entropy

MDPI

Article

Efficient Discretization of Optimal Transport

Junqi Wang [1], Pei Wang [1] and Patrick Shafto [1,2,*]

[1] Department of Math & CS, Rutgers University, Newark, NJ 07102, USA; junqi.wang@rutgers.edu (J.W.); peiwang@rutgers.edu (P.W.)
[2] School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540, USA
* Correspondence: patrick.shafto@rutgers.edu or pshafto@ias.edu

Abstract: Obtaining solutions to optimal transportation (OT) problems is typically intractable when marginal spaces are continuous. Recent research has focused on approximating continuous solutions with discretization methods based on i.i.d. sampling, and this has shown convergence as the sample size increases. However, obtaining OT solutions with large sample sizes requires intensive computation effort, which can be prohibitive in practice. In this paper, we propose an algorithm for calculating discretizations with a given number of weighted points for marginal distributions by minimizing the (entropy-regularized) Wasserstein distance and providing bounds on the performance. The results suggest that our plans are comparable to those obtained with much larger numbers of i.i.d. samples and are more efficient than existing alternatives. Moreover, we propose a local, parallelizable version of such discretizations for applications, which we demonstrate by approximating adorable images.

Keywords: optimal transport; entropy regularization; discretization; gradient descent

Citation: Wang, J.; Wang, P.; Shafto, P. Efficient Discretization of Optimal Transport. *Entropy* **2023**, *25*, 839. https://doi.org/10.3390/e25060839

Academic Editors: Udo Von Toussaint and Nikolai Leonenko

Received: 15 November 2022
Revised: 21 April 2023
Accepted: 25 April 2023
Published: 24 May 2023

1. Introduction

Optimal transport is the problem of finding a coupling of probability distributions that minimizes cost [1], and it is a technique applied across various fields and literatures [2,3]. Although many methods exist for obtaining optimal transference plans for distributions on discrete spaces, computing the plans is not generally possible for continuous spaces [4]. Given the prevalence of continuous spaces in machine learning, this is a significant limitation for theoretical and practical applications.

One strategy for approximating continuous OT plans is based on discrete approximation via sample points. Recent research has provided guarantees on the fidelity of discrete, sample-location-based approximations for continuous OT as the sample size $N \to \infty$ [5]. Specifically, by sampling large numbers of points S_i from each marginal, one may compute a discrete optimal transference plan on $S_1 \times S_2$, with the cost matrix being derived from the pointwise evaluation of the cost function on $S_1 \times S_2$.

Even in the discrete case, obtaining minimal cost plans is computationally challenging. For example, Sinkhorn scaling, which computes an entropy-regularized approximation for OT plans, has a complexity that scales with $|S_1 \times S_2|$ [6]. Although many comparable methods exist [7], all of them have a complexity that scales with the product of sample sizes, and they require the construction of a cost matrix that also scales with $|S_1 \times S_2|$.

We have developed methods for optimizing both sampling locations and weights for small N approximations of OT plans (see Figure 1). In Section 2, we formulate the problem of fixed size approximation and reduce it to discretization problems on marginals with theoretical guarantees. In Section 3, the gradient of entropy-regularized Wasserstein distance between a continuous distribution and its discretization is derived. In Section 4, we present a stochastic gradient descent algorithm that is based on the optimization of the locations and weights of the points with empirical demonstrations. Section 5 introduces a parellelizable algorithm via decompositions of the marginal spaces, which reduce the computational complexity by exploiting intrinsic geometry. In Section 6, we analyze time

and space complexity. In Section 7, we illustrate the advantage of including weights for sample points by providing a comparison with an existing location that is only based on discretization.

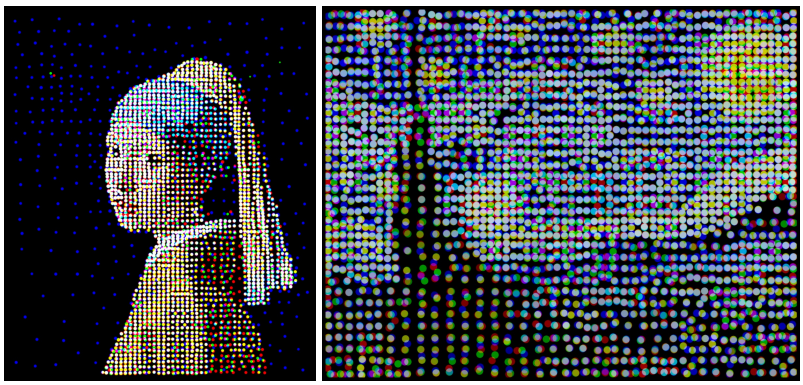

Figure 1. Discretization of "Girl with a Pearl Earring" and "Starry Night" using EDOT with 2000 discretization points for each RGB channel. $k = 2$, $\zeta = 0.01 \times \text{diam}^2$.

2. Efficient Discretizations

Optimal transport (OT): Let (X, d_X), (Y, d_Y) be compact Polish spaces (complete separable metric spaces), $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ be probability distributions on their Borel-algebras, and $c : X \times Y \to \mathbb{R}$ be a cost function. Denote the set of all joint probability measures (couplings) on $X \times Y$ with marginals μ and ν by $\Pi(\mu, \nu)$. For the cost function c, the optimal transference plan between μ and ν is defined as in [1]: $\gamma(\mu, \nu) := \text{argmin}_{\pi \in \Pi(\mu,\nu)} \langle c, \pi \rangle$, where $\langle c, \pi \rangle := \int_{X \times Y} c(x, y) \mathrm{d}\pi(x, y)$.

When $X = Y$, the cost $c(x, y) = d_X^k(x, y)$, $W_k(\mu, \nu) = \langle c, \gamma(\mu, \nu) \rangle^{1/k}$ defines the k-Wasserstein distance between μ and ν for $k \geq 1$. Here, $d_X^k(x, y)$ is the k-th power of the metric d_X on X.

Entropy regularized optimal transport (EOT) [5,8] was introduced to estimate OT couplings with reduced computational complexity: $\gamma_\lambda(\mu, \nu) := \text{argmin}_{\pi \in \Pi(\mu,\nu)} \langle c, \pi \rangle + \lambda \mathrm{KL}(\pi || \mu \otimes \nu)$, where $\lambda > 0$ is a regularization parameter, and the regularization term $\mathrm{KL}(\pi || \mu \otimes \nu) := \int \log(\frac{\mathrm{d}\pi}{\mathrm{d}\mu \otimes \mathrm{d}\nu}) \mathrm{d}\pi$ is the Kullback–Leibler divergence. The EOT objective is smooth and convex, and its unique solution with a given discrete (μ, ν, c) can be obtained using a Sinkhorn iteration (SK) [9].

However, for large-scale discrete spaces, the computational cost of SK can still be unfeasible [6]. Even worse, to even apply the Sinkhorn iteration, one must know the entire cost matrix over the large-scale spaces, which itself can be a non-trivial computational burden to obtain; in some cases, for example, where the cost is derived from a probability model [10], it may require intractable computations [11,12].

The Framework: We propose the optimization of the location and weights of a fixed size discretization to estimate the continuous OT. The discretization on $X \times Y$ is completely determined by those on X and Y to respect the marginal structure in the OT. Let $m, n \in \mathbb{Z}^*$, $\mu_m \in \mathcal{P}(X)$, $\nu_n \in \mathcal{P}(Y)$ be a discrete approximation of μ and ν, respectively, with $\mu_m = \sum_{i=1}^m w_i \delta_{x_i}$, $\nu_n = \sum_{j=1}^n u_j \delta_{y_j}$, $x_i \in X$, $y_j \in Y$, and $w_i, u_j \in \mathbb{R}^+$. Then, the EOT plan $\gamma_\lambda(\mu, \nu) \in \Pi(\mu, \nu)$ for the OT problem (μ, ν, c) can be approximated by the EOT plan $\gamma_\lambda(\mu_m, \nu_n) \in \Pi(\mu_m, \nu_n)$ for the OT problem (μ_m, ν_n, c). There are three distributions that have their discrete counterparts; thus, with a fixed size $m, n \in \mathbb{Z}^*$, a naive idea about the objective to be optimized can be

$$\Omega_{k,\rho}(\mu_m, \nu_n) = W_k^k(\mu, \mu_m) + W_k^k(\nu, \nu_n) + \rho W_k^k(\gamma_\lambda(\mu, \nu), \gamma_\lambda(\mu_m, \nu_n)), \tag{1}$$

where $W_k^k(\phi, \psi)$ represents the k-th power of k-Wasserstein distance between measures ϕ and ψ. The hyperparameter $\rho > 0$ balances between the estimation accuracy over marginals and that of the transference plan, while the weights on marginals are equal.

To properly compute $W_k^k(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n))$, a metric $d_{X\times Y}$ on $X\times Y$ is needed. We expect $d_{X\times Y}$ on X-slices or Y-slices to be compatible with d_X or d_Y, respectively; furthermore, we may assume that there exists a constant $A>0$ such that:

$$\max\{d_X^k(x_1,x_2),d_Y^k(y_1,y_2)\}\le d_{X\times Y}^k((x_1,y_1),(x_2,y_2))\le A(d_X^k(x_1,x_2)+d_Y^k(y_1,y_2)). \quad (2)$$

For instance, (2) holds when $d_{X\times Y}$ is the p-product metric for $1\le p\le\infty$.

The objective $\Omega_{k,\rho}(\mu_m,\nu_n)$ is estimated by its entropy regularized approximation $\Omega_{k,\zeta,\rho}(\mu_m,\nu_n)$ for efficient computation, where ζ is the regularization parameter, as follows:

$$\Omega_{k,\zeta,\rho}(\mu_m,\nu_n)=W_{k,\zeta}^k(\mu,\mu_m)+W_{k,\zeta}^k(\nu,\nu_n)+\rho W_{k,\zeta}^k(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n)). \quad (3)$$

Here, $W_k^k(\mu,\mu_m)=\langle d_X^k,\gamma(\mu,\mu_m)\rangle^{1/k}$ is estimated by $W_{k,\zeta}^k(\mu,\mu_m)=\langle d_X^k,\gamma_\zeta(\mu,\mu_m)\rangle^{1/k}$. $\gamma_\zeta(\mu,\mu_m)$ is computed by optimizing $\widehat{W}_{k,\zeta}^k(\mu,\mu_m)=\langle d_X^k,\gamma_\zeta(\mu,\mu_m)\rangle+\lambda\mathrm{KL}(\gamma_\zeta(\mu,\mu_m)||\mu\otimes\mu_m)$.

One major difficulty in optimizing $\Omega_{k,\zeta,\rho}(\mu_m,\nu_n)$ is to evaluate $W_{k,\zeta}^k(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n))$. In fact, obtaining $\gamma_\lambda(\mu,\nu)$ is intractable, which is the original motivation for the discretization. To overcome this drawback, by utilizing the dual formulation of EOT, the following are shown (see proof in Appendix A):

Proposition 1. *When X and Y are two compact spaces, and the cost function c is $\mathcal{C}^\infty$, there exists a constant $C_1\in\mathbb{R}^+$ such that*

$$\max\{W_k^k(\mu,\mu_m),W_k^k(\nu,\nu_n)\}\le W_{k,\zeta}^k(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n))\le C_1[W_{k,\zeta}^k(\mu,\mu_m)+W_{k,\zeta}^k(\nu,\nu_n)].$$

Notice that Proposition 1 indicates that $W_{k,\zeta}^k(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n))$ is bounded above by multiples of $W_{k,\zeta}^k(\mu,\mu_m)+W_{k,\zeta}^k(\nu,\nu_n)$, i.e., when the continuous marginals μ and ν are properly approximated, so is the optimal transference plan between them. Therefore, to optimize $\Omega_{k,\zeta,\rho}(\mu_m,\nu_n)$, we focus on developing algorithms to obtain μ_m^*,ν_n^* that minimize $W_{k,\zeta}^k(\mu,\mu_m)$ and $W_{k,\zeta}^k(\nu,\nu_n)$.

Remark 1. *The regularizing parameters (λ and ζ above) introduce smoothness, together with an error term, into the OT problem. To make an accurate approximation, we need λ and ζ to be as small as possible. However, when parameters become too small, the matrices to be normalized in the Sinkhorn algorithm lead to an overflow or underflow problem of numerical data types (32-bit or 64-bit floating point numbers). Thus, the value for regularizing the constant threshold is proportional to the k-th power of the diameter of the supported region. In this work, we try our best to control the value (mainly on ζ), which ranges from 10^{-4} to 0.01 when the diameter is 1 in different examples.*

3. Gradient of the Objective Function

Let $\nu=\sum_{i=1}^m w_i\delta_{y_i}$ be a discrete probability measure in the position of "μ_m" in the last section. For a fixed (continuous) μ, the objective now is to obtain a discrete target $\nu^*=\operatorname{argmin} W_{k,\zeta}^k(\mu,\nu)$.

In order to apply a stochastic gradient descent (SGD) to both the positions $\{y_i\}_{i=1}^m$ and their weights $\{w_i\}_{i=1}^m$ to achieve ν^*, we now derive the gradient of $W_{k,\zeta}^k(\mu,\nu)$ about ν by following the discrete discussions of [13,14]. The SGD on X is either derived through an exponential map, or by treating X as (part of) an Euclidean space.

Let $g(x,y):=d_X^k(x,y)$, and denote the joint distribution minimizing $\widehat{W}_{k,\zeta}^k$ as π with the differential form at (x,y_i) being $\mathrm{d}\pi_i(x)$, which is used to define $W_{k,\zeta}^k$ in Section 2.

By introducing the Lagrange multipliers $\alpha\in L^\infty(X),\beta\in\mathbb{R}^m$i, we have $\widehat{W}_{k,\zeta}^k(\mu,\nu)=\max_{\alpha,\beta}\mathcal{L}(\mu,\nu;\alpha,\beta)$,

where $\mathcal{L}(\mu,\nu;\alpha,\beta) = \int_X \alpha(x)\mathrm{d}\mu(x) + \sum_{i=1}^n \beta w_i - \zeta \int_X \sum_{i=1}^n w_i E_i(x)\mathrm{d}\mu(x)$ with $E_i(x) = e^{(\alpha(x)+\beta_i g(x,y_i))/\zeta}$ (see [5]). Let α^*, β^* be the argmax; then, we have

$$W_{k,\zeta}^k(\mu.\nu) = \int_X \sum_{i=1}^n g(x,y_i)E_i^*(x)w_i\mathrm{d}\mu(x)$$

with $E_i^*(x) = e^{(\alpha^*(x)+\beta_i^*-g(x,y_i))/\zeta}$. Since $\alpha'(x) := \alpha(x) + t$ and $\beta_i' := \beta_i - t$ produce the same $E_i(x)$ for any $t \in \mathbb{R}$, the representative with $\beta_n = 0$ that is equivalent to β (as well as β^*) is denoted by $\overline{\beta}$ (similarly $\overline{\beta}^*$) below in order to obtain uniqueness and make the differentiation possible.

From a direct differentiation of $W_{k,\zeta}^k$, we have

$$\begin{aligned}\frac{\partial W_{k,\zeta}^k}{\partial w_i} &= \int_X g(x,y_i)E_i^*(x)\mathrm{d}\mu(x) + \\ &\frac{1}{\zeta}\int_X \sum_{j=1}^n g(x,y_j)\left(\frac{\partial\alpha^*(x)}{\partial w_i} + \frac{\partial\beta_j^*}{\partial w_i}\right)w_j E_j^*(x)\mathrm{d}\mu(x).\end{aligned} \tag{4}$$

$$\begin{aligned}\nabla_{y_i}W_{k,\zeta}^k &= \int_X \nabla_{y_i}g(x,y_i)\left(1-\frac{g(x,y_i)}{\zeta}\right)E_i^*(x)w_i\mathrm{d}\mu(x) + \\ &\frac{1}{\zeta}\int_X \sum_{j=1}^n g(x,y_j)\left(\nabla_{y_i}\alpha^*(x) + \nabla_{y_i}\beta_j^*\right)w_j E_j^*(x)\mathrm{d}\mu(x).\end{aligned} \tag{5}$$

With the transference plan $\mathrm{d}\pi_i(x) = w_i E_i^*(x)\mathrm{d}\mu(x)$ and the derivatives of α^*, β^*, $g(x,y_i)$ calculated, the gradient of $W_{k,\zeta}^k$ can be assembled.

Assume that g is a Lipschitz constant that is differentiable almost everywhere (for $k \geq 1$ and a d_X Euclidean distance in $\mathbb{R}^d$, differentiability fails to hold only when $k = 1$ and $y_i = x$) and that $\nabla_y g(x,y)$ is calculated. The derivatives of α^* and $\overline{\beta}^*$ can then be calculated thanks to the Implicit Function Theorem for Banach spaces (see [15]).

The maximality of $\mathcal{L}$ at α^* and $\overline{\beta}^*$ induces $\mathcal{N} := \nabla_{\alpha,\overline{\beta}}\mathcal{L}|_{(\alpha^*,\overline{\beta}^*)} = 0 \in (L^\infty(X)\otimes\mathbb{R}^{m-1})^\vee$, and the Fréchet derivative vanishes. By differentiating (in the sense of Fréchet) again on $(\alpha,\overline{\beta})$ and y_i, w_i, respectively, we get

$$\nabla_{(\alpha,\overline{\beta})}\mathcal{N} = -\frac{1}{\zeta}\begin{bmatrix} \mathrm{d}\mu(x)\delta(x,x') & \mathrm{d}\pi_j(x') \\ \mathrm{d}\pi_i(x) & w_i\delta_{ij}\end{bmatrix} \tag{6}$$

as a bilinear functional on $L^\infty(X)\times\mathbb{R}^{m-1}$ (note that, in Equation (6), the index i of $\mathrm{d}\pi_i$ cannot be m). The bilinear functional $\nabla_{(\alpha,\overline{\beta})}\mathcal{N}$ is invertible, and we denote its inverse by $\mathbf{M}$ as a bilinear form on $\left(L^\infty(X)\otimes\mathbb{R}^{m-1}\right)^\vee$. The last ingredient for the Implicit Function Theorem is $\nabla_{w_i,y_i}\mathcal{N}$:

$$\nabla_{w_i}\mathcal{N} = \left(-\frac{1}{w_i}\int_X(\,\cdot\,)\mathrm{d}\pi_i(x), \vec{0}\right) \tag{7}$$

$$\nabla_{y_i}\mathcal{N} = \left(\frac{1}{\zeta}\int_X(\cdot)\nabla_{y_i}g(x,y_i)\mathrm{d}\pi_i(x),\ \frac{\delta_{ij}}{\zeta}\int_X \nabla_{y_i}g(x,y_i)\mathrm{d}\pi_i(x)\right). \tag{8}$$

Then, $\nabla_{w_i,y_i}(\alpha^*,\overline{\beta}^*) = \mathbf{M}(\nabla_{w_i,y_i}\mathcal{N})$. Therefore, we have gradient $\nabla_{w_i,y_i}W_{k,\zeta}^k$ calculated.

Moreover, we can differentiate Equations (4)–(8) to get a Hessian matrix of $W_{k,\zeta}^k$ on w_i's and y_i's to provide a better differentiability of $g(x,y)$ (which may enable Newton's method, or a mixture of Newton's method and minibatch SGD to accelerate the convergence). More details about the claims, calculations, and proofs are provided in the Appendix B.

4. The Discretization Algorithm

Here, we provide a description of an algorithm for the efficient discretizations of optimal transport (EDOT) from a distribution μ to μ_m with integer m, which is a given cardinality of support. In general, μ does not need not be explicitly accessible, and, even if it is accessible, computing the exact transference plan is not feasible. Therefore, in this construction, we assume that μ is given in terms of a random sampler, and we apply a minibatch stochastic gradient descent (SGD) through a set of samples that are independently drawn from μ of size N on each step to approximate μ.

To calculate the gradient $\nabla_{\mu_m} W^k_{k,\zeta}(\mu,\mu_m) = \left(\nabla_{x_i} W^k_{k,\zeta}(\mu,\mu_m), \nabla_{w_i} W^k_{k,\zeta}(\mu,\mu_m)\right)_{i=1}^m$, we need: (1). $\pi_{X,\zeta}$, the EOT transference plan between μ and μ_m, (2). the cost $g = d_X^k$ on X, and (3). its gradient on the second variable $\nabla_{x'} d_X^k(x,x')$. From N samples $\{y_i\}_{i=1}^N$, we can construct $\mu_N = \frac{1}{N}\sum_{i=1}^N \delta_{y_i}$ and calculate the gradients with μ replaced by μ_N as an estimation, whose effectiveness (convergence as $N \to \infty$) is proved in [5].

We call this discretization algorithm the *Simple EDOT* algorithm. The pseudocode is stated in the Appendix C.

Proposition 2 (Convergence of the Simple EDOT)**.** *The Simple EDOT generates a sequence* $(\mu_m^{(i)})$ *in the compact set* $X^m \times \Delta$*. If the set of limit points of* $(\mu_m^{(i)})$ *does not intersect with* $X^m \times \partial\Delta$*, then* $(\mu_m^{(i)})$ *converges to a stationary point in* $X^m \times Int(\Delta)$ *where* $Int(\cdot)$ *represents the interior.*

In simulations, we fixed $k = 2$ to reduce the computational complexity and fixed the regularizer $\zeta = 0.01$ for X of diameter 1 and scales proportional with $\mathrm{diam}(X)^k$ (see next section). Such a choice for ζ is not only small enough to reduce the error between the EOT estimation $W_{k,\zeta}$ and the true W_k, but also ensures that $e^{-g(x,y)/\zeta}$ and its byproduct in the SK are distinguishable from 0 in a *double* format.

Examples of discretization: We demonstrated our algorithm on the following:

E.g., (1). μ is the uniform distribution on $X = [0,1]$.

E.g., (2) μ is the mixture of two truncated normal distributions on $X = [0,1]$, and the PDF is $f(x) = 0.3\phi(x;0.2,0.1) + 0.7\phi(x;0.7,0.2)$, where $\phi(x;\xi,\sigma)$ is the density of the truncated normal distribution on $[0,1]$ with the expectation ξ and standard deviation σ.

E.g., (3) μ is the mixture of two truncated normal distributions on $X = [0,1]^2$, where the two distributions are $\phi(x;0.2,0.1)\phi(y;0.3,0.2)$ of weight 0.3 and $\phi(x;0.7,0.2)\phi(y;0.6,0.15)$ of weight 0.7.

Let $N = 100$ for all plots in this section. Figure 2a–c plots the discretizations (μ_m) for E.g., (1)–(3) with $m = 5, 5$, and 7, respectively.

Figure 2f illustrates the convergence rate of $W^k_{k,\zeta}(\mu,\mu_m)$ versus the SGD steps for Example (2) with μ_m obtained by a 5-point EDOT. Figure 2d,e plot the entropy-regularized Wasserstein $W^k_{k,\zeta}(\mu,\mu_m)$ versus m, thereby comparing EDOT and naive sampling for Examples (1) and (2). Here, the μ_ms are: (a) from the EDOT with $3 \le m \le 7$ in Example 1 and $3 \le m \le 8$ in Example 2, which are shown by $\times$s in the figures. (b) from naive sampling, which is simulated using a Monte Carlo of volume 20,000 on each size from 3 to 200. Figure 2d,e demonstrate the effectiveness of the EDOT: as indicated by the orange horizontal dashed line, even 5-point EDOT discretization in these two examples outperformed 95% of the naive samplings of size 40, as well as 75% of the naive samplings of size over 100 (the orange dash and dot lines).

An example of a transference plan: In Figure 3a, we illustrate the efficiency of the EDOT on an OT problem: $X = Y = [0,1]$, where the marginal μ and ν are truncated normal (mixtures), and μ has two components (shown in red curve on the left), while ν has only one component (shown in red curve on the top). The cost function is the squared Euclidean distance, and $\lambda = \zeta = 0.01$.

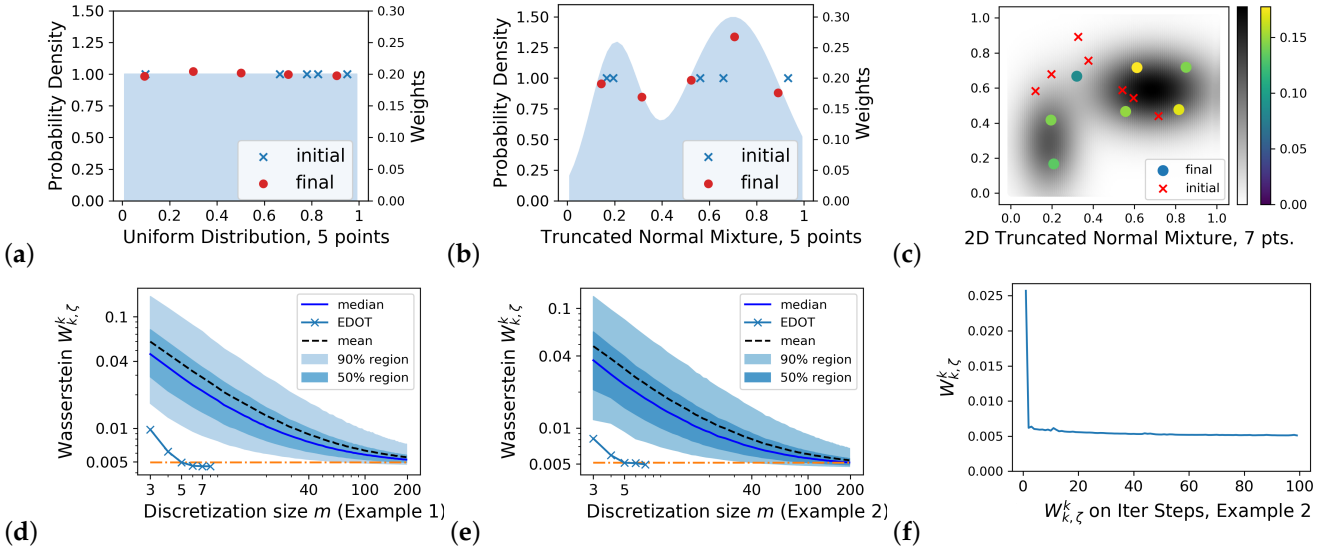

Figure 2. (**a**–**c**) Plots of EDOT discretizations of the Examples (1)–(3). In (**c**), the x-axis and y-axis are the 2D coordinates, and the probability density of μ and weights of μ_m are encoded by color. (**d**,**e**) show comparison between EDOT and i.i.d. sampling for Examples (1) and (2). EDOT are calculated with $m = 3$ to 7 (3 to 8). The 4 boundary curves of the shaded region are 5%-, 25%-, 75%-, and 95%-percentile curves; the orange line represents the level of $m = 5$; (**f**) plots the entropy regularized Wasserstein distance $W^k_{k,\zeta}(\mu, \mu_m)$ versus the SGD steps for Example (2) with μ_m optimized by 5-point EDOT. $\zeta = 0.01$ in all cases.

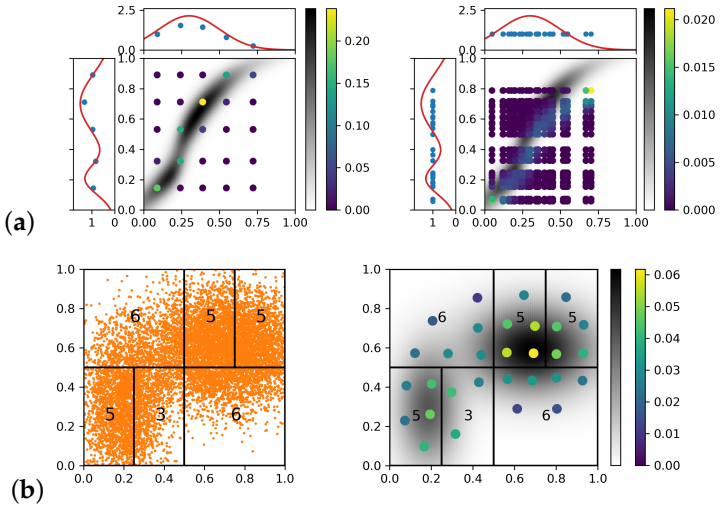

Figure 3. (**a**): Approximation of a transference plan. **Left**: 5×5 EDOT approximation. **Right**: 25×25 naive approximation. In both figures, magnitudes of each point is color coded, the background grayscale density represents the true EOT plan. (**b**): An example of adaptive refinement on a unit square. Left: division of 10,000 sample S approximating a mixture of two truncated Gaussian distributions and the refinement for 30 discretization points. Number of discretization points assigned to each region is marked by black numbers. E.g., upper left regaion needs 6 points. Right: the discretization optimized locally and combined as a probability measure with $k = 2$.

The left of Figure 3a shows a 5×5 EDOT approximation with $W^k_{k,\zeta}(\mu, \mu_5) = 4.792 \times 10^{-3}$, $W^k_{k,\zeta}(\nu, \nu_5) = 5.034 \times 10^{-3}$, and $W^k_{k,\zeta}(\gamma, \gamma_{5,5}) = 8.446 \times 10^{-3}$. The high density area of the EOT plan is correctly covered by EDOT estimating points with high weights. The right shows a 25×25 naive approximation with $W^k_{k,\zeta}(\mu, \mu_7) = 5.089 \times 10^{-3}$, $W^k_{k,\zeta}(\nu, \nu_7) = 2.222 \times 10^{-2}$, and $W^k_{k,\zeta}(\gamma, \gamma_{7,7}) = 2.563 \times 10^{-2}$. The points of the naive estimating with the highest weights missed the region where the true EOT plan was of the most density.

5. Methods of Improvement

I. Adaptive EDOT: The computational cost of a simple EDOT increases with the dimensionality and diameter of the underlying space. Discretization with a large m is needed to capture higher dimensional distributions, which result an increase in parameters for calculating the gradient of $W^k_{k,\zeta}$: md for the y_i positions and $m-1$ for the w_i weights. Such an increment will not only increase the complexity in each step, but also require more steps for the SGD to converge. Furthermore, the calculation will have a higher complexity ($\mathcal{O}(mN)$ for each normalization in Sinkhorn).

We proposed to reduce the computational complexity using a "divide and conquer" approach. The Wasserstein distance took the k-th power of the distance function d^k_X as a cost function. The locality of distance d_X made the solution to the OT / EOT problem local, meaning that the probability mass was more likely to be transported to a close destination than to a remote one. Thus, we can "divide and conquer"—thereby cutting the space X into small cells and solve the discretization problem independently.

To develop a "divide and conquer" algorithm, we need: (1) an adaptive dividing procedure that is able to partition $X = X_1 \sqcup \cdots \sqcup X_{\mathcal{I}}$, which balances the accuracy and computational intensity among the cells; (2) to determine the discretization size m_i and choose a proper regularizer ζ_i for each cell X_i. The pseudocodes for all variations are shown in the Appendix C Algorithms A2 and A3.

Choosing size m: An appropriate choice of m_i will balance contributions to the Wasserstein among the subproblems as follows: Let X_i be a manifold of dimension d, let $\mathrm{diam}(X_i)$ be its diameter, and let $p_i = \mu(X_i)$ be the probability of X_i. The entropy-regularized Wasserstein distance can be estimated as $W^k_{k,\zeta} = \mathcal{O}(p_i m_i^{-k/d}\mathrm{diam}(X_i)^k)$ [16,17]. The contribution to $W^k_{k,\zeta}(\mu,\mu_m)$ per point in support of μ_m is $\mathcal{O}(p_i m_i^{-(k+d)/d}\mathrm{diam}(X_i)^k)$. Therefore, to balance each point's contribution to the Wasserstein among the divided subproblems, we set $m_i \approx \frac{(p_i\mathrm{diam}(X_i)^k)^{d/(k+d)}}{\sum_{j=1}^{\mathcal{I}}(p_j\mathrm{diam}(X_j)^k)^{d/(k+d)}}$.

Occupied volume (Variation 1):A cell could be too vast (e.g., large in size with few points in a corner), thus resulting in obtaining a larger m_i than needed. To fix it, we may replace the $\mathrm{diam}(X_i)$ above with $\mathrm{Vol}(X_i)^{1/d}$, where $\mathrm{Vol}(X_i)$ is the occupied volume calculated by counting the number of nonempty cells in a certain resolution (levels in previous binary division). The algorithm (Variation 1) becomes a binary tree to resolve and obtain the occupied volume for each cell, then there is tree traversal to assign m_i.

Adjusting the regularizer ζ: In the $W^k_{k,\zeta}$, the SK on $e^{-g(x,y)/\zeta}$ is calculated. Therefore, ζ should scale with d^k_X to ensure that the transference plan is not affected by the scaling of d_X. Precisely, we may choose $\zeta_i = \mathrm{diam}(X_i)^k\zeta_0$ for some constant ζ_0.

The division: Theoretically, any refinement procedure that proceeds iteratively and eventually makes the diameter of each cell approach 0 can be applied for division. In our simulation, we used an adaptive kd-tree-style cell refinement in a Euclidean space $\mathbb{R}^d$. Let X be embedded into $\mathbb{R}^d$ within an axis-aligned rectangular region. We chose an axis $\mathbf{x}_l$ in $\mathbb{R}^d$ and evenly split the region along a hyperplane orthogonal to $\mathbf{x}_l$ (e.g., cut square $[0,1]^2$ along the line $x = 0.5$); thus, we constructed X_1 and X_2. With the sample set S given, we split it into two sample sizes S_1 and S_2 according to which subregion each sample was located in. Then, the corresponding m_i and ζ_i could be calculated as discussed above. Thus, two cells and their corresponding subproblems were constructed. If some of the m_i was still too large, the cell was cut along another axis to construct two other cells. The full list of cells and subproblems could be constructed recursively. In addition, another cutting method (variation 2) that chooses the most sparse point as a cutting point through a sliding window is sometimes useful in practice.

After having the set of subproblems, we could apply the EDOT for the solutions in each cell, then combine the solutions $\mu^{(i)}_{m_i} = \sum_{j=1}^{m_i} w^{(i)}_j \delta_{y^{(i)}_j}$ into the final result $\mu_m := \sum_{i=1}^{\mathcal{I}}\sum_{j=1}^{m_i} p_i w^{(i)}_j \delta_{y^{(i)}_j}$.

Figure 3b shows the optimal discretization for the example in Figure 2c with $m = 30$, which was obtained by applying the EDOT with adaptive cell refinement, or $\zeta = 0.01 \times \text{diam}^2$.

II. On embedded CW complexes: Although the samples on space X are usually represented as a vector in $\mathbb{R}^d$, inducing an embedding $X \hookrightarrow \mathbb{R}^d$, the space X usually has its own structure as a CW complex (or simply a manifold) with a more intrinsic metric. Thus, if the CW complex structure is known, even piecewise, we may apply the refinement on X with respect to its own metric, whereas direct discretization as a subset in $\mathbb{R}^d$ may result in a low expressing efficiency.

We now illustrate the adaptive EDOT by an example on a mixture normal distribution of a sphere mapped through stereographic projection. More examples of a truncated normal mixture over a Swiss roll and the discretization of a 2D optimal transference plan are detailed in the Appendix D.5.

On the sphere: The underlying space X_{sphere} is the unit sphere in $\mathbb{R}^3$. μ_{sphere} is the pushforward of a normal mixture distribution on $\mathbb{R}^2$ by stereographic projection. The sample set $S_{\text{sphere}} \sim \mu_{\text{sphere}}$ over X_{sphere} is shown on Figure 4 on the left. Consider a (3D) Euclidean metric on the X_{sphere} induced by the embedding. Figure 4a (right) plots the EDOT solution with refinement for μ_m with $m = 40$. The resulting cell structure is shown as colored boxes.

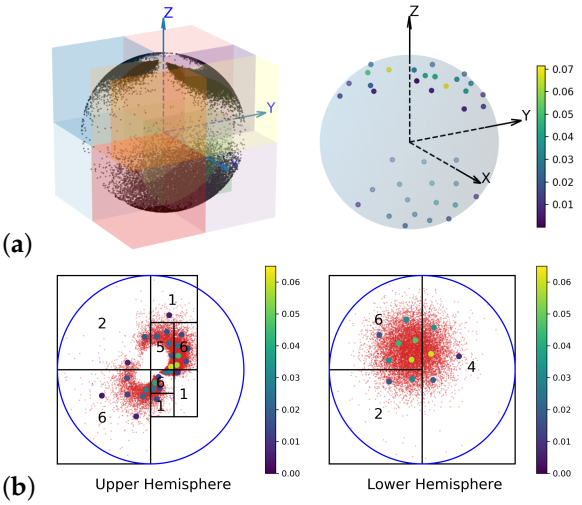

Figure 4. (**a**) **Left**: 30,000 samples from μ_{sphere} and the 3D cells under divide-and-conquer algorithm **Right**: 40-point EDOTs in 3D. (**b**) The 40-point CW-EDOTs in 2D. Red dots: samples, other dots: discrete atoms with weights represented in colors. **Left**: upper hemisphere. **Right**: lower hemisphere, stereographic projections about poles. $\zeta = 0.01 \times \text{diam}^2$.

To consider the intrinsic metric, a CW complex was constructed about a point on the equator as a 0-cell structure; the rest of the equator was constructed as a 1-cell, and the upper hemisphere and lower hemisphere were constructed as two dimension 2- (open) cells. We took the upper and lower hemispheres and mapped them onto a unit disk through stereographic projection with respect to the south and north pole, respectively. Then, we took the metric from spherical geometry and rewrote the distance function and its gradient using the natural coordinate on the unit disk. Figure 4b shows the refinement of the EDOT on the samples (in red) and the corresponding discretizations in colored points. More figures can be found in the Appendices.

6. Analysis of the Algorithms

In this section, we derive the complexity of the simple EDOT and the adaptive EDOT. In particular, we show the following:

Proposition 3. *Let μ be a (continuous) probability measure on a space X. A simple EDOT of size m has time complexity $\mathcal{O}((N+m)^2 mdL + NmL\log(1/\epsilon))$ and space complexity $\mathcal{O}((N+m)^2)$,*

where N is the minibatch size (to construct μ_N in each step to approximate μ), d is the dimension of X, L is the maximal number of iterations for SGD, and ϵ is the error bound in the Sinkhorn calculation for the entropy-regularized optimal transference plan between μ_N and μ_m.

Proposition 3 quantitatively shows that, when the adaptive EDOT is applied, the total complexities (in time and space) are reduced, because the magnitudes of both N and m are much smaller in each cell.

The procedure of dividing sample set S into subsets through the adaptive EDOT is similar to Quicksort; thus, the space and time complexities are similar. The similarity comes from the binary divide-and-conquer structure, as well as that each split action is based on comparing each sample with a target.

Proposition 4. *For the preprocessing (job list creation) for the adaptive EDOT, the time complexity is $\mathcal{O}(N_0 \log N_0)$ in the best and average case and $\mathcal{O}(N_0^2)$ in the worst case, where N_0 is the total number of sample points, and the space complexity is $\mathcal{O}(N_0 d + m)$, or simply $\mathcal{O}(N_0 d)$ as $m \ll N_0$.*

Remark 2. *Complexity is the same as Quicksort. The set of N_0 sample points in the algorithm are treated as the "true" distribution in the adaptive EDOT, since, in the later EDOT steps for each cell, no further samples are taken, as it is hard for a sampler to produce a sample in a given cell. Postprocessing of the adaptive EDOT has $\mathcal{O}(m)$ complexity in both time and space.*

Remark 3. *For the two algorithm variations in Section 5, the occupied volume estimation works in the same way as the original preprocessing step, which has the same time complexity as before (by itself, since dividing must happen after knowing the occupied volume of all cells), but, with the tree built, the original preporcessing becomes a tree traversal and has (additional) time complexity $\mathcal{O}(N_0)$ and (additional) space complexity $\mathcal{O}(N_0)$ for the space storing occupied volume.*

For details on choosing cut points with window sliding, the discussion can be seen in the Appendix C.5.

Comparison with naive sampling: After having a size m discretization on X and a size n discretization on Y, the EOT solution (Sinkhorn algorithm) has time complexity $\mathcal{O}(mn \log(1/\epsilon))$. In the EDOT, two discretization problems must be solved before applying the Sinkhorn, while the naive sampling requires nothing but sampling.

According to Proposition 3, solving a single continuous EOT problem using a size m simple EDOT method may result in higher time complexity than naive sampling with an even larger sample size N (than m). However, unlike the EDOT, which only requires access to a distance function d_X and d_Y on X and Y, respectively, a known cost function $c : X \times Y \to \mathbb{R}$ is necessary for naive sampling. In real applications, the cost function may be from real world experiments (or from extra computations) done for each pair (x, y) in the discretization; thus, the size of discretized distribution is critical for cost control. d_X and d_Y usually come along with the spaces X and Y, respectively, and are easy to compute. An additional application of the EDOT is necessary when the marginal distributions μ_X and ν_Y are fixed for different cost functions; then, discretizations can be reused. Thus, the cost of discretization is calculated one time, and the improvement it brings accumulates in each repeat.

7. Related Work and Discussion

Our original problem was the optimal transport problem between general distributions as samplers (instead of integration oracles). We translated that into a discretization problem and an OT problem between discretizations.

I. Comparison with other discretization methods: There are several other methods that generate discrete distributions from arbitrary distributions in the literature, which are obtained via semi-continuous optimal transport where the calculation of a weighted Voronoi diagram is needed. Calculating the weighted Voronoi diagrams usually requires 1. that the cost function be a squared Euclidean distance and 2. the application of Delaunay triangulation, which is expensive in more than two dimensions. Furthermore,

semi-continuous discretization may only optimize one aspect between the position and weights of the atoms, and this process is mainly based on [18] (the optimized position) and [19] (the optimized weights).

We mainly compared the prior work of [18], which focuses on the barycenter of a set of distributions under the Wasserstein metric. This work resulted in a discrete distribution called the Lagrangian discretization, which is of the form $\frac{1}{m}\sum_{i=1}^{m}\delta_{x_i}$ [2]. Other works, such as [20,21], find barycenters but do not create a discretization. Refs. [19,22] studied the discrete estimation of a 2-Wasserstein distance locating discrete points through a clustering algorithm *k-means++* and a weighted Voronoi diagram refinement, respectively. Then, they assigned weights and made them non-Lagrangian discretizations. Ref. [19] (comparison in Figure 5) roughly followed a "divide-and-conquer" approach in selecting positions, but the discrete positions were not tuned according to Wasserstein distance directly. Ref. [22] converged as the number of discrete points increased. However, it lacked a criterion (such as the Wasserstein in the EDOT) to show that the choice is not just one among all possible converging algorithms, but, rather, it is a special one.

By projecting the gradient in the SGD to the tangent space of the submanifold $X^m \times \{\mathbf{e}_m/m\} = \{\frac{1}{m}\sum\delta_{x_i}\}$, or by equivalently fixing the learning rate on the weights to zero, the EDOT can estimate a Lagrangian discretization (denoted by EDOT-Equal). A comparison among the methods is held on the map of the Canary islands, which is shown in Figure 6. This example shows that our method can get a similar result using Lagrangian discretization as the methods in the literature, while, in general, this type of EDOT can work better.

Moreover, the EDOT can be used to solve barycenter problems.

Note that, to apply adaptive EDOT for barycenter problems, compatible divisions of the target distributions are needed (i.e., a cell A from one target distribution transports onto a discrete subset D thoroughly, and D transports onto a cell B from another target distribution, etc.).

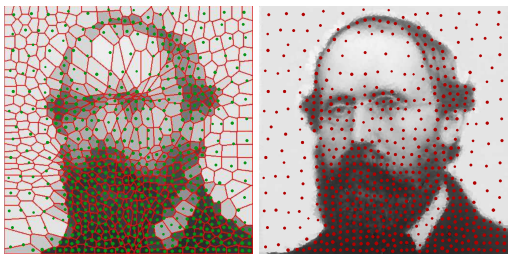

Figure 5. EDOT of an example from [19]. Potrait of Riemann, discretization of size 625. **Left**: green dots show position and weights of EDOT discretization (same as right); cells in background are discretization of the same size in the original [19]. **Right**: A size 10,000 discretization from [19]; we directly applied EDOT to this picture, treating it as the continuous distribution. $\zeta = 0.01 \times \text{diam}^2$.

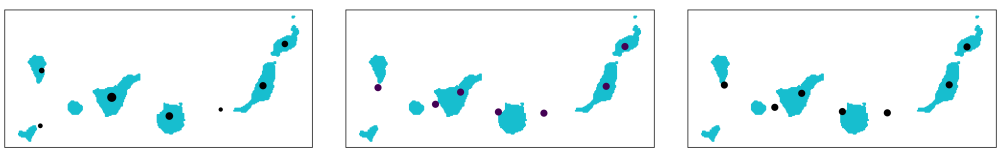

Figure 6. A comparison of EDOT (**left**), EDOT-Equal (**mid**), and [18] (**right**) on the Canary islands, treated as a binary distribution with a constant density on islands and 0 in the sea. Discretizations for each method is shown by black bullets. Wasserstein distances: EDOT: $W^2_{0.005} = 0.02876$, EDOT-Equal: $W^2_{0.005} = 0.05210$, Claici: $W^2_{0.005} = 0.05288$. Map size is 3.13×1.43.

We also tested these algorithms on discretizing gray/colored scale pictures. The comparison of discretization with points varying from 10 to 4000 for a kitty image between EDOT, EDOT-equal, [18] and estimations of their Wasserstein distances to the original image are shown in Figures 7 and 8.

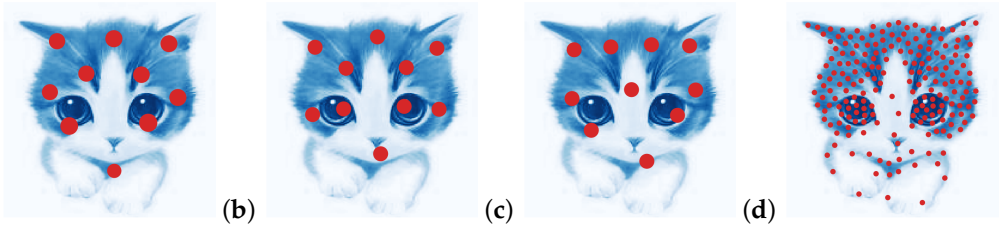

Figure 7. Discretization of a kitty. Discretization by each method is shown in red bullets on top of the Kitty image. (**a**) EDOT, 10 points, $W^2_{0.001} = 0.009176$, radius represents weight; (**b**) EDOT-Equal, 10 points, $W^2_{0.001} = 0.008960$; (**c**) [18], 10 points, $W^2_{0.001} = 0.009832$; (**d**) [18], 200 points. Figure size 1×1, $\zeta = 0.01 \times \text{diam}^2$.

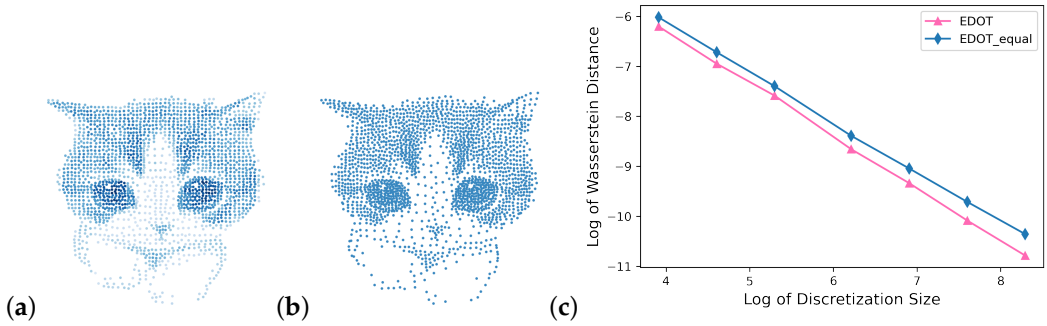

Figure 8. 2000-Point Discretizations, (**a**). EDOT (weight plotted in color), (**b**). EDOT-Equal, (**c**). Relations between $\log(W^2)$ and $\log m$ (all with divide and conquer); it can be seen that the advantage of W_{EDOT} over W_{Equal} grows with the size of discretization.

Furthermore, the EDOT may be applied on RGB channels of an image independently, which then combine plots of discretizations in the corresponding color. The results are shown in Figure 1 at the beginning of this paper.

Lagrangian discretization may have a disadvantage in representing repetitive patterns with incompatible discretization points.

In Figure 9, we can see that discretizing 16 objects with 24 points caused weight incompatibility locally for the Lagrangian discretization, thus making points locate between objects and increasing the Wasserstein distance. With the EDOT, the weights of points that lie outside of the blue object were much smaller. The patterned structure was better represented by the EDOT. In practice, patterns often occur as part of the data (e.g., pictures of nature), and it is easy to get an incompatible number in Lagrangian discretization, since the equal weight-requirement is rigid; consequently, patterns cannot be properly captured.

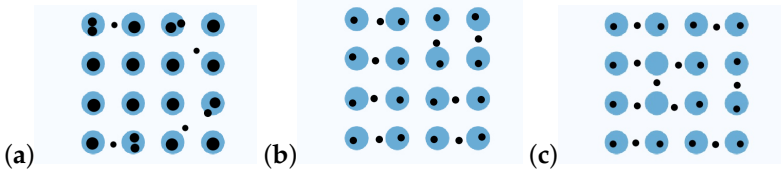

Figure 9. Discretization of 16 blue disks in a unit square with 24 points (black). (**a**) EDOT, $W^2_\zeta = 0.001398$; (**b**) EDOT-Equal, $W^2_\zeta = 0.002008$; (**c**) [18], $W^2_\zeta = 0.002242$. $\zeta = 10^{-4}$. Figure size is 1×1.

II. General k and deneral distance d_X: Our algorithms (Simple EDOT, adaptive EDOT, and EDOT-Equal) work for a general choice of parameter $k > 1$ and C^2 distance d_X on X. For example, in Figure 4 part (b), the distance used on each disk was spherical (arc length along the big circle passing through two points), which could not be isometrically reparametrized into a plane with Euclidean metrics because of the difference in curvatures.

III. Other possible impacts: As the OT problem widely exists in many other areas, our algorithm can be applied accordingly, e.g., the location and size of supermarkets or

electrical substations in an area, or even air conditioners in the rooms of supercomputers. Our divide-and-conquer methods are suitable for solving these real-world applications.

IV. OT for discrete distributions: Many algorithms have been developed to solve OT problems between two discrete distributions [3]. Linear programming algorithms were first developed, but their applications have been restricted by high computational complexity. Other methods such as [23], with a cost of form $c(x, y) = h(x - y)$ for some h, which applies the "back-and-forth" method by hopping between two forms of a Kantorovich dual problem (on the two marginals, respectively) to get a gradient of the total cost over the dual functions, usually solve problems with certain conditions. In our work, we chose to apply an EOT developed by [8] for an estimated OT solution of the discrete problem.

8. Conclusions

We developed methods for efficiently approximating OT couplings with fixed size $m \times n$ approximations. We provided bounds on the relationship between a discrete approximation and the original continuous problem. We implemented two algorithms and demonstrated their efficacy as compared to naive sampling and analyzed computational complexity. Our approach provides a new approach to efficiently compute OT plans.

Author Contributions: Conceptualization, J.W. and P.W.; methodology, J.W.; software, J.W.; validation, P.W., P.S. and J.W.; formal analysis, J.W. and P.W.; investigation, P.W.; writing—original draft preparation, J.W., P.W. and P.S.; writing—review and editing, J.W., P.W. and P.S.; supervision, P.S.; project administration, P.S.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by DARPA grant number HR00112020039, W912CG22C0001, W911NF2020001 and NSF MRI 1828528.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OT	Optimal Transport
EOT	Entropy-Regularized Optimal Transport
EDOT	Efficient Discretization of Optimal Transport
SGD	Stochastic Gradient Descent

Appendix A. Proof of Proposition 1

Proof. We will adopt the notations $Z = X \times Y$ and $z_i = (x_i, y_i) \in Z$. Furthermore, recall the condition:

$$\max\{d_X(x, x'), d_Y(y, y')\} \leq d_Z(z, z') \leq d_X(x, x') + d_Y(y, y') \tag{A1}$$

For inequality (i) without loss, assume that

$$\max\{W_k^k(\mu, \mu_m), W_k^k(\nu, \nu_n)\} = W_k^k(\mu, \mu_m)$$

Denote the optimal $\pi_Z \in \Pi(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n))$ that achieves $W_k^k(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n))$ by π_Z^* and similarly for π_X^*. Then, we have:

$$\begin{aligned} W_k^k(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n)) &= \int_{Z^2} d_Z^k(z_1,z_2)\mathrm{d}\pi_Z^* \geq A\int_{Z^2} d_X^k(x_1,x_2)\mathrm{d}\pi_Z^* \\ &= A\int_{X^2} d_X^k(x_1,x_2)\int_{Y^2}\mathrm{d}\pi_Z^* \overset{(a)}{=} A\int_{X^2} d_X^k(x_1,x_2)\mathrm{d}\pi_X' \\ &\overset{(b)}{\geq} A\int_{X^2} d_X^k(x_1,x_2)\mathrm{d}\pi_X^* = W_k^k(\mu,\mu_m) \end{aligned}$$

Here, $\pi_X' \in \Pi(\mu,\mu_m)$ and eq (a) hold since $\pi_Z^* \in \Pi(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n))$, and ineq (b) holds since π_X^* is the optimal choice.

For inequality (ii), we use the following to simplify the notations: $\mathrm{d}\gamma \otimes \mathrm{d}\gamma_{mn} \coloneqq \mathrm{d}\gamma_\lambda(\mu,\nu)\otimes\mathrm{d}\gamma_\lambda(\mu_m,\nu_n)$ and $d_Z^k \coloneqq d_Z^k(z_1,z_2), d_X^k \coloneqq d_X^k(x_1,x_2), d_Y^k \coloneqq d_Y^k(y_1,y_2)$

$$\begin{aligned} &W_{k,\zeta}^k(\gamma_\lambda(\mu,\nu),\gamma_\lambda(\mu_m,\nu_n)) = \int_{Z^2} d_Z^k(z_1,z_2)\mathrm{d}\pi_{z,\zeta}^* \\ &\overset{(a)}{=} \int_{Z^2} d_Z^k\cdot\exp(\frac{\alpha(z_1)+\beta(z_2)-d_Z^k(z_1,z_2)}{\zeta})\mathrm{d}\gamma\otimes\mathrm{d}\gamma_{mn} \\ &\overset{(b)}{\leq} A\int_{Z^2}(d_X^k+d_Y^k)\cdot\exp(\frac{\alpha(z_1)+\beta(z_2)-d_Z^k}{\zeta})\mathrm{d}\gamma\otimes\mathrm{d}\gamma_{mn} \\ &\overset{(c)}{\leq} C_1\int_{Z^2}(d_X^k+d_Y^k)\cdot\exp(\frac{-d_Z^k}{\zeta})\mathrm{d}\gamma\otimes\mathrm{d}\gamma_{mn} \\ &\overset{(d)}{\leq} C_1[\int_{Z^2} d_X^k\cdot\exp(\frac{-d_X^k}{\zeta})+d_Y^k\cdot\exp(\frac{-d_Y^k}{\zeta})\mathrm{d}\gamma\otimes\mathrm{d}\gamma_{mn}] \\ &\overset{(e)}{=} C_1[\int_{X^2} d_X^k\cdot\exp(\frac{-d_X^k}{\zeta})\mathrm{d}\mu\otimes\mathrm{d}\mu_m+\int_{Y^2} d_Y^k\cdot\exp(\frac{-d_Y^k}{\zeta})\mathrm{d}\nu\otimes\mathrm{d}\nu_n] \\ &\overset{(f)}{\leq} C_1\int_{X^2} d_X^k\cdot\exp(\frac{s(x_1)+t(x_2)-d_X^k}{\zeta})\mathrm{d}\mu\otimes\mathrm{d}\mu_m \\ &\qquad + C_1\int_{Y^2} d_Y^k\cdot\exp(\frac{s'(y_1)+t'(y_2)-d_Y^k}{\zeta})\mathrm{d}\nu\otimes\mathrm{d}\nu_n \\ &= C_1[W_{k,\zeta}^k(\mu,\mu_m)+W_{k,\zeta}^k(\nu,\nu_n)] \end{aligned}$$

Justifications for the derivations:

(a) Based on the dual formulation, it is shown in [5] Proposition 1 that, for $\zeta > 0$, there exist $\alpha(z_1),\beta(z_2)\in\mathcal{C}(Z)$ such that:

$\mathrm{d}\pi_{z,\zeta}^* = \exp(\frac{\alpha(z_1)+\beta(z_2)-d_Z^k(z_1,z_2)}{\zeta})\mathrm{d}\gamma\otimes\mathrm{d}\gamma_{mn}$;

(b) Inequality (ii) of Equation (A1);

(c) According to Ref. [24] Theorem 2, when X and Y are compact and c is smooth, α and β are uniformly bounded; moreover, both d_X^k and d_Y^k are uniformly bounded by the diameter of X and Y, respectively; hence, the constant B exists;

(d) Inequality (ii) of Equation (A1);

(e) $\gamma_\lambda(\mu,\nu)\in\Pi(\mu,\nu)$ and $\gamma_\lambda(\mu_m,\nu_n)\in\Pi(\mu_m,\nu_n)$;

(f) Similarly as in (a), for $\zeta > 0$, there exist $s(x_1),t(x_2)\in\mathcal{C}(X)$, and $s'(y_1),t'(y_2)\in\mathcal{C}(Y)$ such that $\exp(\frac{-d_X^k}{\zeta})\mathrm{d}\mu\otimes\mathrm{d}\mu_m = \mathrm{d}\pi_X^*$ and $\exp(\frac{-d_Y^k}{\zeta})\mathrm{d}\nu\otimes\mathrm{d}\nu_n = \mathrm{d}\pi_Y^*$. Moreover, $\int_{X^2} d_X^k\cdot\exp(\frac{s(x_1)+t(x_2)}{\zeta})\mathrm{d}\mu\otimes\mathrm{d}\mu_m \geq 0$, and $\int_{Y^2} d_Y^k\cdot\exp(\frac{s'(y_1)+t'(y_2)}{\zeta})\mathrm{d}\nu\otimes\mathrm{d}\nu_n \geq 0$. □

Appendix B. Gradient of $W^k_{k,\zeta}$

Appendix B.1. The Gradient

Following the definitions and notations in Sections 2 and 3 of the paper, we calculate the gradient of $W^k_{k,\zeta}(\mu,\mu_m)$ about parameters of $\mu_m := \sum_{i=1}^m w_i\delta_{y_i}$ in detail.

$W^k_{k,\zeta}(\mu,\mu_m) = \int_{X^2} g(x,y)\mathrm{d}\gamma_\zeta(x,y)$, where

$$\gamma_\zeta = \operatorname{argmin}_{\gamma\in\Pi(\mu,\mu_m)} \int_{X^2} g(x,y)\mathrm{d}\gamma(x,y) + \zeta \mathrm{KL}(\gamma||\mu\otimes\mu_m). \tag{A2}$$

Let $\alpha \in L^\infty(X)$ and $\beta\in\mathbb{R}^m$. Denote $\beta = \sum_{i=1}^m \beta_i\delta_{y_i}$, and let

$$\begin{aligned}
\mathcal{F}(\gamma;\mu,\mu_m,\alpha,\beta) :=& \int_{X^2} g(x,y)\mathrm{d}\gamma(x,y) + \zeta\mathrm{KL}(\gamma||\mu\otimes\mu_m)\\
&+ \int_X \alpha(x)\left(\int_X \mathrm{d}\gamma(x,y) - \mathrm{d}\mu_m(y)\right)\\
&+ \int_X \beta(y)\left(\int_X \mathrm{d}\gamma(x,y) - \mathrm{d}\mu(x)\right)\\
=& \int_{X^2} g(x,y)\mathrm{d}\gamma(x,y) + \zeta\mathrm{KL}(\gamma||\mu\otimes\mu_m)\\
&+ \int_X \alpha(x)\left(\sum_{i=1}^m \mathrm{d}\gamma(x,y_i) - \mathrm{d}\mu_m(y_i)\right)\\
&+ \sum_{i=1}^m \beta_i\left(\int_X \mathrm{d}\gamma(x,y_i) - \mathrm{d}\mu(x)\right)
\end{aligned} \tag{A3}$$

Since γ on the second component X is discrete and supported on $\{y_i\}_{i=1}^m$, we may denote $\mathrm{d}\gamma(x,y_i)$ by $\mathrm{d}\pi_i(x)$; thus,

$$\begin{aligned}
\mathcal{F}(\gamma;\mu,\mu_m,\alpha,\beta) =& \int_X \sum_{i=1}^m g(x,y_i)\mathrm{d}\pi_i(x)\\
&+ \zeta\sum_{i=1}^m \int_X \left(\ln\frac{\mathrm{d}\pi_i(x)}{\mathrm{d}\mu(x)} - \ln(\mu_m(y_i))\right)\mathrm{d}\pi_i(x)\\
&+ \int_X \alpha(x)\left(\sum_{i=1}^m \mathrm{d}\pi_i(x) - \mathrm{d}\mu_m(y_i)\right)\\
&+ \sum_{i=1}^m \beta_i\left(\int_X \mathrm{d}\pi_i(x) - \mathrm{d}\mu(x)\right)
\end{aligned} \tag{A4}$$

Then, the Fenchel duality of Problem (A2) is

$$\begin{aligned}
&\mathcal{L}(\mu,\mu_m;\alpha,\beta)\\
=& \int_X \alpha(x)\mathrm{d}\mu(x) + \sum_{i=1}^m \beta_i w_i - \zeta\int_X \sum_{i=1}^m e^{(\alpha(x)+\beta_i - g(x,y_i))/\zeta} w_i \mathrm{d}\mu(x).
\end{aligned} \tag{A5}$$

Let α^* and β^* be the argmax of the Fenchel dual (A5). The primal is solved by $\mathrm{d}\gamma^*(x,y_i) = e^{(\alpha^*(x)+\beta_i^* - g(x,y_i))/\zeta} w_i \mathrm{d}\mu(x)$. To make the solution unique, we restrict the freedom of the solution (where we see that $\mathcal{L}(\mu,\mu_m;\alpha,\beta) = \mathcal{L}(\mu,\mu_m;\alpha+t,\beta-t)$ for any $t\in\mathbb{R}$). We use the condition $\beta_m = 0$ to narrow the choices down to only one, and denote the dual variable β having the property $\overline{\beta}$ and $\overline{\beta}^*$.

We first calculate $\nabla_{w_i,y_i}\mathcal{L}(\mu,\mu_m;\alpha^*,\overline{\beta}^*)$ with α^* and $\overline{\beta}^*$ as functions of μ_m. (from the paper).

$$\frac{\partial \mathcal{L}}{\partial w_i} = \int_X g(x, y_i) E_i^*(x) \mathrm{d}\mu(x) + \frac{1}{\zeta} \int_X \sum_{j=1}^{n} g(x, y_j) \left(\frac{\partial \alpha^*(x)}{\partial w_i} + \frac{\partial \beta_j^*}{\partial w_i} \right) w_j E_j^*(x) \mathrm{d}\mu(x). \tag{A6}$$

$$\nabla_{y_i} \mathcal{L} = \int_X \nabla_{y_i} g(x, y_i) \left(1 - \frac{g(x, y_i)}{\zeta} \right) E_i^*(x) w_i \mathrm{d}\mu(x) + \frac{1}{\zeta} \int_X \sum_{j=1}^{n} g(x, y_j) \left(\nabla_{y_i} \alpha^*(x) + \nabla_{y_i} \beta_j^* \right) w_j E_j^*(x) \mathrm{d}\mu(x). \tag{A7}$$

Next, we calcuate the derivatives of α^* and $\overline{\beta}^*$ by finding their defining equation and then using the Implicit Function Theorem.

The optimal solution to the dual variables α^* and β^* is obtained by solving the stationary state equation $\nabla_{\alpha,\beta} \mathcal{L} = 0$. The derivatives are taken in the sense of the Fréchet derivative. The Fenchel dual function on α and $\overline{\beta}$, has its domain and codomain $\mathcal{L}(\mu, \mu_m; \cdot, \cdot) : L^\infty(X) \times \mathbb{R}^{m-1} \to \mathbb{R}$. The derivatives are

$$\nabla_\alpha \mathcal{L}(\mu, \mu_m; \alpha, \overline{\beta}) = \int_X \left(1 - \sum_{i=1}^{m} w_i E_i(x) \right) (\,\cdot\,) \mathrm{d}\mu(x), \tag{A8}$$

$$\frac{\partial}{\partial \overline{\beta}_i} \mathcal{L}(\mu, \mu_m; \alpha, \overline{\beta}) = w_i \left(1 - \int_X E_i(x) \mathrm{d}\mu(x) \right) \tag{A9}$$

where $E_i(x) = e^{(\alpha(x) + \beta_i - g(x, y_i))/\zeta}$ is defined as in the paper, $\nabla_\alpha \mathcal{L}(\mu, \mu_m; \alpha, \overline{\beta}) \in (L^\infty(X))^\vee$ (as a linear functional), and $\frac{\partial}{\partial \overline{\beta}_i} \mathcal{L}(\mu, \mu_m; \alpha, \overline{\beta}) \in \mathbb{R}$. Next, we need to show that $\mathcal{L}$ is differentiable in the sense of the Fréchet derivative, i.e.,

$$\lim_{||h|| \to 0} \frac{1}{||h||} \left(\mathcal{L}(\mu, \mu_m; \alpha + h, \overline{\beta}) - \mathcal{L}(\mu, \mu_m; \alpha, \overline{\beta}) - \nabla_\alpha \mathcal{L}(\mu, \mu_m; \alpha, \overline{\beta})(h) \right) = 0. \tag{A10}$$

By the definition of $\mathcal{L}$ (we write $\mathcal{L}(\alpha)$ for $\mathcal{L}(\mu, \mu_m; \alpha, \overline{\beta})$),

$$\begin{aligned}
&\mathcal{L}(\alpha + h) - \mathcal{L}(\alpha) - \nabla_\alpha \mathcal{L}(\alpha)(h) \\
=& \int_X h(x) \mathrm{d}\mu(x) - \zeta \int_X \sum_{i=1}^{m} \left(e^{h(x)/\zeta} - 1 \right) w_i E_i(x) \mathrm{d}\mu(x) - \int_X \left(1 - \sum_{i=1}^{m} w_i E_i(x) \right) h(x) \mathrm{d}\mu(x) \\
=& \zeta \int_X \sum_{i=1}^{m} \left(1 + \frac{h(x)}{\zeta} - e^{h(x)/\zeta} \right) E_i(x) w_i \mathrm{d}\mu(x) \\
=& \zeta \int_X \left(\sum_{k=2}^{\infty} \frac{1}{k!} \frac{h(x)^k}{\zeta^k} \right) \sum_{i=1}^{m} E_i(x) w_i \mathrm{d}\mu(x),
\end{aligned} \tag{A11}$$

The last equality is from a Taylor expansion of the exponential function. Consider that $||h||_\infty = \text{ess sup } |_{x \in X} h(x)|$ the essential supremum of $|h(x)|$ for $x \in X$ given measure μ.

Denote $\mathcal{N} := \nabla_{\alpha,\overline{\beta}}\mathcal{L}$,

$$\begin{aligned}
&\frac{1}{||h||}(\mathcal{L}(\alpha+h)-\mathcal{L}(\alpha)-\nabla_\alpha\mathcal{L}(\alpha)(h))\\
\leq&\frac{\zeta}{||h||}\int_X\left(\sum_{k=2}^{\infty}\frac{1}{k!}\frac{|h(x)|^k}{\zeta^k}\right)\sum_{i=1}^{m}E_i(x)w_i\mathrm{d}\mu(x)\\
\leq&\frac{\zeta}{||h||}\int_X\left(\sum_{k=2}^{\infty}\frac{1}{k!}\frac{||h||^k}{\zeta^k}\right)\sum_{i=1}^{m}E_i(x)w_i\mathrm{d}\mu(x)\\
=&\zeta\left(\sum_{k=2}^{\infty}\frac{1}{k!}\frac{||h||^{k-1}}{\zeta^k}\right)\int_X\sum_{i=1}^{m}E_i(x)w_i\mathrm{d}\mu(x)\\
=&\zeta\left(\sum_{k=2}^{\infty}\frac{1}{k!}\frac{||h||^{k-1}}{\zeta^k}\right)
\end{aligned}\tag{A12}$$

Therefore,

$$\lim_{||h||\to 0}\frac{1}{||h||}(\mathcal{L}(\alpha+h)-\mathcal{L}(\alpha)-\nabla_\alpha\mathcal{L}(\alpha)(h))=0,\tag{A13}$$

which shows that the expression of $\nabla_\alpha\mathcal{L}(\alpha)$ in Equation (A8) gives the correct Fréchet derivative. Note here that $\alpha\in L^\infty(X)$ is critical in Equation (A12).

Let $\mathcal{N}:=\nabla_{\alpha,\overline{\beta}}\mathcal{L}$ values in $(L^\infty(X))^\vee\times\mathbb{R}^{m-1}$. Then, $\mathcal{N}=0$ defines α^* and $\overline{\beta}^*$, which makes it possible to differentiate them about μ_m using the Implicit Function Theorem for Banach spaces. From now on, $\mathcal{N}$ take values at $\alpha=\alpha^*$, $\overline{\beta}=\overline{\beta}^*$, i.e., the marginal conditions on $\mathrm{d}\pi_i(x)=w_iE_i(x)\mathrm{d}\mu(x)$ hold.

Thus, we need $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$ and $\nabla_{w_i,y_i}\mathcal{N}$ calculated, and prove that $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$ is invertible (and give the inverse).

It is necessary to make sure which form $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$ is in according to the Fréchet derivative. Start from the map $\mathcal{N}(\mu,\mu_m;\cdot,\cdot):(L^\infty(X))\times\mathbb{R}^{m-1}\to(L^\infty(X))^\vee\times(\mathbb{R}^{m-1})^\vee$, where $\mathbb{R}^{m-1}$ is isomorphic to its dual Banach space $(\mathbb{R}^{m-1})^\vee$. Then, $\nabla_{\alpha,\overline{\beta}}\mathcal{N}\in\mathrm{Hom}^b_{\mathbb{R}}(L^\infty(X)\times\mathbb{R}^{m-1},(L^\infty(X))^\vee\times(\mathbb{R}^{m-1})^\vee)$, where Hom^b represents the set of bounded linear operators. Moreover, recall that $(\cdot)\otimes A$ is the left adjoint functor of $\mathrm{Hom}^b_{\mathbb{R}}(A,\cdot)$; then, for $\mathbb{R}$-vector spaces, $\mathrm{Hom}^b_{\mathbb{R}}(L^\infty(X)\times\mathbb{R}^{m-1},(L^\infty(X))^\vee\times(\mathbb{R}^{m-1})^\vee)\cong\mathrm{Hom}^b_{\mathbb{R}}\left((L^\infty(X)\times\mathbb{R}^{m-1})^{\otimes 2},\mathbb{R}\right)$. Thus, we can write $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$ in terms of a bilinear form on vector space $L^\infty(X)\times\mathbb{R}^{m-1}$.

From the expression of $\mathcal{N}$, we may differentiate (similarly as the calculations (A11) to (A13)):

$$\nabla_\alpha\mathcal{N}=\left(-\frac{1}{\zeta}\int_X(\cdot)(-)\sum_{i=1}^{m}w_iE_i(x)\mathrm{d}\mu(x)\,,-\frac{1}{\zeta}\int_X(\cdot)w_iE_i(x)\mathrm{d}\mu(x)\right)\tag{A14}$$

$$\nabla_{\overline{\beta}}\mathcal{N}=\left(-\frac{1}{\zeta}\int_X(\cdot)w_iE_i(x)\mathrm{d}\mu(x)\,,-\frac{\delta_{ij}}{\zeta}\int_X w_iE_i(x)\mathrm{d}\mu(x)\right)\tag{A15}$$

Consider the boundary conditions $\sum_{i=1}^m w_iE_i(x)\mathrm{d}\mu(x)=\sum_{i=1}^m\mathrm{d}\pi_i(x)=\mu(x)$ and $\int_X w_iE_i(x)\mathrm{d}\mu(x)=\int_X\pi_i(x)=w_i$. The $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$ as the Hessian form of $\mathcal{L}$ can be written as

$$\nabla_{\alpha,\overline{\beta}}\mathcal{N}=-\frac{1}{\zeta}\begin{bmatrix}\langle-,\cdot\rangle & \langle\pi_j,\cdot\rangle\\ \langle-,\pi_i\rangle & w_i\delta_{ij}\end{bmatrix}\tag{A16}$$

with $\langle\phi_1,\phi_2\rangle=\int_X\phi_1(x)\phi_2(x)\mu(x)$, or further as

$$\nabla_{\alpha,\overline{\beta}}\mathcal{N}=-\frac{1}{\zeta}\begin{bmatrix}\mathrm{d}\mu(x)\mathrm{d}\mu(x')\delta(x,x') & \mathrm{d}\pi_j(x)\\ \mathrm{d}\pi_i(x') & w_i\delta_{ij}\end{bmatrix}\tag{A17}$$

over the basis $\{\delta(x), \mathbf{e}_i\}_{x\in X, i<m}$.

By the inverse of $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$, we mean the element in $\mathrm{Hom}^b_{\mathbb{R}}\left((L^\infty(X))^\vee \times (\mathbb{R}^{m-1})^\vee, L^\infty(X)\times\mathbb{R}^{m-1}\right)$ which composes with $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$ (on the left and on the right) as identities. By the natural identity between double dual $V^{\vee\vee}\cong V$ and the tensor hom adjunction,

$$\begin{aligned}&\mathrm{Hom}^b_{\mathbb{R}}\left((L^\infty(X))^\vee\times(\mathbb{R}^{m-1})^\vee, L^\infty(X)\times\mathbb{R}^{m-1}\right)\\ \cong&\mathrm{Hom}^b_{\mathbb{R}}\left((L^\infty(X))^\vee\times(\mathbb{R}^{m-1})^\vee, (L^\infty(X)\times\mathbb{R}^{m-1})^{\vee\vee}\right)\\ \cong&\mathrm{Hom}^b_{\mathbb{R}}\left(((L^\infty(X))^\vee\times(\mathbb{R}^{m-1})^\vee)^{\otimes 2}, \mathbb{R}\right),\end{aligned} \tag{A18}$$

we can write the inverse of $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$ as a bilinear form again.

Denote $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$ in the block form $\begin{bmatrix} A & B\\ B^T & D\end{bmatrix}$. According to the block-inverse formula

$$\begin{bmatrix} A & B\\ B^T & D\end{bmatrix}^{-1} = \begin{bmatrix} A^{-1}+A^{-1}BF^{-1}B^TA^{-1} & -A^{-1}BF^{-1}\\ -F^{-1}B^TA^{-1} & F^{-1}\end{bmatrix}, \tag{A19}$$

where $F = D - B^TA^{-1}B$, whose invertibility determines the invertibility of $\begin{bmatrix} A & B\\ B^T & D\end{bmatrix}$.

Consider that $A^{-1}\in\mathrm{Hom}^b_{\mathbb{R}}((L^\infty(X)^\vee)^{\otimes 2},\mathbb{R})$; explicitly, $A^{-1}(x,y)=\delta(x,y)$. Therefore, from Equation (A17),

$$\begin{aligned}F_{ij} =&\delta_{ij}w_i - \int_{X^2}\delta(x,x')\pi_i(x)\pi_j(x')\\ =&\delta_{ij}w_i - \int_X w_iw_jE_i(x)E_j(x)\mu(x).\end{aligned} \tag{A20}$$

The matrix F is symmetric, of rank $m-1$, and strictly diagonally dominant; therefore, it is invertible. To see the strictly diagonal dominance, consider $\sum_{j=1}^m\int_X w_iw_jE_i(x)E_j(x)\mu(x) = \int_X w_iE_i(x)\sum_{j=1}^m w_jE_j(x)\mu(x) = \int_X w_iE_i(x)\mu(x) = w_i$ by applying the marginal conditions. The matrix F is of size $(m-1)\times(m-1)$ (there is no $i=m$ or $j=m$ for F_{ij}). Then, the matrix F is strictly diagonally dominant.

With all ingradients known in formula (A19), we can calculate the inverse of $\nabla_{\alpha,\overline{\beta}}\mathcal{N}$.

Following the implicit function theorem, we need $\nabla_{w_i,y_i}\mathcal{N}$; each partial derivative is an element in $L^\infty(X)^\vee\times\mathbb{R}^{m-1}$.

$$\begin{aligned}\frac{\partial\mathcal{N}}{\partial w_i} =&\left(-\int_X E_i(x)(\cdot)\mathrm{d}\mu(x), \delta_{ij}\left(1-\int_X E_i(x)\mathrm{d}\mu(x)\right)\right)\\ =&\left(-\int_X(\cdot)E_i(x)\mathrm{d}\mu(x), 0\right).\end{aligned} \tag{A21}$$

Note that if we apply the constraint $\sum_{i=1}^m w_i = 1$ to the w_is, we may set $w_m = 1-\sum_{i=1}^{m-1}w_i$ and recalculate the above derivatives as $\nabla_{w_i'}\mathcal{N} = \nabla_{w_i}\mathcal{N} - \nabla_{w_m}\mathcal{N}$ when $i\neq m$ and $\nabla_{w_m'}\mathcal{N} = \sum_{i=1}^{m-1}\nabla_{w_i}\mathcal{N}$.

$$\nabla_{y_i}\mathcal{N} = \left(\frac{1}{\zeta}\int_X(\cdot)\nabla_{y_i}g(x,y_i)w_iE_i(x)\mathrm{d}\mu(x),\ \frac{\delta_{ij}}{\zeta}\int_X\nabla_{y_i}g(x,y_i)w_iE_i(x)\mathrm{d}\mu(x)\right) \tag{A22}$$

Finally, by the Implicit Function Theorem,

$$\nabla_{w_i,y_j}(\alpha^*,\overline{\beta}^*) = -\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)^{-1}\left(\nabla_{w_i,y_j}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)$$

which fits in (A6) and (A7).

Appendix B.2. Second Derivatives

In this part, we calculate the second derivatives of $W^k_{k,\zeta}(\mu,\mu_m)$ with respect to the ingredients of μ_m, i.e., w_is and y_is, for the potential of applying Newton's method to the EDOT (which we have not implemented yet).

Using the previous results, we can further calculate the second derivatives of $W^k_{k,\zeta}$ about w_is and y_is. Differentiating (A6) and (A7) results in

$$\begin{aligned}\frac{\partial^2\mathcal{L}}{\partial w_i\partial w_j} =&\frac{1}{\zeta}\int_X g(x,y_j)\sum_{k=i,j}\left(\frac{\partial\alpha(x)}{\partial w_k}+\frac{\partial\overline{\beta}_k}{\partial w_k}\right)E_k(x)\mathrm{d}\mu(x)\\ &+\frac{1}{\zeta}\int_X\sum_{k=1}^{n}\left(\frac{\partial^2\alpha(x)}{\partial w_i\partial w_j}+\frac{\partial^2\overline{\beta}_k}{\partial w_i\partial w_j}\right)w_kE_k(x)\mathrm{d}\mu(x)\\ &+\frac{1}{\zeta^2}\int_X\sum_{k=1}^{m}\prod_{l=i,j}\left(\frac{\partial\alpha(x)}{\partial w_l}+\frac{\partial\overline{\beta}_k}{\partial w_l}\right)w_kE_k(x)\mathrm{d}\mu(x)\end{aligned}\tag{A23}$$

$$\begin{aligned}\nabla_{y_j}\frac{\partial\mathcal{L}}{\partial w_i} =&\delta_{ij}\left[\int_X\left(1-\frac{g(x,y_i)}{\zeta}\right)\nabla_{y_i}E_i(x)\mathrm{d}\mu(x)\right]\\ &+\frac{1}{\zeta}\int_X\nabla_{y_j}g(x,y_j)\left(\frac{\partial\alpha(x)}{\partial w_i}+\frac{\partial\overline{\beta}_j}{\partial w_i}\right)w_jE_j(x)\mathrm{d}\mu(x)\\ &+\frac{1}{\zeta}\int_X g(x,y_j)\nabla_{y_j}\left(\frac{\partial\alpha(x)}{\partial w_i}+\frac{\partial\overline{\beta}_j}{\partial w_i}\right)w_jE_j(x)\mathrm{d}\mu(x)\\ &+\frac{1}{\zeta^2}\int_X\sum_{k=1}^{m}g(x,y_k)\left(\frac{\partial\alpha(x)}{\partial w_i}+\frac{\partial\overline{\beta}_k}{\partial w_i}\right)\left(\nabla_{y_j}\alpha(x)+\nabla_{y_j}\overline{\beta}_k\right)w_kE_k(x)\mathrm{d}\mu(x)\end{aligned}\tag{A24}$$

$$\begin{aligned}\nabla_{y_j}\nabla_{y_i}\mathcal{L} =&\delta_{ij}\left[\int_X\nabla^2_{y_i}g(x,y_i)(1-\frac{g(x,y_i)}{\zeta})w_iE_i(x)\mathrm{d}\mu(x)\right.\\ &\left.-\frac{1}{\zeta}\int_X\left(\nabla_{y_i}g(x,y_i)\right)^2(2-\frac{g(x,y_i)}{\zeta})w_iE_i(x)\mathrm{d}\mu(x)\right]\\ &+\frac{1}{\zeta}\int_X\left(1-\frac{g(x,y_i)}{\zeta}\right)\nabla_{y_i}g(x,y_i)\cdot\left(\nabla_{y_j}\alpha(x)+\nabla_{y_j}\overline{\beta_i}\right)w_iE_i(x)\mathrm{d}\mu(x)\\ &+\frac{1}{\zeta}\int_X\left(1-\frac{g(x,y_j)}{\zeta}\right)\nabla_{y_j}g(x,y_j)\cdot\left(\nabla_{y_i}\alpha(x)+\nabla_{y_i}\overline{\beta_j}\right)w_jE_j(x)\mathrm{d}\mu(x)\\ &+\frac{1}{\zeta}\int_X\sum_{k=1}^{m}g(x,y_k)\left(\nabla_{y_i}\nabla_{y_j}\alpha(x)+\nabla_{y_i}\nabla_{y_j}\overline{\beta}_k\right)\cdot w_kE_k(x)\mathrm{d}\mu(x)\\ &+\frac{1}{\zeta^2}\int_X\sum_{k=1}^{m}g(x,y_k)\prod_{l=i,j}\left(\nabla_l\alpha+\nabla_l\overline{\beta}_k\right)\cdot w_kE_k(x)\mathrm{d}\mu(x)\end{aligned}\tag{A25}$$

Once we have the second derivatives of $g(x,y)$ on y_is, we need the second derivatives of α^* and $\overline{\beta}^*$ to build the above second derivatives. From the formula $\nabla_{w_i,y_j}(\alpha^*,\overline{\beta}^*) = -\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)^{-1}\left(\nabla_{w_i,y_j}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)$, we can differentiate

$$\begin{aligned}&\nabla_{w_k,y_l}\nabla_{w_i,y_j}(\alpha^*,\overline{\beta}^*)\\ =&-\nabla_{w_k,y_l}\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)^{-1}\left(\nabla_{w_i,y_j}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)-\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)^{-1}\left(\nabla_{w_k,y_l}\nabla_{w_i,y_j}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right).\end{aligned}\tag{A26}$$

Here, from the formula that $\nabla A^{-1} = -A^{-1}\nabla AA^{-1}$ (this is the product rule for $AA^{-1}=I$), we have

$$\nabla_{w_k,y_l}\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)^{-1} = -\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)^{-1}\nabla_{w_k,y_l}\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)\quad\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)^{-1} \tag{A27}$$

and

$$\nabla_{w_k}\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right) = -\frac{1}{\zeta}\begin{bmatrix} 0 & \delta_{jk}E_k(x)\mathrm{d}\mu(x) \\ \delta_{ik}E_k(x')\mathrm{d}\mu(x') & \delta_{ij}\delta_{jk}\end{bmatrix} \tag{A28}$$

$$\nabla_{y_k}\left(\nabla_{\alpha,\overline{\beta}}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right) = \frac{1}{\zeta^2}\begin{bmatrix} 0 & \delta_{jk}\nabla_{y_k}g(x,y_k)\mathrm{d}\pi_k(x) \\ \delta_{ik}\nabla_{y_k}g(x',y_k)\mathrm{d}\pi_k(x') & 0\end{bmatrix}. \tag{A29}$$

The last piece we need is $\left(\nabla_{w_k,y_l}\nabla_{w_i,y_j}\mathcal{N}|_{\alpha^*,\overline{\beta}^*}\right)$:

$$\frac{\partial^2\mathcal{N}}{\partial w_j\partial w_i} = (0,0), \tag{A30}$$

$$\nabla_{y_j}\frac{\partial\mathcal{N}}{\partial w_i} = \left(\frac{1}{\zeta}\int_X(\cdot)\nabla_{y_j}g(x,y_j)E_i(x)\mathrm{d}\mu(x),0\right), \tag{A31}$$

$$\begin{aligned}\nabla_{y_j}\nabla_{y_i}\mathcal{N} =& \frac{\delta_{ij}}{\zeta}\left(\int_X(\cdot)\nabla^2_{y_i}g(x,y_i)w_iE_i(x)\mathrm{d}\mu(x) + \int_X(\cdot)\left(\nabla_{y_i}g(x,y_i)\right)^2w_iE_i(x)\mathrm{d}\mu(x),\right.\\ &\left.\delta_{ik}\int_X\nabla^2_{y_i}g(x,y_i)w_iE_i(x)\mathrm{d}\mu(x) + \int_X\left(\nabla_{y_i}g(x,y_i)\right)^2w_iE_i(x)\mathrm{d}\mu(x)\right),\end{aligned} \tag{A32}$$

where in the last one, k, represents the k-th component in $\mathcal{N}$'s second part (about $\overline{\beta}$).

Appendix C. Algorithms

Appendix C.1. Algorithm: Simple EDOT

The following states the Algorithm A1 of Simple EDOT.

Algorithm A1 Simple EDOT using minibatch SGD

1: **input:** μ, k, m, ζ, N batch size, ϵ for stopping criterion, $\alpha = 0.2$ for momentum, $\beta = 0.2$ for learning rate.
2: **output:** $x_i \in X$, $w_i > 0$ such that $\mu_m = \sum_{i=1}^m w_i\delta_{x_i}$.
3: **initialize:** randomly choose $\sum_{i=1}^m w_i^{(0)}\delta_{x_i^{(0)}}$; set $t = 0$; set learning rate $\eta_t = 0.5(1+\beta t)^{-s}$ (for $t > 0$ and $s \in (0.5, 1]$).
4: **repeat**
5: Set $t \leftarrow t + 1$;
6: Sample N points $\{y_j\}_{j=1}^N \subseteq X$ from μ independently;
7: Construct $\mu_N^{(t)} = \frac{1}{N}\sum_{j=1}^N\delta_{y_j}$;
8: Calculate gradients $\nabla_{\mathbf{x}}\widehat{W}^k_{k,\zeta}(\mu_N^{(t)},\mu_m^{(t)})$ and $\nabla_{\mathbf{w}}\widehat{W}^k_{k,\zeta}(\mu_N^{(t)},\mu_m^{(t)})$;
9: Update $D\mathbf{x}_t \leftarrow \alpha D\mathbf{x}_{t-1} + \nabla_{\mathbf{x}}\widehat{W}^k_{k,\zeta}(\mu_N^{(t)},\mu_m^{(t)})$, $D\mathbf{w}_t \leftarrow \alpha D\mathbf{w}_{t-1} + \nabla_{\mathbf{w}}\widehat{W}^k_{k,\zeta}(\mu_N^{(t)},\mu_m^{(t)})$;
10: Update $\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t-1)} - \eta_t D\mathbf{x}_t$, $\mathbf{w}^{(t)} \leftarrow \mathbf{w}^{(t-1)} - \eta_t D\mathbf{w}_t$;
11: **until** $|\nabla_{\mathbf{x}}\widehat{W}^k_{k,\zeta}| + |\nabla_{\mathbf{w}}\widehat{W}^k_{k,\zeta}| < \epsilon$;
12: Set output $x_i \leftarrow x_i^{(t)}$, $w_i \leftarrow w_i^{(t)}$.

Appendix C.2. Proof of Proposition 2

Remark A1. *The convergence to a stationary point in expectation means that the liminf of the expected norm of the gradient over all the sequences considered approaches to* 0.

Proof. Discrete distributions of size m can be parameterized by $X^m \times \Delta$ in terms of $\sum_{i=1}^m p_i \delta_{y_i}$ with $(p_1, p_2, \ldots, p_m) \in \Delta$ and $y_i \in X$.

To make the SGD work, we assume that X is a path-connected subset of $\mathbb{R}^d$ of dimension d.

For $\epsilon > 0$, let $\Delta_\epsilon = \Delta_\epsilon^{m-1} = \{(p_1, p_2, \ldots, p_m) : \sum_{i=1}^m p_i = 1, p_i \geq \epsilon\}$. First, we prove the claim by assuming that $(*)$ the set of limit points of $(\mu_m^{(i)})$ is contained in $X^m \times \Delta_\epsilon$.

According to Theorem 4.10 and Corollary 4.12 of [25], to show that Algorithm A1 converges to a stationary point in expectation, i.e., $\liminf_{i\to 0} \mathbb{E}[||\nabla W_{k,\zeta}^k(\mu, \mu_m^{(i)})||^2] = 0$, one needs to check: (1). $W_{k,\zeta}^k(\mu, \mu_m^{(i)})$ is second-differentiable; (2). $\nabla W_{k,\zeta}^k(\mu, \mu_m^{(i)})$ is Lipschitz continuous; and (3). $\mathbb{E}[||\nabla \widehat{W}_{k,\zeta}^k(\mu_N^{(i)}, \mu_m^{(i)}) - \nabla W_{k,\zeta}^k(\mu, \mu_m^{(i)})||^2]$ is bounded.

(1). The second differentiablity of $W_{k,\zeta}^k(\mu, \mu_m^{(i)})$ is shown in Appendix B.2 with the second derivative calculated.

(2). As a consequence of (a), we have that $\nabla W_{k,\zeta}^k(\mu, \mu_m^{(i)})$ is continuous. Moreover, by checking each factor of $\nabla^2 W_{k,\zeta}^k(\mu, \mu_m^{(i)})$ shown in Appendix B.2, we can see that $\nabla^2 W_{k,\zeta}^k(\mu, \mu_m^{(i)})$ is bounded. (Actually, we need $\det(\nabla_{\alpha,\bar{\beta}} \mathcal{N})$ finite to make the SIM bounded, which is true in Δ_ϵ.) Therefore, $\nabla W_{k,\zeta}^k(\mu, \mu_m^{(i)})$ is Lipschitz continuous.

(3). Noise has bounded variance: Equivalently, we just need to check that $\mathrm{Var}(||\nabla W_{k,\zeta}^k(\mu_N, \mu_m)||^2)$ is finite, where μ_N is the empirical distribution with N samples taken (which is stochastic), and μ_m is the fixed discretization in $X^m \times \Delta_\epsilon$ (this need not to be the "optimal" one). $\nabla W_{k,\zeta}^k(\mu_N^{(i)}, \mu_m^{(i)})$ is continuous with respect to both $\mu_N^{(i)}$ and $\mu_m^{(i)}$; hence, it is continuous over compact space $X^N \times X^m \times \Delta_\epsilon$; hence, it is bounded by a constant C.

Thus, the proposition holds with assumption $(*)$.

Further suppose that assumption $(*)$ does not hold. Then, for any sequence $\epsilon_1, \epsilon_2 \cdots \to 0$, there always exist infinite limit points of $(\mu_m^{(i)})$ that lie outside Δ_{ϵ_k} for any $k > 0$. Therefore, we can construct a subsequence of $\mu_m^{(i)}$ converging to a point $p \in X^m \times \partial\Delta$. Thus, p is also a limit point. This contradicts the assumption that the set of limit points of $(\mu_m^{(i)})$ does not intersect with $X^m \times \partial\Delta$. The proof is then complete. □

Appendix C.3. Proof of Proposition 3

Proof. First, for each iteration in the minibatch SGD, let N be the sample (minibatch) size of μ_N for approximating μ. Let m be the size of target discretization μ_m (the output). Furthermore, let d be the dimension of X and ϵ be the error bound in the Sinkhorn calculation for the entropy-regularized optimal transference plan between μ_N and μ_m . The Sinkhorn algorithm for the positive matrix $e^{-g(x,y)/\zeta}$ (of size $N \times m$) converges linearly, which takes $\mathcal{O}(\log(1/\epsilon))$ steps to fall into a region of radius ϵ, thus contributing $\mathcal{O}(Nm \log(1/\epsilon))$ in time complexity. The inverse matrix $\mathbf{M}$ of $\nabla_{(\alpha,\bar{\beta})} \mathcal{N} = -\frac{1}{\zeta} \begin{bmatrix} A & B \\ B^T & D \end{bmatrix}$ (Equation (6)) is taken block-wise

$$\mathbf{M} = -\zeta \begin{bmatrix} A^{-1} + A^{-1}BE^{-1}B^TA^{-1} & -A^{-1}BE^{-1} \\ -E^{-1}B^TA^{-1} & E^{-1} \end{bmatrix},$$

where $E = D - B^TA^{-1}B$. Block E is constructed in $\mathcal{O}(Nm^2)$ and inverted in $\mathcal{O}(m^3)$; block $A^{-1}BE^{-1}$ takes $\mathcal{O}(Nm^2)$, as A is diagonal; and the block $A^{-1} + A^{-1}BE^{-1}B^TA^{-1}$ takes $\mathcal{O}(N^2m)$ to construct. When $m \ll N$, the time complexity in constructing $\mathbf{M}$ is $\mathcal{O}(N^2m)$. From $\mathbf{M}$ to the gradient of dual variables, the tensor contractions have complexity $\mathcal{O}((N+m)^2md)$. Finally, to get the gradient, the complexity is dominated by the second term of $\nabla_{y_i} W_{k,\zeta}^k$ (see Equation (5)), which is a contraction between a matrix Nm (i.e., $g\mathrm{d}\pi(x)$) with tensors of sizes Nmd and m^2d (two gradients on the dual variables α and β) along N and m, respectively. Thus, the final step contributes $\mathcal{O}((N+m)md)$.

The time complexity of increment steps in the SGD turns out to be $\mathcal{O}(md)$. Therefore, for L steps of the minibatch SGD, the time complexity is $\mathcal{O}((N+m)^2mdL + NmL\log(1/\epsilon))$.

For space complexity of the simple EDOT, the Sinkhorn algorithm (which can be done in position $\mathcal{O}(Nm)$) is the only iterative computation in a single SGD step, and between two SGD steps, only the resulting distribution is passed to the next step. Therefore, the space complexity is $\mathcal{O}((N+m)^2)$ coming from the $\mathbf{M}$; others are, at most, of size $\mathcal{O}(m(N+m))$. □

Appendix C.4. Adaptive Refinement via DFS: Midpoints

The pseudocode for the division algorithm of the adaptive EDOT using KD-tree refinement cutting at the midpoints is shown in Algorithm A2. The $Round_{0.5\uparrow}$ means the rounding method with 0.5 rounded up, and $Round_{0.5\downarrow}$ is that with 0.5 rounded down; thus, the discretization point is correctly partitioned.

Algorithm A2 Adaptive Refinement via Depth First Search

input: m, μ, N_0 sample size, m_* max number of points in a cell, $\mathbf{a} = (a_0, a_1, \ldots, a_{d-1})$ and $\mathbf{b} = (b_0, b_1, \ldots, b_{d-1})$ as lower/upper bounds of the region;
output: *out*: stack of subproblems (S_i, m_i, p_i) with $\sum p_i = 1, \sum m_i = m, \sum |S_i| = N_0$;
initialization: $p_0 \leftarrow 1$; T, *out* be empty stacks;
Sample N_0 points $S_0 \sim \mu$;
T.push((S_0, m, p_0, $\mathbf{a}$, $\mathbf{b}$));
while T is not empty **do**
    (S, m, p, $\mathbf{a}$, $\mathbf{b}$) $\leftarrow T$.pop();
    $l \leftarrow \text{argmax}\,\{b_i - a_i\}$, $mid \leftarrow (a_l + b_l)/2$;
    $\mathbf{a}^{(1)} \leftarrow \mathbf{a}$, $\mathbf{a}^{(2)} \leftarrow (a_0, \ldots, mid, \ldots, a_{d-1})$;
    $\mathbf{b}^{(1)} \leftarrow (b_0, \ldots, mid, \ldots, b_{d-1})$, $\mathbf{b}^{(2)} \leftarrow \mathbf{b}$;
    $S_1 \leftarrow \{\mathbf{x} \in S : x_l <= mid\}$, $S_2 \leftarrow \{\mathbf{x} \in S : x_l > mid\}$, $N_1 \leftarrow |S_1|$, $N_2 \leftarrow |S_2|$;
    $m_1 \leftarrow Round_{0.5\uparrow}\left(\frac{N_1^{d/(k+d)}}{N_1^{d/(k+d)}+N_2^{d/(k+d)}}\right)$;
    $m_2 \leftarrow Round_{0.5\downarrow}\left(\frac{N_2^{d/(k+d)}}{N_1^{d/(k+d)}+N_2^{d/(k+d)}}\right)$;
    if $m_1 = 0$ **then**
        $p_1 \leftarrow 0$, $p_2 \leftarrow (N_1 + N_2)/N_0$;
    else if $m_2 = 0$ **then**
        $p_1 \leftarrow (N_1 + N_2)/N_0$, $p_2 \leftarrow 0$;
    else
        $p_1 \leftarrow N_1/N_0$, $p_2 \leftarrow N_2/N_0$;
    end if
    for $i \leftarrow 1, 2$ **do**
        if $m_i > m_*$ **then**
            T.push(($S_i, m_i, p_i, \mathbf{a}^{(i)}, \mathbf{b}^{(i)}$));
        else if $m_i > 0$ **then**
            out.push((S_i, m_i, p_i));
        end if
    end for
end while

Algorithm A3 Adaptive EDOT Variation 1

input: m, μ, N_0 sample size, m_* max number of points in a cell, $\mathbf{a} = (a_0, a_1, \ldots, a_{d-1})$ and $\mathbf{b} = (b_0, b_1, \ldots, b_{d-1})$ as lower/upper bounds of the region, let R be resolution for occupied volume;
output: *out*: stack of subproblems (S_i, m_i, p_i) with $\sum p_i = 1, \sum m_i = m$;
initialization: $p_0 \leftarrow 1$;
procedure BUILDTREE(*node*)
    $\mathbf{a} \leftarrow node.lower$, $\mathbf{b} \leftarrow node.upper$
    $l \leftarrow \text{argmax}\,\{b_i - a_i\}$, $mid \leftarrow (a_l + b_l)/2$;
    $\mathbf{a}^{(0)} \leftarrow \mathbf{a}$, $\mathbf{a}^{(1)} \leftarrow (a_0, \ldots, a_{l-1}, mid, a_{l+1} \ldots, a_{d-1})$;
    $\mathbf{b}^{(0)} \leftarrow (b_0, \ldots, b_{l-1}, mid, \ldots, b_{l+1}, b_{d-1})$, $\mathbf{b}^{(1)} \leftarrow \mathbf{b}$;
    $S_0 \leftarrow \{\mathbf{x} \in S : x_l <= mid\}$, $S_1 \leftarrow \{\mathbf{x} \in S : x_l > mid\}$;
    for $i \leftarrow 0, 1$; **do**
        if $\prod_{j=0}^{d-1}(\mathbf{b}_j - \mathbf{a}_j) > R$ and $|S_i| > 0$; **then**
            $node.child[i] = \{occVol : 0, lower : \mathbf{a}^{(i)}, upper : \mathbf{b}^{(i)}, sample : S_i, child : Array(2), discSize : 0\}$;
            BUILDTREE($node.child[i]$)
        else if $|S_i| = 0$; **then**
            $node.child[i] = \{occVol : 0, lower : \mathbf{a}^{(i)}, upper : \mathbf{b}^{(i)}, sample : S_i, child : Array(2), discSize : 0\}$
        else
            $node.child[i] = \{occVol : \prod_{j=0}^{d-1}(\mathbf{b}_j^{(i)} - \mathbf{a}_j^{(i)}), lower : \mathbf{a}^{(i)}, upper : \mathbf{b}^{(i)}, sample : S_i, child : Array(2), discSize : 0\}$
        end if
    end for
    $node.occVol \leftarrow node.child[0].occVol + node.child[1].occVol$;
end procedure
Sample N_0 points $S_0 \sim \mu$;
$Root = \{occVol : 0, lower : \mathbf{a}, upper : \mathbf{b}, sample : S_0, child : Array(2), discSize : 0\}$;
BUILDTREE(*Root*);
$T, out \leftarrow$ empty stack;
T.push($Root$)
while T is not empty **do**
    $node \leftarrow T$.pop();
    $P_0 \leftarrow |node.child[0].sample|$, $P_1 \leftarrow |node.child[1].sample|$;
    $V_0 \leftarrow node.child[0].occVol$, $V_1 \leftarrow node.child[1].occVol$;
    $M_0 \leftarrow P_0 \cdot V_0^{k/d}$, $M_1 \leftarrow P_1 \cdot V_1^{k/d}$;
    $m_0 \leftarrow Round_{0.5\uparrow}\left(\frac{M_0^{d/(k+d)}}{M_0^{d/(k+d)} + M_1^{d/(k+d)}}\right)$;
    $m_1 \leftarrow Round_{0.5\downarrow}\left(\frac{M_1^{d/(k+d)}}{M_0^{d/(k+d)} + M_1^{d/(k+d)}}\right)$;
    if $m_0 = 0$ **then**
        $p_0 \leftarrow 0$, $p_1 \leftarrow (M_0 + M_1)/N_0$;
    else if $m_1 = 0$ **then**
        $p_0 \leftarrow (M_0 + M_1)/N_0$, $p_1 \leftarrow 0$;
    else
        $p_0 \leftarrow M_0/N_0$, $p_1 \leftarrow M_1/N_0$;
    end if
    for $i \leftarrow 0, 1$; **do**
        if $m_i > m_*$ **then**
            T.push($node.child[i]$);
        else if $m_i > 0$ **then**
            out.push((S_i, m_i, p_i));
        end if
    end for
end while

Appendix C.5. Adaptive Refinement: Variation 1

The original division algorithm of the adaptive EDOT (Algorithm A2) cuts a cell into two based on the balance of the averaged contribution of W^k per discretization point

between the divided cells. However, when the mass of μ is not distributed evenly in a cell to be cut, especially if it concentrates on a few small regions, the estimation of W^k by the diameter of a cell becomes far greater than the actual one, thereby resulting in assigning much more discretizaiton points to a cell (Figure A1). Thus, we develop the division algorithm of Variation 1 to elevate the performance in this situation by estimating the Wasserstein distance using an occupied volume of a set of sample points (usually the same sample points we used in Algorithm A2): Given a resolution $R > 0$ (the upper bound of a cell's volume can be taken as $\mathrm{Vol}(X)/N_0$, with N_0 as the size of sample points), we keep cutting the region X until each cell is either of a volume smaller than R or contains no sample points; then, we call the total volume of those nonempty cells by the occupied volume $V_{Occ}(X)$. Similar definition applies to each cell X_i.

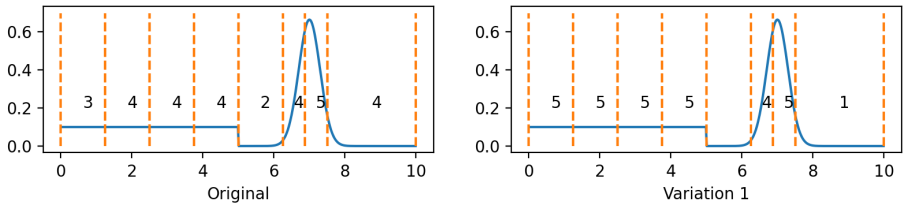

Figure A1. Divide-and-conquer strategies: original and Variation 1. Original tends to assign more atoms to vast region with small weights, while Variation 1 does better. Example: distribution on $[0, 10]$, pdf is plotted in blue curve, discretization size of each cell is in black.

After having the occupied volume of each cell, we may proceed to assign a number of discretization points to each cell. The only improvement of division algorithm Variation 1 in this part is on the Wasserstein estimation step (line 12 and 13 in Algorithm A2), where the estimated Wasserstein W^k of cell i is changed from $\mu(X_i) \cdot m_i^{-1/d} \cdot (\mathrm{diam} X_i)^k$ to $\mu(X_i) \cdot m_i^{-1/d} \cdot (V_{\mathrm{Occ}}(X_i))^{k/d}$.

It is considered that the algorithm assigning the discretization size depends on the estimation of the Wasserstein distance; however, this estimation in Variation 1 requires the occupied volume, which is calculated from leaf to root, meaning that the binary tree for occupied volume has to be built before starting Algorithm A2 with a modified estimation of W^k. Fortunately, as the cutting points in this step have to coincide with the occupied-volume-calculation step, and the sample points belonging to each cell are both needed, we may save the sample points partition in the binary tree building for occupied volume and reuse them in discretization size assigning. Therefore, the discretization size assigning step works as a tree traversal (on a subtree with the same root, which is defined by the stopping conditions in depth along each path) of the binary tree built for occupied volume calculation.

Therefore, the time complexity for the occupied volume calculation is again $\mathcal{O}(N_0 \log N_0)$, as the algorithm works in the same way as Quicksort again, and the time complexity for the rest (assigning discretization sizes) is $\mathcal{O}(m)$ as traversal on a tree of, at most, m leaves (m discretization points in total).

For space complexity, it is still $\mathcal{O}(N_0 + m)$, since after the calculation of occupied volume, the rest is only adding constant size decorations onto the subtree with, at most, m leaves mentioned above.

Appendix C.6. Adaptive Refinement: Variation 2

The "cutting in the middle" method is easy to implement and guarantee the volume decreasing while going deeper in the tree (so the depth of getting under the resolution is guaranteed). However, it is also too rigid to fit the natural gaps of the original distribution, which may critically affect the optimal location of discretization points.

Our Variation 2 is on the dimension of redefining the cutting points from midpoints along the corresponding axis to the most sparse points. The sparsity is calculated by the moving window method along an axis / component of the d-coordinates; by applying the

moving window method, we may have to sort the data points every time (since at each node, the sorting axis / component may be different). Since we still want to control the depth of the tree, a correction must be added to avoid the cutting point from locating too close to the boundaries (usually, the function $-\frac{C}{(x-a)^k(x-b)^k}$ with a and b the boundaries and $C > 0$ as a constant). One influence is that now each cell's volume (not the occupied volume) has to be calculated using the rectangular boundaries instead of being indicated only from its depth as before.

Thus, the influence on the time complexity is the following: 1. Changing the tree-building step to $N_0^2(\log N_0)^2$ in the average case, N_0^4 in worst case (if Quicksort is applied) on each node's moving-window method), and 2. Introducing a $\mathcal{O}(N_0)$ for calculating the volume on each node in the binary tree. Furthermore, it introduces, at most, $\mathcal{O}(N_0)$ additional space complexity, since each cell's volume has to be stored instead of being calculated directly from the depth.

Variation 2 can be applied together with Variation 1, since they are aiming at different parts of the algorithm. An example is shown in Figure A6.

Appendix D. Empirical Parts

Appendix D.1. Estimate $W_{k,\zeta}^k$: Richardson Extrapolation and Others

In the analysis, we may need $W_{k,\zeta}^k(\mu, \mu_m)$ to compare how discretization methods behave. However, when the μ is not discrete, we are generally not able to obtain the analytical solution to the Wasserstein distance.

In certain cases, including all examples this paper contains, the Wasserstein can be estimated by finite samples (with a large size). According to [26], for $\mu \in \mathcal{P}(X)$ in our setup (a probability measure on a compact Polish space with Borel algebra) and with $g = d_X^k \in \mathcal{C}(X^2)$ being a continuous function, the Online Sinkhorn method can be used to estimate $W_{k,\zeta}^k$. The Online Sinkhorn needs a large number of samples for μ (in batch) to be accurate.

In our paper, as X are compact subsets in $\mathbb{R}^n$, and μ has a continuous probability density function, we may use the Richardson Extrapolation method to estimate the Wasserstein distance between μ and μ_m, which may require fewer samples and fewer computations (the Sinkhorn twice with different sizes).

Our examples are on intervals or rectangles, in which two grids Λ_1 of N points and Λ_2 of rN points (N and rN are both integers) can be constructed naturally for each. With μ determined by a smooth probability density function ρ, let $\mu_{(N)}$ be the normalization of $\sum_{i=1}^N \rho(\Lambda_i)\delta_{\Lambda_i}$ (this may not be a probability distribution, so we use its normalization). From a continuity of ρ and the boundedness of the dual variables α and β, we can conclude that

$$\lim_{N\to\infty} W_{k,\zeta}^k(\mu_{(N)}, \mu_m) = W_{k,\zeta}^k(\mu, \mu_m).$$

Let $W_{k,\zeta}^k(\mu_{(N)}, \mu_m)$ be a function of $1/N$; to apply Richardson extrapolation, we need the exponent of the lowest term of $1/N$ in the expansion $W_{k,\zeta}^k(\mu_{(N)}, \mu_m) = W^* + \mathcal{O}(1/N^h) + \mathcal{O}((1/N)^{h+1})$, where $W^* = W_{k,\zeta}^k(\mu, \mu_m)$.

Consider that

$$\left|W_{k,\zeta}^k(\mu, \mu_m) - W_{k,\zeta}^k(\mu_{(N)}, \mu_m)\right| \le W_{k,\zeta}^k(\mu, \mu_{(N)}).$$

Since $W_{k,\zeta}(\mu_{(N)}, \mu) \propto N^{-1/d}$, we may conclude that $h = k/d$, where d is the dimension of X. Figure A2 shows an empirical example in a $d = 1$ and $k = 2$ situation.

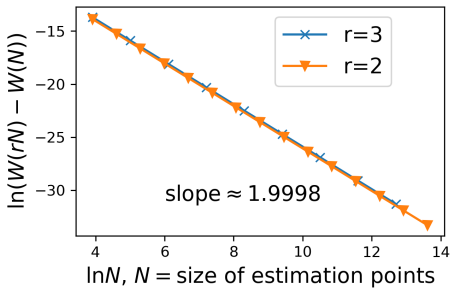

Figure A2. Richardson Extrapolation: the power of the expansion about N^{-1}. We take the EDOT μ_5 of example 2 (1-dim truncated normal mixture) as the target μ_m and use evenly positioned μ_N for different Ns to estimate. The y-axis is $\ln(W^2_{2,0.01}(\mu_{(rN)}, \mu_5) - W^2_{2,0.01}(\mu_{(N)}, \mu_5))$, where $r = 2$ and $r = 3$ are calculated. With $\ln N$ as x-axis, linearity can be observed. The slopes are both about -1.9998, which represent the exponent of the leading non-constant term of $W^2_{2,0.01}(\mu_{(N)}, \mu_5)$ on N, while the theoretical result is $r = -k/d = -2$. The differences are from higher order terms on N.

Appendix D.2. Example: The Sphere

The CW complex structure of the unit sphere S^2 is constructed as follows: let $(1, 0, 0)$, the point on the equator, be the only dimension-0 structure, and let the equator be the dimension-1 structure (line segment $[0, 1]$ attached to the dimension-0 structure by identifying both end points to the only point $(1, 0, 0)$). The dimension-2 structure is the union of two unit discs, which is identified to the south/north hemisphere of S^2 by stereographic projection:

$$\pi_{\pm N} : (X, Y) \to \frac{1}{1 + X^2 + Y^2}\left(2X, 2Y, \pm(X^2 + Y^2 - 1)\right) \tag{A33}$$

with respect to the north / south pole.

Spherical Geometry

The spherical geometry is the Riemannian manifold structure induced by embedding onto the unit sphere in $\mathbb{R}^3$.

The geodesic between two points is the shorter arc along the great circle determined by the two points. In their $\mathbb{R}^3$ coordinates, $d_{S^2}(\mathbf{x}, \mathbf{y}) = \arccos(\langle \mathbf{x}, \mathbf{y} \rangle)$. Composed with stereographic projections, the distance in terms of CW complex coordinates can be calculated (and be differentiated).

The gradient about $\mathbf{y}$ (or its CW coordinate) can be calculated via the above formulas. In practice, the only problem is when $\mathbf{x} = \pm\mathbf{y}$ function arccos at ± 1 is singular. From the symmetry of sphere on the rotation along axis $\mathbf{x}$, the derivatives of distance along all directions are the same. Therefore, we may choose the radial direction on the CW coordinate (unit disc). Furthermore, the differentiations are primary to calculate.

Appendix D.3. A Note on the Implementation of SGD with Momentum

There is a slight difference between our implementation of the SGD and the algorithm provided in the paper. In the implementation, we give two different learning rates to the positions (y_is) and the weights (w_is), as moving along positions is usually observed much slower than moving along weights. Empirically, we make the learning rates on the positions be exactly three times the learning rates on the weights at each SGD iteration. With this change, the convergence is faster, but we do not have a theory or empirical evidence to show that a fixed ratio of three is the best choice.

Implementing and testing the Newton's method (hybrid with SGD) and other improved SGD methods could be good problems to work on.

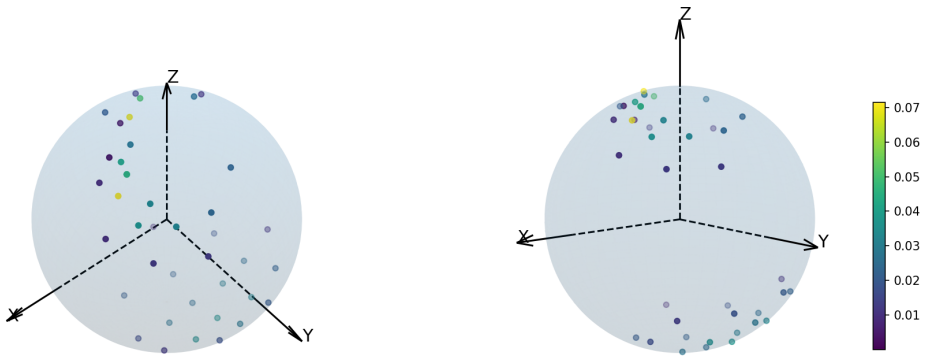

Figure A3. The sphere example with 3D discretization (same as the paper) on two view directions. Colors of dots represent the weights of each atom in the distribution.

Appendix D.4. An Example on Transference Plan with Adaptive EDOT

We now illustrate the performance of the adaptive EDOT on a 2D optimal transport task. Let $X = Y = [0,1]^2$, $c = d_X$ be the Euclidean distance, $g = d_X^2$, and the marginal μ, ν be truncated normal (mixtures), where μ has only two components and ν has five components. Figure A4 plots the McCann Interpolation of the OT plan between μ and ν (shown in red dots) and its discrete approximations (weights are color coded) with $m = n = 30$. With $m = n = 10$, the adaptive EDOT results were as follows: $W^k_{k,\zeta}(\mu, \mu_{10}) = 1.33 \times 10^{-2}$, $W^k_{k,\zeta}(\nu, \nu_{10}) = 1.30 \times 10^{-2}$, and $W^k_{k,\zeta}(\gamma, \gamma_{10,10}) = 2.71 \times 10^{-2}$. With $m = n = 30$, the adaptive EDOT results were as follows: $W^k_{k,\zeta}(\mu, \mu_{30}) = 9.62 \times 10^{-3}$, $W^k_{k,\zeta}(\nu, \nu_{30}) = 9.18 \times 10^{-3}$, and $W^k_{k,\zeta}(\gamma, \gamma_{30,30}) = 1.516 \times 10^{-2}$. The naive sampling results were as follows: $W^k_{k,\zeta}(\mu, \mu'_{30}) = 1.75 \times 10^{-2}$, $W^k_{k,\zeta}(\nu, \nu'_{30}) = 1.58 \times 10^{-2}$, and $W^k_{k,\zeta}(\gamma, \gamma'_{30,30}) = 3.95 \times 10^{-2}$. The adaptive EDOT approximated the quality of 900 naive samples with only 100 points on a four-dimensional transference plan.

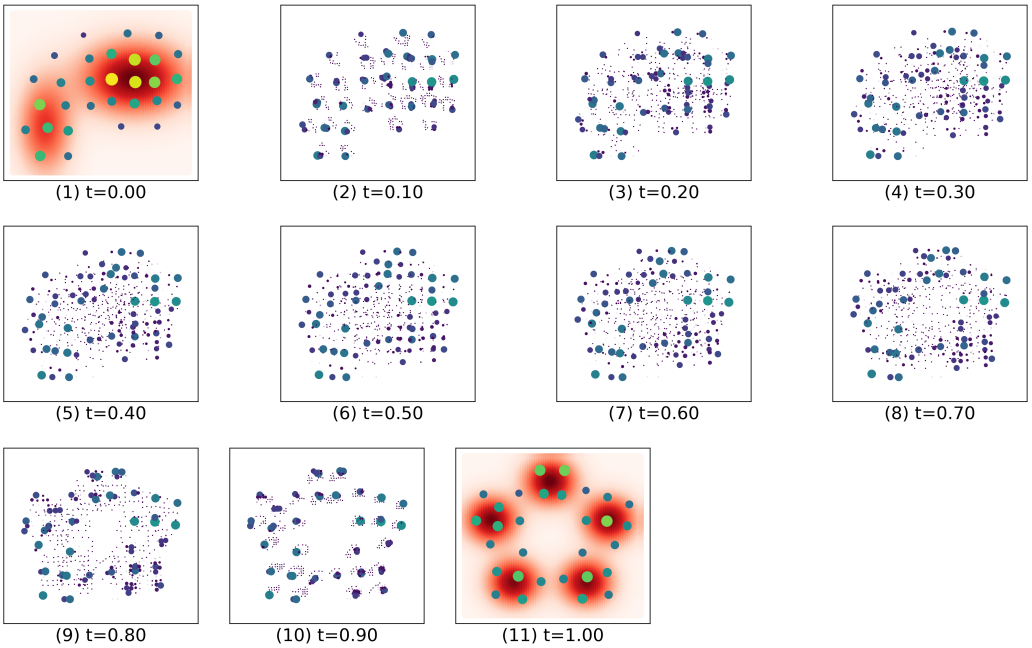

Figure A4. The McCann interpolation figures in finer time resolution for visualizing the transference plan from (**1**)–(**11**). It is a refined figure of the original paper. We see can see that the larger bubbles (representing a large probability mass) moved in a short distance, and smaller pieces moved longer.

Appendix D.5. Example: Swiss Roll

In this case, the underlying space X_{swiss} is a the Swiss Roll, which is a 2D rectangular strip embedded in $\mathbb{R}^3$: $(\theta, z) \mapsto (\theta, \theta, z)$ in cylindrical coordinates. μ_{swiss} is a truncated normal mixture on a (θ, z)-plane. Samples $S_{\text{swiss}} \sim \mu_{\text{swiss}}$ over X_{swiss} are shown in Figure A5 (left) embedded in 3D and in Figure A6a as isometric into $\mathbb{R}^2$.

By following the Euclidean metric in $\mathbb{R}^3$, Figure A5 (right) plots the EDOT solution μ_m through adaptive cell refinement (Algorithm A2) with $m = 50$. The resulting cell structure is shown as colored boxes. The corresponding partition of S_{swiss} is shown on Figure A6a, with samples contained in a cell marked by the same color. According to Figure A5 (right), the points in μ_m were mainly located on the strip, with only one point off in the most sparse cell (yellow cell located in the bottom in the figure).

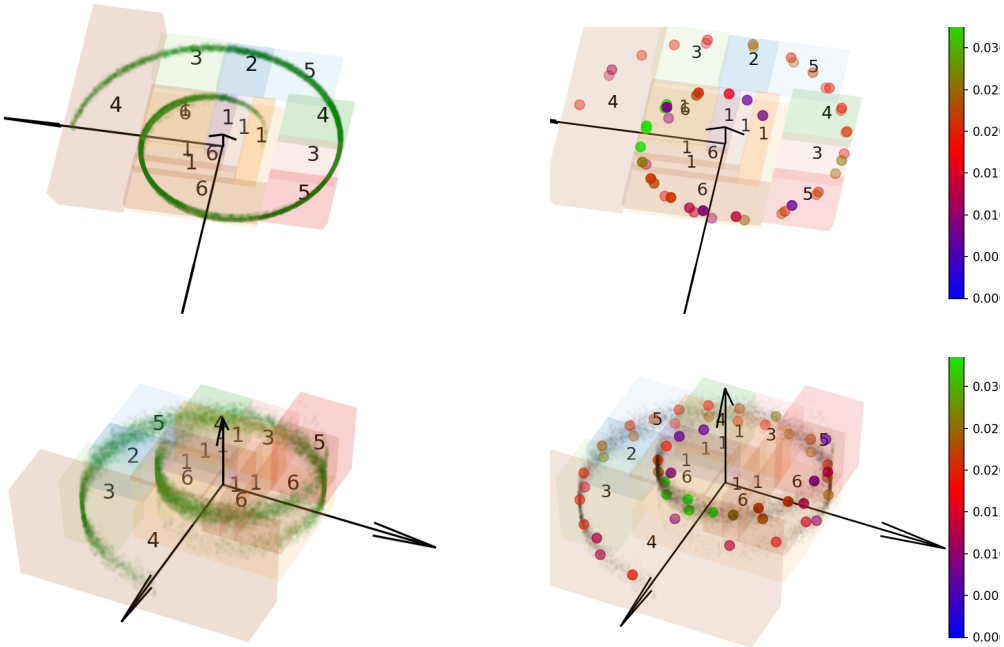

Figure A5. Discretization of a distribution supported on a Swiss Roll. **Left**: A total of 15,000 samples from the truncated normal mixture distribution μ_{swiss} over X_{swiss}. **Right**: A 50-point 3D discretization using Variation 2 of Algorithm A2; the refinement cells are shown in colored boxes.

On the other hand, consider the metric on X_{swiss} induced by the isometry from the Swiss Roll as a manifold to a strip on $\mathbb{R}^2$. A more intrinsic discretization of μ_{swiss} can be obtained by applying the EDOT through a refinement on the coordinate space—the (2D) strip. The partition of S_{swiss} is shown on Figure A6b, and the resulting discretization μ_{50} is shown in Figure A6c. Notice that all 50 points were located on the (locally) high density region of the Swiss Roll. We observe from Figure A6a,b that the 3D partition pulled disconnected and intrinsically remote regions together, while the 2D partition maintained the intrinsic structure.

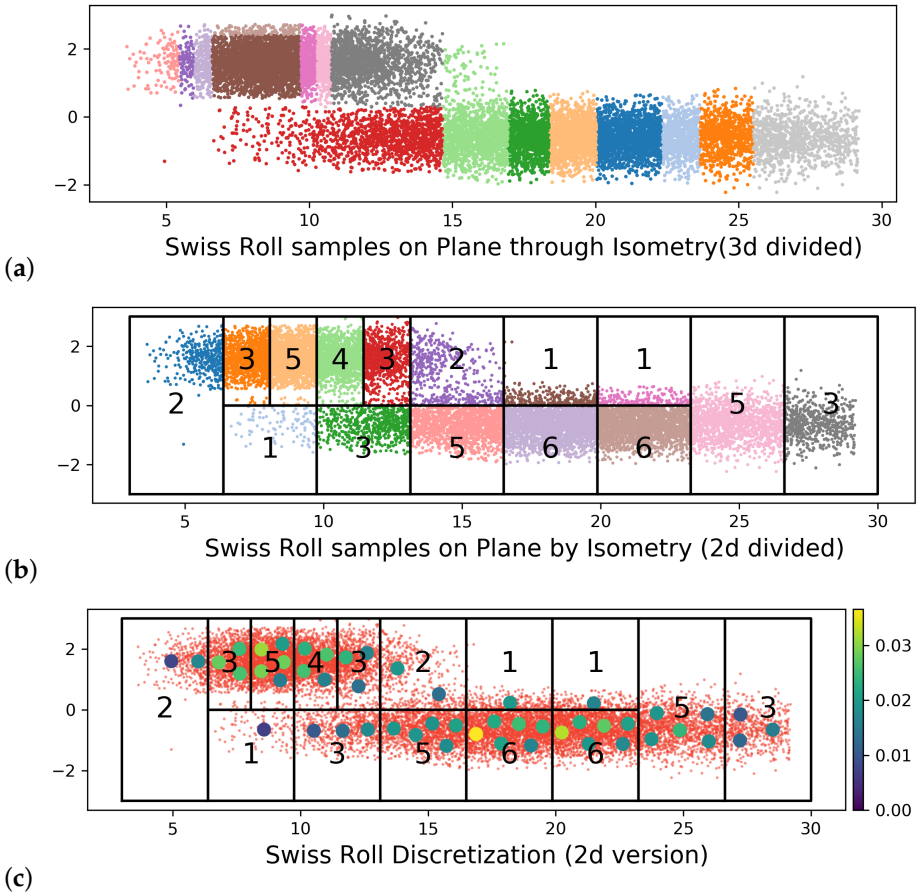

Figure A6. Swiss Roll under isometry. (**a**) Refinement cells under 3D Euclidean metric (one color per samples from a refinement cell). (**b**) Refinement cells under 2D metric induced by the isometry. (**c**) EDOT of 50 points with respect to the 2D refinement. $\zeta = 0.01 \times \text{diam}^2$ for all.

Appendix D.6. Example: Figure Densities

We used OpenCV to process the figures. The cat figure is a gray-scale figure of size 180×180. Variation 1 was used in cutting the figure into pieces, since the figure contains some sparse regions on which the original division algorithm does not work well.

For the colored figures (Starry Night and Girl with a Pearl Earring) in Figure 1, we process the three channels independently, then plotted the colored dots, and finally combined them as corresponding channels in a colored file. In the reconstruction of Starry Night, we made the size of the colored dots of same size with a modified color value according to the weights. Furthermore, for Girl with a Pearl Earring, we used pure color ((255,0,0) as red, etc.) and changed the size of the dots (with an area proportional to the weights).

Appendix D.7. Example: Simple Barycenter Problems

The EDOT in simple form (no divide and conquer) can solve barycenter problems. The idea is simple: the gradient of a sum of functions is the sum of gradients of each function. Thus, to find the discrete barycenter of size m for several distributions μ_i, we take the objective to be the sum of Wasserstein distances (raised to power k for rationality), whose gradient-to-target distribution is the sum of gradients between the discretization and each target distribution. This method only works for the simple EDOT, since there is no locality in barycenter problems. After a division, the weights of each target distribution in each cell of the partition may be different, so there is inter-cell transport in the optimal transport plan, which the current algorithm cannot deal with.

We can see in Figure A7 that the simple EDOT-Equal (no divide and conquer) achieved similar results as the non-regularized discretization in [18], whereas the EDOT produced a better approximation of the barycenter by taking advantage of changing weights freely.

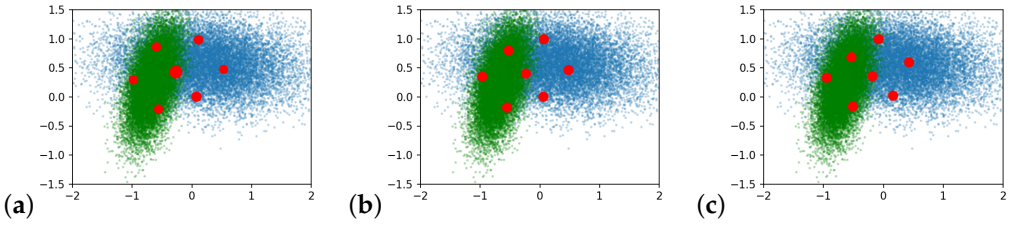

Figure A7. A 7-point barycenter of two Gaussian distributions: (**a**): EDOT, area of dots represent the weights, $W^2_{sum} = 0.7322$; (**b**): EDOT-Equal, $W^2_{sum} = 0.7380$; (**c**): [18], $W^2_{sum} = 0.7389$; W are regularized with $\zeta = 0.04$.

References

1. Kantorovich, L.V. On the translocation of masses. *J. Math. Sci.* **2006**, *133*, 1381–1382. [CrossRef]
2. Peyré, G.; Cuturi, M. Computational optimal transport. *Found. Trends Mach. Learn.* **2019**, *11*, 355–607. [CrossRef]
3. Villani, C. *Optimal Transport: Old and New*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008; Volume 338.
4. Janati, H.; Muzellec, B.; Peyré, G.; Cuturi, M. Entropic Optimal Transport between Unbalanced Gaussian Measures has a Closed Form. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 10468–10479.
5. Aude, G.; Cuturi, M.; Peyré, G.; Bach, F. Stochastic optimization for large-scale optimal transport. *arXiv* **2016**, arXiv:1605.08527.
6. Allen-Zhu, Z.; Li, Y.; Oliveira, R.; Wigderson, A. Much faster algorithms for matrix scaling. In Proceedings of the 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), Berkeley, CA, USA, 15–17 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 890–901.
7. Lin, T.; Ho, N.; Jordan, M.I. On the efficiency of the Sinkhorn and Greenkhorn algorithms and their acceleration for optimal transport. *arXiv* **2019**, arXiv:1906.01437.
8. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In Proceedings of the Advances in Neural Information Processing Systems, Harrahs and Harveys, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2292–2300.
9. Sinkhorn, R.; Knopp, P. Concerning nonnegative matrices and doubly stochastic matrices. *Pac. J. Math.* **1967**, *21*, 343–348. [CrossRef]
10. Wang, J.; Wang, P.; Shafto, P. Sequential Cooperative Bayesian Inference. In Proceedings of the International Conference on Machine Learning, PMLR, Online/Vienna Austria, 12–18 July 2020; pp. 10039–10049.
11. Tran, M.N.; Nott, D.J.; Kohn, R. Variational Bayes with intractable likelihood. *J. Comput. Graph. Stat.* **2017**, *26*, 873–882. [CrossRef]
12. Overstall, A.; McGree, J. Bayesian design of experiments for intractable likelihood models using coupled auxiliary models and multivariate emulation. *Bayesian Anal.* **2020**, *15*, 103–131. [CrossRef]
13. Wang, P.; Wang, J.; Paranamana, P.; Shafto, P. A mathematical theory of cooperative communication. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17582–17593.
14. Luise, G.; Rudi, A.; Pontil, M.; Ciliberto, C. Differential properties of sinkhorn approximation for learning with wasserstein distance. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–8 December 2018; pp. 5859–5870.
15. Accinelli, E. A Generalization of the Implicit Function Theorems. 2009. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1512763 (accessed on 3 February 2021)
16. Weed, J.; Bach, F. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli* **2019**, *25*, 2620–2648. [CrossRef]
17. Dudley, R.M. The speed of mean Glivenko-Cantelli convergence. *Ann. Math. Stat.* **1969**, *40*, 40–50. [CrossRef]
18. Claici, S.; Chien, E.; Solomon, J. Stochastic Wasserstein Barycenters. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 999–1008.
19. Mérigot, Q. A multiscale approach to optimal transport. In *Proceedings of the Computer Graphics Forum*; Blackwell Publishing Ltd.: Oxford, UK, 2011; Volume 30, pp. 1583–1592.
20. Solomon, J.; de Goes, F.; Peyré, G.; Cuturi, M.; Butscher, A.; Nguyen, A.; Du, T.; Guibas, L. Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains. *ACM Trans. Graph.* **2015**, *34*, 1–11. [CrossRef]
21. Staib, M.; Claici, S.; Solomon, J.M.; Jegelka, S. Parallel Streaming Wasserstein Barycenters. In *Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Long Beach, CA, USA, 2017; Volume 30.
22. Beugnot, G.; Genevay, A.; Greenewald, K.; Solomon, J. Improving Approximate Optimal Transport Distances using Quantization. *arXiv* **2021**, arXiv:2102.12731.
23. Jacobs, M.; Léger, F. A fast approach to optimal transport: The back-and-forth method. *Numer. Math.* **2020**, *146*, 513–544. [CrossRef]

24. Genevay, A.; Chizat, L.; Bach, F.; Cuturi, M.; Peyré, G. Sample complexity of sinkhorn divergences. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, Japan, 16–18 April 2019; pp. 1574–1583.
25. Bottou, L.; Curtis, F.E.; Nocedal, J. Optimization methods for large-scale machine learning. *Siam Rev.* **2018**, *60*, 223–311. [CrossRef]
26. Mensch, A.; Peyré, G. Online sinkhorn: Optimal transport distances from sample streams. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1657–1667.