AVERAGED MIRROR DESCENT AND DUAL GRADIENT METHODS: CONVERGENT ALGORITHMS FOR ENTROPIC GROMOV–WASSERSTEIN PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

The Gromov–Wasserstein (GW) distance measures the discrepancy between metric measure (mm) spaces and identifies optimal alignments between them based solely on their intrinsic structure. Since it identifies isomorphic mm spaces, it provides a natural notion of distance for heterogeneous datasets which may admit isomorphic representations. In order to accelerate computation of GW distances, many practitioners employ entropic regularization to obtain an Entropic GW (EGW) problem. The most popular EGW solver is the Mirror Descent (MD) algorithm, which reduces EGW computations to an iterative process where an entropic optimal transport (EOT) problem is solved at each iteration. Despite its widespread use, the convergence of MD for this problem has only been established for restricted classes of costs. On the other hand, a recently proposed dual gradient method is available for general costs, but requires a choice of step size which depends on the regularization parameter. To address these two issues, we introduce Averaged Mirror Descent (AMD), which averages consecutive MD steps, and prove its convergence for arbitrary costs. Then, we establish that the dual gradient method with a fixed step size also converges for arbitrary costs at the cost of a more complicated iteration. In both cases, we also account for inexact iterations which are inescapable in practice. We compare the empirical performance of these methods across various settings and, in particular, show that AMD and the dual gradient method both converge on an example where classical MD fails.

1 INTRODUCTION

Optimal Transport (OT) provides a principled framework for comparing and transforming probability distributions according to the geometry of the underlying metric space (Villani, 2008). Computing OT between finitely discrete distributions supported on N points requires solving a linear program in N^2 variables which can be done in $O(N^3 \log(N))$ operations via the network simplex algorithm (Peyré & Cuturi, 2019). This undesirable cubic scaling can be circumvented by regularizing OT using an entropic penalty yielding the Entropic Optimal Transport (EOT) problem which can be solved in $O(N^2 \log(N))$ operations via Sinkhorn's diagonal scaling algorithm (Cuturi, 2013). This has since driven considerable interest in the ML community, with applications spanning generative modeling (Arjovsky et al., 2017; Gulrajani et al., 2017; Genevay et al., 2018; De Bortoli et al., 2021; Lipman et al., 2023), sampling (Vargas et al., 2023; Liu et al., 2025) and image recognition (Li et al., 2013; Rubner et al., 2000), among others.

An important consideration when using OT in applications is that it requires specifying a cost of transporting points between the supports of the underlying distributions which captures a desired correspondence. In cases where the distributions represent datasets of different types, the cost is generally chosen in an *ad hoc* manner and may fail to account for intra-domain relationships between the underlying objects. This is particularly important when comparing objects which may have many natural representations within the same space such as in aligning shapes related by an unknown rotation or matching graphs across different node representations. The Gromov–Wasserstein (GW) distance, introduced in Mémoli (2011), enables comparing metric measure spaces by comparing their intra-domain distortions. Crucially, the GW distance vanishes if and only if the two metric measure spaces are isomorphic and provides a coupling $\pi^\star$ which can be thought of as an optimal

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

fuzzy alignment of the spaces. These properties have made GW distances a versatile tool across many domains including language modeling (Alvarez-Melis & Jaakkola, 2018; Kawakita et al., 2024), generative modeling (Bunne et al., 2019; Klein et al., 2024), graph matching (Brogat-Motte et al., 2022; Xu et al., 2019; Rioux et al., 2024b), and single-cell genomics (Demetci et al., 2022; Klein et al., 2024).

Despite these appealing properties, computing the GW distance generally requires solving a non-convex Quadratic Program (QP) which is computationally hard. Nevertheless, fast iterative methods for approximately solving regularized GW problems have been proposed in Peyré et al. (2016); Solomon et al. (2016) and appear to work well in practice. These methods regularize the GW problem using the entropic penalty and linearize the objective so that each iteration only requires solving an EOT problem; this is the core of the Mirror Descent (MD) algorithm. Although this method is widely used in practice, the convergence analysis of this method is limited and the impact of inexactness in Sinkhorn's method for solving the EOT problem is unaccounted for in the literature. Related approaches to accelerating GW computations include convex/conic relaxations (Séjourné et al., 2021; Vincent-Cuaz et al., 2022), semi-definite relaxations (Chen et al., 2024), and slicing-based methods (Vayer et al., 2019; Gong et al., 2026), but Entropic Gromov–Wasserstein (EGW) remains the most popular method for practical applications.

Another line of work, Rioux et al. (2024a), introduced provably convergent algorithms for the EGW problem between Euclidean distributions by leveraging a variational formulation introduced in Zhang et al. (2024). Subsequently, Houry et al. (2026) extended this dual formulation to costs of conditionally negative type (CNT) and showed that standard gradient methods with a fixed step size (independent of the regularization parameter) converge and coincide with MD for CNT costs. Moreover, their work shows, via a numerical example, that MD iterations can oscillate and fail to converge if the cost is not CNT. Most recently, Rioux et al. (2026) established a variational form of EGW which covers arbitrary finitely discrete metric measure spaces and developed convergent algorithms with a stepsize that decays to 0 as the regularization parameter is decreased; the case of CNT costs effectively corresponds to conditionally negative semidefinite GW cost matrices in that work. Hence, two natural questions arise: (1) can a simple modification of MD be shown to converge for arbitrary costs and (2) can a dual gradient method for arbitrary costs be shown to converge with a stepsize that does not depend on the regularization parameter.

The present work answers both of these questions positively and additionally shows that the error from inexact resolution of the EOT problems can be accounted for in the analysis of these algorithms. To address (1) we introduce a variant of the mirror descent algorithm which takes less aggressive steps and is provably convergent. We dub this method Averaged Mirror Descent (AMD), as we average the standard MD step with the previous iterate. AMD can be shown to coincide with a conditional gradient method, enabling us to obtain non-asymptotic convergence bounds. In addition, we go beyond the classical analysis of conditional gradient methods by establishing its convergence when subproblems are solved inexactly. As for (2), we leverage the dual formulation of EGW from Rioux et al. (2026) and the proof technique from Houry et al. (2026) to show that a standard gradient method with fixed stepsize (independent of the regularization parameter) converges for arbitrary costs. Crucially, the iterates for the dual gradient method do not coincide with the MD updates for indefinite GW cost matrices which indicates that the failure of MD to converge on some numerical examples is due to its coarse approximation of the true gradient step.

2 Optimal transport and Gromov–Wasserstein problems

Optimal transport. Fix two finite sets $\mathcal{X}_0$ and $\mathcal{X}_1$, and let μ_0, μ_1 be probability distributions on $\mathcal{X}_0, \mathcal{X}_1$ respectively (denoted as $\mu_0 \in \mathcal{P}(\mathcal{X}_0)$, $\mu_1 \in \mathcal{P}(\mathcal{X}_1)$). In addition, let $c : \mathcal{X}_0 \times \mathcal{X}_1 \to \mathbb{R}$ be a cost function. The OT problem reads $\mathsf{OT}_c(\mu_0, \mu_1) \coloneqq \inf_{\pi \in \Pi(\mu_0, \mu_1)} \int c d\pi$, where $\Pi(\mu_0, \mu_1)$ denotes the collection of all couplings of (μ_0, μ_1), i.e., distributions on $\mathcal{X}_0 \times \mathcal{X}_1$ whose $\mathcal{X}_i$-marginal is μ_i for $i \in \{0, 1\}$. The EOT problem regularizes the OT problem as in Cuturi (2013)

$$\mathsf{EOT}_c^{\varepsilon}(\mu_0, \mu_1) \coloneqq \inf_{\pi \in \Pi(\mu_0, \mu_1)} \int c d\pi + \varepsilon \mathrm{KL}(\pi \| \mu_0 \otimes \mu_1). \tag{1}$$

where $\varepsilon > 0$ is a regularization parameter and $\mathrm{KL}(\pi \| \mu_0 \otimes \mu_1)$ is the Kullback-Leibler (KL) divergence between π and the product measure $\mu_0 \otimes \mu_1$ defined by $\int \log(d\pi / d(\mu_0 \otimes \mu_1)) d\pi$ when

$\pi \ll \mu_0 \otimes \mu_1$ and takes the value $+\infty$ otherwise.[1] The added penalty makes the resulting problem strictly convex and enforces a certain structure on solutions of Equation 1; this structure is at the heart of Cuturi (2013) which proposed to solve the EOT problem via Sinkhorn iterations, see Section A for details.

Gromov-Wasserstein Problems. Given finite sets $\mathcal{X}_0, \mathcal{X}_1, \mu_0 \in \mathcal{P}(\mathcal{X}_0), \mu_1 \in \mathcal{P}(\mathcal{X}_1)$, and kernels $\kappa_0 : \mathcal{X}_0 \times \mathcal{X}_0 \to \mathbb{R}$ and $\kappa_1 : \mathcal{X}_1 \times \mathcal{X}_1 \to \mathbb{R}$, the p-GW problem ($p > 0$) is given by Mémoli (2011)

$$\mathsf{GW}_p(\mu_0, \mu_1) \coloneqq \inf_{\pi \in \Pi(\mu_0, \mu_1)} \iint |\kappa_0(x, x') - \kappa_1(y, y')|^p \, d\pi \otimes \pi(x, y, x', y').$$

As with EOT it has been proposed in Peyré et al. (2016); Solomon et al. (2016) to penalize GW using the KL divergence, viz.

$$\mathsf{EGW}_p^\varepsilon(\mu_0, \mu_1) \coloneqq \inf_{\pi \in \Pi(\mu_0, \mu_1)} \iint |\kappa_0(x, x') - \kappa_1(y, y')|^p \, d\pi \otimes \pi(x, y, x', y') + \varepsilon \mathrm{KL}(\pi \| \mu_0 \otimes \mu_1),$$

where $p > 0$ and $\varepsilon > 0$. The resulting problem is called the EGW problem and is generally non-convex unless $\varepsilon > 0$ is sufficiently large. However, in applications $\varepsilon > 0$ is taken to be small in order to approximate solutions of GW. Concretely, Karumanchi et al. (2025) established that solutions of the regularized problem converge to those of the original problem exponentially fast (in ε) if the quadratic form above is concave, but that this convergence is as slow as linear otherwise.

Given that $\mathcal{X}_0 = \{x_0^{(i)}\}_{i=1}^{N_0}, \mathcal{X}_1 = \{x_1^{(j)}\}_{j=1}^{N_1}$ are finite sets, solving EGW is equivalent to solving a regularized QP in $k = N_0 N_1$ variables which we write as

$$\mathsf{EGW}_p^\varepsilon(\mu_0, \mu_1) = \inf_{\substack{\mathrm{A}x = b \\ x \geq 0}} \left\{ \frac{1}{2} x^\mathsf{T} \mathrm{C} x - \varepsilon \mathsf{H}(x) + \varepsilon \mathsf{H}_{\mu_0} + \varepsilon \mathsf{H}_{\mu_1} \right\}, \tag{2}$$

where $\{x \in \mathbb{R}^k : \mathrm{A}x = b, x \geq 0\}$ accounts for the coupling constraints, C is the cost matrix with entries $\mathrm{C}_{i,j} = 2 \left| \kappa_0(x_0^{(i_0)}, x_0^{(j_0)}) - \kappa_1(x_1^{(i_1)}, x_1^{(j_1)}) \right|^p$, where (i_0, i_1) and (j_0, j_1) are pairs such that $i = (i_1 - 1)N_0 + i_0$ and $j = (j_1 - 1)N_0 + j_0$, $\mathsf{H}(x) = -\sum_{i=1}^k x_i \log(x_i)$ is Shannon's entropy which we treat as a function on the non-negative orthant, and $\mathsf{H}_{\mu_0} = -\sum_{x \in \mathcal{X}_0} \mu_0(\{x\}) \log(\mu_0(\{x\}))$ and similarly for H_{μ_1}. Throughout, we opt to work with the symmetric matrix $\mathrm{M} = \frac{1}{2}(\mathrm{C} + \mathrm{C}^\mathsf{T})$ which satisfies $z^\mathsf{T} \mathrm{M} z = z^\mathsf{T} \mathrm{C} z$ for every $z \in \mathbb{R}^k$.

The problem formulations in Peyré et al. (2016) and Scetbon et al. (2022) are equivalent to this setup under the transformation $x = \mathrm{vec}(\mathrm{P})$ where $\mathrm{P} \in \mathbb{R}^{N_1 \times N_0}$ is a coupling matrix with entries $\mathrm{P}_{lm} = \pi\left(\left\{\left(x_0^{(m)}, x_1^{(l)}\right)\right\}\right)$ for some $\pi \in \Pi(\mu_0, \mu_1)$ and $\mathrm{vec}(\mathrm{P}) \in \mathbb{R}^k$ is the vector obtained by stacking the columns of P. In the following, we let $\mathcal{K} \coloneqq \{x \in \mathbb{R}^k : \mathrm{A}x = b, x \geq 0\}$ denote the set of admissible couplings and write $\|\mathrm{M}\|_{\max} \coloneqq \max_{i,j=1}^k |\mathrm{M}_{ij}|$

3 Averaged Mirror Descent

Here and in the sequel, we assume that μ_0, μ_1 assign positive mass to each point in $\mathcal{X}_0, \mathcal{X}_1$. As described previously, our proposed algorithm, Algorithm 1, consists of two steps. First, we solve a linearization of Equation 2 at the previous iterate (which is simply an EOT problem) and then we average the solution of the linearized problem with the previous iterate. The averaging step effectively introduces a step size and therefore prevents excessively large steps at a given iteration.

Algorithm 1: Averaged Mirror Descent (AMD)

Input: initialization $x_0 \in \mathcal{K}$, maximum iteration number J, step sequence $(\alpha_j)_{j \in \mathbb{N}} \subset (0, 1]$
for $j = 0, 1, \ldots, J - 1$ **do**
  $\tilde{x}_j \leftarrow$ approximate solution of $\min_{x \in \mathcal{K}} \left\{ x_j^\mathsf{T} \mathrm{M} x - \varepsilon \mathsf{H}(x) \right\}$;
  $x_{j+1} \leftarrow (1 - \alpha_j) x_j + \alpha_j \tilde{x}_j$;

[1] The notation $\pi \ll \mu_0 \otimes \mu_1$ indicates that π is absolutely continuous with respect to $\mu_0 \otimes \mu_1$, that is, $\mu_0 \otimes \mu_1(A) = 0$ implies that $\pi(A) = 0$ for any measurable set A.

As noted above, the subproblem defining $\tilde{x}_j$ is simply an EOT problem with the cost vector $\mathrm{M}^\mathsf{T} x_j$ and x_{j+1} is obtained by taking a convex combination of x_j and $\tilde{x}_j$. We underscore that the MD algorithm (Scetbon et al., 2022, Algorithm 1) performs the same iterations as Algorithm 1 with the fixed step sequence $\alpha_j = 1$ for each $j \in \mathbb{N}$, but the convergence properties of MD for this problem for arbitrary cost matrices are unknown. Nevertheless, it remains the most popular EGW solver for practical applications.

As $\min_{x \in \mathcal{K}} \left\{ x_j^\mathsf{T} \mathrm{M} x - \varepsilon \mathsf{H}(x) \right\}$ is an EOT problem, it is solved using Sinkhorn iterations in practice (see Section A for details on Sinkhorn's method). An important limitation of this algorithm is that it does not return an exact solution in a finite number of steps. Taking this into account, we first analyze Algorithm 1 assuming that each subproblem is solved exactly and, later, account for inexactness.

3.1 Convergence under exact solutions

We now establish the non-asymptotic convergence properties of Algorithm 1 assuming that all EOT subproblems are solved exactly. Our analysis hinges on the fact that Algorithm 1 is simply a conditional gradient type algorithm as analyzed in Ghadimi (2019). In effect, the algorithm studied in that work applies to the composite problem $\inf_{x \in X} \{f(x) + h(x)\}$ where $X \subset \mathbb{R}^d$ is a closed convex set, $f : X \to \mathbb{R}$ is continuously differentiable with Hölder continuous gradient, and $h : X \to \mathbb{R}$ is μ-strongly convex. Starting from some $x_0 \in X$, their iterations read

$$\tilde{x}_j \leftarrow \operatorname*{arg\,min}_{u \in X} \left\{ \nabla f(x_j)^\mathsf{T} u + h(u) \right\}, \quad x_{j+1} \leftarrow (1 - \alpha_j) x_j + \alpha_j \tilde{x}_j.$$

Notably, these iterations coincide exactly with Algorithm 1 by setting $X = \mathcal{K}$, $f(x) = \frac{1}{2} x^\mathsf{T} \mathrm{M} x$, and $h(x) = -\varepsilon \mathsf{H}(x)$ which is shown to be ε-strongly convex on the probability simplex Δ_k with respect to $\|\cdot\|_1$. This connection is central to the proof of convergence, see Section B.1 for full details.

Theorem 1 (Convergence of Algorithm 1). *Fix the step sequence $\alpha_j = \min\left\{ \frac{\varepsilon}{4\|\mathrm{M}\|_{\max}}, 1 \right\}$ and assume that $\min_{x \in \mathcal{K}} \left\{ \tilde{x}^\mathsf{T} \mathrm{M} x - \varepsilon \mathsf{H}(x) \right\}$ can be solved exactly for each $\tilde{x} \in \mathcal{K}$. Then, the iterates $\tilde{x}_j, x_j$ of Algorithm 1 satisfy*

$$\min_{j=0}^{J-1} \|x_j - \tilde{x}_j\|_1^2 \leq \frac{8}{3J\varepsilon} \max\{4\|\mathrm{M}\|_{\max}/\varepsilon, 1\} \Big(f(x_0) - \varepsilon \mathsf{H}(x_0) - \inf_{\mathcal{K}} \{f - \varepsilon \mathsf{H}\} \Big).$$

We now discuss the interpretation of the convergence metric $\min_{j=0}^{J-1} \|x_j - \tilde{x}_j\|_1^2$.

Remark 2 (Convergence criterion). *Following the discussion surrounding Equation (2.5) in Ghadimi (2019) and using the properties of the problem derived in Section B.1, if $\|x_j - \tilde{x}_j\|_1 \leq \gamma / (\|\mathrm{M}\|_{\max} \sup_{x,y \in \mathcal{K}} \|x - y\|_1)$ for some $\gamma > 0$, there exists $p \in \partial h(\tilde{x}_j)$ (the subdifferential of h at $\tilde{x}_j$) for which $\langle -(p + \nabla f(\tilde{x}_j)), x - \tilde{x}_j \rangle \leq \gamma$ for each $x \in \mathcal{K}$. $\tilde{x}_j$ is then interpreted as a γ-approximate stationary point since a minimizer $\bar{x} \in \mathcal{K}$ for this problem is such that there exists $\bar{p} \in \partial h(\bar{x})$ for which $\langle -(\bar{p} + \nabla f(\bar{x})), x - \bar{x} \rangle \leq 0$ for every $x \in \mathcal{K}$. We prove this assertion, show that h is classically differentiable at each iterate, and provide further context in Section C.1.*

3.2 Convergence under inexact solutions

Next, we show that a similar convergence result holds when the EOT subproblems are not solved exactly. The following convergence result is crucial for this analysis, see Section A.1 for its proof.

Lemma 3 (Sinkhorn convergence). *Let $x^\star$ be the unique solution of $\mathsf{EOT}_c^\varepsilon(\mu_0, \mu_1)$. Then, for each $\delta > 0$, there exists a number of steps n depending on $\delta, \varepsilon, \|c\|_\infty, \mu_0$, and μ_1 after which Sinkhorn's algorithm, as written in Algorithm 3, outputs a vector, $\tilde{x}$, satisfying $\|\tilde{x} - x^\star\|_1 \leq e^\delta - 1$.*

An exact expression for n can be found in Equation (28) of Rioux et al. (2024a). Since each $\tilde{x}$ is a simplex vector (see Section A) the infinity norm of the cost for the EOT subproblems in Algorithm 1 is always bounded above by $\|\mathrm{M}\|_{\max}$ so that a fixed number of Sinkhorn iterations can be used to solve all problems to within a desired accuracy in the 1-norm.

Theorem 4 (Convergence under inexactness). *Fix $\delta > 0$ and suppose that all EOT problems are solved to an accuracy $e^\delta - 1 \leq \frac{1}{2}$ in the 1-norm and set $\alpha_j = \min\{\frac{\varepsilon}{2(\|\mathrm{M}\|_{\max} + \varepsilon)}, 1\}$. Then, the*

iterates $\tilde{x}_j, x_j$ in Algorithm 1 satisfy

$$\min_{j=0}^{J-1} \|x_j - \tilde{x}_j\|_1^2 \leq \frac{4}{J\varepsilon}\frac{1}{\alpha_j}\left(f(x_0) - \varepsilon\mathsf{H}(x_0) - \inf_{\mathcal{K}}\{f - \varepsilon\mathsf{H}\} + \omega_\varepsilon(e^\delta - 1)\right) + \frac{8}{\varepsilon}\omega_\varepsilon(e^\delta - 1).$$

where $\omega_\varepsilon : p \in [0,2] \mapsto \|\mathrm{M}\|_{\max} p + \varepsilon\left(\left(\frac{1}{2}\log(k-1)\right)p - \frac{p}{2}\log(\frac{p}{2}) - (1-\frac{p}{2})\log(1-\frac{p}{2})\right)$.

Notably, the optimization error decouples into a term scaling as $O(1/J)$ as in Theorem 1 and an approximation error depending on $\omega_\varepsilon(e^\delta - 1)$. As $e^\delta - 1 \leq \frac{3}{2}\delta$, since $e^\delta - 1 \leq \frac{1}{2}$ by assumption, and $\log(1-x) \geq \frac{-x}{1-x}$ for $x < 1$ so that $-(1-x)\log(1-x) \leq x$ for $x \leq 1$ (the equality case corresponds to $0 \leq 1$), we obtain $\omega_\varepsilon(e^\delta - 1) \leq \frac{3}{2}\left(\|\mathrm{M}\|_{\max}\delta + \frac{\delta\varepsilon}{2}[\log((4(k-1))/(3\delta)) + 1]\right)$ so that $\omega_\varepsilon(e^\delta - 1) = O(\delta\varepsilon\log(k/\delta))$ as $\delta \downarrow 0$ and the error can be made arbitrarily small. The proof of Theorem 4 is included in B.2.

Remark 5 (Inexact convergence criterion)**.** *Due to inexactness, both x_j and $\tilde{x}_j$ may not belong to $\mathcal{K}$ so that the interpretation of the convergence metric in Theorem 4 is more opaque than in Theorem 1. Nevertheless, by leveraging the approximation results from the proof of Theorem 4 we establish in Section C.2 that if $\|x_j - \tilde{x}_j\|_1 \leq \tau$ for some $\tau > 0$, there exists some $\tilde{x}_j^\star \in \mathcal{K}$ for which $\tilde{x}_j^\star$ is a $\|\mathrm{M}\|_{\max}\sup_{x,y\in\mathcal{K}}\|x-y\|_1(\tau + e^\delta - 1)$-approximate stationary point and $\|\tilde{x}_j^\star - \tilde{x}_j\|_1 \leq e^\delta - 1$ provided that the EOT subproblems are solved to within an accuracy of $e^\delta - 1$ in the 1-norm.*

4 Dual gradient method

Dual gradient methods (Rioux et al., 2024a; Houry et al., 2026; Rioux et al., 2026) rely on a certain reformulation of the GW problem which trades off the quadratic program over $\mathcal{K}$ for nested optimization problems which depend on the coupling only in a linear manner. In effect, the recent work Rioux et al. (2026) uses the fact that M can always be expressed as $\mathrm{M} = \mathrm{B}_1^\mathsf{T}\mathrm{B}_1 - \mathrm{B}_0^\mathsf{T}\mathrm{B}_0$ for matrices $\mathrm{B}_0 \in \mathbb{R}^{r_0\times k}$ and $\mathrm{B}_1 \in \mathbb{R}^{r_1\times k}$ with $r_0 + r_1 \leq k$ to establish that the generic EGW problem $\inf_{x\in\mathcal{K}}\left\{\frac{1}{2}x^\mathsf{T}\mathrm{M}x - \varepsilon\mathsf{H}(x)\right\}$ is equivalent to the problem

$$\inf_{u\in\mathbb{R}^{r_0}}\sup_{v\in\mathbb{R}^{r_1}}\left\{\frac{1}{2}\|u\|^2 - \frac{1}{2}\|v\|^2 + \inf_{x\in\mathcal{K}}\left\{\left(\mathrm{B}_1^\mathsf{T}v - \mathrm{B}_0^\mathsf{T}u\right)^\mathsf{T}x - \varepsilon\mathsf{H}(x)\right\}\right\} \tag{3}$$

in the sense that their optimal values coincide and, if $x^\star$ solves Equation 2 then $(u^\star, v^\star) = (\mathrm{B}_0x^\star, \mathrm{B}_1x^\star)$ solves Equation 3. Conversely, if $(u^\star, v^\star)$ solve the inf-sup problem then there exists $x^\star \in \arg\min_{x\in\mathcal{K}}\left\{\left(\mathrm{B}_1^\mathsf{T}v - \mathrm{B}_0^\mathsf{T}u\right)^\mathsf{T}x - \varepsilon\mathsf{H}(x)\right\}$ which solves Equation 2 and $(u^\star, v^\star) = (\mathrm{B}_0x^\star, \mathrm{B}_1x^\star)$. Proposition 8 in Rioux et al. (2026) shows that Equation 3 can be written as

$$\inf_{\mathbb{R}^{r_0}}\ell_\varepsilon \text{ for } \ell_\varepsilon : u \in \mathbb{R}^{r_0} \mapsto \frac{1}{2}\|u\|^2 + \inf_{x\in\mathcal{K}}\left\{\frac{1}{2}x^\mathsf{T}\mathrm{B}_1^\mathsf{T}\mathrm{B}_1x - u^\mathsf{T}\mathrm{B}_0x - \varepsilon\mathsf{H}(x)\right\},$$

where ℓ_ε is smooth for any $\varepsilon > 0$ with gradient

$$\nabla\ell_\varepsilon(u) = u - \mathrm{B}_0x_u \text{ where } x_u \in \arg\min_{x\in\mathcal{K}}\left\{\frac{1}{2}x^\mathsf{T}\mathrm{B}_1^\mathsf{T}\mathrm{B}_1x - u^\mathsf{T}\mathrm{B}_0x - \varepsilon\mathsf{H}(x)\right\}.$$

The same reference proposes to minimize ℓ_ε using projected gradient methods with and without acceleration and derives convergence bounds for the two methods under the inexactness of Sinkhorn's solutions. However, the convergence rates depend inversely on ε and require setting the stepsize as a function of ε, leading to potentially small stepsizes.

Algorithm 2: Dual gradient method

Input: initialization $u_0 = \mathrm{B}_0x_0$ for $x_0 \in \mathcal{K}$, maximum iteration count J
for $j = 0, \ldots, J-1$ **do**
  $x_{u_j} \leftarrow$ approximate solution of $\min_{x\in\mathcal{K}}\left\{\frac{1}{2}x^\mathsf{T}\mathrm{B}_1^\mathsf{T}\mathrm{B}_1x - u_j^\mathsf{T}\mathrm{B}_0x - \varepsilon\mathsf{H}(x)\right\}$;
  $u_{j+1} \leftarrow u_j - (u_j - \mathrm{B}_0x_{u_j})$;

We now introduce an inexact gradient method (Algorithm 2) for minimizing ℓ_ε which does not require setting a stepsize. However, each iteration requires solving the strongly convex problem

$\min_{x\in\mathcal{K}} \left\{\frac{1}{2}\|\mathrm{B}_1 x\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x - \varepsilon \mathsf{H}(x)\right\}$ at least approximately. Algorithm 1 can thus be thought of as taking coarse approximations of the gradient; this can explain why the original MD method can fail on some examples, as it takes large steps in a direction which is not necessarily a gradient.

Algorithm 2 is related to the framework analyzed in Houry et al. (2026) for CNT costs. Crucially, a CNT cost κ_i on $\mathcal{X}_i \times \mathcal{X}_i$ ($i \in \{0,1\}$) is such that $\kappa_i(x,x') = \|\varphi_i(x) - \varphi_i(x')\|^2_{\mathcal{H}_i}$ where $\mathcal{H}_i$ is a Hilbert space with norm $\|\cdot\|_{\mathcal{H}_i}$ and $\varphi_i : \mathcal{X}_i \to \mathcal{H}_i$ is a continuous map. With this, the quadratic form $\iint |\kappa_0(x,x') - \kappa_1(y,y')|^2 d\pi \otimes \pi(x,y,x',y')$ decomposes as $C(\mu_0,\mu_1) - 8\|\int \varphi_0(x)\varphi_1(y)^\mathsf{T} d\pi(x,y)\|^2_{\mathrm{HS}} - 4\int \|\varphi_0(x)\|^2_{\mathcal{H}_0}\|\varphi_1(y)\|^2_{\mathcal{H}_1} d\pi(x,y)$ for any $\pi \in \Pi(\mu_0,\mu_1)$, where $\|\cdot\|_{\mathrm{HS}}$ is the Hilbert-Schmidt norm and $C(\mu_0,\mu_1)$ depends only on moments of the marginals (see Lemma C.1 in Houry et al. (2026)). It follows that CNT costs fall within the variational framework for the special case $\mathrm{B}_1 = 0$ when the marginals are finitely discrete, but with an added linear term which can be absorbed into the minimization over $\mathcal{K}$.

4.1 Convergence under exact solutions

We first derive convergence bounds for Algorithm 2 assuming that we have access to exact solutions of $\min_{x\in\mathcal{K}} \left\{\frac{1}{2}\|\mathrm{B}_1 x\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x - \varepsilon \mathsf{H}(x)\right\}$. The proof follows by adapting the result of Houry et al. (2026) for CNT costs to the present dual formulation which holds for arbitrary costs, see Section B.3.

Theorem 6 (Dual method convergence)**.** *Fix $J \geq 1$, $\varepsilon > 0$, and let $(u_j)_{j=0}^{J-1}$ be the iterates from Algorithm 2 assuming that $\min_{x\in\mathcal{K}} \left\{\frac{1}{2}\|\mathrm{B}_1 x\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x - \varepsilon \mathsf{H}(x)\right\}$ can be solved exactly. Then,*

$$\min_{j=0}^{J-1} \|\nabla \ell_\varepsilon(u_j)\|^2 = \min_{j=0}^{J-1} \|u_j - u_{j+1}\|^2 \leq \frac{2}{J}\left(\ell_\varepsilon(u_0) - \inf_{\mathbb{R}^{r_0}} \ell_\varepsilon\right)$$

Remark 7 (Comparison with Theorem 1)**.** *Compared with Theorem 1, the convergence metric is straightforward; a point which is arbitrarily close to stationary will be met once J is large enough. Furthermore, the rate does not depend on ε^{-1} and so does not deteriorate as ε becomes small, which is often the region of interest for practitioners. However, this algorithm requires a more complicated gradient call as noted previously.*

Remark 8 (The case of CNT costs)**.** *As noted previously, the CNT costs analyzed in Houry et al. (2026) correspond to the case $\mathrm{B}_1 = 0$ in the current setting. In this case, the iterations coincide with the MD algorithm, and the analysis is the same as in the aforementioned work.*

Naturally, a point $u \in \mathbb{R}^{r_0}$ is called η-stationary for ℓ_ε with $\eta \geq 0$ if $\|\nabla \ell_\varepsilon(u)\| \leq \eta$. The notions of primal and dual approximate stationarity are related as follows, see Section B.5.

Theorem 9 (Primal vs. dual stationarity)**.** *Fix $\varepsilon > 0$, $\eta \geq 0$, and suppose that $\bar{u} \in \mathbb{R}^{r_0}$ is η-stationary for ℓ_ε. Then, $x_{\bar{u}}$, the unique solution of $\arg\min_{x\in\mathcal{K}} \left\{\frac{1}{2}x^\mathsf{T}\mathrm{B}_1^\mathsf{T}\mathrm{B}_1 x - \bar{u}^\mathsf{T}\mathrm{B}_0 x - \varepsilon \mathsf{H}(x)\right\}$ is $2\|\mathrm{B}_0\|_{1,2}\eta$-stationary for the primal problem as defined in Remark 2.*

4.2 Convergence under inexact solutions

Since the solution to $\min_{x\in\mathcal{K}} \left\{\frac{1}{2}\|\mathrm{B}_1 x\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x - \varepsilon \mathsf{H}(x)\right\}$ is not available exactly in practice, we extend Theorem 6 to the setting where we only have access to an inexact gradient oracle. Complete details can be found in Section B.4.

Theorem 10 (Dual convergence under inexactness)**.** *Fix $\tau > 0$ and suppose that the problem $\min_{x\in\mathcal{K}} \left\{\frac{1}{2}\|\mathrm{B}_1 x\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x - \varepsilon \mathsf{H}(x)\right\}$ can be solved to an accuracy τ, i.e., $\|x_{u_j} - x^\star_{u_j}\|_1 < \tau$ for all j where $x^\star_{u_j}$ denotes the true solution of the problem. Fix $J \geq 1$, $\varepsilon > 0$, and let $(u_j)_{j=0}^{J-1}$ be the iterates obtained from Algorithm 2. Then,*

$$\min_{j=0}^{J-1} \|\nabla \ell_\varepsilon(u_j)\|^2 \leq \frac{2}{J}\left(\ell_\varepsilon(u_0) - \inf_{\mathbb{R}^{r_0}} \ell_\varepsilon\right) + \|\mathrm{B}_0\|^2_{1,2}\tau^2 \text{ where } \|\mathrm{B}_0\|_{1,2} = \sup_{\|z\|_1 \leq 1} \|\mathrm{B}_0 z\|_2.$$

We first describe how to solve these quadratic subproblems and, afterwards, provide more context on the derived convergence rate.

Remark 11 (Gradient computation)**.** *As $\min_{x\in\mathcal{K}} \left\{\frac{1}{2}\|\mathrm{B}_1 x\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x - \varepsilon \mathsf{H}(x)\right\}$ is simply a regularized QP, we may apply dual gradient methods, e.g., Algorithms 5 and 6 in Rioux et al. (2026),*

to solve it.[2] *In fact, Propositions 11 and 12 of that work already establish how close the iterates get to the true solution so that the conditions of Theorem 10 can be shown to hold. However, the number of iterations to get a τ-approximate solution scales as ε^{-1}. It is noteworthy that the proof of Theorem 10 does not apply for this convex subproblem as the inequalities are the "wrong" direction.*

Remark 12 (Approximation error)**.** *As in Theorem 4, the rate decouples into a term scaling as $O(1/J)$ and an approximation error which depends on the approximation accuracy τ. The error and overall runtime then depend on the solver used to approximate the gradient.*

While we cannot directly track $\|\nabla \ell_\varepsilon(u_j)\|^2$, $\|\nabla \ell_\varepsilon(u_j)\| \leq \|\mathrm{B}_0(x_{u_j} - x^\star_{u_j})\| + \|u_j - \mathrm{B}_0 x_{u_j}\| \leq \|\mathrm{B}_0\|_{1,2}\tau + \|u_{j+1} - u_j\|$ so that terminating the algorithm when the approximate gradient $u_{j+1} - u_j = -(u_j - \mathrm{B}_0 x_{u_j})$ has small magnitude is reasonable.

5 Experiments

In this section we empirically validate the convergence rates derived in Sections 3 and 4 across a variety of different settings. We also study the performance of these algorithms on a reconstruction of an example from Houry et al. (2026), where the original MD algorithm fails to converge for a cost which is not CNT.

In the sequel, the implementation of the dual gradient method is identical to Algorithm 4 in Rioux et al. (2026) with the stepsize taken to 1. We call this algorithm Variational (1) to emphasize this fact. The regularized convex quadratic subproblems used to approximate the gradients at each iteration are solved using Algorithm 6 from that work with the stepsize chosen per experiment. We stress, however, that any other solver could be used to solve this inner problem as long as it can be shown to satisfy $\|x_{u_j} - x^\star_{u_j}\|_1 < \tau$ for some $\tau > 0$. We also compare Variational (1) to Algorithms 2 and 4 in Rioux et al. (2026) with theoretical stepsizes computed from a Lipschitz constant L which we hence refer to as Accelerated variational (L) and Variational (L). The regularized quadratic subproblems are solved in the same manner. We report the values of L per experiment in Section E.2.

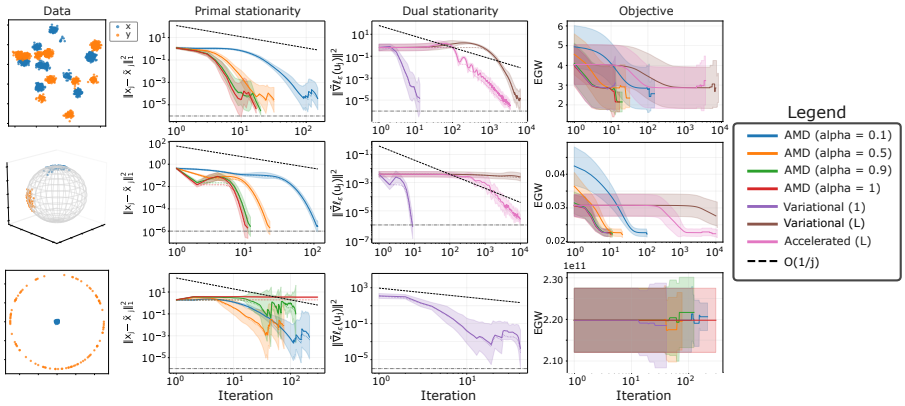


Figure 1: Convergence of AMD (Algorithm 1) for $\alpha \in \{0.1, 0.5, 0.9, 1\}$ ($\alpha = 1$ recovering classical MD), and the dual gradient method (Algorithm 2) across three datasets, which are visualized in the left-most panel. Primal stationarity records $\|x_j - \tilde{x}_j\|_1^2$ for AMD. Dual stationarity records $\|u_j - u_{j+1}\|^2$ for simple variational methods: Variational (1) and Variational (L), and $\|u_j - z_j\|^2$ for Accelerated (L) where z_j is a second sequence used for momentum (see Algorithm 2 in Rioux et al. (2026)). The right-most plot shows the corresponding EGW value. In both stationarity panels, solid lines show the per-iteration value and dotted lines its running minimum, with the $O(1/j)$ rate overlaid in black. Solid lines report the geometric mean for stationarity and the arithmetic mean for the objective over 10 seeds; shaded regions indicate one standard deviation in log and non-log space respectively.

Firstly, we illustrate convergence of AMD for $\alpha \in \{0.1, 0.5, 0.9, 1\}$ as well as the dual gradient method in three settings: a 10-component Gaussian mixture with $x \in \mathbb{R}^{10}$ and $y \in \mathbb{R}^{15}$ under the

[2]Both of these methods are iterative and only require solving EOT problems at each iteration. This can be done in $O(N_0 N_1)$ operations (up to log factors) via Sinkhorn's algorithm (Algorithm 3).

Method	Gaussian mixture		Sphere		Gaussian-ring	
	Runtime (s)	Conv.	Runtime (s)	Conv.	Runtime (s)	Conv.
AMD ($\alpha = 0.1$)	38.2 ± 13.1	$10/10$	0.62 ± 0.49	$10/10$	9.42 ± 3.39	$8/10$
AMD ($\alpha = 0.5$)	7.69 ± 2.74	$10/10$	0.117 ± 0.090	$10/10$	4.66 ± 3.76	$9/10$
AMD ($\alpha = 0.9$)	4.19 ± 1.55	$10/10$	0.061 ± 0.048	$10/10$	7.85 ± 6.18	$6/10$
AMD ($\alpha = 1$)	3.66 ± 1.41	$10/10$	0.054 ± 0.042	$10/10$	15.10 ± 3.19	$0/10$
Variational (1)	9.06 ± 2.94	$10/10$	0.128 ± 0.113	$10/10$	106.1 ± 57.8	$10/10$
Variational (L)	2366 ± 694	$8/10$	53 ± 58	$0/10$	—	—
Accelerated variational (L)	1155 ± 300	$10/10$	92 ± 74	$7/10$	—	—

Table 1: Mean solver wall-clock runtime in seconds ($\pm$ one standard deviation) over 10 seeds as well as the number of runs that reached the prescribed tolerance before reaching the maximum number of iterations. The measured runtime does not include the setup needed for the methods and in particular does not take into account the cost decompositions needed for both methods (for details see Section D.2).

squared Euclidean cost which is CNT, following Scetbon et al. (2022), rotated point clouds on a sphere under the geodesic cost which is also CNT, and a Gaussian mapped to a ring both centered at 0 under Euclidean cost raised to the sixth power which is not CNT. We use fixed values of α rather than the step sizes required by Theorems 1 and 4 as they scale as $\varepsilon/\|\mathrm{M}\|_{\max}$ (similarly to the variational methods of Rioux et al. (2026)) and are prohibitively small in all three settings. We report their values together with the corresponding stationarity plots in Section E.2 for completeness.

The entries of μ_0, μ_1 are drawn uniformly at random and normalized to sum to one. In the sphere example μ_1 is assigned the same weights as μ_0, so that the two spaces are isomorphic. We set $\varepsilon = 0.1$ for the Gaussian mixture, $\varepsilon = 0.01$ for the sphere and $\varepsilon = 1$ for the Gaussian-to-ring example. See Section D for dataset generation and other experimental details.

Figure 1 shows the primal approximate stationarity $\|x_j - \tilde{x}_j\|_1^2$ for AMD, the approximate dual gradient $\tilde{\nabla}\ell(u_j)$ for the variational methods, and the EGW value across iterations. In the two CNT examples, AMD with $\alpha < 1$ converges more slowly than classical MD, although $\alpha = 0.5$ and 0.9 require a comparable number of iterations. In the non-CNT Gaussian-ring example, classical MD fails to converge in 300 iterations for all 10 seeds whereas the smaller values of α are seen to converge on at least some of the runs. By contrast, Variational (1) converges for all experiments albeit slower than AMD. These methods are seen to abide by the theoretical $O(1/j)$ rate when they converge. As for the dual gradient methods, Variational (1) is seen to converge much faster than Variational (L) and its accelerated counterpart, as the stepsize $1/L$ used for those methods is very small. This also explains why those methods fail to converge in $10\,000$ iterations for some of the CNT examples; we thus omit these methods for the non-CNT example as they converge too slowly.

We underscore that all methods attain similar objective values. The large EGW values obtained in the Gaussian-ring example are due to the fact that the sixth-power cost produces a large constant term that hides the decrease of the (effective) objective. Since only the interaction term depends on the coupling (Section E.1), we plot it separately in Figure 3, where both the decrease and the oscillations of classical MD are clearly visible. The dual methods require a similar number of iterations to AMD with $\alpha \in \{0.9, 1\}$ on the CNT examples, and fewer than every AMD variant on the non-CNT example. However, each iteration requires a more complex inner solve, so their total runtime is consistently higher (Table 1). We report the average number of Sinkhorn calls in Table 4 and discuss the complexity of the methods in Section D.2.2. Finally, we note that the same stopping tolerance is applied to different quantities for the primal and dual methods, so their iteration counts are not directly comparable.

Next, we consider a reconstruction of the example introduced by Houry et al. (2026), which compares squared Euclidean costs, which are CNT, with sixth-power Euclidean costs, which are not CNT. Their example illustrates that the classical MD iteration can oscillate between two distinct couplings for non-CNT costs. We use two uniformly weighted point clouds containing 15 points each and study two regularization regimes, $\varepsilon = 0.1$ and $\varepsilon = 0.01$. The details of the point cloud generation and implementation are described in Section D.3.

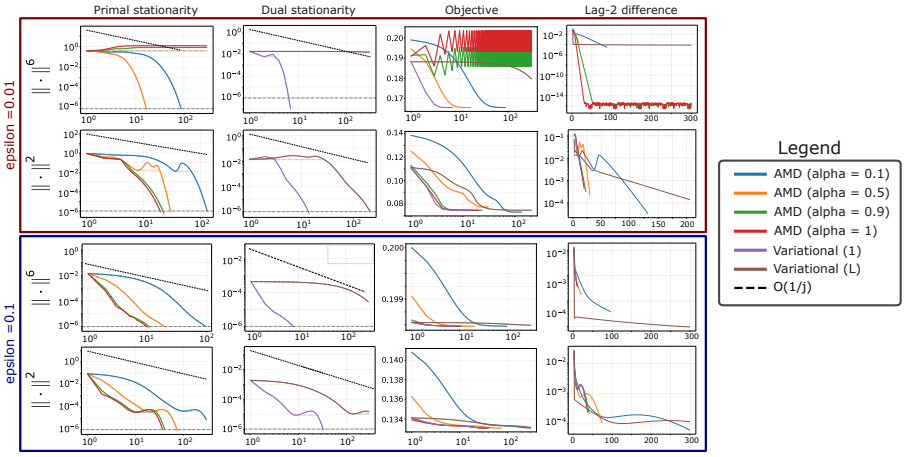


Figure 2: Primal stationarity i.e. $\|x_j - \tilde{x}_j\|_1^2$ for AMD for $\alpha \in \{0.1, 0.5, 0.9, 1\}$, dual stationarity i.e. $\|u_j - u_{j+1}\|^2$ for Variational (1) and Variational (L), the EGW value and the lag-two difference between couplings at each iteration i.e. $\|x_j - x_{j+2}\|$ for all methods. The dotted line denotes the $O(1/j)$ reference curve. The top block shows the results for $\varepsilon = 0.01$ and the bottom for $\varepsilon = 0.1$.

Figure 2 reports the primal approximate stationarity for AMD, the dual stationarity for Variational (1) and Variational (L), the EGW value, and the lag-2 difference $\|x_j - x_{j+2}\|_F$ for the various problems. When $\varepsilon = 0.01$ with the sixth-power cost, the objective oscillates for $\alpha \in \{0.9, 1\}$, whereas it decreases monotonically for $\alpha \in \{0.1, 0.5\}$ and for both dual methods. For the oscillating runs, the stationarity proxy remains bounded away from zero while the lag-2 difference becomes negligible, showing that the couplings oscillate with cycle two. This can also be seen in Section E, where we visualize the different couplings. Every other run in Figure 2 decreases monotonically and meets the stopping criterion, including all runs at $\varepsilon = 0.1$: the oscillation is therefore a joint effect of a non-CNT cost and small regularization value.

6 Conclusion

This work studied two provably convergent algorithms for solving Entropic Gromov–Wasserstein problems and established their connection to the mirror descent scheme of Peyré et al. (2016) which is most commonly used by practitioners. First, we introduced Averaged Mirror Descent, a generalization of the mirror descent scheme for solving the EGW problem which incorporates a step size parameter. This modification places AMD within the conditional gradient framework of Ghadimi (2019) which enabled us to obtain non-asymptotic convergence guarantees when exact iterates are assumed and to account for inexactness stemming from approximate EOT solutions obtained via Sinkhorn's algorithm. Secondly, we showed that the dual gradient method from Rioux et al. (2026) is provably convergent with a fixed regularization independent stepsize. We have also discussed the connection between our dual method and the results obtained by Houry et al. (2026) and showed that, for non-CNT costs, the MD scheme does not take true gradient steps explaining why it can fail to converge in practice on some examples.

In addition, we have empirically validated the convergence rates for both methods across a variety of settings, demonstrating that AMD attains performance comparable to classical MD. Moreover, the dual gradient method requires a similar number of steps as AMD, but each iteration requires solving a regularized convex quadratic program in place of an EOT problem so that the dual method is often slower in practice.

An important research direction stemming from this line of work is to speed up the gradient computation in the dual gradient method. While different gradient methods were proposed in Rioux et al. (2026), their theoretically justified stepsizes become small as ε decreases so that the convergence of these methods may be too slow for practical applications.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets and provide feedback on research methodology or experiments. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, implement methods, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, support qualitative and thematic data analysis, interpret results and the rest of the required disclosure tasks are not applicable to this work.

Additionally, we used generative AI tools to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, edit a research paper to improve readability. We have reviewed all AI-assisted work. LLM-generated code for data generation, plotting, and multi-seed runs was verified and tested for correctness by one of the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI. We want to highlight that LLMs were not used to implement the main methods in the paper; however, they were used to add additional features to the methods code such as more efficient storing modes as well as check the implemented code against the algorithm formulation to catch potential implementation bugs.

REPRODUCIBILITY STATEMENT

To ensure reproducibility of our results we submit a zip of the code we used to implement methods, to create datasets and to run the included experiments. The code contains both a demo notebook which allows to run the methods on all of the included datasets as well as scripts that allow to reproduce the figures included in the manuscript. We highlight that Section D.2 as well as Section D.3 mention the exact hyperparameters used for the figure generation as well as explains how the methods are implemented. Section D.1 also contains explanations of how the synthetic data has been generated.

The proofs of the results presented in the paper can be found in Section B and appropriate references are cited for claims that are used but not proved as part of this manuscript.

REFERENCES

David Alvarez-Melis and Tommi Jaakkola. Gromov-Wasserstein alignment of word embedding spaces. In *Conference on Empirical Methods in Natural Language Processing*, pp. 1881–1890. Association for Computational Linguistics, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223. PMLR, 06–11 Aug 2017.

Koenraad MR Audenaert. A sharp continuity estimate for the von Neumann entropy. *Journal of Physics A: Mathematical and Theoretical*, 40(28):8127–8136, 2007.

Luc Brogat-Motte, Rémi Flamary, Céline Brouard, Juho Rousu, and Florence D'Alché-Buc. Learning to predict graphs with fused Gromov-Wasserstein barycenters. In *International Conference on Machine Learning*, pp. 2321–2335. PMLR, 2022.

Charlotte Bunne, David Alvarez-Melis, Andreas Krause, and Stefanie Jegelka. Learning generative models across incomparable spaces. In *International conference on machine learning*, pp. 851–861. PMLR, 2019.

Junyu Chen, Binh T Nguyen, Shang H Koh, and Yong S Soh. Semidefinite relaxations of the Gromov-Wasserstein distance. *Advances in Neural Information Processing Systems*, 37:69814–69839, 2024.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Pinar Demetci, Rebecca Santorella, Björn Sandstede, William Stafford Noble, and Ritambhara Singh. SCOT: single-cell multi-omics alignment with optimal transport. *Journal of Computational Biology*, 29(1):3–18, 2022.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617, 2018.

Saeed Ghadimi. Conditional gradient type methods for composite nonlinear and stochastic optimization. *Mathematical Programming*, 173(1):431–464, 2019.

Xiaoyun Gong, Gabriel Rioux, and Ziv Goldfeld. Sliced inner product Gromov-Wasserstein distances. *arXiv preprint arXiv:2605.08546*, 2026.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. *Advances in neural information processing systems*, 30, 2017.

Guillaume Houry, Jean Feydy, and François-Xavier Vialard. Gromov-Wasserstein at scale, beyond squared norms. In *Forty-third International Conference on Machine Learning*, 2026.

Venkatkrishna Karumanchi, Gabriel Rioux, and Ziv Goldfeld. Approximation analysis of the entropic penalty in quadratic programming. *arXiv preprint arXiv:2509.20031*, 2025.

Genji Kawakita, Ariel Zeleznikow-Johnston, Naotsugu Tsuchiya, and Masafumi Oizumi. Gromov–Wasserstein unsupervised alignment reveals structural correspondences between the color similarity structures of humans and large language models. *Scientific Reports*, 14(1):15917, 2024.

Dominik Klein, Théo Uscidda, Fabian J Theis, and Marco Cuturi. GENOT: Entropic (Gromov) Wasserstein flow matching with applications to single-cell genomics. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Peihua Li, Qilong Wang, and Lei Zhang. A novel earth mover's distance methodology for image matching with Gaussian mixture models. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1689–1696, 2013.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin K Miller, and Ricky TQ Chen. Adjoint Schrödinger bridge sampler. *Advances in Neural Information Processing Systems*, 38:15673–15708, 2025.

Facundo Mémoli. Gromov–Wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487, 2011.

Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pp. 2664–2672, 2016.

Gabriel Rioux, Ziv Goldfeld, and Kengo Kato. Entropic Gromov-Wasserstein distances: Stability and algorithms. *Journal of Machine Learning Research*, 25(363):1–52, 2024a.

Gabriel Rioux, Ziv Goldfeld, and Kengo Kato. Limit laws for Gromov-Wasserstein alignment with applications to testing graph isomorphisms. *arXiv preprint arXiv:2410.18006*, 2024b.

Gabriel Rioux, Joanna Marks, Riccardo Passeggeri, and Ziv Goldfeld. Discrete Gromov-Wasserstein duality: Algorithms and isomorphism testing. *arXiv preprint arXiv:2609.03094*, 2026.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

Meyer Scetbon, Gabriel Peyré, and Marco Cuturi. Linear-time Gromov Wasserstein distances using low rank couplings and costs. In *International Conference on Machine Learning*, pp. 19347–19365, 2022.

Thibault Séjourné, François-Xavier Vialard, and Gabriel Peyré. The unbalanced gromov Wasserstein distance: Conic formulation and relaxation. *Advances in Neural Information Processing Systems*, 34:8766–8779, 2021.

Justin Solomon, Gabriel Peyré, Vladimir G Kim, and Suvrit Sra. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics*, 35(4):1–13, 2016.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023.

Titouan Vayer, Rémi Flamary, Romain Tavenard, Laetitia Chapel, and Nicolas Courty. Sliced Gromov-Wasserstein. *Advances in Neural Information Processing Systems*, 32, 2019.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2008.

Cédric Vincent-Cuaz, Rémi Flamary, Marco Corneli, Titouan Vayer, and Nicolas Courty. Semi-relaxed Gromov-Wasserstein divergence and applications on graphs. In *International Conference on Learning Representations*, 2022.

Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin. Gromov-Wasserstein learning for graph matching and node embedding. In *International Conference on Machine Learning*, pp. 6932–6941, 2019.

Zhengxin Zhang, Ziv Goldfeld, Youssef Mroueh, and Bharath K. Sriperumbudur. Gromov–Wasserstein distances: Entropic regularization, duality and sample complexity. *The Annals of Statistics*, 52(4):1616 – 1645, 2024. doi: 10.1214/24-AOS2406.

A SINKHORN'S ALGORITHM

Sinkhorn's algorithm, Algorithm 3, solves an EOT problem $\mathsf{EOT}_c^{\varepsilon}(\mu_0, \mu_1)$ between marginals supported on finite sets $\{x_0^{(i)}\}_{i=1}^{N_0}$ and $\{x_1^{(j)}\}_{j=1}^{N_1}$ respectively by using the fact that its solutions, $\pi^\star$, satisfy

$$\frac{\pi^\star(\{x_0^{(i)}, x_1^{(j)}\})}{\mu_0(\{x_0^{(i)}\})\mu_1(\{x_1^{(j)}\})} = e^{\frac{u_i^\star + v_j^\star - c(x_0^{(i)}, x_1^{(j)})}{\varepsilon}} \text{ for each } i = 1, \dots, N_0, j = 1, \dots, N_1, \tag{4}$$

for some vectors $u^\star = (u_1^\star, \dots, u_{N_0}^\star)$ and $v^\star = (v_1^\star, \dots, v_{N_1}^\star)$ which are unique up to adding a constant vector to $u^\star$ and subtracting the same constant vector from $v^\star$. With this representation, Sinkhorn's method fixes the cost matrix

$$(\boldsymbol{K})_{ij} = e^{\frac{-c(x_0^{(i)}, x_1^{(j)})}{\varepsilon}} \text{ for each } i = 1, \dots, N_0, j = 1, \dots, N_1,$$

and the marginals $r_0 = \Big(\mu_0(\{x_0^{(1)}\}), \dots, \mu_0(\{x_0^{(N_0)}\})\Big)$, $r_1 = \Big(\mu_1(\{x_1^{(1)}\}), \dots, \mu_1(\{x_1^{(N_1)}\})\Big)$ and performs iterations on a matrix with positive entries summing to 1 which consist of iteratively fixing its marginals (the vectors obtained by left or right multiplication with the vector of ones) as follows.

Algorithm 3: Sinkhorn Algorithm

Input: stopping threshold γ, max iteration count K, cost matrix $\boldsymbol{K}$, marginal vectors r_0, r_1
$u_0 \leftarrow \mathbb{1}_{N_0}/N_0$;
$k \leftarrow 1$;
for $k = 0, 1, \ldots, K-1$ **do**
  $v_k \leftarrow r_1/(\boldsymbol{K}^\mathsf{T} u_{k-1})$;
  $u_k \leftarrow r_0/(\boldsymbol{K} v_k)$;
  $\boldsymbol{\Pi}^k \leftarrow \mathrm{diag}(u_k)\boldsymbol{K}\mathrm{diag}(v_k)$;
  if $\|(\boldsymbol{\Pi}^k)^\mathsf{T}\mathbb{1}_{N_0} - r_1\|_1 < \gamma$ **then**
    break;
  $k \leftarrow k+1$;

The convergence properties of Algorithm 3 are well-established as noted in Lemma 3. Although the output, $\boldsymbol{\Pi}$, of Algorithm 3 does not necessarily satisfy both marginal constraints simultaneously, $\boldsymbol{\Pi}$ is evidently positive by construction and satisfies the marginal constraint of $\boldsymbol{\Pi}\mathbb{1}_{N_1} = r_0$ by construction. It follows that $\boldsymbol{\Pi}$ can be thought of as a probability matrix and hence can be identified with a probability distribution on $\mathcal{X}_0 \times \mathcal{X}_1$.

We conclude this section with the proof of Lemma 3 now that the setting is clear.

A.1 Proof of Lemma 3

The proof of Proposition 8 in Rioux et al. (2024a) establishes that, for each $\delta > 0$, there exists a number of steps n depending on $\delta, \varepsilon, \|c\|_\infty, \mu_0$, and μ_1 after which Sinkhorn's algorithm, as written in Algorithm 3, outputs a vector, $\tilde{x}$, satisfying $\mathsf{d}(\tilde{x}, x^\star) \leq \delta$ where d is a certain pseudometric. The proof of Lemma 26 in the same reference shows that if $\mathsf{d}(\tilde{x}, x^\star) \leq \delta$,

$$|\tilde{x}_i - x_i^\star| \leq \tilde{x}_i(e^\delta - 1) \text{ for each } i \in \{1, \ldots, k\}.$$

Conclude that $\|\tilde{x} - x^\star\|_1 \leq e^\delta - 1$ as claimed. □

B Proofs of main results

First, we establish the strong convexity of the negative entropy on the simplex $\Delta_k := \{x \in \mathbb{R}^k : x \geq 0, \|x\|_1 = 1\}$ which, notably, contains $\mathcal{K}$.

Lemma 13 (Strong convexity of negative entropy)**.** $-\mathsf{H}$ *is* 1*-strongly convex on* Δ_k *with respect to* $\|\cdot\|_1$*. Thus,* $h = -\varepsilon\mathsf{H}$ *is* ε*-strongly convex on* Δ_k*.*

Proof. Fix $t \in (0,1)$ and $x, x' \in \Delta_k$, and write $m := tx + (1-t)x'$ which again lies in Δ_k by convexity. Expanding the definitions of H and of the KL divergence gives the identity

$$-t\mathsf{H}(x) - (1-t)\mathsf{H}(x') = t\,\mathrm{KL}(x\|m) + (1-t)\,\mathrm{KL}(x'\|m) - \mathsf{H}(m).$$

Applying Pinsker's inequality gives,

$$t\,\mathrm{KL}(x\|m) + (1-t)\,\mathrm{KL}(x'\|m) \;\geq\; \tfrac{1}{2}\Big(t\|x-m\|_1^2 + (1-t)\|x'-m\|_1^2\Big),$$

and since $x - m = (1-t)(x-x')$ and $x' - m = -t(x-x')$, the right-hand side equals

$$\tfrac{1}{2}\Big(t(1-t)^2 + (1-t)t^2\Big)\|x-x'\|_1^2 = \tfrac{1}{2}\,t(1-t)\|x-x'\|_1^2.$$

Combining the two displays,

$$-\mathsf{H}\big(tx + (1-t)x'\big) \;\leq\; -t\mathsf{H}(x) - (1-t)\mathsf{H}(x') - \tfrac{1}{2}t(1-t)\|x-x'\|_1^2,$$

which proves 1-strong convexity of $-\mathsf{H}$ on Δ_k. □

B.1 Proof of Theorem 1

The proof will follow by applying the results from Ghadimi (2019). As noted previously in Section 3, Algorithm 1 coincides with the conditional gradient type method studied in that work for the choices $X = \mathcal{K}$, $f(x) = \frac{1}{2}x^{\mathsf{T}}\mathrm{M}x$, and $h(x) = -\varepsilon\mathsf{H}(x)$. We note that the gradient of f satisfies

$$\|\nabla f(x) - \nabla f(x')\|_\infty = \|\mathrm{M}(x - x')\|_\infty \leq \max_{i,j} |\mathrm{M}_{i,j}| \, \|x - x'\|_1$$

so that the gradient of f is 1-Hölder continuous (i.e., Lipschitz continuous) with respect to $\|\cdot\|_\infty$ with constant $\|\mathrm{M}\|_{\max} := \max_{i,j} |\mathrm{M}_{i,j}|$. Moreover, by Lemma 13, h is ε-strongly convex on $\mathcal{K} \subset \Delta_k$ with respect to $\|\cdot\|_1$, so that $\mu = \varepsilon$.

By Theorem 1 (b) in Ghadimi (2019), with $\alpha_j = \min\left\{\frac{\varepsilon}{4\|\mathrm{M}\|_{\max}}, 1\right\}$,

$$\sum_{j=0}^{J-1} \alpha_j \left(\nabla f(x_j)^{\mathsf{T}}(x_j - \tilde{x}_j) + h(x_j) - h(\tilde{x}_j)\right) \leq \frac{4}{3}\left(f(x_0) + h(x_0) - \min_{\mathcal{K}}(f + h)\right).$$

By Lemma 1 from the same reference,

$$\|x_j - \tilde{x}_j\|_1^2 \leq \frac{2}{\varepsilon}\left(\nabla f(x_j)^{\mathsf{T}}(x_j - \tilde{x}_j) + h(x_j) - h(\tilde{x}_j)\right),$$

so that, using the fact that α_j is constant,

$$\min_{j=0}^{J-1} \|x_j - \tilde{x}_j\|_1^2 \leq \frac{8}{3J\varepsilon}\frac{1}{\alpha_j}\left(f(x_0) - \varepsilon\mathsf{H}(x_0) - \min_{\mathcal{K}}\{f - \varepsilon\mathsf{H}\}\right),$$

which proves the claim upon inserting the expression for α_j^{-1}. □

B.2 Proof of Theorem 4

As noted in Section A, the output of Sinkhorn's algorithm can be identified with an element of Δ_k upon vectorizing the matrix $\mathbf{\Pi}$, that is, $\tilde{x}_j \in \Delta_k$. It follows that the iterates x_j of Algorithm 1 are all elements of Δ_k due to convexity of Δ_k.

Throughout we write $\phi_{x_j}(x) = \nabla f(x_j)^{\mathsf{T}}x - \varepsilon\mathsf{H}(x)$ for the function minimized in the exact update, $\tilde{x}_j^\star := \arg\min_{\mathcal{K}} \phi_{x_j}$ and $L := \|\mathrm{M}\|_{\max}$, so that $\tilde{x}_j$ is the approximation of $\tilde{x}_j^\star$ returned by Sinkhorn's algorithm.

We now establish that ϕ_{x_j} admits a certain modulus of continuity over Δ_k which is a key ingredient in the proof of Theorem 4.

Lemma 14 (Modulus of continuity)**.** *For any x_j with $\|x_j\|_1 = 1$ and $x, x' \in \Delta_k$,*

$|\phi_{x_j}(x) - \phi_{x_j}(x')| \leq \omega_\varepsilon(\|x - x'\|_1)$ *and* $|(f(x) - \varepsilon\mathsf{H}(x)) - (f(x') - \varepsilon\mathsf{H}(x'))| \leq \omega_\varepsilon(\|x - x'\|_1)$

where $\omega_\varepsilon : [0, 2] \to \mathbb{R}$ is defined by

$$\omega_\varepsilon(p) = \|\mathrm{M}\|_{\max} p + \varepsilon\left[\frac{1}{2}p\log(k-1) - \frac{1}{2}p\log\left(\frac{1}{2}p\right) - \left(1 - \frac{1}{2}p\right)\log\left(1 - \frac{1}{2}p\right)\right],$$

which is increasing on $[0, 1]$.

Proof. Equation (11) in Audenaert (2007) asserts that

$$|\mathsf{H}(x) - \mathsf{H}(x')| \leq \frac{1}{2}\|x - x'\|_1\log(k-1) - \frac{1}{2}\|x - x'\|_1\log\left(\frac{1}{2}\|x - x'\|_1\right) - \left(1 - \frac{1}{2}\|x - x'\|_1\right)\log\left(1 - \frac{1}{2}\|x - x'\|_1\right)$$

for each non-negative vector with 1-norm equal to 1 of size k, noting that $\|x - x'\|_1 \leq 2$. On the other hand, we have that

$$|\nabla f(x_j)^{\mathsf{T}}x - \nabla f(x_j)^{\mathsf{T}}x'| = |x_j^{\mathsf{T}}\mathrm{M}(x - x')| \leq \|\mathrm{M}x_j\|_\infty\|x - x'\|_1 \leq \|\mathrm{M}\|_{\max}\|x - x'\|_1.$$

Altogether we have that

$$\begin{aligned}|\phi_{x_j}(x) - \phi_{x_j}(x')| &\leq \|\mathrm{M}\|_{\max} \|x - x'\|_1 + \frac{\varepsilon}{2}\|x - x'\|_1 \log(k-1) - \frac{\varepsilon}{2}\|x - x'\|_1 \log\left(\frac{1}{2}\|x - x'\|_1\right) \\ &\quad - \varepsilon\left(1 - \frac{1}{2}\|x - x'\|_1\right)\log\left(1 - \frac{1}{2}\|x - x'\|_1\right) \\ &= \omega_\varepsilon(\|x - x'\|_1)\end{aligned}$$

which proves the first claim. Observe that

$$\begin{aligned}|f(x) - f(x')| = \left|\frac{1}{2}(x - x')^\mathsf{T}\mathrm{M}x + \frac{1}{2}(x')^\mathsf{T}\mathrm{M}(x - x')\right| &\leq \frac{1}{2}\left(\|\mathrm{M}x\|_\infty + \|\mathrm{M}x'\|_\infty\right)\|x - x'\|_1 \\ &\leq \|\mathrm{M}\|_{\max}\|x - x'\|_1\end{aligned}$$

so that the same argument as above establishes that

$$|(f(x) - \varepsilon\mathsf{H}(x)) - (f(x') - \varepsilon\mathsf{H}(x'))| \leq \omega_\varepsilon(\|x - x'\|_1).$$

To see that ω_ε is increasing on $[0, 1]$, note that $p \in [0, 1] \mapsto -p\log(p) - (1-p)\log(1-p)$ is increasing on $[0, 1/2]$ and that the linear terms are evidently increasing. □

With these results in hand, we are in position to prove Theorem 4.

Proof of Theorem 4 Since the gradient of f is L-Lipschitz continuous with respect to $\|\cdot\|_1$ and $x_{j+1} - x_j = \alpha(\tilde{x}_j - x_j)$,

$$f(x_{j+1}) \leq f(x_j) + \alpha\nabla f(x_j)^\mathsf{T}(\tilde{x}_j - x_j) + \frac{L\alpha^2}{2}\|\tilde{x}_j - x_j\|_1^2.$$

As noted above $\tilde{x}_j \in \Delta_k$ and, since $x_0 \in \mathcal{K} \subset \Delta_k$, $x_j \in \Delta_k$ for all j. Lemma 13 yields that

$$-\varepsilon\mathsf{H}(x_{j+1}) \leq -(1-\alpha)\varepsilon\mathsf{H}(x_j) - \alpha\varepsilon\mathsf{H}(\tilde{x}_j) - \frac{\alpha(1-\alpha)}{2}\varepsilon\|x_j - \tilde{x}_j\|_1^2.$$

Summing these two displayed equations, we obtain that

$$\begin{aligned}&\alpha\left(\nabla f(x_j)^\mathsf{T}(x_j - \tilde{x}_j) + \varepsilon\mathsf{H}(\tilde{x}_j) - \varepsilon\mathsf{H}(x_j)\right) \\ &\qquad \leq (f(x_j) - \varepsilon\mathsf{H}(x_j)) - (f(x_{j+1}) - \varepsilon\mathsf{H}(x_{j+1})) + \frac{\alpha}{2}\left(L\alpha - (1-\alpha)\varepsilon\right)\|x_j - \tilde{x}_j\|_1^2,\end{aligned}$$

and, since $\alpha = \frac{\varepsilon}{2(L+\varepsilon)}$, $L\alpha - (1-\alpha)\varepsilon = -\varepsilon/2$, so that

$$\begin{aligned}&\frac{\alpha\varepsilon}{4}\|x_j - \tilde{x}_j\|_1^2 + \alpha\left(\nabla f(x_j)^\mathsf{T}(x_j - \tilde{x}_j) + \varepsilon\mathsf{H}(\tilde{x}_j) - \varepsilon\mathsf{H}(x_j)\right) \\ &\qquad\qquad \leq (f(x_j) - \varepsilon\mathsf{H}(x_j)) - (f(x_{j+1}) - \varepsilon\mathsf{H}(x_{j+1}))\,.\end{aligned} \tag{5}$$

We now analyze the term $\nabla f(x_j)^\mathsf{T}(x_j - \tilde{x}_j) + \varepsilon\mathsf{H}(\tilde{x}_j) - \varepsilon\mathsf{H}(x_j) = \phi_{x_j}(x_j) - \phi_{x_j}(\tilde{x}_j)$. Let $\tilde{x}_j^\star$ be the true minimizer of ϕ_{x_j} over $\mathcal{K}$ and let $z_{j+1} = (1-\alpha)z_j + \alpha\tilde{x}_j^\star \in \mathcal{K}$ for each $j \in \{0, \ldots, J-1\}$ with the convention $z_0 = x_0$. We prove by induction that $\|z_J - x_J\|_1 \leq (e^\delta - 1)\sum_{j=0}^{J-1}(1-\alpha)^j\alpha$. The base case is straightforward since $z_0 = x_0$. Now, by the triangle inequality

$$\begin{aligned}\|z_{J+1} - x_{J+1}\|_1 &= \|(1-\alpha)(z_J - x_J) + \alpha(\tilde{x}_J^\star - \tilde{x}_J)\|_1 \\ &\leq (e^\delta - 1)\alpha(1-\alpha)\sum_{j=0}^{J-1}(1-\alpha)^j + \alpha(e^\delta - 1) = (e^\delta - 1)\alpha\sum_{j=0}^{J}(1-\alpha)^j,\end{aligned}$$

where we have applied Lemma 3 to obtain that $\|\tilde{x}_J^\star - \tilde{x}_J\|_1 \leq e^\delta - 1$.

With these points, we note that

$$\phi_{x_j}(x_j) - \phi_{x_j}(\tilde{x}_j) = \underbrace{\phi_{x_j}(x_j) - \phi_{x_j}(z_j)}_{(i)} + \underbrace{\phi_{x_j}(z_j) - \phi_{x_j}(\tilde{x}_j^\star)}_{(ii)} + \underbrace{\phi_{x_j}(\tilde{x}_j^\star) - \phi_{x_j}(\tilde{x}_j)}_{(iii)}.$$

Note that expression (ii) is non-negative since $\tilde{x}_j^\star$ is the global minimizer of ϕ_{x_j}. On the other hand, since $x_j, z_j, \tilde{x}_j^\star$, and $\tilde{x}_j$ are elements of Δ_k as noted previously, Lemma 14 yields that

$$\phi_{x_j}(x_j) - \phi_{x_j}(\tilde{x}_j) \geq -\omega_\varepsilon(\|x_j - z_j\|_1) - \omega_\varepsilon(\|\tilde{x}_j^\star - \tilde{x}_j\|_1) \geq -2\omega_\varepsilon(e^\delta - 1),$$

where the final inequality follows from the fact that ω_ε is increasing on $[0,1]$ and that $e^\delta - 1 \leq \frac{1}{2}$.

Returning to Equation 5, we obtain that

$$\sum_{j=0}^{J-1} \|x_j - \tilde{x}_j\|_1^2 \leq \frac{4}{\alpha\varepsilon}\left(f(x_0) - \varepsilon\mathsf{H}(x_0) - (f(x_J) - \varepsilon\mathsf{H}(x_J))\right) + \frac{8}{\varepsilon}J\omega_\varepsilon(e^\delta - 1).$$

Recalling $z_J \in \mathcal{K}$ from the previous step which satisfies $\|x_J - z_J\|_1 \leq e^\delta - 1$, we apply Lemma 14 again to obtain that

$$f(x_J) - \varepsilon\mathsf{H}(x_J) \geq f(z_J) - \varepsilon\mathsf{H}(z_J) - \omega_\varepsilon(e^\delta - 1) \geq \inf_{\mathcal{K}}\{f - \varepsilon\mathsf{H}\} - \omega_\varepsilon(e^\delta - 1)$$

whereby

$$\sum_{j=0}^{J-1} \|x_j - \tilde{x}_j\|_1^2 \leq \frac{4}{\alpha\varepsilon}\left(f(x_0) - \varepsilon\mathsf{H}(x_0) - \inf_{\mathcal{K}}\{f - \varepsilon\mathsf{H}\} + \omega_\varepsilon(e^\delta - 1)\right) + \frac{8}{\varepsilon}J\omega_\varepsilon(e^\delta - 1).$$

We conclude by lower bounding the sum on the left hand side by J times the minimum summand,

$$\min_{j=0}^{J-1} \|x_j - \tilde{x}_j\|_1^2 \leq \frac{8(L+\varepsilon)}{J\varepsilon^2}\left(f(x_0) - \varepsilon\mathsf{H}(x_0) - \inf_{\mathcal{K}}\{f - \varepsilon\mathsf{H}\} + \omega_\varepsilon(e^\delta - 1)\right) + \frac{8}{\varepsilon}\omega_\varepsilon(e^\delta - 1).$$

□

B.3 Proof of Theorem 6

From the proof of Lemma 2 in Rioux et al. (2026)

$$\ell_\varepsilon(u) = \frac{1}{2}\|u\|^2 + \inf_{x\in\mathcal{K}}\left\{\frac{1}{2}x^\mathsf{T}\mathrm{B}_1^\mathsf{T}\mathrm{B}_1 x - u^\mathsf{T}\mathrm{B}_0 x - \varepsilon\mathsf{H}(x)\right\}$$

and so, noting that $u_{j+1} = u_j - \nabla\ell_\varepsilon(u_j) = \mathrm{B}_0 x_{u_j}$,

$$\begin{aligned}
\ell_\varepsilon(u_j) - \ell_\varepsilon(u_{j+1}) &\geq \left(\frac{1}{2}\|u_j\|^2 + \frac{1}{2}\|\mathrm{B}_1 x_{u_j}\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x_{u_j} - \varepsilon\mathsf{H}(x_{u_j})\right) \\
&\quad - \left(\frac{1}{2}\|u_{j+1}\|^2 + \frac{1}{2}\|\mathrm{B}_1 x_{u_j}\|^2 - u_{j+1}^\mathsf{T}\mathrm{B}_0 x_{u_j} - \varepsilon\mathsf{H}(x_{u_j})\right) \\
&= \frac{1}{2}\|u_j\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x_{u_j} - \frac{1}{2}\|\mathrm{B}_0 x_{u_j}\|^2 + \|\mathrm{B}_0 x_{u_j}\|^2.
\end{aligned}$$

Conclude that $\ell_\varepsilon(u_j) - \ell_\varepsilon(u_{j+1}) \geq \frac{1}{2}\|u_j - \mathrm{B}_0 x_{u_j}\|^2 = \frac{1}{2}\|u_j - u_{j+1}\|^2$. Summing this expression over the iterates gives

$$\frac{1}{2}\sum_{j=0}^{J-1}\|u_j - u_{j+1}\|^2 \leq \ell_\varepsilon(u_0) - \ell_\varepsilon(u_J) \leq \ell_\varepsilon(u_0) - \inf_{\mathbb{R}^{r_0}}\ell_\varepsilon$$

which yields that

$$\min_{j=0}^{J-1}\|\nabla\ell_\varepsilon(u_j)\|^2 = \min_{j=0}^{J-1}\|u_j - u_{j+1}\|^2 \leq \frac{2}{J}\left(\ell_\varepsilon(u_0) - \inf_{\mathbb{R}^{r_0}}\ell_\varepsilon\right).$$

□

B.4 Proof of Theorem 10

Following the proof of Theorem 6,

$$\ell_\varepsilon(u_j) = \frac{1}{2}\|u_j\|^2 + \frac{1}{2}\|\mathrm{B}_1 x_j^\star\|^2 - u_j^\mathsf{T}\mathrm{B}_0 x_j^\star - \varepsilon\mathsf{H}(x_j^\star)$$
$$\ell_\varepsilon(u_{j+1}) \leq \frac{1}{2}\|u_{j+1}\|^2 + \frac{1}{2}\|\mathrm{B}_1 x_j^\star\|^2 - u_{j+1}^\mathsf{T}\mathrm{B}_0 x_j^\star - \varepsilon\mathsf{H}(x_j^\star).$$

Consequently,

$$\begin{aligned}
\ell_\varepsilon(u_j) - \ell_\varepsilon(u_{j+1}) &\geq \frac{1}{2}\|u_j\|^2 - \frac{1}{2}\|u_{j+1}\|^2 - (u_j - u_{j+1})^\mathsf{T}\mathrm{B}_0 x_j^\star \\
&= \frac{1}{2}\|u_j\|^2 - \frac{1}{2}\|u_{j+1}\|^2 - (u_j - u_{j+1})^\mathsf{T}\mathrm{B}_0 x_j^\star - \frac{1}{2}\|\mathrm{B}_0 x_j^\star\|^2 + \frac{1}{2}\|\mathrm{B}_0 x_j^\star\|^2 \\
&= \frac{1}{2}\|u_j - \mathrm{B}_0 x_j^\star\|^2 - \frac{1}{2}\|u_{j+1} - \mathrm{B}_0 x_j^\star\|^2 \\
&= \frac{1}{2}\|\nabla\ell_\varepsilon(u_j)\|^2 - \frac{1}{2}\|\mathrm{B}_0 x_{u_j} - \mathrm{B}_0 x_j^\star\|^2,
\end{aligned}$$

where we recall that $u_j - \mathrm{B}_0 x_j^\star = \nabla\ell_\varepsilon(u_j)$ is the true gradient of ℓ_ε. Now,

$$\|\mathrm{B}_0 x_{u_j} - \mathrm{B}_0 x_j^\star\| \leq \|\mathrm{B}_0\|_{1,2}\|x_{u_j} - x_j^\star\|_1 \leq \|\mathrm{B}_0\|_{1,2}\tau,$$

whereby $\sum_{j=0}^{J-1}\|\nabla\ell_\varepsilon(u_j)\|^2 \leq 2\left(\ell_\varepsilon(u_0) - \ell_\varepsilon(u_J)\right) + J\|\mathrm{B}_0\|_{1,2}^2\tau^2$, and, since $\ell_\varepsilon(u_J) \geq \inf_{\mathbb{R}^{r_0}}\ell_\varepsilon$,

$$\min_{j=0}^{J-1}\|\nabla\ell_\varepsilon(u_j)\|^2 \leq \frac{2}{J}\left(\ell_\varepsilon(u_0) - \inf_{\mathbb{R}^{r_0}}\ell_\varepsilon\right) + \|\mathrm{B}_0\|_{1,2}^2\tau^2,$$

proving the claim. □

B.5 Proof of theorem 9

Suppose that $\bar{u}$ is an η-stationary point for ℓ_ε and some $\eta \geq 0$, that is, $\|\bar{u} - \mathrm{B}_0 x_{\bar{u}}\| \leq \eta$ where $x_{\bar{u}}$ solves

$$\inf_{x\in\mathcal{K}}\left\{\frac{1}{2}x^\mathsf{T}\mathrm{B}_1^\mathsf{T}\mathrm{B}_1 x - u^\mathsf{T}\mathrm{B}_0 x - \varepsilon\mathsf{H}(x)\right\}.$$

Following the same arguments as in Section C.1, we necessarily have that

$$\langle \mathrm{B}_1^\mathsf{T}\mathrm{B}_1 x_{\bar{u}} - \mathrm{B}_0^\mathsf{T}\bar{u} + \nabla h(\bar{x}_u), x - \bar{x}_u\rangle \geq 0 \text{ for each } x \in \mathcal{K}.$$

Applying the Cauchy-Schwarz inequality and the fact that $\|\bar{u} - \mathrm{B}_0 x_{\bar{u}}\| \leq \eta$,

$$\langle -\mathrm{B}_0^\mathsf{T}\bar{u}, x - x_{\bar{u}}\rangle = \langle -\mathrm{B}_0^\mathsf{T}(\bar{u} - \mathrm{B}_0 x_{\bar{u}} + \mathrm{B}_0 x_{\bar{u}}), x - x_{\bar{u}}\rangle \leq \langle -\mathrm{B}_0^\mathsf{T}\mathrm{B}_0 x_{\bar{u}}, x - x_{\bar{u}}\rangle + \|x - x_{\bar{u}}\|_1\|\mathrm{B}_0\|_{1,2}\eta.$$

Since $\|x - x_{\bar{u}}\|_1 \leq 2$ as $x, x_{\bar{u}}$ are probability vectors, we obtain that, for each $x \in \mathcal{K}$,

$$0 \leq \langle \mathrm{B}_1^\mathsf{T}\mathrm{B}_1 x_{\bar{u}} - \mathrm{B}_0^\mathsf{T}\bar{u} + \nabla h(x_{\bar{u}}), x - x_{\bar{u}}\rangle \leq \langle \mathrm{B}_1^\mathsf{T}\mathrm{B}_1 x_{\bar{u}} - \mathrm{B}_0^\mathsf{T}\mathrm{B}_0 x_{\bar{u}} + \nabla h(x_{\bar{u}}), x - x_{\bar{u}}\rangle + 2\|\mathrm{B}_0\|_{1,2}\eta.$$

Conclude that $\langle -(\nabla h(x_{\bar{u}}) + \nabla f(x_{\bar{u}})), x - x_{\bar{u}}\rangle \leq 2\|\mathrm{B}_0\|_{1,2}\eta$ so that $x_{\bar{u}}$ is $2\|\mathrm{B}_0\|_{1,2}\eta$-stationary for the primal problem according to the discussion in Remark 2. □

C Additional details on the convergence metrics

This section provides additional context for the convergence metrics used in Theorems 1 and 4. We begin with the case of exact iterates and later comment on the case where iterates are inexact.

C.1 Exact iterates

We first establish that every local or global minimizer of $f + h$ over $\mathcal{K}$ has strictly positive entries. We denote this property by $x \in \mathbb{R}^k_{>0}$.

Lemma 15. *Every local or global minimizer of $f + h$ over $\mathcal{K}$ lies in $\mathbb{R}^k_{>0}$.*

Proof. By definition $\bar{x}$ is a local minimizer of $f+h$ over $\mathcal{K}$ if and only if there exists a neighborhood, N, of $\bar{x}$ for which $(f+h)(x) \geq (f+h)(\bar{x})$ for every $x \in N \cap \mathcal{K}$. It follows that,

$$\liminf_{t \downarrow 0} \frac{(f+h)(\bar{x}+t(x-\bar{x})) - (f+h)(\bar{x})}{t} = \liminf_{t \downarrow 0} \frac{(f+h)(x_t) - (f+h)(\bar{x})}{t} \geq 0 \tag{6}$$

for any choice of $x \in N \cap \mathcal{K}$ noting that $x_t = (1-t)\bar{x} + tx \in \mathcal{K}$ for each $t \in [0,1]$ by convexity of $\mathcal{K}$ so that the limit is well-defined and $x_t \in N$ once t is sufficiently small. Since f is smooth,

$$\lim_{t \downarrow 0} \frac{f(x_t) - f(\bar{x})}{t} = \bar{x}^{\mathsf{T}} \mathrm{M}(x - \bar{x}).$$

As for h, observe that

$$\begin{aligned} -\mathsf{H}(x_t) + \mathsf{H}(\bar{x}) &= \sum_{i=1}^{k} (\bar{x}_i + t(x_i - \bar{x}_i)) \log(\bar{x}_i + t(x_i - \bar{x}_i)) - \bar{x}_i \log(\bar{x}_i) \\ &= \sum_{i=1}^{k} (\bar{x}_i (\log(\bar{x}_i + t(x_i - \bar{x}_i)) - \log(\bar{x}_i)) + t(x_i - \bar{x}_i) \log(\bar{x}_i + t(x_i - \bar{x}_i))). \end{aligned}$$

Now, if $\bar{x}_I = 0$ for some $I \in \{1, \ldots, k\}$ we obtain that

$$\bar{x}_I (\log(\bar{x}_I + t(x_I - \bar{x}_I)) - \log(\bar{x}_I)) + t(x_I - \bar{x}_I) \log(\bar{x}_I + t(x_I - \bar{x}_I)) = tx_I \log(tx_I).$$

and $\lim_{t \downarrow 0} x_I \log(tx_I) = -\infty$ if $x_I > 0$. As noted in Equation 4, there always exists some $x' \in \mathcal{K} \cap \mathbb{R}^k_{>0}$ and, by convexity of $\mathcal{K}$, $x = (1-\alpha)\bar{x} + \alpha x' \in (\mathcal{K} \cap N) \cap \mathbb{R}^k_{>0}$ for some $\alpha \in (0,1)$ whereby

$$\lim_{t \downarrow 0} \frac{(f+h)(\bar{x}+t(x-\bar{x})) - (f+h)(\bar{x})}{t} = -\infty$$

for this choice of x, contradicting optimality of $\bar{x}$.

Conclude that a local minimizer, $\bar{x}$, must also have strictly positive entries and hence this proviso must also be met for a global minimizer. □

With this, we obtain a necessary condition for optimality for the problem $\inf_{x \in \mathcal{K}} \{f+h\}$.

Proposition 16. *Suppose that $\bar{x}$ is a global minimizer of $f+h$ over $\mathcal{K}$. Then, h is differentiable at $\bar{x}$ with derivative $\nabla h(\bar{x})$ defined componentwise as*

$$(\nabla h(\bar{x}))_i = \varepsilon(1 + \log(\bar{x}_i)) \text{ for } i \in \{1, \ldots, k\}$$

and $\bar{x}$ must satisfy the property that

$$\langle \nabla f(\bar{x}) + \nabla h(\bar{x}), x - \bar{x} \rangle \geq 0 \text{ for each } x \in \mathcal{K}. \tag{7}$$

Proof. By Lemma 15 $\bar{x}$ is an element of $\mathcal{K} \cap \mathbb{R}^k_{>0}$ so that the formula for the gradient of h at $\bar{x}$ follows by a computation. Since $\bar{x}$ is a global minimizer, Equation 6 must hold at each $x \in \mathcal{K}$, that is, $\langle \nabla f(\bar{x}) + \nabla h(\bar{x}), x - \bar{x} \rangle \geq 0$ for each $x \in \mathcal{K}$ as claimed. □

With this result, it is natural to define the notion of a near stationary point in terms of how much Equation 7 is violated. As noted in the main text, we can thus call $\tilde{x} \in \mathcal{K} \cap \mathbb{R}^k_{>0}$ γ-approximately stationary for this problem if $\langle \nabla f(\tilde{x}) + \nabla h(\tilde{x}), x - \tilde{x} \rangle \geq -\gamma$ for some $\gamma > 0$ and for each $x \in \mathcal{K}$.

We conclude this section by recalling that the solution of each EOT subproblem is an element of $\mathbb{R}^k_{>0}$ (see Equation 4) so that the approximate stationarity condition above makes sense at each $\tilde{x}_j$ as the derivatives are well-defined.

C.2 Inexact iterates

As noted in Remark 5, the convergence metric $\min_{j=0}^{J-1} \|x_j - \tilde{x}_j\|_1^2$ is more opaque in the inexact setting since $\tilde{x}_j$ is not an exact solution of the EOT subproblem and is not an element of $\mathcal{K}$.

Proposition 17. *Fix $\delta > 0$ and suppose that every EOT subproblem is solved to within an accuracy of $e^\delta - 1$ in the 1-norm. Furthermore, let $\tilde{x}_j^\star$ be the unique solution of $\min_{x\in\mathcal{K}}\{x_j^\mathsf{T}\mathrm{M}x - \varepsilon\mathsf{H}(x)\}$ for each $j \in \{0,\dots,J-1\}$ and hence satisfies $\|\tilde{x}_j^\star - \tilde{x}_j\|_1 \le e^\delta - 1$. Then, if $\|x_j - \tilde{x}_j\|_1 \le \tau$ it holds that*

$$\langle -\nabla f(\tilde{x}_j^\star) - \nabla h(\tilde{x}_j^\star), x - \tilde{x}_j^\star\rangle \le \|\mathrm{M}\|_{\max} \sup_{x,y\in\mathcal{K}} \|x-y\|_1 \left(\tau + e^\delta - 1\right),$$

that is, $\tilde{x}_j^\star$ is an $\|\mathrm{M}\|_{\max}\sup_{x,y\in\mathcal{K}}\|x-y\|_1\left(\tau+e^\delta-1\right)$-approximate stationary point which is a distance at most $e^\delta - 1$ from $\tilde{x}_j$.

Proof. Since $\tilde{x}_j^\star$ solves $\min_{x\in\mathcal{K}}\{x_j^\mathsf{T}\mathrm{M}x - \varepsilon\mathsf{H}(x)\}$ we have from the same logic as the proof of Lemma 15 and Proposition 16 that $\tilde{x}_j^\star \in \mathbb{R}^k_{>0}$ and

$$\langle -(\nabla f(x_j) + \nabla h(\tilde{x}_j^\star)), x - \tilde{x}_j^\star\rangle \le 0 \text{ for each } x \in \mathcal{K}.$$

Since f has $\|\mathrm{M}\|_{\max}$-Lipschitz continuous gradient in the ∞-norm,

$$\begin{aligned}\langle \nabla f(x_j) - \nabla f(\tilde{x}_j^\star), x - \tilde{x}_j^\star\rangle &\le \|\nabla f(x_j) - \nabla f(\tilde{x}_j^\star)\|_\infty \|x - \tilde{x}_j^\star\|_1 \\ &\le \sup_{x,y\in\mathcal{K}} \|x-y\|_1 \|\mathrm{M}\|_{\max} \|x_j - \tilde{x}_j^\star\|_1 \\ &\le \sup_{x,y\in\mathcal{K}} \|x-y\|_1 \|\mathrm{M}\|_{\max} (\tau + e^\delta - 1).\end{aligned}$$

Combining the two displayed equations above we obtain that

$$\langle -(\nabla f(\tilde{x}_j^\star) + \nabla h(\tilde{x}_j^\star)), x - \tilde{x}_j^\star\rangle \le \sup_{x,y\in\mathcal{K}} \|x-y\|_1 \|\mathrm{M}\|_{\max} (\tau + e^\delta - 1)$$

which shows that $\tilde{x}_j^\star$ is an $\sup_{x,y\in\mathcal{K}}\|x-y\|_1\|\mathrm{M}\|_{\max}(\tau+e^\delta-1)$-approximate stationary point of the EGW problem. □

D Experimental details

D.1 Dataset generation

Gaussian mixture. We independently generate 1000 points from two 10-component Gaussian mixtures, one in $\mathbb{R}^{10}$ and the other in $\mathbb{R}^{15}$. Component weights are sampled from a symmetric $\mathrm{Dirichlet}(2)$ distribution, component centres from $\mathcal{N}(0, 9I)$, and each component has standard deviation 0.45. For visualization, each point cloud is projected onto its first two principal components. The cost is set to be the squared Euclidean distance. Both cost matrices are rescaled by the common factor $3/\max\{\mathrm{median}(\mathrm{K}_0), \mathrm{median}(\mathrm{K}_1)\}$ so that they are on a comparable scale; since the same factor is applied to both, this is equivalent to rescaling ε.

Spherical point clouds. We sample 100 points on $\mathbb{S}^2$ from a von Mises–Fisher distribution with mean direction $(0,0,1)^\top$ and concentration $\kappa = 40$. The second point cloud is obtained by applying a random rotation to the first. The cost is chosen to be the geodesic distance.

Gaussian–ring. We generate two point clouds of 100 points in $\mathbb{R}^2$. The first is sampled independently from $x_i \sim \mathcal{N}(0, 0.1^2 I_2)$. For the second, we draw $\theta_i \sim \mathrm{Unif}[0, 2\pi)$ independently and set $y_i = 5(\cos\theta_i, \sin\theta_i)$, so that the points are uniformly distributed on a circle of radius 5 centered at the origin. The cost is set to be the Euclidean distance raised to the sixth power.

Example from Houry et al. (2026). We reconstruct the example from Houry et al. (2026) using two fixed clouds of 15 points in $\mathbb{R}^2$. The pixel coordinates were approximated by an LLM from the published visualization, as the exact figure reproducing code was not publicly available. The sampled points are defined as

$$x_i = (q_i^X, -r_i^X)/150, \qquad y_i = (q_i^Y, -r_i^Y)/150,$$

using the entries in the following coordinate table and the weights are taken to be uniform.

i	q_i^X	r_i^X	q_i^Y	r_i^Y
1	105	131	244	48
2	120	138	260	43
3	98	197	279	50
4	144	202	282	60
5	126	225	267	74
6	114	238	230	88
7	133	242	290	87
8	145	230	284	103
9	187	219	304	109
10	180	250	291	119
11	203	234	255	119
12	233	266	245	140
13	211	150	344	115
14	214	139	344	130
15	122	238	313	25

To verify that the approximation was accurate we run the `EntropicGW` function for the squared Euclidean cost and the Euclidean cost raised to the sixth power from the repository supplied by Houry et al. (2026) and obtained the same oscillatory effect for the sixth power cost.

D.2 Implementation details and computational complexity

D.2.1 Implementation of AMD

Our implementation of AMD adapts the MD method used in Scetbon et al. (2022) which leverages low-rank factorizations of the cost matrices to speed up computations, see Algorithm 2 in Scetbon et al. (2022). Once the low-rank factorizations of the costs are given (we discuss similar decompositions in the following section), the complexity of this algorithm is $\mathrm{O}(N_0 N_1 S J)$ up to log factors where J denotes the number of iterations needed to reach approximate stationarity and S denotes the maximum number of Sinkhorn's iterations. We note that the averaging step can impact J which can be both larger and smaller than for the classical MD depending on the exact experimental setting as we have shown in Section 5.

D.2.2 Implementation of the variational methods

Matrix decomposition. In order to implement the variational methods we first need to find the decomposition $\mathrm{M} = \mathrm{B}_1^\intercal \mathrm{B}_1 - \mathrm{B}_0^\intercal \mathrm{B}_0$. A generic approach to construct B_0 and B_1 described in Rioux et al. (2026), is to compute the eigendecomposition of M and let $\mathrm{B}_1 = \mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_{r_1}})\mathbf{V}_1$, where $(\lambda_1, \ldots, \lambda_{r_1})$ is the collection of all positive eigenvalues of M and $\mathbf{V}_1 \in \mathbb{R}^{r_1 \times k}$ has i-th row given by the (transpose of the) i-th eigenvector v_i. B_0 is constructed similarly using the absolute value of the negative eigenvalues. Eigenvalues with absolute value at most 10^{-10} are discarded. As noted by Rioux et al. (2026) for the case $p = 2$, it suffices to solve the regularized quadratic program, equation 2, with $\mathrm{M} = -4\mathrm{K}_0 \otimes \mathrm{K}_1$, where $\mathrm{K}_0 \otimes \mathrm{K}_1$ is the Kronecker product of the pairwise cost matrices $\mathrm{K}_0 \in \mathbb{R}^{N_0 \times N_0}, \mathrm{K}_1 \in \mathbb{R}^{N_1 \times N_1}$ whose ij-th entries are given by $\kappa_0\left(x_0^{(i)}, x_0^{(j)}\right)$ and $\kappa_1\left(x_1^{(i)}, x_1^{(j)}\right)$ respectively. They note that the complexity of this decomposition is $O(\max\{N_0^3, N_1^3\})$. They further note that the memory-efficiency of the implementation can be enhanced using some properties of Kronecker product and hence in all of our runs we use the memory-efficient implementation as described in Rioux et al. (2026) which applies B_0, B_1 and their transposes through matrix products using their derived memory efficient formulas without explicitly forming the full matrices B_0, B_1.

Hyperparameters. For all of the variational solvers, the inner maximization is set to a simple gradient method (Algorithm 6 in Rioux et al. (2026)) which requires solving an EOT problem at each iteration. The number of gradient ascent steps and the step size varies by experiment and is described in the Section D.3. The gradient ascent is terminated early if the squared

norm of the gradient is below 10^{-6}. For the Variational (1) method we set the outer minimization stepsize to 1, which is theoretically justified by our convergence analysis. For the Variational (L) and Accelerated variational (L) we set the outer step sizes based on the value of $L_{\text{var}} = \frac{\lambda_{\max}(\mathrm{B}_0^\intercal \mathrm{B}_0)}{\lambda_{\min}(\mathrm{B}_1^\intercal \mathrm{B}_1)+\varepsilon} + 1$ for Variational (L) and the update sequences for Accelerated variational (L) are set using $L_{\text{acc}} = \max\left\{1, \frac{\lambda_{\max}(\mathrm{B}_0^\intercal \mathrm{B}_0)}{\lambda_{\min}(\mathrm{B}_1^\intercal \mathrm{B}_1)+\varepsilon} - 1\right\}$ where $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues respectively. These values come from Proposition 10 and Theorem 5 in Rioux et al. (2026). We initialize all solvers from the product coupling $\mu_0 \otimes \mu_1$.

Complexity. Assuming that B_0 and B_1 are available, Rioux et al. (2026) report that the overall computational complexity of their simple gradient method with a first order oracle used for the inner solve scales as $O(J_{\text{outer}} J_{\text{inner}}(S + \max\{r_0, r_1\})N_0 N_1)$, where J_{outer} denotes the number of outer iterations, J_{inner} denotes the number of inner oracle iterations and S is the maximum number of Sinkhorn's iterations. The r_0 and r_1 denote the dimensions of B_0 and B_1 respectively.

D.3 Experimental details

Sinkhorn implementation and parameters. All EOT problems are solved using POT (Flamary et al., 2021). We either use the standard `sinkhorn_knopp` or `sinkhorn_epsilon_scaling` which is more stable, but slower. For the three datasets, the termination threshold for these methods is set to 10^{-9}. For `sinkhorn_knopp` we set the maximum number of iterations to 1000 and, for `sinkhorn_epsilon_scaling`, a maximum of 20 000 continuation iterations with POT's default of 100 inner scaling iterations per continuation step is used.

The settings used for the methods are compiled in Table 2.

Data	ε	J_{outer}	J_{inner}	Inner stepsize	Sinkhorn method
Gaussian mixture	0.1	10 000	100	0.5	`sinkhorn_knopp`
Sphere	0.01	10 000	100	0.5	`sinkhorn_knopp`
Gaussian-ring	1	300	200	0.05	`sinkhorn_epsilon_scaling`

Table 2: Settings for the experiments. J_{outer} is the total number of iterations used for AMD (Algorithm 1) and the dual gradient method (Algorithm 2). J_{inner} is the maximum number of gradient ascent iterations used to solve each convex quadratic subproblem in Algorithm 2 and inner stepsize refers to the stepsize used for this gradient ascent algorithm. Sinkhorn method refers to what method was used to solve the EOT problems.

The `sinkhorn-epsilon-scaling` was used for the Gaussian-ring example, as `sinkhorn-knopp` encounters division by zero errors.

Stopping criteria. We terminate AMD when $\|\tilde{x}_j - x_j\|_1^2 < 10^{-6}$ or when the maximum number of iterations is reached. For all experiments except for the Gaussian-ring example this tolerance is always met before reaching the maximum number of iterations. For the Gaussian-ring examples 8/10 runs for $\alpha = 0.1$ reach the stopping criterion within the prescribed number of iterations, 9/10 runs for $\alpha = 0.5$, 6/10 runs for $\alpha = 0.9$ and none of the classical mirror descent ($\alpha = 1$) runs reach the prescribed tolerance (see Table 1 for details).

All of the dual gradient methods terminate once the requisite metric is less than 10^{-6} or a maximum number of iterations is met. For the Variational (1) and Variational (L) methods, the metric of interest is $\|\tilde{\nabla}\ell(u_j)\|$ and, for the Accelerated variational (L) method, it is $\|u_j - z_j\|^2$, where z_j is a second sequence used to add momentum (see Algorithm 2 in Rioux et al. (2026)). For the Variational (1) method the outer tolerance is always met first, but this is not always the case for Variational (L) and Accelerated variational (L) as the theoretical stepsizes are too conservative.

For the Gaussian-to-ring example, the gradient ascent method does not meet the prescribed tolerance for solving the regularized convex subproblem for the first iterate, but does meet it first for the subsequent iterations. For all other examples this inner tolerance is always met before the maximum number of inner iterations is reached.

Aggregation and plotting. The stationarity curves show the geometric mean of the iterate differences plus one standard deviation in base-10 log space. Objective curves show the arithmetic mean of the EGW values over runs and one standard deviation. Running minima are computed within each seed before aggregation. The aggregate curves are only plotted when at least half of the seeds converge to avoid misleading conclusions.

Table 1 compiles the average runtime for each solver excluding steps such as decomposing the matrix for the variational methods. The large standard deviation for the Gaussian-ring example stems from the fact that not all of the runs reach the prescribed tolerance in the chosen number of iterations.

Example from Houry et al. (2026) For this example, we use the standard `sinkhorn-knopp` method with the stopping threshold set to 10^{-9} and at most $10\,000$ iterations. All methods use a maximum of $J = 300$ iterations and terminate early when $\|x_j - \tilde{x}_j\|_1^2 < 10^{-6}$ for AMD and $\|u_j - u_{j+1}\|^2 < 10^{-6}$ for the dual gradient method. The regularized quadratic subproblems in the dual gradient method are solved using gradient ascent with a stepsize of 0.5 and a maximum number of iterations set to 100.

The cost matrices are set to

$$(\mathrm{K}_0^{(p)})_{ii'} = \|x_i - x_{i'}\|_2^p, \qquad (\mathrm{K}_1^{(p)})_{kk'} = \|y_k - y_{k'}\|_2^p, \qquad p \in \{2, 6\},$$

where the support points are described in Section D.1. When $p = 2$ the costs are CNT and are not CNT when $p = 6$. We also consider two choices of regularization parameter, $\varepsilon = 0.01$ and $\varepsilon = 0.1$.

We visualize the couplings at different iterations for these settings in Figures 5 and 6; these plots connect each point in the two datasets with a line whose width and opacity scale proportionally to the weight that the given coupling matrix assigns to that pair.

E Additional results

E.1 Gaussian-ring interaction objective

When $p = 2$,

$$\mathsf{EGW}_2^{\varepsilon}(\mu_0, \mu_1) = \int \kappa_0(x, x')^2 d\mu_0 \otimes \mu_0(x, x') + \int \kappa_1(y, y')^2 d\mu_1 \otimes \mu_1(y, y') + \inf_{\pi \in \Pi(\mu_0, \mu_1)} \left\{ -2 \int \kappa_0(x, x')\kappa_1(y, y') d\pi \otimes \pi(x, y, x', y') + \varepsilon \mathsf{KL}(\pi \| \mu_0 \otimes \mu_1) \right\},$$

so that the EGW objective decomposes into a constant term that depends only on the marginals and an interaction term. It suffices, therefore, to solve the minimization problem for the interaction term and hence only the interaction part is updated as the optimization progresses.

As explained in the main text, for the Gaussian-ring example these constant terms are excessively large as the cost is taken to be $\|\cdot\|^6$. Thus, in Figure 3, we plot the progress of the interaction term only as a function of the number of iterations for the various methods to get a clearer picture of the optimization progress. In particular, we note that the progress for the classical MD method stalls and appears to oscillate.

E.2 Theoretical stepsizes

As mentioned in the main text, the theoretical stepsize from Theorem 1 or Theorem 4 can be too conservative for AMD and the method is seen to converge for larger step sizes as shown in Section 5. In this section, we report the theoretical stepsize choices for AMD (α) and the variational methods ($1/L$) for the Gaussian mixture and sphere examples. We highlight that we used the theoretical α from Theorem 4. In addition, we show in Figure 4 how AMD with the theoretical stepsize compares with the other values of α considered in the text. Overall, the theoretical stepsize leads to slower convergence, but still reaches approximate stationarity in the prescribed number of iterations in the sphere example. Experimental details are compiled in Section D.3.

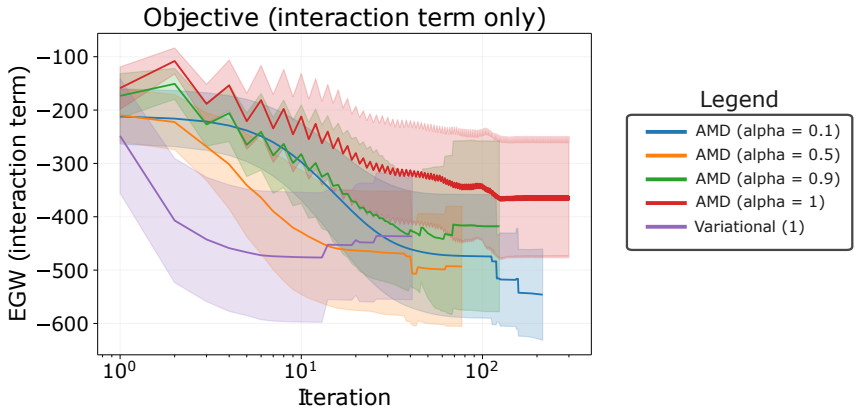


Figure 3: The interaction term from the EGW objective plotted for increasing number of iterations computed using the outputs of AMD for $\alpha \in \{0.1, 0.5, 0.9, 1\}$ and the Variational (1) method.

Method	Quantity	Gaussian mixture	Sphere
AMD	α	$(3.30 \pm 0.53) \times 10^{-4}$	$(1.53 \pm 0.42) \times 10^{-3}$
Variational (L)	L_{var}	706 ± 66	15355 ± 10740
Accelerated variational (L)	L_{acc}	704 ± 66	15353 ± 10740

Table 3: Theoretical step sizes, mean $\pm$ one standard deviation over 10 seeds.

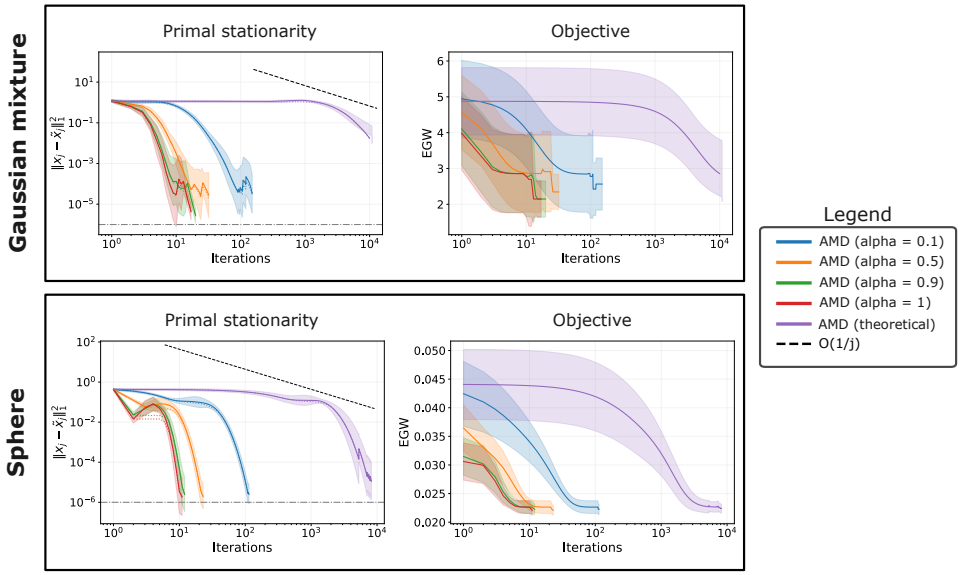


Figure 4: The primal stationarity and objective plots for AMD for $\alpha \in \{0.1, 0.5, 0.9, 1, \text{theoretical}\}$ for the Gaussian mixture and sphere examples. The theoretical step sizes per experiment are reported in Table 3.

E.3 Coupling visualizations

Below we visualize the coupling at the last few iterations for the example adapted from Houry et al. (2026). We recall that the lines between the datasets are plotted such that their thickness and opacity are equal to the mass that the coupling assigns to that particular pair.

When the intra-domain cost is $\|\cdot\|^6$ which is not CNT and $\varepsilon = 0.01$, the couplings for AMD with $\alpha = 0.9$ and $\alpha = 1$ are seen to alternate between two configurations, indicating that the method has failed to converge. When $\varepsilon = 0.1$, this issue does not occur, indicating that the failure to converge

is also caused by the choice of regularization parameter. When the intra-domain cost is set to $\|\cdot\|^2$ which is CNT, all methods converge as expected.

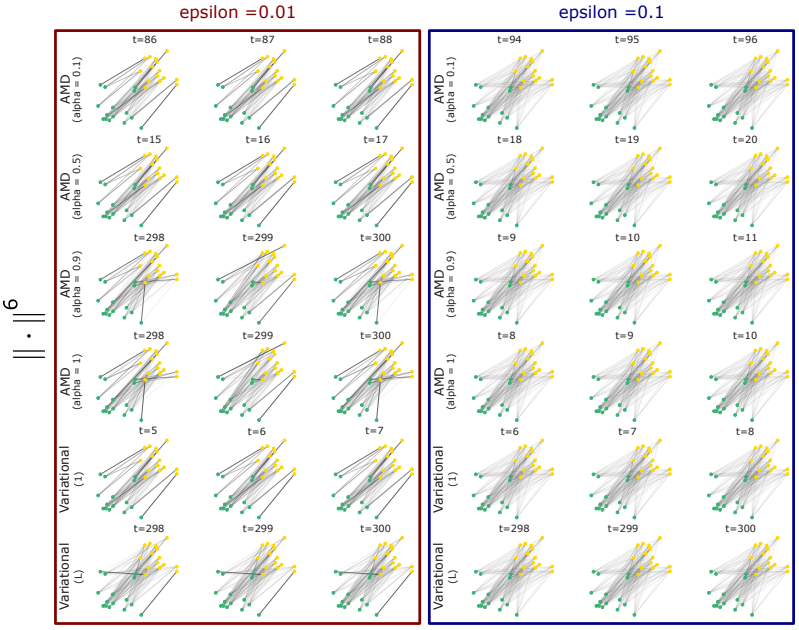


Figure 5: Couplings at the final iterations for different algorithms applied to the example adapted from Houry et al. (2026) when the intra-domain cost is set to $\|\cdot\|^6$. The support points of μ_0 are in green and those from μ_1 are in yellow. On the left, the regularization is set to $\varepsilon = 0.01$ and $\varepsilon = 0.1$ on the right.

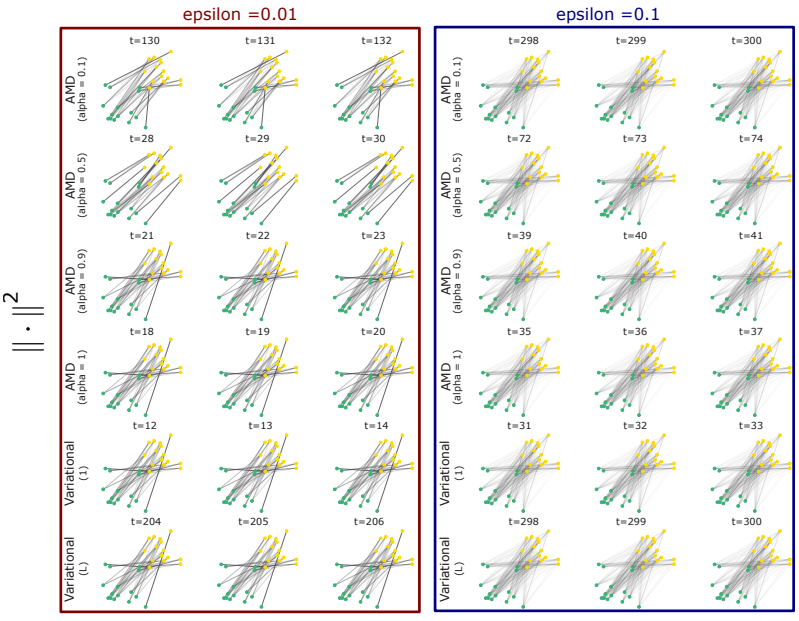


Figure 6: Couplings at the final iterations for different algorithms applied to the example adapted from Houry et al. (2026) when the intra-domain cost is set to $\|\cdot\|^2$. The support points of μ_0 are in green and those from μ_1 are in yellow. On the left, the regularization is set to $\varepsilon = 0.01$ and $\varepsilon = 0.1$ on the right

E.4 Complexity in terms of average Sinkhorn calls

Table 4 reports the average number of calls to the Sinkhorn solver for each algorithm together with one standard deviation.

Method	Gaussian mixture	Sphere	Gaussian-ring
AMD ($\alpha = 0.1$)	149 ± 49	118 ± 9	200 ± 72
AMD ($\alpha = 0.5$)	30.8 ± 10.0	23.4 ± 2.1	99.1 ± 77.6
AMD ($\alpha = 0.9$)	17.0 ± 5.4	12.5 ± 1.4	158 ± 118
AMD ($\alpha = 1$)	15.0 ± 5.0	11.1 ± 1.1	300 ± 0
Variational (1)	68.6 ± 21.0	32.2 ± 2.4	2458 ± 1350
Variational (L)	14517 ± 3407	20006 ± 0.5	—
Accelerated variational (L)	6938 ± 1544	15913 ± 4641	—

Table 4: Mean number of Sinkhorn calls ($\pm$ one standard deviation) over 10 seeds.