \{(i,j)\}_{i,j=1}^{n}$. The sparsified matrix and its associated Hessian approximation are given by

$$\left(\widetilde{P}_{\mathcal{S}} \right)_{ij} = \begin{cases} P_{ij} & (i,j) \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}, \qquad \widetilde{\boldsymbol{H}}_{\mathcal{S}} = \lambda^{-1} \begin{bmatrix} \operatorname{diag}\left(\boldsymbol{P}\boldsymbol{1}_n \right) & \widetilde{\boldsymbol{P}}_{\mathcal{S}} \\ \widetilde{\boldsymbol{P}}_{\mathcal{S}}^{\top} & \operatorname{diag}\left(\boldsymbol{P}^{\top}\boldsymbol{1}_n \right) \end{bmatrix}.$$

Wang and Qiu (2025) establishes a theoretical result characterizing how the condition number of $\widetilde{\boldsymbol{H}}_{\mathcal{S}}$ evolves as the sparsity pattern changes. Specifically, under a certain regularity condition, they show that the increased sparsity level can lead to improved numerical conditioning. Formally, let $\mathcal{S}_1 \subset \mathcal{S}_0$, and suppose there exists a positive integer $p > 0$ such that $\left(\widetilde{\boldsymbol{H}}_{\mathcal{S}_1} \right)^{p} > 0$. Then, the condition number of $\widetilde{\boldsymbol{H}}_{\mathcal{S}_1}$ is smaller than that of $\widetilde{\boldsymbol{H}}_{\mathcal{S}_0}$. In particular, setting $\mathcal{S}_0$ as the full index set $\{(i,j)\}_{i,j=1}^{n}$, and this result implies that any sparsification pattern $\mathcal{S}_1$ satisfying the regularity condition preserves the positive definiteness of $\boldsymbol{H}$ and improves its numerical stability. As a concrete example, they show that the sparsity pattern $\mathcal{S}_* = \{(i,j) \mid i = 1 \,\text{or}\, j = 1\}$ satisfies the regularity condition with exponent $p = 4$, that is, $\left(\widetilde{\boldsymbol{H}}_{\mathcal{S}_*} \right)^{4} > 0$.

ALGORITHM 7 | Sparsifying the Hessian matrix in SPLR.

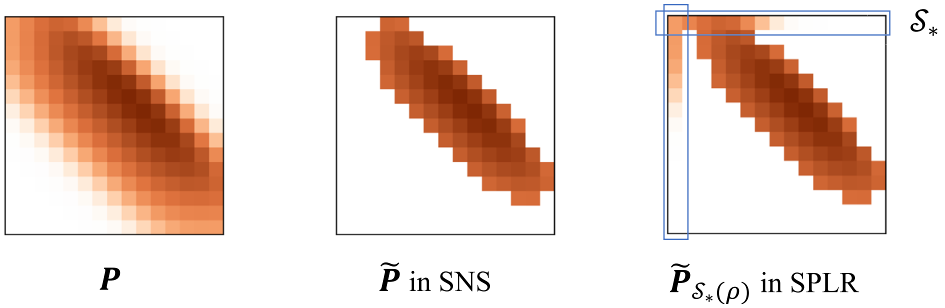
```
1: Input: Cost matrix $\boldsymbol{C}$, probability mass vectors $\boldsymbol{a}$, $\boldsymbol{b}$, dual
   variables $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, proportion parameter $0 < \rho < 1$
2: Compute $\boldsymbol{P} \leftarrow \exp\left\{ \frac{\boldsymbol{\alpha}\boldsymbol{1}_n^{\top} + \boldsymbol{1}_n\boldsymbol{\beta}^{\top} - \boldsymbol{C}}{\lambda} \right\}$
3: Identify the index set $\mathcal{S}(\rho)$ corresponding to the $\lfloor \rho n^2 \rfloor$
   largest entries in $\boldsymbol{P}$
4: Construct $\mathcal{S}_*(\rho) = \mathcal{S}_* \cup \mathcal{S}(\rho)$
5: Output: $\widetilde{\boldsymbol{H}}_{\mathcal{S}_*(\rho)} \leftarrow \lambda^{-1} \begin{bmatrix} \operatorname{diag}\left( \boldsymbol{P}\boldsymbol{1}_n \right) & \widetilde{\boldsymbol{P}}_{\mathcal{S}_*(\rho)} \\ \widetilde{\boldsymbol{P}}_{\mathcal{S}_*(\rho)}^{\top} & \operatorname{diag}\left( \boldsymbol{P}^{\top}\boldsymbol{1}_n \right) \end{bmatrix}$
```

Building upon this insight, the sparsification scheme $\mathcal{S}_*(\rho)$ in SPLR consists of the structurally important index set $\mathcal{S}_*$ and a data-dependent component $\mathcal{S}(\rho)$, where $\mathcal{S}(\rho)$ contains the $\lfloor \rho n^2 \rfloor$ largest-magnitude entries in the matrix $\boldsymbol{P}$. The full sparsification procedure is described in Algorithm 7. Remark that the hard-thresholding rule used in SNS selects the $\lfloor \rho n^2 \rfloor$ th largest entry of the Hessian matrix $\boldsymbol{H}$ as the cutoff threshold η, which leads to similar sparsity pattern as $\widetilde{\boldsymbol{H}}_{\mathcal{S}}$ in SPLR. Therefore, the primary distinction between the SNS and SPLR sparsification schemes lies in: (1) the off-diagonal sparsification pattern, since SNS directly sparsifies the full Hessian matrix instead of the off-diagonal submatrix $\boldsymbol{P}$; (2) the inclusion of the index set $\mathcal{S}_*$, as illustrated in Figure 7. This additional structure plays a critical role in guaranteeing the improved condition number after sparsification, which further ensures the positive definiteness of the sparsified Hessian, as justified by the theoretical analysis discussed earlier.

Considering the scenarios in which the Hessian matrix $\boldsymbol{H}$ is relatively dense, the SPLR method augments the sparse approximation $\widetilde{\boldsymbol{H}}_{\mathcal{S}}$ with additional low-rank terms to recover essential curvature information lost during sparsification. The final Hessian approximation $\widetilde{\boldsymbol{H}}$ takes the following form:

$$\widetilde{\boldsymbol{H}} = \widetilde{\boldsymbol{H}}_{\mathcal{S}} + \left(m\boldsymbol{s}\boldsymbol{s}^{\top} + n\boldsymbol{t}\boldsymbol{t}^{\top} \right) + h\boldsymbol{I}, \tag{20}$$

where $m\boldsymbol{s}\boldsymbol{s}^{\top} + n\boldsymbol{t}\boldsymbol{t}^{\top}$ is a low-rank correction of rank two, and $h\boldsymbol{I}$ is a shift term added to improve the numerical stability during matrix inversion. Under this construction, SPLR proceeds within a quasi-Newton optimization framework, where the dual variables are updated by computing the search direction $\Delta\boldsymbol{z}$ via solving the linear system $\widetilde{\boldsymbol{H}}\Delta\boldsymbol{z} = -\nabla f$. While the augmentation in Equation (20) makes $\widetilde{\boldsymbol{H}}$ dense again, its structure allows for efficient matrix inversion using sparse and low-rank arithmetic techniques. The parameters $m, n, h, \boldsymbol{s}$, and $\boldsymbol{t}$ are determined based on the classical secant condition from the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update rule (Liu



$\boldsymbol{P}$

$\widetilde{\boldsymbol{P}}$ in SNS

$\widetilde{\boldsymbol{P}}_{\mathcal{S}_*(\rho)}$ in SPLR

FIGURE 7 | Comparison of the sparsification procedures for matrix $\boldsymbol{P}$ in SNS and SPLR. The SNS scheme retains only the entries with the largest magnitudes, whereas SPLR additionally involves the index set $\mathcal{S}_*$ to achieve better theoretical properties on the sparsified matrix, such as the improved condition number.

and Nocedal 1989; Cuturi and Peyré 2018). Detailed algorithmic procedures and implementation considerations are further elaborated in Wang and Qiu (2025).

Convergence analysis shows that the SPLR algorithm enjoys a global convergence guarantee, with a worst-case convergence rate that is at least linear. In practice, empirical results indicate that SPLR often exhibits super-linear-like convergence behavior across a range of problem instances. Wang and Qiu (2025) asserts that SPLR achieves rapid convergence with low computational cost, making it a practical alternative to fully dense second-order methods. The per-iteration complexity of SPLR is $\mathcal{O}(n^2)$, identical to that of SNS, as it preserves the same order of non-zero elements in the sparsified Hessian matrix. Combined with its global linear convergence rate, this leads to an overall computational cost of $\mathcal{O}(n^2 \log(1/\varepsilon))$ for a given precision tolerance ε.

5 | Other Related Works

Other related works such as partial-update OT and mini-batch OT also randomly or selectively sample a few rows/columns of the transport plan or data samples to accelerate OT computations, which also can be viewed as structured sparsification strategies.

In particular, partial-update OT methods update only part of the transport plan in each iteration according to certain sampling or selection criteria. Representative examples include the greedy Sinkhorn (GREENKHORN, Altschuler et al. 2017), stochastic OT (Genevay et al. 2016), screening Sinkhorn (SCREENKHORN, Alaya et al. 2019) algorithms and others. Specifically, the GREENKHORN algorithm performs coordinate-wise updates, where only a single row or column of the transport plan $\boldsymbol{P}$ is updated at each coordinate descent iteration. Instead of alternating between full row and column updates as in the classical Sinkhorn algorithm, GREENKHORN greedily selects the row or column that most violates the marginal constraints under a suitable divergence measure, and updates only that part to reduce the violation most efficiently. In contrast, Genevay et al. (2016) employed the stochastic averaged gradient method, which uniformly and randomly selects a data point (sample) from the empirical distribution to update the average gradient with respect to the dual variables in Equation (18). The SCREENKHORN method adapts the static screening test from sparse supervised learning to the dual formulation (18) of entropic-regularized OT. By imposing a threshold on the coordinates of the dual variables, SCREENKHORN reduces the optimization scale of the original OT problem, directing computational resources towards the active variables.

Moreover, another relevant work is mini-batch OT and widely used in several situations Damodaran et al. (2018), Liutkus et al. (2019), and Tong et al. (2024). Mini-batch OT methods substitute the original large-scale OT computation in Equation (4)/(5) with more computationally efficient procedures on subsets of the full dataset. This is achieved by splitting the problem into smaller sub-problems and aggregating the results of these sub-problems to approximate the original transport plan. The statistical properties of mini-batch OT have been explored by Fatras et al. (2020), Bernton et al. (2019), Sommerfeld et al. (2019), and Fatras, Zine, et al. (2021). The subsampling process can involve either uniform or random sampling, with or without replacement, as well as other sampling schemes under specific constraints, as discussed in Fatras, Zine, et al. (2021). However, the issue of misspecified mappings has been noted, wherein the transport plan estimated from mini-batch OT may produce transport mappings that deviate from the original solution of the OT problem (Nguyen and Luu 2022). To address this issue, Fatras, Séjourné, et al. (2021) proposed replacing the OT formulation with the UOT formulation between empirical measures derived from mini-batches, while Nguyen and Luu (2022) suggested the use of partial OT for transportation between mini-batches.

This review focuses on element-based sparsification strategies for entropic-regularized OT problems, analyzing them from both kernel-based and Hessian-based perspectives. These methods leverage the inherent sparse structures of the kernel and Hessian matrices, respectively, to improve computational efficiency. In contrast, methods such as partial-update OT and mini-batch OT reduce computational costs without explicitly considering the matrix structure, instead relying on selective sampling of data or coordinates based on specific criteria. Moreover, mini-batch OT involves sampling multiple subsets (mini-batches) from the original dataset and solving the corresponding sub-problems over these subsets. Each sub-problem can be treated as a sparse subsample, though the overall procedure merely splits the full dataset into smaller chunks. The computational efficiency of mini-batch OT is primarily derived from the parallelizability of these sub-problems, rather than from a reduction in the total computation across the entire procedure.

6 | Conclusion

6.1 | Summary

In this survey, we review recent sparsification techniques developed to improve the scalability of entropic-regularized OT and its variants. These methods are categorized into two major classes based on the formulation they operate on: (1) kernel-based sparsification, which directly sparsifies the kernel matrix in the primal formulation, and (2) Hessian-based sparsification, which focuses on sparsifying the Hessian matrix in the dual formulation. Kernel-based methods reduce the per-iteration computational complexity while maintaining similar convergence rates as the standard Sinkhorn algorithm. In contrast, Hessian-based methods are designed to accelerate convergence by leveraging Newton-type updates. They seek to alleviate the cubic computational burden of dense Newton steps through the sparsification analysis of the Hessian matrix. However, many of these methods currently lack rigorous analysis regarding algorithmic accuracy and computational complexity.

We provide a summary of the accelerated methods discussed in our review in Table 1, which includes the category of each method, its computational complexity per iteration, and convergence rate. A "—" indicates that the method is not

TABLE 1 | Comparison of various accelerated Sinkhorn methods in terms of method category, per-iteration computational complexity, and convergence rate.

Method	Method category	Computational complexity	Convergence rate
Sinkhorn	—	$\mathcal{O}(n^2)$	$\widetilde{\mathcal{O}}(1/\varepsilon^2)$
Spar-Sink	Kernel-based	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(1/\varepsilon^2)$
Spar-GW	Kernel-based	$\mathcal{O}(n^{2+\delta})$	—
LCN-Sinkhorn	Kernel-based	$\mathcal{O}(n\log(n) + nl^2)$	$\widetilde{\mathcal{O}}(1/\varepsilon^2)$
SNS	Hessian-based	$\mathcal{O}(n^2)$	—
SSNS	Hessian-based	—	$\mathcal{O}(\log(\log(1/\varepsilon)))$
SPLR	Hessian-based	$\mathcal{O}(n^2)$	$\mathcal{O}(\log(1/\varepsilon))$
Stochastic OT	Partial-update	$\mathcal{O}(n)$	—
GREENKHORN	Partial-update	$\mathcal{O}(n^2)$	$\widetilde{\mathcal{O}}(1/\varepsilon^2)$
SCREENKHORN	Partial-update	—	—

discussed with respect to that particular aspect. The parameter ε represents the precision tolerance, quantifying the distance between the original OT distance and the estimated OT distance obtained through each acceleration method. From the table, we observe that kernel-based methods maintain the same convergence rate as the original Sinkhorn algorithm, while reducing the computational complexity per iteration. In contrast, Hessian-based methods significantly improve the overall convergence rate of the original Sinkhorn algorithm, though they do not alter the computational complexity of each iteration. Thus, kernel-based and Hessian-based methods accelerate the original Sinkhorn algorithm from complementary perspectives.

However, existing sparsity-driven Sinkhorn methods also face several limitations, which suggest potential directions for further improvement. For instance, the Spar-Sink and Spar-GW methods primarily rely on sparsity patterns derived from marginal distributions under the assumption that the elements of the cost matrix are bounded by a constant, thereby neglecting the valuable structural information encoded in the cost matrix itself. Moreover, when applied to entropic-regularized UOT, the sampling probabilities depend on the regularization parameters (τ, λ), whereas for entropic-regularized OT, they are independent of the regularization parameter. Hence, developing sparsification strategies that adapt to both the cost matrix and the regularization parameters could enhance robustness and efficiency. In addition, these methods introduce stochasticity in the subsampling process, as Spar-Sink and Spar-GW perform importance sampling on the kernel matrix only once. To mitigate this randomness, it may be beneficial to apply the subsampling procedure multiple times and aggregate the resulting sparse kernel matrices, which could lead to more stable and reliable approximations. Second, some sparsity-driven Sinkhorn methods rely on a large number of manually specified hyperparameters. For example, the number of landmarks l in the k-means Nyström method, the parameters involved in the AND–OR construction of LSH within the LCN-Sinkhorn approximation, and the shift parameter h as well as the density parameter ρ in the SPLR method all require careful tuning. Developing adaptive strategies that can dynamically adjust sparsity patterns according to the cost structure or data geometry would alleviate the burden of extensive manual hyperparameter tuning.

Furthermore, all of the sparsity-aware Sinkhorn methods discussed in this paper are highly versatile and applicable to a wide range of learning tasks. Several sparsity-aware Sinkhorn methods have already been successfully applied across various domains. By replacing the original Sinkhorn distance computation with its sparsity-aware variants, these applications achieve more efficient training while preserving accuracy. For example, LCN-Sinkhorn has been used for unsupervised word embedding alignment and for graph distance regression on graph transport networks; Spar-GW can be directly applied to graph distance regression problems; and Spar-Sink has been used for tasks such as color transfer between images, echocardiogram similarity analysis, and approximating the Sinkhorn divergence during the training of auto-encoders in Sinkhorn-based generative modeling. We are also eager to explore further applications of these methods, including Wasserstein barycenters, multi-marginal OT, and Wasserstein gradient flows, where replacing the original Sinkhorn or Wasserstein distance with sparsity-aware alternatives may lead to significant computational improvements.

6.2 | Future Work

Future research can proceed along several promising directions. Building upon the limitations and potential improvements discussed in the previous subsection, we further outline several prospective avenues that may extend and enrich sparsification-based OT methods.

First, extending these sparsification strategies to a wider range of OT problems, including Wasserstein barycenters (Rabin et al. 2011; Cuturi and Doucet 2014), multi-marginal OT (Haasler et al. 2021; Beier et al. 2023; Hu et al. 2025), and Wasserstein gradient flows (Peyré 2015; Mokrov et al. 2021), could significantly broaden their applicability.

Second, existing acceleration techniques primarily focus on the entropic-regularized OT problem. However, quadratically regularized OT (Blondel et al. 2018; Lorenz et al. 2021) has gained increasing importance in modern applications (Daniels et al. 2021; Xu and Cheng 2023) due to its ability to produce sparse transport plans, which are often more desirable than the dense transport plans obtained from entropic-regularized OT. The lack of sparsity in the latter can be problematic, particularly when the transport plan itself is of interest. Many studies have introduced quadratic regularization into various OT problems, including classical OT (Blondel et al. 2018), graph OT (Essid and Solomon 2018), UOT (Nguyen et al. 2023), and partial OT (Tran et al. 2025). Migrating existing acceleration techniques, such as sparsification strategies and optimization methods (e.g., the Nyström method and Nesterov's method), to quadratically regularized OT is an exciting avenue for future work. Analyzing the properties of quadratically regularized OT, such as sparsity or low-rank structures in the matrices involved in the optimization process, and exploring the feasibility of applying acceleration techniques to this domain remain a promising research direction.

Third, combining different sparsification heuristics may yield further improvements. For example, Spar-Sink focuses mainly on sparsity patterns derived from marginal distributions, while LCN-Sinkhorn emphasizes spatial locality. Integrating both perspectives or enriching them with additional structural priors, such as low-rank approximations or graph-based constraints, may lead to more effective and generalizable sparsification frameworks. For example, similar to the LCN method, which first sparsifies the kernel matrix using hard-thresholding based on LSH and subsequently locally corrects the sparse kernel matrix using the Nyström method to capture local interactions more effectively, we are motivated to apply a similar approach. Specifically, we can propose locally correcting the sparse kernel or Hessian matrix generated by a specific sparsification strategy using a low-rank approximation method. Alternatively, one could first perform a low-rank decomposition of the kernel matrix and then apply sparse sampling on the decomposed low-rank matrix.

Finally, closer integration with application domains (e.g., computational biology, graphics) and design of algorithms that account for task-specific constraints will help translate sparsification advances into practical impact.

Author Contributions

Xiaxue Ouyang: investigation (equal), methodology (equal), writing – original draft (equal). **Hao Zheng:** investigation (equal), visualization (equal). **Haoxian Liang:** investigation (equal), software (equal), visualization (equal). **Jingyi Zhang:** supervision (equal). **Yixuan Qiu:** investigation (equal), methodology (equal), supervision (equal), writing – review and editing (equal). **Cheng Meng:** supervision (equal), writing – review and editing (equal). **Mengyu Li:** supervision (equal), writing – review and editing (equal).

Acknowledgments

This work is supported by Beijing Municipal Natural Science Foundation No. 1232019, National Natural Science Foundation of China Grant No. 12301381, 72571163, 12271522, National Key Research and Development Program of China No. 2021YFA1001300.

Funding

This work was supported by the National Natural Science Foundation of China, No. 12301381, 72571163, 12271522; National Key Research and Development Program of China, No. 2021YFA1001300; Natural Science Foundation of Beijing Municipality, No. 1232019.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are openly available in https://people.csail.mit.edu/sumner/research/deftransfer/data.html#caveats at https://doi.org/10.1145/1015706.1015736, reference number Sumner and Popović (2004).

Related WIREs Articles

Projection-based techniques for high-dimensional optimal transport problems

References

Alaya, M. Z., M. Berar, G. Gasso, and A. Rakotomamonjy. 2019. "Screening Sinkhorn Algorithm for Regularized Optimal Transport." *Advances in Neural Information Processing Systems* 32: 12191–12201.

Altschuler, J., F. Bach, A. Rudi, and J. Niles-Weed. 2019. "Massively Scalable Sinkhorn Distances via the Nyström Method." *Advances in Neural Information Processing Systems* 32: 4427–4437.

Altschuler, J., J. Niles-Weed, and P. Rigollet. 2017. "Near-Linear Time Approximation Algorithms for Optimal Transport via Sinkhorn Iteration." *Advances in Neural Information Processing Systems* 30: 1965–1975.

Andoni, A., P. Indyk, T. Laarhoven, I. Razenshteyn, and L. Schmidt. 2015. "Practical and Optimal LSH for Angular Distance." *Advances in Neural Information Processing Systems* 28: 1225–1233.

Arjovsky, M., S. Chintala, and L. Bottou. 2017. "Wasserstein Generative Adversarial Networks." In *International Conference on Machine Learning*, 214–223. PMLR.

Balikas, G., C. Laclau, I. Redko, and M.-R. Amini. 2018. "Cross-Lingual Document Retrieval Using Regularized Wasserstein Distance." In *European Conference on Information Retrieval*, 398–410. Springer.

Beier, F., J. von Lindheim, S. Neumayer, and G. Steidl. 2023. "Unbalanced Multi-Marginal Optimal Transport." *Journal of Mathematical Imaging and Vision* 65, no. 3: 394–413.

Bernton, E., P. E. Jacob, M. Gerber, and C. P. Robert. 2019. "On Parameter Estimation With the Wasserstein Distance." *Information and Inference: A Journal of the IMA* 8, no. 4: 657–676.

Blanchet, J., K. Murthy, and V. A. Nguyen. 2021. "Statistical Analysis of Wasserstein Distributionally Robust Estimators." In *Tutorials in Operations Research: Emerging Optimization Methods and Modeling Techniques With Applications*, 227–254. INFORMS.

Blondel, M., V. Seguy, and A. Rolet. 2018. "Smooth and Sparse Optimal Transport." In *International Conference on Artificial Intelligence and Statistics*, 880–889. PMLR.

Brauer, C., C. Clason, D. Lorenz, and B. Wirth. 2017. "A Sinkhorn-Newton Method for Entropic Optimal Transport." arXiv Preprint arXiv:1710.06635.

Braverman, V., R. Krauthgamer, A. R. Krishnan, and S. Sapir. 2021. "Near-Optimal Entrywise Sampling of Numerically Sparse Matrices." In *Conference on Learning Theory*, 759–773. PMLR.

Brenier, Y. 1991. "Polar Factorization and Monotone Rearrangement of Vector-Valued Functions." *Communications on Pure and Applied Mathematics* 44, no. 4: 375–417.

Carlier, G. 2022. "On the Linear Convergence of the Multimarginal Sinkhorn Algorithm." *SIAM Journal on Optimization* 32, no. 2: 786–794.

Chizat, L., G. Peyré, B. Schmitzer, and F.-X. Vialard. 2018. "Scaling Algorithms for Unbalanced Optimal Transport Problems." *Mathematics of Computation* 87, no. 314: 2563–2609.

Chowdhury, S., D. Miller, and T. Needham. 2021. "Quantized Gromov-Wasserstein." In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 811–827. Springer.

Claici, S., E. Chien, and J. Solomon. 2018. "Stochastic Wasserstein Barycenters." In *International Conference on Machine Learning*, 999–1008. PMLR.

Courty, N., R. Flamary, and D. Tuia. 2014. "Domain Adaptation With Regularized Optimal Transport." In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 274–289. Springer.

Courty, N., R. Flamary, D. Tuia, and A. Rakotomamonjy. 2016. "Optimal Transport for Domain Adaptation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 9: 1853–1865.

Cuturi, M. 2013. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport." *Advances in Neural Information Processing Systems* 26: 2292.

Cuturi, M., and A. Doucet. 2014. "Fast Computation of Wasserstein Barycenters." In *International Conference on Machine Learning*, 685–693. PMLR.

Cuturi, M., and G. Peyré. 2018. "Semidual Regularized Optimal Transport." *SIAM Review* 60, no. 4: 941–965.

Damodaran, B. B., B. Kellenberger, R. Flamary, D. Tuia, and N. Courty. 2018. "Deepjdot: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 447–463. Springer.

Daniels, M., T. Maunu, and P. Hand. 2021. "Score-Based Generative Neural Networks for Large-Scale Optimal Transport." *Advances in Neural Information Processing Systems* 34: 12955–12965.

Dembo, R. S., S. C. Eisenstat, and T. Steihaug. 1982. "Inexact Newton Methods." *SIAM Journal on Numerical Analysis* 19, no. 2: 400–408.

Dennis, J. E., Jr., and J. J. Moré. 1977. "Quasi-Newton Methods, Motivation and Theory." *SIAM Review* 19, no. 1: 46–89.

Deshpande, I., Y.-T. Hu, R. Sun, et al. 2019. "Max-Sliced Wasserstein Distance and Its Use for Gans." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10648–10656. IEEE.

Drineas, P., R. Kannan, and M. W. Mahoney. 2006. "Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication." *SIAM Journal on Computing* 36, no. 1: 132–157.

Dvurechensky, P., A. Gasnikov, and A. Kroshnin. 2018. "Computational Optimal Transport: Complexity by Accelerated Gradient Descent is Better Than by Sinkhorn's Algorithm." In *International Conference on Machine Learning*, 1367–1376. PMLR.

Eisenberger, M., A. Toker, L. Leal-Taixé, F. Bernard, and D. Cremers. 2022. "A Unified Framework for Implicit Sinkhorn Differentiation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 509–518. IEEE.

Essid, M., and J. Solomon. 2018. "Quadratically Regularized Optimal Transport on Graphs." *SIAM Journal on Scientific Computing* 40, no. 4: A1961–A1986.

Fang, L., E. Latif, H. Lu, Y. Zhou, P. Ma, and X. Zhai. 2025. "Efficient Multi-Task Inferencing: Model Merging With Gromov-Wasserstein Feature Alignment." In *International Conference on Artificial Intelligence in Education*, 192–200. Springer.

Farnia, F., A. Reisizadeh, R. Pedarsani, and A. Jadbabaie. 2022. "An Optimal Transport Approach to Personalized Federated Learning." *IEEE Journal on Selected Areas in Information Theory* 3, no. 2: 162–171.

Fatras, K., T. Séjourné, R. Flamary, and N. Courty. 2021. "Unbalanced Minibatch Optimal Transport; Applications to Domain Adaptation." In *International Conference on Machine Learning*, 3186–3197. PMLR.

Fatras, K., Y. Zine, R. Flamary, R. Gribonval, and N. Courty. 2020. "Learning With Minibatch Wasserstein: Asymptotic and Gradient Properties." In *AISTATS 2020-23nd International Conference on Artificial Intelligence and Statistics*, vol. 108, 1–20. PMLR.

Fatras, K., Y. Zine, S. Majewski, R. Flamary, R. Gribonval, and N. Courty. 2021. "Minibatch Optimal Transport Distances; Analysis and Applications." arXiv Preprint arXiv:2101.01792.

Flamary, R., N. Courty, D. Tuia, and A. Rakotomamonjy. 2016. "Optimal Transport for Domain Adaptation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1, no. 2: 1–40.

Fournier, N., and A. Guillin. 2015. "On the Rate of Convergence in Wasserstein Distance of the Empirical Measure." *Probability Theory and Related Fields* 162, no. 3: 707–738.

Franklin, J., and J. Lorenz. 1989. "On the Scaling of Multidimensional Matrices." *Linear Algebra and Its Applications* 114: 717–735.

Gasteiger, J., M. Lienen, and S. Günnemann. 2021. "Scalable Optimal Transport in High Dimensions for Graph Distances, Embedding Alignment, and More." In *International Conference on Machine Learning*, 5616–5627. PMLR.

Genevay, A., M. Cuturi, G. Peyré, and F. Bach. 2016. "Stochastic Optimization for Large-Scale Optimal Transport." *Advances in Neural Information Processing Systems* 29: 3440–3448.

Golub, G. H., and C. F. Van Loan. 2013. *Matrix Computations*. JHU Press.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, et al. 2014. "Generative Adversarial Nets." *Advances in Neural Information Processing Systems* 27: 2672–2680.

Grave, E., A. Joulin, and Q. Berthet. 2019. "Unsupervised Alignment of Embeddings With Wasserstein Procrustes." In *The 22nd International Conference on Artificial Intelligence and Statistics*, 1880–1890. PMLR.

Gulrajani, I., F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. 2017. "Improved Training of Wasserstein GANs." *Advances in Neural Information Processing Systems* 30: 5769–5779.

Guminov, S., P. Dvurechensky, N. Tupitsa, and A. Gasnikov. 2021. "On a Combination of Alternating Minimization and Nesterov's Momentum." In *International Conference on Machine Learning*, 3886–3898. PMLR.

Gupta, N., and A. Sidford. 2018. "Exploiting Numerical Sparsity for Efficient Learning: Faster Eigenvector Computation and Regression." *Advances in Neural Information Processing Systems* 31: 5274–5283.

Haasler, I., R. Singh, Q. Zhang, J. Karlsson, and Y. Chen. 2021. "Multi-Marginal Optimal Transport and Probabilistic Graphical Models." *IEEE Transactions on Information Theory* 67, no. 7: 4647–4668.

Horowitz, J., and R. L. Karandikar. 1994. "Mean Rates of Convergence of Empirical Measures in the Wasserstein Metric." *Journal of Computational and Applied Mathematics* 55, no. 3: 261–273.

Hu, Y., M. Li, X. Liu, and C. Meng. 2025. "Sampling-Based Methods for Multi-Block Optimization Problems Over Transport Polytopes." *Mathematics of Computation* 94, no. 353: 1281–1322.

Kahn, H., and A. W. Marshall. 1953. "Methods of Reducing Sample Size in Monte Carlo Computations." *Journal of the Operations Research Society of America* 1, no. 5: 263–278.

Kantorovich, L. 1942. "On Translation of Mass." *Comptes Rendus Proceedings of the USSR Academy of Sciences* 37: 199–201.

Kawano, S., and J. K. Mason. 2021. "Classification of Atomic Environments via the Gromov-Wasserstein Distance." *Computational Materials Science* 188: 110144.

Klatt, M., C. Tameling, and A. Munk. 2020. "Empirical Regularized Optimal Transport: Statistical Theory and Applications." *SIAM Journal on Mathematics of Data Science* 2, no. 2: 419–443.

Kroshnin, A., V. Spokoiny, and A. Suvorikova. 2021. "Statistical Inference for Bures-Wasserstein Barycenters." *Annals of Applied Probability* 31, no. 3: 1264–1298.

Lei, N., K. Su, L. Cui, S.-T. Yau, and X. D. Gu. 2019. "A Geometric View of Optimal Transportation and Generative Model." *Computer Aided Geometric Design* 68: 1–21.

Li, D.-H., M. Fukushima, L. Qi, and N. Yamashita. 2004. "Regularized Newton Methods for Convex Minimization Problems With Singular Solutions." *Computational Optimization and Applications* 28, no. 2: 131–147.

Li, H., W. Huang, J. Wang, and Y. Shi. 2024. "Global and Local Prompts Cooperation via Optimal Transport for Federated Learning." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12151–12161. IEEE.

Li, M., J. Yu, T. Li, and C. Meng. 2023. "Importance Sparsification for Sinkhorn Algorithm." *Journal of Machine Learning Research* 24, no. 247: 1–44.

Li, M., J. Yu, H. Xu, and C. Meng. 2023. "Efficient Approximation of Gromov-Wasserstein Distance Using Importance Sparsification." *Journal of Computational and Graphical Statistics* 32, no. 4: 1512–1523.

Lin, T., N. Ho, and M. I. Jordan. 2022. "On the Efficiency of Entropic Regularized Algorithms for Optimal Transport." *Journal of Machine Learning Research* 23, no. 137: 1–42.

Liu, D. C., and J. Nocedal. 1989. "On the Limited Memory BFGS Method for Large Scale Optimization." *Mathematical Programming* 45, no. 1: 503–528.

Liu, J. S., and J. S. Liu. 2001. *Monte Carlo Strategies in Scientific Computing*. Vol. 10. Springer.

Liutkus, A., U. Simsekli, S. Majewski, A. Durmus, and F.-R. Stöter. 2019. "Sliced-Wasserstein Flows: Nonparametric Generative Modeling via Optimal Transport and Diffusions." In *International Conference on Machine Learning*, 4104–4113. PMLR.

Lorenz, D. A., P. Manns, and C. Meyer. 2021. "Quadratically Regularized Optimal Transport." *Applied Mathematics & Optimization* 83, no. 3: 1919–1949.

Luo, D., Y. Wang, A. Yue, and H. Xu. 2022. "Weakly-Supervised Temporal Action Alignment Driven by Unbalanced Spectral Fused Gromov-Wasserstein Distance." In *Proceedings of the 30th ACM International Conference on Multimedia*, 728–739. ACM.

Luo, D., H. Xu, and L. Carin. 2022. "Differentiable Hierarchical Optimal Transport for Robust Multi-View Learning." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, no. 6: 7293–7307.

Mahoney, M. W. 2011. "Randomized Algorithms for Matrices and Data." *Foundations and Trends in Machine Learning* 3, no. 2: 123–224.

Mémoli, F. 2011. "Gromov-Wasserstein Distances and the Metric Approach to Object Matching." *Foundations of Computational Mathematics* 11, no. 4: 417–487.

Meng, C., J. Yu, J. Zhang, P. Ma, and W. Zhong. 2020. "Sufficient Dimension Reduction for Classification Using Principal Optimal Transport Direction." *Advances in Neural Information Processing Systems* 33: 4015–4028.

Mokrov, P., A. Korotin, L. Li, A. Genevay, J. M. Solomon, and E. Burnaev. 2021. "Large-Scale Wasserstein Gradient Flows." *Advances in Neural Information Processing Systems* 34: 15243–15256.

Monge, G. 1781. "Mémoire Sur La Théorie Des Déblais Et Des Remblais." *Histoire de L'academie Royale Des Sciences de Paris*: 666–704.

Montavon, G., K.-R. Müller, and M. Cuturi. 2016. "Wasserstein Training of Restricted Boltzmann Machines." *Advances in Neural Information Processing Systems* 29: 3718–3726.

Musco, C., and C. Musco. 2017. "Recursive Sampling for the Nyström Method." *Advances in Neural Information Processing Systems* 30: 3836–3848.

Muzellec, B., and M. Cuturi. 2019. "Subspace Detours: Building Transport Plans That Are Optimal on Subspace Projections." *Advances in Neural Information Processing Systems* 32: 6917–6928.

Nguyen, Q. M., H. H. Nguyen, Y. Zhou, and L. M. Nguyen. 2023. "On Unbalanced Optimal Transport: Gradient Methods, Sparsity and Approximation Error." *Journal of Machine Learning Research* 24, no. 384: 1–41.

Nguyen, T. T., and A. T. Luu. 2022. "Improving Neural Cross-Lingual Abstractive Summarization via Employing Optimal Transport Distance for Knowledge Distillation." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 11103–11111. AAAI Press.

Oglic, D., and T. Gärtner. 2017. "Nyström Method With Kernel k-Means++ Samples as Landmarks." In *International Conference on Machine Learning*, 2652–2660. PMLR.

Owen, A. B. 2013. "Monte Carlo Theory, Methods and Examples." https://artowen.su.domains/mc/.

Paulevé, L., H. Jégou, and L. Amsaleg. 2010. "Locality Sensitive Hashing: A Comparison of Hash Function Types and Querying Mechanisms." *Pattern Recognition Letters* 31, no. 11: 1348–1358.

Pele, O., and M. Werman. 2009. "Fast and Robust Earth Mover's Distances." In *2009 IEEE 12th International Conference on Computer Vision*, 460–467. IEEE.

Petric Maretic, H., M. El Gheche, G. Chierchia, and P. Frossard. 2019. "Got: An Optimal Transport Framework for Graph Comparison." *Advances in Neural Information Processing Systems* 32: 13899–13910.

Peyré, G. 2015. "Entropic Approximation of Wasserstein Gradient Flows." *SIAM Journal on Imaging Sciences* 8, no. 4: 2323–2351.

Peyré, G., and M. Cuturi. 2019. "Computational Optimal Transport: With Applications to Data Science." *Foundations and Trends in Machine Learning* 11, no. 5–6: 355–607.

Peyré, G., M. Cuturi, and J. Solomon. 2016. "Gromov-Wasserstein Averaging of Kernel and Distance Matrices." In *International Conference on Machine Learning*, 2664–2672. PMLR.

Pham, K., K. Le, N. Ho, T. Pham, and H. Bui. 2020. "On Unbalanced Optimal Transport: An Analysis of Sinkhorn Algorithm." In *International Conference on Machine Learning*, 7673–7682. PMLR.

Rabin, J., G. Peyré, J. Delon, and M. Bernot. 2011. "Wasserstein Barycenter and Its Application to Texture Mixing." In *International Conference on Scale Space and Variational Methods in Computer Vision*, 435–446. Springer.

Ramdas, A., N. García Trillos, and M. Cuturi. 2017. "On Wasserstein Two-Sample Testing and Related Families of Nonparametric Tests." *Entropy* 19, no. 2: 47.

Rolet, A., M. Cuturi, and G. Peyré. 2016. "Fast Dictionary Learning With a Smoothed Wasserstein Loss." In *Artificial Intelligence and Statistics*, 630–638. PMLR.

Rubner, Y., L. J. Guibas, and C. Tomasi. 1997. "The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image

Retrieval." In *Proceedings of the ARPA Image Understanding Workshop*, vol. 661, 668. Springer.

Scetbon, M., G. Peyré, and M. Cuturi. 2022. "Linear-Time Gromov-Wasserstein Distances Using Low Rank Couplings and Costs." In *International Conference on Machine Learning*, 19347–19365. PMLR.

Séjourné, T., F.-X. Vialard, and G. Peyré. 2021. "The Unbalanced Gromov-Wasserstein Distance: Conic Formulation and Relaxation." *Advances in Neural Information Processing Systems* 34: 8766–8779.

Shrivastava, A., and P. Li. 2014. "Asymmetric LSH (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS)." *Advances in Neural Information Processing Systems* 27: 2321–2329.

Si, N., J. Blanchet, S. Ghosh, and M. Squillante. 2020. "Quantifying the Empirical Wasserstein Distance to a Set of Measures: Beating the Curse of Dimensionality." *Advances in Neural Information Processing Systems* 33: 21260–21270.

Sinkhorn, R. 1964. "A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices." *Annals of Mathematical Statistics* 35, no. 2: 876–879.

Sinkhorn, R., and P. Knopp. 1967. "Concerning Nonnegative Matrices and Doubly Stochastic Matrices." *Pacific Journal of Mathematics* 21, no. 2: 343–348.

Solomon, J., G. Peyré, V. G. Kim, and S. Sra. 2016. "Entropic Metric Alignment for Correspondence Problems." *ACM Transactions on Graphics* 35, no. 4: 1–13.

Sommerfeld, M., J. Schrieber, Y. Zemel, and A. Munk. 2019. "Optimal Transport: Fast Probabilistic Approximation With Exact Solvers." *Journal of Machine Learning Research* 20, no. 105: 1–23.

Sumner, R. W., and J. Popović. 2004. "Deformation Transfer for Triangle Meshes." *ACM Transactions on Graphics* 23, no. 3: 399–405.

Tameling, C., and A. Munk. 2018. "Computational Strategies for Statistical Inference Based on Empirical Optimal Transport." In *2018 IEEE Data Science Workshop (DSW)*, 175–179. IEEE.

Tang, X., M. Shavlovsky, H. Rahmanian, et al. 2024. "Accelerating Sinkhorn Algorithm With Sparse Newton Iterations." arXiv Preprint arXiv:2401.12253.

Tang, Z., and Y. Qiu. 2024. "Safe and Sparse Newton Method for Entropic-Regularized Optimal Transport." *Advances in Neural Information Processing Systems* 37: 129914–129943.

Thibault, A., L. Chizat, C. Dossal, and N. Papadakis. 2021. "Overrelaxed Sinkhorn-Knopp Algorithm for Regularized Optimal Transport." *Algorithms* 14, no. 5: 143.

Titouan, V., N. Courty, R. Tavenard, and R. Flamary. 2019. "Optimal Transport for Structured Data With Application on Graphs." In *International Conference on Machine Learning*, 6275–6284. PMLR.

Titouan, V., R. Flamary, N. Courty, R. Tavenard, and L. Chapel. 2019. "Sliced Gromov-Wasserstein." *Advances in Neural Information Processing Systems* 32: 14753–14763.

Tolstikhin, I., O. Bousquet, S. Gelly, and B. Schoelkopf. 2018. "Wasserstein Auto-Encoders." In *International Conference on Learning Representations*. IEEE.

Tong, A., K. Fatras, N. Malkin, et al. 2024. "Improving and Generalizing Flow-Based Generative Models With Minibatch Optimal Transport." *Transactions on Machine Learning Research*: 1–34.

Tran, K., K. Nguyen, A. Nguyen, et al. 2025. "Sparse Partial Optimal Transport via Quadratic Regularization." *Journal of Computer Science* 21: 2508.08476.

Vayer, T., L. Chapel, R. Flamary, R. Tavenard, and N. Courty. 2020. "Fused Gromov-Wasserstein Distance for Structured Objects." *Algorithms* 13, no. 9: 212.

Villani, C. 2008. *Optimal Transport: Old and New, Volume 338*. Springer Science & Business Media.

Vincent-Cuaz, C., R. Flamary, M. Corneli, T. Vayer, and N. Courty. 2022. "Semi-Relaxed Gromov-Wasserstein Divergence With Applications on Graphs." In *ICLR 2022-10th International Conference on Learning Representations*, 1–28. IEEE.

Wang, C., and Y. Qiu. 2025. "The Sparse-Plus-Low-Rank Quasi-Newton Method for Entropic-Regularized Optimal Transport." In *Proceedings of the 42nd International Conference on Machine Learning*. PMLR.

Wang, T., M. Li, G. Zeng, C. Meng, and Q. Zhang. 2025. "Gaussian Herding Across Pens: An Optimal Transport Perspective on Global Gaussian Reduction for 3DGS." *Advances in Neural Information Processing Systems* 39: 1–13.

Wang, W., H. Xu, G. Wang, W. Wang, and L. Carin. 2021. "Zero-Shot Recognition via Optimal Transport." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3471–3481. IEEE.

Wang, Y., H. Xu, and D. Luo. 2023. "Self-Supervised Video Summarization Guided by Semantic Inverse Optimal Transport." In *Proceedings of the 31st ACM International Conference on Multimedia*, 6611–6622. ACM.

Wang, Z., D. Zhou, M. Yang, Y. Zhang, C. Rao, and H. Wu. 2020. "Robust Document Distance With Wasserstein-Fisher-Rao Metric." In *Asian Conference on Machine Learning*, 721–736. PMLR.

Weed, J., and F. Bach. 2019. "Sharp Asymptotic and Finite-Sample Rates of Convergence of Empirical Measures in Wasserstein Distance." *Bernoulli* 25, no. 4A: 2620–2648.

Werner, M., and E. Laber. 2020. "Speeding Up Word Mover's Distance and Its Variants via Properties of Distances Between Embeddings." In *Proceedings of the 2020th European Conference on Artificial Intelligence*, 2204–2211. IOS Press.

Williams, C., and M. Seeger. 2000. "Using the Nyström Method to Speed Up Kernel Machines." *Advances in Neural Information Processing Systems* 13: 682–688.

Xu, H., and M. Cheng. 2023. "Regularized Optimal Transport Layers for Generalized Global Pooling Operations." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, no. 12: 15426–15444.

Xu, H., D. Luo, and L. Carin. 2019. "Scalable Gromov-Wasserstein Learning for Graph Partitioning and Matching." *Advances in Neural Information Processing Systems* 32: 3052–3062.

Xu, H., D. Luo, L. Carin, and H. Zha. 2021. "Learning Graphons via Structured Gromov-Wasserstein Barycenters." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 10505–10513. AAAI Press.

Xu, H., W. Wang, W. Liu, and L. Carin. 2018. "Distilled Wasserstein Learning for Word Embedding and Topic Modeling." *Advances in Neural Information Processing Systems* 31: 1723–1732.

Yang, W., J. Yang, and Y. Liu. 2023. "Multimodal Optimal Transport Knowledge Distillation for Cross-Domain Recommendation." In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2959–2968. ACM.

Yu, W., Z. Sun, J. Xu, et al. 2022. "Explainable Legal Case Matching via Inverse Optimal Transport-Based Rationale Extraction." In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 657–668. ACM.

Yule, G. U. 1912. "On the Methods of Measuring Association Between Two Attributes." *Journal of the Royal Statistical Society* 75, no. 6: 579–652.

Yurochkin, M., S. Claici, E. Chien, F. Mirzazadeh, and J. M. Solomon. 2019. "Hierarchical Optimal Transport for Document Representation." *Advances in Neural Information Processing Systems* 32: 1601–1611.

Zehlike, M., P. Hacker, and E. Wiedemann. 2020. “Matching Code and Law: Achieving Algorithmic Fairness With Optimal Transport.” *Data Mining and Knowledge Discovery* 34, no. 1: 163–200.

Zemel, Y., and V. M. Panaretos. 2019. “Fréchet Means and Procrustes Analysis in Wasserstein Space.” *Bernoulli* 25, no. 2: 932–976.

Zhang, J., P. Ma, W. Zhong, and C. Meng. 2023. “Projection-Based Techniques for High-Dimensional Optimal Transport Problems.” *WIREs Computational Statistics* 15, no. 2: e1587.

Zhang, J., C. Meng, J. Yu, M. Zhang, W. Zhong, and P. Ma. 2023. “An Optimal Transport Approach for Selecting a Representative Subsample With Application in Efficient Kernel Density Estimation.” *Journal of Computational and Graphical Statistics* 32, no. 1: 329–339.

Zhang, J., W. Zhong, and P. Ma. 2021. “A Review on Modern Computational Optimal Transport Methods With Applications in Biomedical Research.” In *Modern Statistical Methods for Health Research*, 279–300. Springer International Publishing.

Zhao, P., and T. Zhang. 2015. “Stochastic Optimization With Importance Sampling for Regularized Loss Minimization.” In *International Conference on Machine Learning*, 1–9. PMLR.