Semidefinite Relaxations of the Gromov-Wasserstein Distance

Junyu Chen * **Binh T. Nguyen** * **Shang Hui Koh** **Yong Sheng Soh**

Department of Mathematics
National University of Singapore
chenjunyu@u.nus.edu,binhnt@nus.edu.sg,matsys@nus.edu.sg

Abstract

The Gromov-Wasserstein (GW) distance is an extension of the optimal transport problem that allows one to match objects between incomparable spaces. At its core, the GW distance is specified as the solution of a non-convex quadratic program and is not known to be tractable to solve. In particular, existing solvers for the GW distance are only able to find locally optimal solutions. In this work, we propose a semi-definite programming (SDP) relaxation of the GW distance. The relaxation can be viewed as the Lagrangian dual of the GW distance augmented with constraints that relate to the linear and quadratic terms of transportation plans. In particular, our relaxation provides a tractable (polynomial-time) algorithm to compute globally optimal transportation plans (in some instances) together with an accompanying proof of global optimality. Our numerical experiments suggest that the proposed relaxation is strong in that it frequently computes the globally optimal solution. Our Python implementation is available at https://github.com/tbng/gwsdp.

1 Introduction

The optimal transport (OT) problem concerns the task of finding a transportation plan between two probability distributions to minimize some costs. The problem has applications in a wide range of scientific and engineering applications. For instance, in the context of machine learning, the OT problem forms the backbone of recent breakthroughs in generative modeling (Arjovsky et al., 2017; Liu et al., 2022; Lipman et al., 2022), natural language processing (Kusner et al., 2015), domain adaptation (Courty et al., 2017), and single-cell alignment (Schiebinger et al., 2019; Bunne et al., 2023, 2024).

Let $\alpha \in \Sigma_m$ and $\beta \in \Sigma_m$ be discrete probability distributions over a metric space – here $\Sigma_m := \{\alpha \in \mathbb{R}^m_+, \sum_{i=1}^m \alpha_i = 1\}$ denotes the probability simplex. Let $C \in \mathbb{R}^{m\times n}$ be the matrix such that $C_{i,j}$ models the transportation cost between points $x_i \sim \alpha$ and $y_j \sim \beta$. The (Kantorovich) formulation of the discrete OT problem (Kantorovich, 1942; Villani et al., 2009; Santambrogio, 2015; Peyré et al., 2019) is defined as the solution of the following convex optimization instance

$$\pi_{\mathcal{W}} \overset{\text{def.}}{=} \operatorname*{argmin}_{\pi\in\Pi(\alpha,\beta)} \langle C, \pi\rangle. \tag{1}$$

Here, $\Pi(\alpha,\beta) = \{\pi \in \mathbb{R}^{m\times n}_+ : \pi\mathbb{1}_n = \alpha, \pi^\top\mathbb{1}_m = \beta\}$ denotes the set of couplings between probability distributions $\alpha, \beta \in \Sigma_m$, while $\mathbb{1}_m \in \mathbb{R}^m$ denotes the vector of ones. The OT problem (1) is an instance of a linear program (LP), and hence admits a global minimizer.

*Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-2231

One limitation of the classical OT formulation in (1) is that the definition of the cost matrix C requires the probability distributions α and β to reside in the same metric space. This is problematic in application domains where we wish to compare probability distributions in different spaces, in which case there is no meaningful way to describe the cost of moving from one location to another. Such settings arise frequently in shape comparison and graph matching applications, for example.

To address such scenarios, the work of Mémoli (2011) formulates an extension of the OT problem known as the Gromov-Wasserstein distance (GW) whereby one can define an analogous OT problem given knowledge of the cost matrices for the respective spaces where α and β reside in. More concretely, let the tuple $(C, \alpha) \in \mathbb{R}^{m\times m} \times \Sigma_m$ denote a discrete metric-measure space. Given a smooth, differentiable function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the Gromov-Wasserstein distance between two discrete metric-measure spaces (C, α) and (D, β) is defined by

$$\mathrm{GW}(C, D, \alpha, \beta) \overset{\text{def.}}{=} \min_{\pi\in\Pi(\alpha,\beta)} \ell(C_{i,k}, D_{j,l})\pi_{i,j}\pi_{k,l} = \min_{\pi\in\Pi(\alpha,\beta)} \langle \mathbf{L}(C, D) \otimes \pi, \pi\rangle. \tag{GW}$$

Here, the transportation cost is specified by the four-way tensor that measures the discrepancy between the metrics C and D

$$\mathbf{L}(C, D)_{i,j,k,l} \overset{\text{def.}}{=} \ell(C_{i,k}, D_{j,l}). \tag{2}$$

The squared loss error, for instance, is a common choice. Following Peyré et al. (2016), we define the tensor-matrix multiplication by

$$[\mathbf{L} \otimes \pi]_{i,j} \overset{\text{def.}}{=} \sum_{k,l} \mathbf{L}_{i,j,k,l}\pi_{k,l}.$$

The GW distance has been applied widely to machine learning tasks, most notably on graph learning (Vayer et al., 2019a; Xu et al., 2019; Vincent-Cuaz et al., 2021, 2022). It is an instance of a quadratic program (QP) – these are optimization instances in which we minimize a quadratic objective subject to some linear inequalities. To see this, one can re-write the objective in (GW) in terms of vectorized matrices

$$\min_{\pi} \ \langle \mathrm{vec}(\pi), L\ \mathrm{vec}(\pi)\rangle \ \text{ s.t. } \ \pi \in \Pi(\alpha, \beta). \tag{GW+}$$

Here, $(L_{ij,kl})_{ij,kl} \in \mathbb{R}^{mn\times mn}$ denotes the flattened 2-dimension tensor of $\mathbf{L}$, while the vectorization of a matrix $\pi \in \mathbb{R}^{m\times n}$ is given by

$$\mathrm{vec}(\pi) \overset{\text{def.}}{=} [\pi_{11}, \pi_{21}, \ldots, \pi_{m1}, \ldots, \pi_{mn}]^\top \in \mathbb{R}^{mn}.$$

The constraint $\pi \in \Pi(\alpha, \beta)$ is convex, and in fact linear. On the other hand, the matrix L need not be positive semidefinite, and as such, the QP instance in (GW+) is non-convex in general. In fact, in the cases where the matrix L arises as the difference of cost matrices (2), L is *never* positive semidefinite as these are zero on the diagonal while the off-diagonal entries are non-negative.

2 Main Contributions

The main contribution of this work is to propose a strong semidefinite programming (SDP)-based relaxation for the Gromov-Wasserstein distance that leads to globally optimal solutions in many instances. Concretely, let (π_{sdp}, P_{sdp}) denote an optimal solution to the following

$$\begin{aligned}
\text{GW-SDP}(C, D, \alpha, \beta) \quad \overset{\text{def.}}{=} \quad & \min_{\substack{\pi\in\mathbb{R}^{m\times n},\\ P\in\mathbb{R}^{mn\times mn}}} \quad \langle L, P\rangle \\
\text{s.t.} \quad & \begin{pmatrix} P & \mathrm{vec}(\pi) \\ \mathrm{vec}(\pi)^\top & 1 \end{pmatrix} \succeq 0 \\
& \pi \in \Pi(\alpha, \beta) \\
& P\mathrm{vec}(e_i \mathbb{1}_n^\top) = \alpha_i \mathrm{vec}(\pi), i \in [m] \\
& P\mathrm{vec}(\mathbb{1}_m e_j^\top) = \beta_j \mathrm{vec}(\pi), j \in [n] \\
& P \geq 0
\end{aligned} \tag{GW-SDP}$$

Here, e_i denotes the standard basis vector whose i-th entry is 1. This relaxation can be viewed as the Lagrangian dual of the GW problem augmented with constraints that relate the linear and quadratic

terms of transportation plans (we discuss these aspects in greater detail in Appendix B). A simpler way to express the condition $P\text{vec}(e_i \mathbb{1}_n^\top) = \alpha_i \text{vec}(\pi)$ is to note that

$$\begin{aligned} P\text{vec}(e_i \mathbb{1}_n^\top) = \alpha_i \text{vec}(\pi) \quad &\Leftrightarrow \quad \textstyle\sum_j P_{(i,j),(k,l)} = \alpha_i \pi_{k,l}, \\ P\text{vec}(\mathbb{1}_m e_j^\top) = \beta_j \text{vec}(\pi) \quad &\Leftrightarrow \quad \textstyle\sum_i P_{(i,j),(k,l)} = \beta_j \pi_{k,l}. \end{aligned}$$

We begin by noting a basic observation: Let $\pi \in \Pi(\alpha, \beta)$ be a transportation plan. Then the tuple $(\pi, P) = (\pi, \text{vec}(\pi)\text{vec}(\pi)^\top)$ is a feasible solution to (GW-SDP) since $\text{vec}(\pi)\text{vec}(\pi)^\top \text{vec}(e_i \mathbb{1}_m^\top) = \text{vec}(\pi)\langle \pi, e_i \mathbb{1}_m^\top \rangle = \text{vec}(\pi)\langle \pi \mathbb{1}_n, e_i \rangle = \alpha_i \text{vec}(\pi)$. The inequalities for β follow analogously. This implies that the optimal value of (GW-SDP) is a lower bound to the GW problem (GW): Let $\pi^\star$ denote an optimal solution to the GW problem (note: this is (GW), which is equivalent to (GW+)). By recalling that the tuple $(\text{vec}(\pi^\star), \text{vec}(\pi^\star)\text{vec}(\pi^\star)^\top)$ is a feasible solution to (GW-SDP), one has

$$\langle P_{sdp}, L \rangle \leq \langle \text{vec}(\pi^\star), L\ \text{vec}(\pi^\star) \rangle = \langle \pi^\star, \mathbf{L} \otimes \pi^\star \rangle. \tag{3}$$

The inequality (3) provides us with a principled way of *certifying* global optimality of a given transportation plan. Let $\pi \in \Pi(\alpha, \beta)$ be an arbitrary transportation plan. A natural approach to quantify the quality of π is to compare its objective value with the optimal choice:

$$\text{Apx. Ratio}(\pi) \ := \ \frac{\langle \pi, \mathbf{L} \otimes \pi \rangle}{\langle \pi^\star, \mathbf{L} \otimes \pi^\star \rangle}.$$

This ratio is at least one and is equal to one if π is also globally optimal. A consequence of (3) is the following upper bound

$$\text{Apx. Ratio}(\pi_{sdp}) \leq \frac{\langle \pi_{sdp}, \mathbf{L} \otimes \pi_{sdp} \rangle}{\langle P_{sdp}, L \rangle}. \tag{4}$$

Note that all the quantities in the RHS can be computed efficiently as the solution of a SDP. Suppose we are able to do so, and in the process evaluate the RHS to be equal to one. *Then, we have a proof that π_{sdp} is the global optimal solution to the GW problem.* In a recent work that appeared during the reviewing process of our work, the GW problem is shown to be intractable in general (Kravtsova, 2024). What our discussion shows is that it is possible, in some instances, to obtain the globally optimal solution via a polynomial-time algorithm by solving (GW-SDP), and with a guarantee that the obtained solution is indeed globally optimal. In fact, the instances for which one can obtain an upper bound equal to one using (GW-SDP) is not as far-fetched as one might think: our numerical experiments in Section 4 show that this happens quite often, and especially so whenever $m = n$. **No restrictions on cost tensor.** One of the strengths of our proposed SDP relaxation is that it is valid for *all* cost tensors $\mathbf{L}$. This stands in contrast with other methods like the entropic GW (Peyré et al., 2016), which is only applicable to the cost that can be decomposed to a specific form such as the ℓ_2 or discrete KL loss.

3 SDP Relaxations of QPs

We motivate the relaxation in (GW-SDP). The starting point is to recognize that the GW problem is an instance of a QPs – these are optimization instances of the form

$$\min_{x \in \mathbb{R}^n} \qquad x^\top A x + 2b^\top x + c \qquad \text{s.t.} \qquad Bx \leq d. \tag{5}$$

QPs are an important class of optimization problems. If the matrix A is PSD, then the objective is convex, and the QP instance can be solved tractably using standard software (Nocedal and Wright, 2006). The problem becomes difficult if A contains negative eigenvalues. The general class of QPs is NP-hard; for instance, it contains the problem of finding the maximum clique of a graph (Motzkin and Straus, 1965). In fact, the presence of a *single* negative eigenvalue in A is sufficient to make the class of QPs NP-hard (Pardalos and Vavasis, 1991). The typical approach to solving a quadratic program exactly is via a branch-and-bound type of algorithm. Other approaches include relating QPs to the class of co-positive programming, mixed integer linear programming, and deploying SDP relaxations (Bomze and de Klerk, 2002) – typically, these methods are used as a sub-routine within a branch-and-bound procedure.

https://doi.org/10.52202/079017-2231

Standard SDP Relaxation. The first step of SDP relaxation is to express the quadratic terms with a PSD matrix whose rank is one. Concretely, the QP instance (5) is equivalent to the following:

$$\begin{aligned} \min_{x\in\mathbb{R}^n, X\in\mathbb{R}^{n\times n}} \quad & \operatorname{tr}(AX) + 2b^\top x + c \\ \text{s.t.} \quad & Bx \le d \\ & \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \succeq 0, \ \operatorname{rank} \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} = 1 \end{aligned} .$$

This optimization instance is not convex because of the rank-one constraint. The second step is to simply omit the rank constraint, which yields a semidefinite program and therefore is convex. This is the standard SDP relaxation for QPs (the technique applies more generally to quadratically constrained quadratic programs – QCQPs).

By applying the same sequence of steps to (GW+), the standard SDP relaxation one arrives at is the following:

$$\begin{aligned} \min_{\pi\in\mathbb{R}^{m\times n}, P\in\mathbb{R}^{mn\times mn}} \quad & \langle L, P\rangle \\ \text{s.t.} \quad & \begin{pmatrix} P & \operatorname{vec}(\pi) \\ \operatorname{vec}(\pi)^\top & 1 \end{pmatrix} \succeq 0 \\ & \pi \in \Pi(\alpha,\beta) \end{aligned} \tag{6}$$

Problem (6) is a tractable convex semidefinite programming, which can be efficiently solved in polynomial time. If the solution to (6) (and the subsequent SDP relaxations we introduce) has a rank equal to one, we would have solved the original GW problem (GW). Unfortunately, the feasible region of P in (6) is not compact, and the optimal value to (6) is unbounded below in general.

Proposition 3.1. *The optimization instance* (6) *is unbounded below.*

Tightening the Relaxation. As such, it is necessary to augment (6) with additional constraints to further strengthen the relaxation. Recall that the relaxation (6) is exact if $P = \operatorname{vec}(\pi)\operatorname{vec}(\pi)^\top$. Therefore, a simple way to improve the relaxation is to add any linear constraints that is satisfied by solutions of the form $(\pi, P) = (\pi, \operatorname{vec}(\pi)\operatorname{vec}(\pi)^\top)$.

First, $\pi \ge 0$ for all $\pi \in \Pi(\alpha,\beta)$, and hence $\operatorname{vec}(\pi)\operatorname{vec}(\pi)^\top \ge 0$. This means we may freely impose

$$P \ge 0. \tag{Nng.}$$

Second, note that $\sum_i \pi_{ij}\pi_{kl} = \pi_{kl}(\sum_i \pi_{ij}) = \pi_{kl}\beta_j$. Subsequently, we may impose $\sum_i P_{(i,j),(k,l)} = \beta_j\pi_{kl}$. This leads to the following set of equalities:

$$P\operatorname{vec}(e_i\mathbb{1}_n^\top) = \alpha_i\operatorname{vec}(\pi), i\in[m], \qquad P\operatorname{vec}(\mathbb{1}_m e_j^\top) = \beta_j\operatorname{vec}(\pi), j\in[n]. \tag{Mar.}$$

The proposed SDP relaxation (GW-SDP) is precisely (6) with the additional constraints (Nng.) and (Mar.). In addition, the set of matrices P satisfying (Mar.) have trace at most one. Hence the feasible region is a subset of PSD matrices with trace at most one, which is compact.

Relation to the QAP. We point out that the constraints (Nng.) and (Mar.) have been previously proposed for a different but closely related problem known as Quadratic Assignment Problem (QAP) (Dym et al., 2017; Kezurer et al., 2015; Zhao et al., 1998). Mathematically, the QAP problem can be viewed as equivalent to (GW+) but with the additional restriction that $m = n$ and that π is a *permutation* matrix. The work in Zhao et al. (1998) proposes a SDP relaxation that is effectively equivalent to (GW-SDP) but with additional linear equalities implied by orthogonality $\pi\pi^\top = \pi^\top\pi = I$. The works in Dym et al. (2017); Kezurer et al. (2015) subsequently build on the ideas in Zhao et al. (1998) and propose more scalable alternatives while providing tight relaxations.

The key difference between the QAP and the GW problem we investigate is that π is not necessarily a permutation matrix in the GW problem, and necessarily so if $m \ne n$. As such, the relaxation in Zhao et al. (1998) is invalid. Our contribution is to recognize that, by omitting the constraints corresponding to orthogonality, one obtains an SDP relaxation that now becomes valid for the GW problem, which leads to good practical performance.

No need for rounding. One important property of the relaxation in (GW-SDP) is that the output will always be a feasible transportation plan in $\Pi(\alpha,\beta)$. This means that no additional rounding

is necessary. This is vastly different from combinatorial optimization problems including the QAP where the optimal solution to the relaxation is not guaranteed to be a feasible solution, and additional rounding steps may be necessary.

4 Numerical Experiments with Off-the-shelf Convex Solvers

In this section, we implement our proposed SDP relaxation using an off-the-shelf solver. We compare our method with the Conditional Gradient (CG-GW) solver for finding local solutions (Vayer et al., 2019a), and the Sinkhorn projections solver for computing solutions to the entropic GW (eGW) problem by Peyré et al. (2016). Both of the latter are implemented in the Python Optimal Transport library (PythonOT, Flamary et al. 2021). The goal is to show that our proposed SDP relaxation frequently computes the global optimal transportation plan whereas existing methods frequently do not.

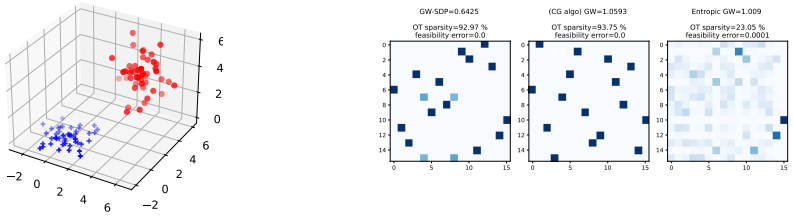
In what follows, we will use the 2-Gromov-Wasserstein distance, *i.e.* the cost function is squared Euclidean norm. We solve the GW-SDP instance implemented in CVXPY (Diamond and Boyd, 2016) using the SCS and MOSEK solvers (ApS, 2022; O'Donoghue et al., 2016).

4.1 Matching Gaussian Distributions

In this example, we estimate the GW distance between two Gaussian point clouds, one in $\mathbb{R}^2$, and the other in $\mathbb{R}^3$. A visualization of this dataset can be found in Figure 1a. The classical optimal transport formulation such as the likes of Wasserstein-2 distance does not apply because the two point clouds belong to different spaces.

As seen in a qualitative demonstration of Figure 1b, our algorithm returns optimal transport plans that are as sparse as the transportation plans obtained via the Conditional Gradient descent solver of Python OT for GW distance (CG-GW). We also vary the number of sample points and calculate the value of the objective function $\langle \pi, \mathbf{L} \otimes \pi \rangle$. As shown in Figure 2a, the transport plans obtained by (GW-SDP) consistently returns smaller objective value (orange line) than those obtained via the GW-CG counterpart from PythonOT (blue line) and its entropic regularization (green line). This shows that the transport plans computed by PythonOT, for instance, are in fact frequently sub-optimal.

In Figure 2b, we plot the estimated approximation ratio across different numbers of sample points. We notice that in this scenario of Gaussian matching, the estimated approximation ratio is close to 1.0 in most instances – this tells us that the (GW-SDP) frequently computes globally optimal transportation plans. In contrast, local methods such as PythonOT often do not. Note that we also observed the sparsity of the SDP-GW transport plans in this varying scenario.



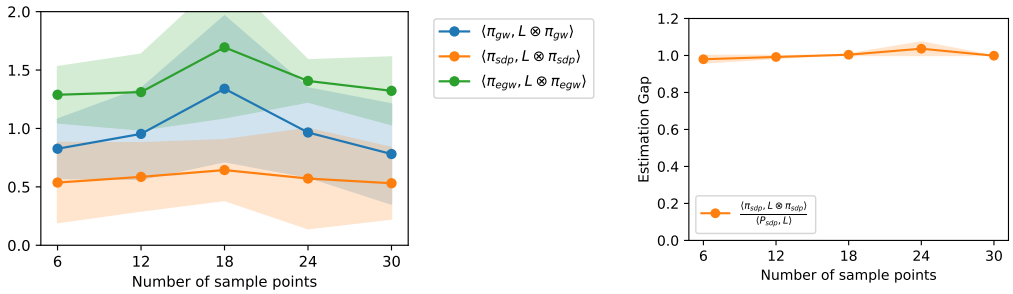
(a) Datasets visualization

(b) Solution of the OT plans

Figure 1: **Left:** source distribution (2D, blue dots) and target distribution (3D, red dots). For ease of visualization, we lift the source $\mathbb{R}^2$ mm-spaces into target $\mathbb{R}^3$ by padding the third coordinate to zero. **Right:** OT solutions of GW-SDP (our algorithm), CG-GW (conditional gradient descent, default solver of PythonOT) and entropic OT solver. The OT plans from GW-SDP is almost sparse in the same manner to CG-GW, while the eGW is not.

Scenario where $\mathbf{m \neq n}$. The bulk of our experiments focus on the setting where $m = n$. We performed an experiment where $m \neq n$: the number of samples in one distribution is fixed ($n = 8$) and we vary the number of samples m in the other distribution. From our results in Figure 3, we notice that the relaxation is exact whenever m is a multiple of n. On the other hand, when m is not a multiple of n, we still observe exactness, but much less frequently.

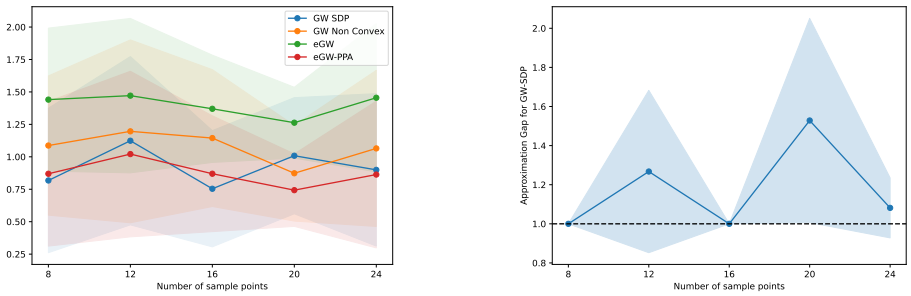
https://doi.org/10.52202/079017-2231



(a) Objective of problem (GW-SDP).

(b) Estimation gap (RHS of (4)).

Figure 2: Value of the objective (left) and approximation ratio (right) with a varying number of sample points, calculated on 10 runs of the Gaussian matching experiment.



(a) GW Distances with different solvers.

(b) Approximation Gap for GW-SDP.

Figure 3: Gaussian Matching experiments with the sample points m from source distribution varying while the sample points from target distribution keeping fixed $n = 8$. Average of 20 runs.

Runtime Comparisons Table 1 presents the run-time of the GW-SDP problem in Experiment 1, running on a PC with 8 cores CPU and 32GB of RAM. In these experiments, the cost matrix C is pre-computed (i.e. assumed given). As such, the run-time is independent of the data dimension. The GW-SDP has a matrix of dimension $mn \times mn$, which is slower than most local and entropic solvers. However, the solvers we implement (SCS and MOSEK) are off-the-shelf and are general SDP solvers that do not exploit special structures in the problem and do not provide options to use initialization of the transport plans. (SCS is a first-order method, but we are otherwise unaware of its complexity). We want to emphasize that in most settings where SDPs are applied, one will always try to develop specialized solvers that exploit the structure of the problem. In our setup, the optimal solution has low rank, and is rank-one if the relaxation is tight. There are numerous well-established methods for exploiting such structure. This is the subject of ongoing work.

Table 1: Average run-time in seconds for experiment in Figure 1a (matching Gaussians with varying number of samples n).

n	**GW-SDP**	**GW-CG**	**eGW** ($\varepsilon = 0.1$)
6	0.2437 (0.0265)	0.0005 (0.000041)	0.226 (0.1145)
12	11.615 (2.4088)	0.0006 (0.00003)	0.2596 (0.0726)
20	216.3645 (14.1123)	0.0014 (0.000017)	0.4923 (0.1500)

Comparisons of GW-SDP solver and GW-CG solver when number of sample points n increase. We increase the number of samples for GW-CG (non-convex GW solver using conditional gradient descent or Frank-Wolfe algorithm) vs our GW-SDP solver for a fixed number of samples. From Table 2, we noticed that the objective value for GW-CG decreases as we increase the number of samples. For 100000 sample points, the GW-CG algorithm is more expensive and has a poorer

objective value than our method with 10 sample points. This suggests that our method can give good approximations of the GW distance with fewer sample points than existing methods.

Table 2: Comparisons of GW-SDP solver and GW-CG solver with a varying number of sample points n.

n	GW-SDP	GW-SDP Runtime (s)	GW-CG	GW-CG runtime (s)
10	0.4577	6.3753	1.135940	0.000389
100			0.629425	0.007571
1000			0.540984	2.520011
10000			0.496796	138.358954

4.2 Graph Community Matching

The objective of this task is to find matching between two random graphs that are drawn from the stochastic block model (SBM) (Abbe, 2017; Holland et al., 1983) with fixed inter/intra-clusters probability (the probability that nodes inside and outside a cluster are connected, respectively). The source is a three-cluster SBM whose intra-cluster probability is $p = \{1.0, 0.95, 0.9\}$, and the target is a two-cluster SBM whose intra-cluster probability is $p = \{1.0, 0.9\}$. The inter-clusters probability is all set to 0.1. The distance matrices on each graph are created first by simulating the node features drawn from Gaussian distributions with uniform weights. Subsequently, we compute the ℓ_2 norm between nodes and shrink the value of disconnected nodes to zero to form the distance matrices.

We compare the transportation plans obtained using our methods with the baseline comparisons GW-CG and eGW in Figure 4a. We note that the (GW-SDP) model typically returns a transport plan with a smaller total transportation cost (i.e., a smaller objective value) $\langle \pi_{sdp}, \mathbf{L} \otimes \pi_{sdp} \rangle$. This trend is consistent with our observations in the previous experiments. Nevertheless, we see a degree of similarity between the transportation plans provided as output by all three methods. In addition, the transportation plans computed by our method and GW-CG are both reasonably sparse. This fact is observed in multiple runs of different seeds and graph sizes.

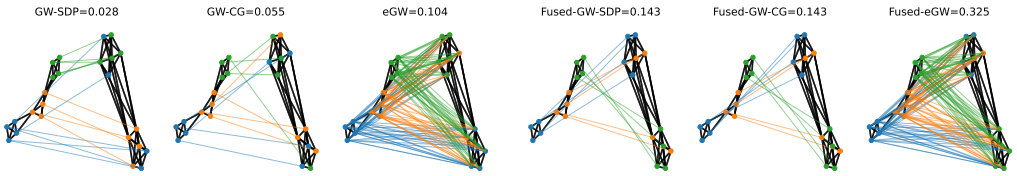
4.3 Extension of GW-SDP to Structured Data

In this example, we consider an extension of the (GW-SDP) to structured data, more specifically graphs with node features similar to the Fused-GW distance in (Vayer et al., 2019a). The discrete metric-measure space is now described by the tuple $(F, C, \alpha) \in \mathbb{R}^{m\times d} \times \mathbb{R}^{m\times m} \times \Sigma_m$, where $F \overset{\text{def.}}{=} (f_i)_i \in \mathbb{R}^d$ encodes the feature information of the sample point. The Fused GW-SDP (FGW-SDP) formulation is given by

$$
\begin{aligned}
\text{FGW-SDP}(M_{FG}, C, D, \alpha, \beta, \xi) \overset{\text{def.}}{=} \min_{\substack{\pi\in\mathbb{R}^{m\times n},\\ P\in\mathbb{R}^{mn\times mn}}} \quad & (1-\xi)\langle M_{\alpha,\beta}, \pi\rangle + \xi\langle \mathbf{L}(C,D), P\rangle \\
\text{s.t.} \quad & \begin{pmatrix} P & \text{vec}(\pi) \\ \text{vec}(\pi)^\top & 1 \end{pmatrix} \succeq 0 \qquad \text{(FGW-SDP)} \\
& \pi \in \Pi(\alpha, \beta) \\
& P\text{vec}(e_i \mathbb{1}_n^\top) = \alpha_i \text{vec}(\pi), i \in [m] \\
& P\text{vec}(\mathbb{1}_m e_j^\top) = \beta_j \text{vec}(\pi), j \in [n] \\
& P \geq 0,
\end{aligned}
$$

with $M_{FG} = d(f_j, g_j)_{i,j}$ encodes the distance between node features, and $\xi \in [0, 1]$ the interpolation parameter. Figure 4b shows the result of matching two SBM graphs with the same setting as in Section 4.2, with the exception that now we input the feature to calculate M_{FG} by ℓ_2 norm, and the structured matrices are the shortest path matrices obtained from the adjacency matrices of the graphs. We set $\xi = 0.8$ for this example. The figure shows that the output OT plans and values of (FGW-SDP) and FGW-CG (using PythonOT) are identical, while entropic Fused-GW returned a higher value and a denser transport plan. This indicates that the SDP relaxation of Fused-GW can be useful in graph matching applications, akin to Fused-GW.

https://doi.org/10.52202/079017-2231

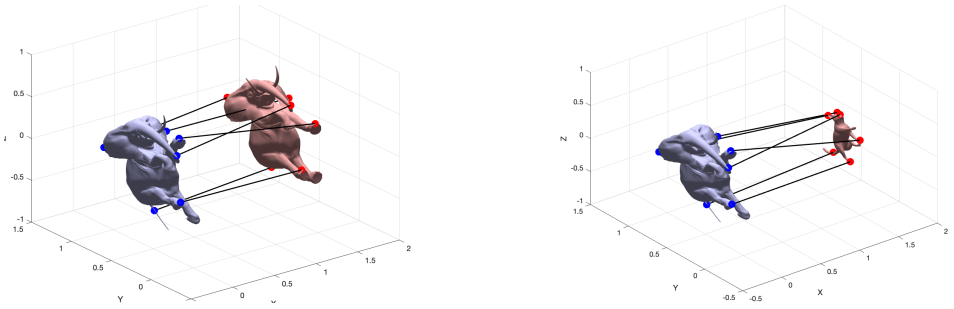


(a) Matching with GW. (b) Matching with FGW.

Figure 4: Value of the objective on the synthetic graph matching task, from the three-block SBM (left) to the two-block SBM (right). **Upper:** calculated using GW. **Lower:** calculated using Fused-GW.

4.4 Using GW-SDP on Realistic Shape-Matching Task

We use a publicly available dataset of triangular meshes (Sumner and Popović, 2004). The dataset comprises 72 objects from seven different classes, from which we chose samples of class cat, elephant, and horse. For each object, we first chose 4 representative points (the right back foot, the left front foot, the nose, and the tail) for each object and then selected another 14 points following the Euclidean farthest point sampling (fps) procedure. The distance matrices C and D are computed using Dijkstra's algorithm. Each object's probability measure is chosen to be uniform. We apply (GW-SDP) to the corresponding metric-measure spaces to determine the correspondence between the selected vertices across different objects. Two representative examples are given in Figure 5. For better visualization, in the representative examples we sampled only 6 points (4 representative points and 2 selected using fps).



(a) (b)

Figure 5: Correspondence between different 3D objects obtained by (GW-SDP). Left: Correspondence between two elephants. Right: Correspondence between an elephant and a cat. For both cases, (GW-SDP) returns one-one mappings.

Table 3 illustrates the results when we perform matching of distance matrices across different objects. In general, we expect shapes of the same animals to have a smaller GW distance than shapes of different animals, which is indeed the case for the three GW formulations. We still notice that GW-SDP consistently returns the smallest value when performing the same matching task.

Table 3: Value of different GW formulations for the realistic 3D shape matching dataset, visualized in Figure 5. GW-SDP consistently returns the smallest value when performing the same matching task.

	GW-SDP	**GW-CG**	**eGW-PPA**
Elephant-Elephant	0.007416	0.043879	0.025688
Elephant-Cat	0.015695	0.050594	0.042214
Cat-Cat	0.006549	0.016634	0.006757
Cat-Horse	0.011040	0.033736	0.011041
Horse-Horse	0.006287	0.033768	0.007395

5 Duality

Given a generic optimization instance, the dual optimization instance concerns the task of finding optimal lower bounds to the primal instance. The (Lagrangian) dual to *any* optimization instance is a convex program in general and provides a principled way of obtaining convex relaxations of difficult optimization instances. We briefly discuss some of these relationships. A more detailed discussion on the duality can be found in Appendix B.

It is possible to obtain the relaxation (GW-SDP) via duality. Concretely, let (GW++) refer to the original GW problem instance (GW+) with the additional and *redundant* constraints (Nng.) and (Mar.). Let (GW-Dual) refer to the Lagrangian dual of the proposed semidefinite relaxation (GW-SDP).

Theorem 5.1. *The optimization instance* (GW-Dual) *is the Lagrangian dual of* (GW++)*, which is* (GW+) *with the additional constraints* (Nng.) *and* (Mar.)*.*

The (duality) gap between (GW-Dual) and (GW++) is non-zero in general, and is equal to zero precisely when the convex relaxation (6) succeeds. These can be characterized by a rank condition:

Proposition 5.2. *Let P_{sdp} and π_{sdp} be the solution to* (GW-SDP)*. Suppose the matrix variable has rank equals to one; that is*

$$rank \begin{pmatrix} P_{sdp} & vec(\pi_{sdp}) \\ vec(\pi_{sdp})^\top & 1 \end{pmatrix} = 1.$$

Then the duality gap is zero; i.e., strong duality holds.

6 Related work

There is a substantial body of prior work concerning the GW problem in the literature. We briefly discuss some of these and explain the novelty of our work.

First, the work in Vayer et al. (2019a) applies an alternating minimization-type approach based on the conditional gradient (Frank-Wolfe) algorithm to find local optima to the GW problem. This algorithm is currently implemented and is the default choice within the Python Optimal Transport package (Flamary et al., 2021). The basic idea is to start by computing the partial derivative of the objective (GW) with respect to π:

$$G(\pi) = 2\,\mathbf{L}(C, D) \otimes \pi,$$

This is a linear OT problem that can be solved using classical OT solvers. One proceeds with an alternating minimization scheme in which one updates the gradient G with respect to $\pi^{(i-1)}$, subsequently solves for $\pi^{(i)}$ with the loss $G(\pi)$ at each ith-iteration, and finally projects $\pi^{(i)}$ into the feasible set by performing a line-search. The Conditional Gradient-based approach is not guaranteed to find globally optimal solutions; in fact, our numerical experiments in Section 4 suggest that this is quite often the case. Last, we briefly note that the work in Kerdoncuff et al. (2021) suggests a similar alternating numerical scheme.

Second, there is a body of work that aims at developing numerical schemes for finding transportation plans that approximately minimize the GW objective without incurring the expensive $O(m^2n^2)$ dependency. For instance, the work in Peyré et al. (2016) introduces an entropic regularization into the GW objective – this leads to a formulation that permits Sinkhorn scaling-like updates, much like the original scheme to solve entropic Wasserstein distance in Cuturi (2013). The work of Vayer et al. (2019b) adapts the ideas from the Wasserstein problem in one dimension in which closed-form solutions are available (this is known as the sliced Wasserstein problem, Rabin et al. 2012) to the GW context. Finally, the work in Sejourne et al. (2021); Vincent-Cuaz et al. (2021) relaxes the constraints on the probability distributions. These numerical schemes frequently lead to numerical schemes that are far more scalable than other existing methods, but they ultimately optimize for an objective that is different from the GW problem.

There is an interesting piece of work in Scetbon et al. (2022), which operates under the assumption that the cost matrices have low-rank structure. While the algorithm does not give guarantees about global optimality, it raises an interesting future direction; namely, could we develop numerical schemes for our proposed SDP relaxation that also exploit similar structures?

https://doi.org/10.52202/079017-2231

Finally, we discuss prior works that do in fact address global optimality (which is the heart of this paper): a recent work is in Mula and Nouy (2022), which suggests the use of moment sum-of-square (SOS) relaxation technique to solve the GW problem. The standard SDP relaxation for QPs on which our work is based on may be viewed, in a suitable sense, as the first level of the SOS hierarchy for polynomial optimization. Unfortunately, and as we note in Section 3, this alone is insufficient – the real novelty in our work is the addition of constraints that substantially strengthen the overall convex relaxation. A piece of related work by Villar et al. (2016) proposes a SDP relaxation of the closely related Gromov-Hausdorff problem, with an extension to the Gromov-Wasserstein problem. The relaxation is primarily designed for the Gromov-Hausdorff problem and is not equivalent to ours. The formulation also requires the probability distributions to be uniform whereas we do not. Another recent work by Ryner et al. (2023) also studies the Gromov-Hausdorff problem, and proposes a Branch-and-Bound approach for solving integer programs. The GW problem does not contain integer constraints, and hence Branch-and-Bound techniques are not applicable. That said, SDP relaxations can be used in conjunction with Branch-and-Bound. It would be interesting to see if our proposed SDP relaxations for GW suggest suitable relaxations for the Gromov-Hausdorff problem, which can be used in conjunction with the Branch-and-Bound techniques in Ryner et al. (2023).

7 Conclusions and Future Directions

In this work, we proposed a semidefinite programming relaxation of the Gromov-Wasserstein distance. Our initial results suggest that the relaxation (GW-SDP) is strong in the sense that π_{sdp} frequently coincides with the globally optimal solution; moreover, we are able to provide a proof when this actually happens. These results are exciting, as it suggests a tractable approach for solving the GW problem – at least for examples of interest – which was previously assumed to be quite difficult.

An interesting future direction is to understand precisely how difficult is an instance of the GW problem. The fact that our convex relaxations work very well for the examples we considered suggests that the GW problem might not be as difficult as we think. It is important to bear in mind that these cost tensors $\mathbf{L}$ have structure – they arise from the difference of actual cost matrices. Could it be that the difficult instances of the GW problem correspond to cost tensors $\mathbf{L}$ that are not realizable as the difference of cost matrices; e.g., they violate the triangle inequality? A concrete question to this end is: Is the GW problem corresponding to cost tensors $\mathbf{L}$ arising in practical instances tractable to solve?

A second important future direction concerns computation. One limitation of our proposed convex relaxation is that it is specified as the solution of an SDP in which the matrix dimension is mn; that is, it is equal to the dimension of the transport plan. The prohibitive dependence on the data dimension means that we are currently only able to apply the relaxation on moderate sized instances using off-the-shelf SDP solvers. It would be of interest to develop specialized algorithms to solve the proposed relaxation (GW-SDP).

Acknowledgements

We thank the anonymous reviewers for their helpful comments and feedback that helped improve our work. B.T. Nguyen is supported by NUS Start-up Grant A-0004595-00-00.

References

Abbe, E. (2017). Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531.

Agueh, M. and Carlier, G. (2011). Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924.

ApS, M. (2022). *The MOSEK optimization toolbox for Python manual. Version 10.0.*

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015). Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138.

Bomze, I. M. and de Klerk, E. (2002). Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *J. Global Optim.*, 24(2):163–185.

Bunne, C., Schiebinger, G., Krause, A., Regev, A., and Cuturi, M. (2024). Optimal transport for single-cell and spatial omics. *Nature Reviews Methods Primers*, 4(1):58.

Bunne, C., Stark, S. G., Gut, G., Del Castillo, J. S., Levesque, M., Lehmann, K.-V., Pelkmans, L., Krause, A., and Rätsch, G. (2023). Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768.

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. (2017). Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems*, 30.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.

Dym, N., Maron, H., and Lipman, Y. (2017). Ds++: A Flexible, Scalable and Provably Tight Relaxation for Matching Problems. *ACM Transactions on Graphics*, 36(184):1–14.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. (2021). Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Kantorovich (1942). On translocation of masses. *USSR AS Doklady New Series, 37, 7-8, 227-229 (in Russian)*.

Kerdoncuff, T., Emonet, R., and Sebban, M. (2021). Sampled gromov wasserstein. *Machine Learning*, 110(8):2151–2186.

Kezurer, I., Kovalsky, S. Z., Basri, R., and Lipman, Y. (2015). Tight relaxation of quadratic matching. In *Computer graphics forum*, volume 34, pages 115–128. Wiley Online Library.

Kong, L., Li, J., Tang, J., and So, A. M.-C. (2024). Outlier-robust gromov-wasserstein for graph data. *Advances in Neural Information Processing Systems*, 36.

Kravtsova, N. (2024). The NP-hardness of the Gromov-Wasserstein Distance. *arXiv*, abs/2408.06525.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2022). Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, X., Gong, C., and Liu, Q. (2022). Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

 https://doi.org/10.52202/079017-2231

Motzkin, T. and Straus, E. G. (1965). Maxima for graphs and a new proof of a theorem of tur´an. *Canadian Journal of Mathematics*.

Mula, O. and Nouy, A. (2022). Moment-sos methods for optimal transport problems. *arXiv preprint arXiv:2211.10742*.

Mémoli, F. (2011). Gromov–Wasserstein Distances and the Metric Approach to Object Matching. *Foundations of Computational Mathematics*, 11(4):417–487.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer.

O'Donoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068.

Pardalos, P. M. and Vavasis, S. A. (1991). Quadratic programming with one negative eigenvalue is np-hard. *Journal of Global optimization*, 1(1):15–22.

Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Peyré, G., Cuturi, M., and Solomon, J. (2016). Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pages 2664–2672. PMLR.

Rabin, J., Peyré, G., Delon, J., and Bernot, M. (2012). Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, pages 435–446. Springer.

Ryner, M., Kronqvist, J., and Karlsson, J. (2023). Globally solving the Gromov-Wasserstein problem for point clouds in low dimensional Euclidean spaces. *arXiv preprint arXiv:2307.09057*.

Santambrogio, F. (2015). Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94.

Scetbon, M., Peyré, G., and Cuturi, M. (2022). Linear-time gromov wasserstein distances using low rank couplings and costs. In *International Conference on Machine Learning*, pages 19347–19365. PMLR.

Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., et al. (2019). Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.

Sejourne, T., Vialard, F.-X., and Peyré, G. (2021). The unbalanced gromov wasserstein distance: Conic formulation and relaxation. In *Advances in Neural Information Processing Systems*, volume 34, pages 8766–8779.

Sumner, R. W. and Popović, J. (2004). Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405.

Vayer, T., Courty, N., Tavenard, R., Laetitia, C., and Flamary, R. (2019a). Optimal transport for structured data with application on graphs. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6275–6284. PMLR.

Vayer, T., Flamary, R., Tavenard, R., Chapel, L., and Courty, N. (2019b). Sliced gromov-wasserstein. In *Advances in Neural Information Processing Systems*, volume 33, pages 14753–14763.

Villani, C. et al. (2009). *Optimal transport: old and new*, volume 338. Springer.

Villar, S., Bandeira, A. S., Blumberg, A. J., and Ward, R. (2016). A polynomial-time relaxation of the gromov-hausdorff distance. *arXiv*, abs/1610.05214.

Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., and Courty, N. (2022). Template based graph neural network with optimal transport distances. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 11800–11814. Curran Associates, Inc.

Vincent-Cuaz, C., Vayer, T., Flamary, R., Corneli, M., and Courty, N. (2021). Online graph dictionary learning. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10564–10574. PMLR.

Xu, H., Luo, D., Zha, H., and Duke, L. C. (2019). Gromov-Wasserstein learning for graph matching and node embedding. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6932–6941. PMLR.

Zhao, Q., Karisch, S. E., Rendl, F., and Wolkowicz, H. (1998). Semidefinite Programming Relaxations for the Quadratic Assignment Problem. *Journal of Combinatorial Optimization*, 2:71–109.

 https://doi.org/10.52202/079017-2231

A Proofs of Main Results

Proof of Proposition 3.1. Let $1 \leq s, t \leq mn$ be coordinates such that $L_{st} > 0$. Let $v := v_{(s,t)} \in \mathbb{R}^{mn}$ be a vector whose s-th entry is 1, whose t-th entry is -1, and whose remaining entries are zeros. Let $\tilde{\pi} \in \Pi(\alpha, \beta)$ be any transportation map and consider the matrix $P_c = \text{vec}(\tilde{\pi})\text{vec}(\tilde{\pi})^\top + cvv^\top$. Notice that $P_c \succeq \text{vec}(\tilde{\pi})\text{vec}(\tilde{\pi})^\top$ for all $c \geq 0$. Hence the choice of variables $\pi = \tilde{\pi}$ and $P = P_c$ are feasible. Then notice that the objective evaluates to

$$\langle L, P_c \rangle = \langle L, \text{vec}(\tilde{\pi})\text{vec}(\tilde{\pi})^\top \rangle + c(L_{ss} - 2L_{st} + L_{tt}) = \langle L, \text{vec}(\tilde{\pi})\text{vec}(\tilde{\pi})^\top \rangle - 2cL_{st}.$$

We obtain the last equality by noting that $L_{ii} = 0$ for all i (this is a property of cost matrices). The result follows by taking $c \to +\infty$. □

B Duality

Given a generic optimization instance, the dual optimization instance concerns the task of finding optimal lower bounds to the primal instance. The (Lagrangian) dual to *any* optimization instance is a convex program in general. As such, the process of deriving the dual to any optimization instance provides a principled way of obtaining convex relaxations of difficult optimization instances.

The objective value of the dual will always be a lower bound to the primal instance – this is precisely *weak duality*. In some cases, however, the objective values of these problems may coincide, and we call such settings *strong duality*.

In this section, we explore these relationships in the context of the GW problem. As we shall see, the proposed SDP relaxation (GW-SDP) can be viewed as being equivalent to the dual of an equivalent form of the GW problem (GW+), augmented with additional constraints that relate the linear and quadratic terms of transportation maps specified by (Nng.) and (Mar.). To simplify notation, we denote

$$\begin{aligned} a_i &= \text{vec}(e_i \mathbb{1}_n^\top), \\ b_j &= \text{vec}(\mathbb{1}_m e_j^\top). \end{aligned}$$

We proceed to describe the dual program. We start by defining the following dual variables:

$$\begin{aligned}
&\begin{pmatrix} Y & y \\ y^\top & t \end{pmatrix} \succeq 0 && : \begin{pmatrix} P & \text{vec}(\pi) \\ \text{vec}(\pi)^\top & 1 \end{pmatrix} \succeq 0 && \text{(a)} \\
&\lambda_i \in \mathbb{R} && : a_i^\top \text{vec}(\pi) = \alpha_i, \quad i \in [m] && \text{(b)} \\
&\mu_j \in \mathbb{R} && : b_j^\top \text{vec}(\pi) = \beta_j, \quad j \in [n] && \text{(c)} \\
&Z \geq 0 && : P \geq 0 && \text{(d)} \\
&\eta^i \in \mathbb{R}^{mn} && : Pa_i = \alpha_i \text{vec}(\pi), \quad i \in [m] && \text{(e)} \\
&\theta^j \in \mathbb{R}^{mn} && : Pb_j = \beta_j \text{vec}(\pi), \quad j \in [n] && \text{(f)}
\end{aligned}$$

As a reminder, the constraints in (b) and (c) specify the transportation map $\Pi(\alpha, \beta)$ while the constraints in (e) and (f) correspond to (Mar.).

Theorem B.1. *The Lagrangian dual of* (GW-SDP) *is given as follows*

$$\begin{aligned}
\max \quad & \lambda^\top \alpha + \mu^\top \beta - t \\
\text{s.t.} \quad & \begin{pmatrix} L - Z + \frac{1}{2}\sum_{i=1}^m (\eta^i a_i^\top + a_i \eta^{i\top}) + \frac{1}{2}\sum_{j=1}^n (\theta^j b_j^\top + b_j \theta^{j\top}) & y \\ y^\top & t \end{pmatrix} \succeq 0, \\
& \sum_{i=1}^m \left(\lambda_i a_i + \alpha_i \eta^i\right) + \sum_{j=1}^n \left(\mu_j b_j + \beta_j \theta^j\right) + 2y \leq 0, \\
& Z \geq 0.
\end{aligned} \tag{GW-Dual}$$

Furthermore, strong duality holds; that is, the duality gap between (GW-SDP) *and* (GW-Dual) *is zero.*

It turns out that it is possible to derive the dual instance (GW-Dual) *directly* from (GW+), an equivalent formulation of the original GW problem, with additional constraints specified by (Nng.) and (Mar.). Concretely, consider

$$\begin{aligned} \min_{\substack{\pi\in\mathbb{R}^{m\times n}\\ P\in\mathbb{R}^{mn\times mn}}} \quad & \langle L, P\rangle \\ \text{s.t.} \quad & P = \text{vec}(\pi)\text{vec}(\pi)^\top \\ & \pi \in \Pi(\alpha,\beta) \\ & P\text{vec}(e_i \mathbb{1}_n^\top) = \alpha_i \text{vec}(\pi), i \in [m] \\ & P\text{vec}(\mathbb{1}_m e_j^\top) = \beta_j \text{vec}(\pi), j \in [n] \\ & P \geq 0. \end{aligned} \tag{GW++}$$

The last three constraints on P are always satisfied so long as $P = \text{vec}(\pi)\text{vec}(\pi)^\top$, where $\pi \in \Pi(\alpha,\beta)$ is a transportation map. Hence these constraints on P and π are technically redundant within (GW++). That is, the optimization instances (GW+) and (GW++) are equivalent. However, the Lagrangian dual of these optimization instances are different, and we summarize this observation in the following.

Theorem B.2. (GW-Dual) *is the Lagrangian dual of* (GW++)*.*

The (duality) gap between (GW-Dual) and (GW++) is non-zero in general, and is equal to zero precisely when the convex relaxation (6) succeeds. These can be characterized by a rank condition satisfied by the optimal solutions, namely:

Proposition B.3. *Let P_{sdp} and π_{sdp} be the solution to GW-SDP. Suppose the matrix variable has rank equals to one, that is*

$$\textit{rank} \begin{pmatrix} P_{sdp} & \textit{vec}(\pi_{sdp}) \\ \textit{vec}(\pi_{sdp})^\top & 1 \end{pmatrix} = 1.$$

Then the duality gap for (GW++) *is zero; i.e., strong duality holds.*

Proof. The rank condition implies $P_{sdp} = \text{vec}(\pi_{sdp})\text{vec}(\pi_{sdp})^\top$. Subsequently, the choice of variables $\pi = \pi_{sdp}$ and $P = \text{vec}(\pi_{sdp})\text{vec}(\pi_{sdp})^\top$ is a feasible solution to (GW++). This means (GW++) attains the same objective value as the (GW-Dual). Recall from Theorem B.2 that (GW-Dual) is the dual of (GW++), and hence in this instance the duality gap is indeed zero. □

We summarize the relationships among the original GW problem, (GW+), (GW++), (GW-SDP), and (GW-Dual) in Figure 6:

- (GW+) is an equivalent reformulation of the original GW problem.
- (GW++) is derived by introducing additional redundant constraints to (GW+). Consequently, (GW+) and (GW++) share the same optimal solutions.
- (GW-SDP) is obtained by applying the standard SDP relaxation to (GW+) and introducing supplementary constraints to tighten the relaxation.
- (GW-Dual) serves as the Lagrangian dual to both (GW++) and (GW-SDP), and strong duality establishes the equivalence between (GW-Dual) and (GW-SDP).

https://doi.org/10.52202/079017-2231

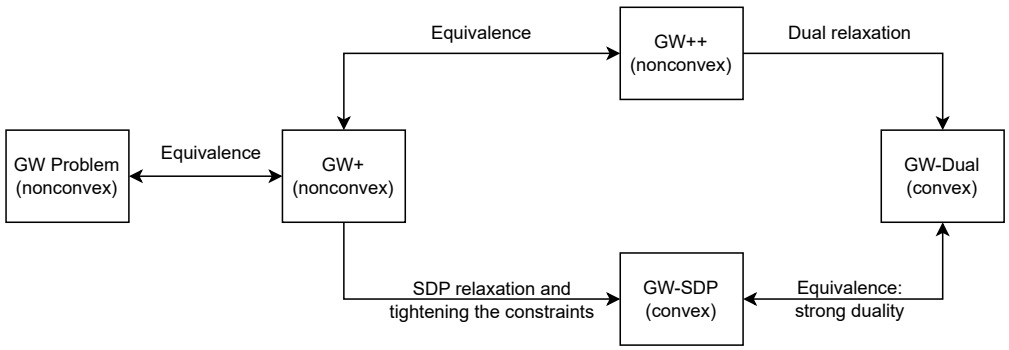


Figure 6: The relationship among the original GW problem, (GW+), (GW++), (GW-SDP) and (GW-Dual).

C Proof of Results Concerning Duality

We begin by defining these functions:

$$\begin{aligned} H(\eta,\theta,Z) &:= L - Z + \frac{1}{2}\sum_{i=1}^{m}(\eta^i a_i^\top + a_i \eta^{i\top}) + \frac{1}{2}\sum_{j=1}^{n}(\theta^j b_j^\top + b_j \theta^{j\top}), \\ g(\lambda,\mu,\eta,\theta,y) &:= \sum_{i=1}^{m}\left(\lambda_i a_i + \alpha_i \eta^i\right) + \sum_{j=1}^{n}\left(\mu_j b_j + \beta_j \theta^j\right) + 2y. \end{aligned}$$

Proof of Theorem B.1. [Deriving the dual program]: The dual function of the GW-SDP problem is given by

$$\begin{aligned} \min_{P,\pi\geq 0}\ & \operatorname{tr}(LP) - \operatorname{tr}(YP) - 2y^\top \operatorname{vec}(\pi) - t \\ & + \sum_{i=1}^{m}\lambda_i(\alpha_i - a_i^\top \operatorname{vec}(\pi)) + \sum_{j=1}^{n}\mu_j(\beta_j - b_j^\top \operatorname{vec}(\pi)) \\ & + \sum_{i=1}^{m}\eta^{i\top}(Pa_i - \alpha_i \operatorname{vec}(\pi)) + \sum_{j=1}^{n}\theta^{j\top}(Pb_j - \beta_j \operatorname{vec}(\pi)) \\ & - \operatorname{tr}(ZP) \\ = \min_{P}\ & \operatorname{tr}\left((H(\eta,\theta,Z) - Y)\,P\right) + \min_{\pi\geq 0}\ \langle -g(\lambda,\mu,\eta,\theta,y), \operatorname{vec}(\pi)\rangle \\ & + \lambda^\top \alpha + \mu^\top \beta - t. \end{aligned}$$

In the above minimization over P, we observe that the objective evaluates to $-\infty$ if the following does not hold

$$H(\eta,\theta,Z) = Y.$$

Similarly, in the minimization over $\pi \geq 0$, the objective evaluates to $-\infty$ if the following does not hold

$$g(\lambda,\mu,\eta,\theta,y) \leq 0.$$

We impose these as constraints, and we add the additional constraint that $Y \succeq 0$ on our dual variable, to obtain the form of the dual problem in (GW-Dual).

[Establishing zero duality gap]: Notice that (GW-SDP) and (GW-Dual) are convex programs. Hence, to show strong duality, it suffices to check that Slater's condition hold; that is, there exists a strictly feasible solution.

Consider $\eta^i = (|\lambda_{\min}(L)| + 2)\mathbb{1}$, $\lambda_i = -2m$ for $i \in [m]$, $\theta^j = 0$, $\mu_j = 0$ for $j \in [n]$, $t = 1$, $y = 0$, and

$$Z = (|\lambda_{\min}(L)| + 2)\mathbb{1}\mathbb{1}^\top - (|\lambda_{\min}(L)| + 1)I$$

for some $\mathbb{1}$, 0 and I of appropriate dimension. Then we have $Z > 0$, and

$$g(\lambda, \mu, \eta, \theta, y) = -2m \sum_{i=1}^{m} a_i + \sum_{i=1}^{m} \mathbb{1} = -m\mathbb{1} < 0.$$

Additionally, since $H(\eta, \theta, Z) = L + (|\lambda_{\min}(L)| + 1)I \succ 0$, $t > 0$ and $y = 0$, it follows that the LHS of the first constraint in (GW-Dual) is positive definite. Therefore, we find a feasible solution of (GW-Dual) such that strict inequality holds for all inequality constraints. Strong duality then follows. □

Proof of Theorem B.2. In addition to the dual variables (b)-(f), we define these additional dual variables:

$$\begin{aligned} Y \in \mathbb{R}^{mn \times mn} \quad &: P = \text{vec}(\pi)\text{vec}(\pi)^\top \\ z \geq 0 \quad &: \text{vec}(\pi) \geq 0 \end{aligned}$$

Then the dual function of (GW++) is given by

$$\begin{aligned} &\min_{\pi, P} \text{tr}(LP) - \text{tr}(Y(P - \text{vec}(\pi)\text{vec}(\pi)^\top)) + \sum_{i=1}^{m} \lambda_i(\alpha_i - a_i^\top \text{vec}(\pi)) + \sum_{j=1}^{n} \mu_j(\beta_j - b_j^\top \text{vec}(\pi)) \\ &\quad + \sum_{i=1}^{m} {\eta^i}^\top (Pa_i - \alpha_i \text{vec}(\pi)) + \sum_{j=1}^{n} {\theta^j}^\top (Pb_j - \beta_j \text{vec}(\pi)) - \text{tr}(ZP) - z^\top \text{vec}(\pi) \\ &= \min_{P} \underbrace{\text{tr}((H(\eta, \theta, Z) - Y)\, P)}_{A_1} \\ &\quad + \min_{\pi} \underbrace{\text{vec}(\pi)^\top Y \text{vec}(\pi) - \left(\sum_{i=1}^{m} (\lambda_i a_i + \alpha_i \eta^i) + \sum_{j=1}^{n} (\mu_j b_j + \beta_j \theta^j) + z \right)^\top \text{vec}(\pi)}_{A_2} \\ &\quad + \lambda^\top \alpha + \mu^\top \beta. \end{aligned}$$

To simplify notation, we denote

$$p := \sum_{i=1}^{m} (\lambda_i a_i + \alpha_i \eta^i) + \sum_{j=1}^{n} (\mu_j b_j + \beta_j \theta^j).$$

Observe that

$$\min_{P \in \mathbb{R}^{mn \times mn}} A_1 = \begin{cases} 0 & , \quad \text{if } Y = H(\eta, \theta, Z) \\ -\infty & , \quad \text{otherwise} \end{cases},$$

and

$$\max_{\pi \in \mathbb{R}^{m \times n}} A_2 = \begin{cases} -\frac{1}{4}(p + z)^\top Y^\dagger (p + z) & , \quad \text{if } Y \succeq 0 \text{ and } (I - YY^\dagger)(p + z) = 0 \\ -\infty & , \quad \text{otherwise} \end{cases}$$

Hence, the dual of (GW++) is given by

$$\begin{aligned} \max_{\lambda, \mu, y, z, Z} \quad & \lambda^\top \alpha + \mu^\top \beta - \frac{1}{4}(p + z)^\top Y^\dagger (p + z) \\ \text{s.t.} \quad & Y \succeq 0 \\ & (I - YY^\dagger)(p + z) = 0 \\ & Y = H(\eta, \theta, Z) \\ & Z \geq 0 \\ & z \geq 0 \end{aligned} \quad .$$

https://doi.org/10.52202/079017-2231

We re-write this as

$$\begin{aligned}\max_{\lambda,\mu,y,z,Z,t} \quad & \lambda^\top \alpha + \mu^\top \beta - t \\ \text{s.t.} \quad & \frac{1}{4}(p+z)^\top Y^\dagger (p+z) \le t \\ & Y \succeq 0 \\ & (I - YY^\dagger)(p+z) = 0 \\ & Y = H(\eta, \theta, Z) \\ & Z \ge 0 \\ & z \ge 0\end{aligned}$$

By taking Schur complements and by replacing Y with $H(\eta, \theta, Z)$, the above optimization instance reduces to

$$\begin{aligned}\max_{\lambda,\mu,z,Z,t} \quad & \lambda^\top \alpha + \mu^\top \beta - t \\ \text{s.t.} \quad & \begin{pmatrix} H(\eta,\theta,Z) & -\frac{1}{2}(p+z) \\ -\frac{1}{2}(p+z)^\top & t \end{pmatrix} \succeq 0 \\ & Z \ge 0 \\ & z \ge 0\end{aligned}$$

Note that $g(\lambda, \mu, \eta, \theta, y) = p + 2y$, the theorem then follows by doing a change of variable $y = -\frac{1}{2}(p+z)$. □

D Extended Applications of GW-SDP

D.1 GW-SDP Barycenters

One popular application of optimal transport is to compute the barycenters of measures that serves as a building block for many learning methods. The notion of barycenter for measures was first proposed in Agueh and Carlier (2011) for Wasserstein space. Akin to barycenter in Euclidean space (Fréchet), the Wasserstein barycenter is defined as the solution of a weighted sum of OT distances over the space of measures. An efficient algorithm to compute the discrete OT barycenter with entropic regularization was proposed in Benamou et al. (2015), and was later extended to discrete metric-measure spaces with entropic GW distance in Peyré et al. (2016).

We show that it is straightforward to extend the GW-SDP formulation to find barycenters of a set of data as Fréchet means. For simplicity, we assume that the base histogram $\bar{\alpha}$, the size of the barycenters $m \in \mathbb{N}$, and $(\lambda_k)_k$ such that $\sum_k \lambda_k = 1$ are fixed. We aim to find a structure matrix $\bar{C}$ that minimizes

$$\min \sum_k \lambda_k \text{GW-SDP}(C_k, \bar{C}, \alpha_k, \bar{\alpha}). \tag{7}$$

We have the following corollary.

Corollary D.1 (Adaptation of Proposition 3 in Peyré et al. (2016))**.** *In the special case of the squared loss $\ell(a,b) = (a-b)^2$, the solution of (7) reads*

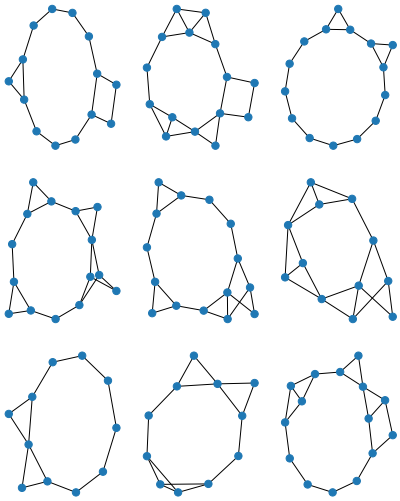
$$\bar{C} = \frac{\sum_k \lambda_k \pi_{sdp,k}^\top C_k \pi_{sdp,k}}{\alpha \alpha^\top}, \tag{8}$$

where $\pi_{sdp,k}$ is the solution to GW-SDP$(C_k, \bar{C}, \alpha_k, \bar{\alpha})$ and the division is entry-wise.

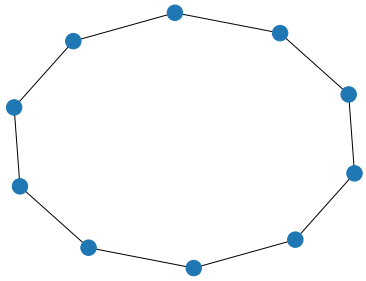
Corollary D.1 shows that we may apply iterative updates to solve for the barycenter $\bar{C}$ via the Block Coordinate Descent (BCD) algorithm. At each iteration, we solve K independent instances of the GW-SDP problem to find $(\pi_{sdp,k})_k$, and then compute $\bar{C}$ using (8) to solve for (7). A pseudocode for the GW-SDP barycenter calculation is provided in Algorithm 1. We demonstrate the effectiveness of the GW-SDP barycenter calculation by applying it to find the barycenter of a graph dataset. The dataset consists of 20 noisy graphs, created by adding random connections from a circular graph. We show a visualization of 9 of these in Figure 7a. The number of nodes ranges from 8-16. We apply the (GW-SDP) barycenters update for 100 iterations, and Figure 7b shows the result for a circular graph of 10 nodes.

Algorithm 1 Computation of GW-SDP barycenters.

Input: dataset $\{C_k, \alpha_k\}_{k=1}^K$; $\{\lambda_k\}_{k=1}^K$.
Initialize $\bar{C}$.
repeat
  for $k = 1$ **to** K **do**
    $\pi_{sdp,k} \leftarrow$ `solve_GW-SDP`$(C_k, \bar{C}, \alpha_k, \bar{\alpha})$.
  end for
  Update $\bar{C}$ using (8).
until convergence



(a) Visualization of noisy circular graphs.



(b) Graph barycenter learned by (GW-SDP).

Figure 7: Application of the (GW-SDP) to find graph barycenter of noisy circular graphs.

D.2 Outlier-Robust GW-SDP

It is generally possible to extend the semidefinite relaxation to variants of the GW problem. We briefly describe the semidefinite relaxation to the outlier-robust GW problem by Kong et al. (2024). Here, (X, d_X) and (Y, d_Y) are two metric spaces with accompanying measures μ and ν. The distance between μ and ν is

$$\begin{aligned} \min \ & \langle L, P \rangle + \tau_1 d_{KL}(\pi 1, \alpha) + \tau_2 d_{KL}(\pi^T 1, \beta) \\ \text{s.t.} & \begin{pmatrix} P & \text{vec}(\pi)^T \\ \text{vec}(\pi) & 1 \end{pmatrix} \succeq 0 \\ & \textstyle\sum_i P_{(i,j),(k,l)} = f_j^{k,l}, \Sigma_j P_{(i,j),(k,l)} = g_i^{k,l} \\ & P \geq 0 \\ & d_{KL}(\mu, \alpha) \leq \rho_1, d_{KL}(\nu, \beta) \leq \rho_2. \end{aligned}$$

There is one technical aspect: In the marginal sums $\sum_i P_{(i,j),(k,l)}$ we set this equal to some constant $f_j^{k,l}$. In GW-SDP, the corresponding RHS term depends on π and α. In the robust set-up, α is an optimization variable, not a constant, which necessitates the above change. We remark that the resulting formulation is convex but not an SDP because of the presence of the KL divergence.

https://doi.org/10.52202/079017-2231

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract highlights the proposed approach and advantages of solving the GW distance problem via a SDP relaxation and talks about the use of numerical algorithms to solve the problem.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The main limitation of the proposed GW-SDP formulation is the runtime. In the conclusion, we discuss possible future directions towards mitigating these issues.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

Answer: [Yes]

Justification: The feasibility and basis of the formulation of the SDP relaxation is shown. The validity of the numerical algorithms is also shown.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

Answer: [Yes]

Justification: The formulation of GW-SDP is compatible with existing convex optimization solvers. The equations and steps taken for the numerical algorithms are provided.

Guidelines:

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

https://doi.org/10.52202/079017-2231

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

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

   Answer: [Yes]

   Justification: The code is made publicly available at https://github.com/tbng/gwsdp.

   Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [NA]

   Justification: The main contribution is algorithmic. There is no testing or training aspect to this problem. The implementation of the algorithm in Section **??** does involve certain parameter tuning. We do not discuss these parameters but these will be specified in code that will be made public.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [NA]

   Justification: The main contribution of this paper is algorithmic. There are no numerical experiments of a statistical nature in this paper. All numerical experiments concern algorithmic performance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

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

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: The details of the computer resources used for the experiments are given in section 6.1.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: The research does not involve human subjects or private data. The libraries used for comparison against existing methodologies are open source.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

https://doi.org/10.52202/079017-2231

Justification: The paper's focus is on improvements towards solving the GW distance problem and is foundational in nature. The societal impact is limited insofar as it improves existing ML techniques that are based on Optimal Transportation techniques.

Guidelines:

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

11. **Safeguards**

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

Answer: [NA]

Justification: There is no risk of misuse for the algorithms in the research. There is no release of data or models.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The open source Python Optimal Transport package, released under the MIT license, is only used for algorithmic comparisons. Other datasets that using in the numerical experiment sections are also cited.

Guidelines:

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

13. **New Assets**

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

    Answer: [NA]

    Justification: The paper does not released new assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

https://doi.org/10.52202/079017-2231

Guidelines:

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