Semidefinite Relaxations of the Gromov-Wasserstein Distance

Abstract

The Gromov-Wasserstein distance (GW) 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 quadractically constrained quadratic program, which is not known to be tractable to solve. In particular, existing solvers are only able to find local optimizers. In this work, we propose a semi-definite programming (SDP) relaxation of the GW distance. Our approach provides the ability to compute the optimality gap of any transport map from the global optimal solution. Our initial numerical experiments suggest that our proposed relaxation is strong in that it frequently computes the global optimal solution, together with a proof of global optimality.

1 Introduction

From classical formulation of optimal transport... The optimal transport (OT) problem concerns the task finding a transportation plan between two probability distributions so as to minimize cost. 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 [1, 3, 13, 12], natural language processing [10], domain adaptation [4].

Let $\alpha \in \Sigma_m$ and $\beta \in \Sigma_m$ be 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 point $x_i \in \alpha$ and $y_j \in \beta$. The (Kantorovich) formulation of the OT problem [16] is defined as the solution of the following convex optimization instance

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

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.

...to Optimal Transport between Incomparable Spaces. 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, which is typical in shape comparion or graph matching, for example.

To address such scenarios, the work in [14] formulates an extension of the OT problem known as the Gromov-Wasserstein (GW) distance 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. The Gromov-Wasserstein distance between two discrete metric-measure spaces (C,α) and (D,β) is defined by

$$\mathrm{GW}(C,D,\alpha,\beta) \stackrel{\text{def.}}{=} \min_{\pi\in\Pi(\alpha,\beta)} (|C_{i,k} - D_{j,l})|)_{i,j,k,l}\pi_{i,j}\pi_{k,l} = \min_{\pi\in\Pi(\alpha,\beta)} \langle \mathbf{L}(C,D)\otimes\pi, \pi\rangle. \tag{2}$$

Preprint. Under review.

Here, the transportation cost is specified by the four-way tensor

$$\mathbf{L}(C, D) \stackrel{\text{def.}}{=} (|C_{i,k} - D_{j,l})|)_{i,j,k,l}, \tag{3}$$

in which the tensor-matrix multiplication is defined by

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

The GW distance has been applied widely to machine learning tasks, most notably on graph learning [20, 22, 23, 25]. The GW-problem is an instance of a quadratically constrained quadratic program (QCQP): to see this, one can re-write the objective in (2) in terms of vectorized matrices

$$\min_{\pi} \quad \langle \text{vec}(\pi), L \, \text{vec}(\pi) \rangle \quad \text{s.t.} \quad \pi \in \Pi(\alpha, \beta). \tag{4}$$

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

The constraint $\pi \in \Pi(\alpha, \beta)$ is convex, and in fact linear. On the other hand, the matrix L is usually not positive semidefinite – and this is typically the case for L's arising from differences of cost matrices (3). As such, the QCQP instance in (4) is typically non-convex.

Existing approaches for solving (2) include alternating minimization techniques, but these are not known to produce globally optimal solutions. An alternative approach is to impose entropic regularization [17]. This leads to a formulation that permits Sinkhorn scaling-like updates, but does not address the inherent non-convexity within the GW-distance. (Entropic regularization also introduces bias in solutions.) Other approaches include low-rank approximation of the cost matrices [18], adapting ideas from the Sliced-Wasserstein problem [21], an unbalanced-analog of the GW-problem [19], and using mini-batch samples [8].

2 Main Results

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. Let (π_{sdp}, P_{sdp}) denote an optimal solution to the following

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

Here, e_i denotes the standard basis vector whose i-th entry is 1.

Optimality gaps. The most valuable aspect of (GW-SDP) (as well as any other suitably defined convex relaxation) is that it provides a principled way to certify global optimality of any computed transporation map. We explain how this is done: Let $\pi^\star$ denote the optimal solution to the GW problem. Given any transportation map $\pi \in \Pi(\alpha, \beta)$, a natural to quantify the quality of π is to compare its objective value with the optimal choice:

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

This ratio is at least one, and is equal to one if π is also globally optimal.

Note that the optimal value of (GW-SDP) will always be a lower bound to (2):

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

This is because the tuple $(\pi^\star, \mathrm{vec}(\pi^\star)\mathrm{vec}(\pi^\star)^\top)$ is a feasible solution to (GW-SDP):

$$\mathrm{vec}(\pi^\star)\mathrm{vec}(\pi^\star)^\top \mathrm{vec}(e_i \mathbb{1}_m^\top) = \mathrm{vec}(\pi^\star)\langle \pi^\star, e_i \mathbb{1}_m^\top \rangle = \mathrm{vec}(\pi^\star)\langle \pi^\star \mathbb{1}_n, e_i \rangle = \alpha_i \mathrm{vec}(\pi^\star).$$

(The inequalities for β follow analogously.) Subsequently, one has

$$\text{Opt. Gap}(\pi_{sdp}) \leq \frac{\langle \pi_{sdp}, \mathbf{L} \otimes \pi_{sdp} \rangle}{\langle P_{sdp}, \mathbf{L} \rangle}.$$

This bound is useful because all the quantites on the RHS can be computed efficiently as the solution of a SDP. If in fact we have an instance where the RHS evaluates to one, then we have a *proof* that π_{sdp} is the *global optimal* solution to the GW problem. In our numerical experiments in Section 3, we observed that this is frequently the case for our experiments.

3 Experiments

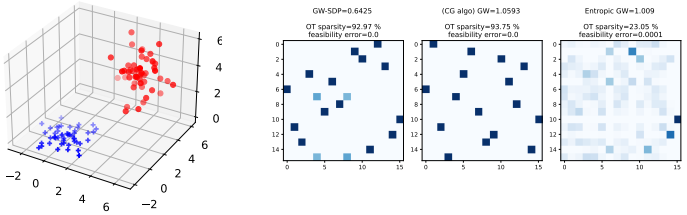
We demonstrate the effectiveness of GW-SDP over two tasks: transporting Gaussian distributions belonging to different spaces, and graph learning. Without further mention, we will use the 2-Gromov-Wasserstein distance; *i.e.* the cost function is squared Euclidean norm.

3.1 Synthetic Dataset: Matching Gaussian Distributions

We aim to find estimated 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. Clearly, using classical optimal transport formulation such as the likes of Wasserstein-2 distance is not viable. We compute our GW-SDP plans and distance, and compare them with the local solvers in Python Optimal Transport (PythonOT [5]). As seen in a qualitative demonstration of Figure 1a, our algorithm return optimal transport plans that is as sparse as the Conditional Gradient descent solver of Python OT for GW distance (CG-GW). Moreover, GW-SDP distance is smaller than the CG-GW value. 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, GW-SDP algorithm consistently returns smaller objective value (orange line) than GW-CG counterpart from PythonOT (blue line) and its entropic regularization (green line). This shows that the two solvers from PythonOT is more prone to stuck in local minima than our GW-SDP solver. In Figure 2b, we plot the value of the estimation gap with different number of sample points. We notice in this scenario of Gaussian matching, the gap is almost 1.0 in most case, which shows that GW-SDP is nearly optimal; *i.e.* $\langle \pi_{sdp}, \mathbf{L} \otimes \pi_{sdp} \rangle \approx \langle P_{sdp}, L \rangle \approx \langle \pi^\star, \mathbf{L} \otimes \pi^\star \rangle$.

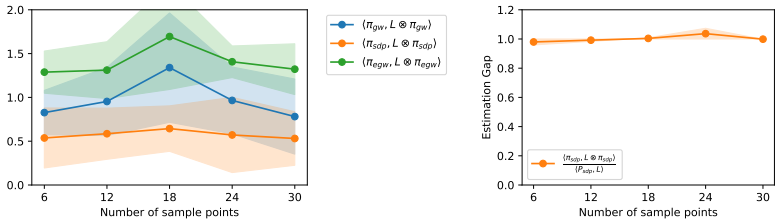
3.2 Synthetic Dataset: Graph Community Matching

The objective of this task is to find a matching between two graphs that follow stochastic block model (SBM, [7, 24]) with fixed inter/intra-clusters probability (the chance that the node inside/outside a clusters connect to each other, respectively). The source is a three-clusters SBM that has the intra-cluster probability $p = \{1.0, 0.95, 0.9\}$, and the target is a two-clusters SBM with intra probability $p = \{1.0, 0.9\}$. The inter-clusters probability are all set to 0.1. The distance matrices on each graph is created by first simulate the node features following Gaussian distributions with uniform weights. We then calculate the l_2 norm between nodes, and shrink the value of disconnected nodes to zero to form the distance matrices. Similar to the previous section, we observed GW-SDP algorithm return a transport plan that give a smaller total transportation cost $\langle \pi, \mathbf{L} \otimes \pi \rangle$ compared to GW-CG and eGW, which can be seen in Figure 3. Still, we notice some level of similarity between the GW-SDP and GW-CG transport plans, most notably both are reasonable sparse, while the eGW is dense which results in the highest transport cost.



(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.



(a) Comparison of objective values (b) Estimated optimality gap

Figure 2: **Left:** Value of the objective with **Right:** Value of the estimation gap. The numbers are calculated on 10 runs of different number of sample points on the Gaussian matching experiment.

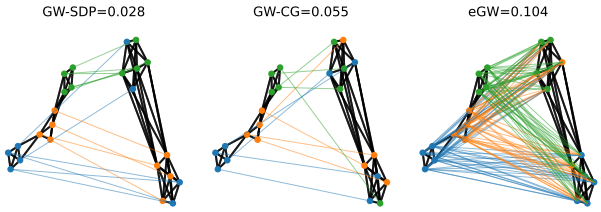


Figure 3: Value of the objective on the synthetic graph matching task, from the three-block SBM (left) to the two-block SBM (right).

4 Conclusion

In this paper, we propose a SDP-relaxation (GW-SDP) for the GW-problem. Our initial results suggest that the relaxation (GW-SDP) is strong in the sense that π_{sdp} frequently coincides with the global optimal solution – moreover, we are able to provide a proof when this actually happens. These results are exciting because it provides because it suggests a tractable approach for solving the GW-problem – at least for examples of interest – that is otherwise assumed to be difficult to compute in general.

It would be interesting to explain the strength of the relaxation in (GW-SDP) . Our initial experiments suggest that the most critical ingredient is the inclusion of the constraint

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

If these constraints are omitted, then the solutions tend to be substantially weaker. It would be interesting to better how the inclusion of these constraints tend to encourage good transportation maps as solutions.

References

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

[2] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220, 2006.

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

[4] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems*, 30, 2017.

[5] 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.

[6] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[7] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[8] Tanguy Kerdoncuff, Rémi Emonet, and Marc Sebban. Sampled gromov wasserstein. *Machine Learning*, 110(8):2151–2186, 2021.

[9] Itay Kezurer, Shahar Z Kovalsky, Ronen Basri, and Yaron Lipman. Tight relaxation of quadratic matching. In *Computer graphics forum*, volume 34, pages 115–128. Wiley Online Library, 2015.

[10] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France, 07–09 Jul 2015. PMLR.

[11] Javad Lavaei and Steven H Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power systems*, 27(1):92–107, 2011.

[12] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[13] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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

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

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

[17] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pages 2664–2672. PMLR, 2016.

[18] Meyer Scetbon, Gabriel Peyré, and Marco Cuturi. Linear-time gromov wasserstein distances using low rank couplings and costs. In *International Conference on Machine Learning*, pages 19347–19365. PMLR, 2022.

[19] Thibault Sejourne, Francois-Xavier Vialard, and Gabriel Peyré. The unbalanced gromov wasserstein distance: Conic formulation and relaxation. In *Advances in Neural Information Processing Systems*, volume 34, pages 8766–8779, 2021.

[20] Vayer Titouan, Nicolas Courty, Romain Tavenard, Chapel Laetitia, and Rémi Flamary. Optimal transport for structured data with application on graphs. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6275–6284. PMLR, 09–15 Jun 2019.

[21] Titouan Vayer, Rémi Flamary, Romain Tavenard, Laetitia Chapel, and Nicolas Courty. Sliced gromov-wasserstein. In *Advances in Neural Information Processing Systems*, volume 33, pages 14753–14763, 2019.

[22] Cédric Vincent-Cuaz, Rémi Flamary, Marco Corneli, Titouan Vayer, and Nicolas Courty. Template based graph neural network with optimal transport distances. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 11800–11814. Curran Associates, Inc., 2022.

[23] Cédric Vincent-Cuaz, Titouan Vayer, Rémi Flamary, Marco Corneli, and Nicolas Courty. Online graph dictionary learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10564–10574. PMLR, 18–24 Jul 2021.

[24] Yuchung J Wang and George Y Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[25] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-Wasserstein learning for graph matching and node embedding. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6932–6941. PMLR, 09–15 Jun 2019.

A SDP LIFTS

We motivate the ideas behind our SDP relaxation (GW-SDP). The starting point is to recognize that the Gromov-Wasserstein is an instance of a QCQPs – these are optimization instances of the form

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & x^\top A_0 x + 2b_0^\top x + c_0 \\ \text{s.t.} & x^\top A_i x + 2b_i^\top x + c_i \leq 0, i \in [m] \end{array} . \tag{5}$$

QCQPs are huge expressive powers, for instance, problems such as Sensors Network Localization in communications [2], Maximum Cut and Quadratic Assignment Problems in combinatorics [6, 9], and Optimal Power Flow in power system [11] can be expressed as instances of QCQPs. If the matrices A_i are psd, then the optimization instance (5) is convex, and can be solved tractably using standard software. The problem becomes difficult if the A_i's contain negative eigenvalues. In fact, the presence of a single negative eigenvalue is sufficient to make these problems NP-hard [15].

The first step of SDP relaxation is to observe that $\text{tr}(x^\top A_i x) = \text{tr}(A_i x x^\top)$ for all i. Hence, by introducing a new variable $X = xx^\top$, (5) is equivalent to

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & \text{tr}(A_0 X) + 2b_0^\top x + c_0 \\ \text{s.t.} & \text{tr}(A_i X) + 2b_i^\top x + c_i \leq 0, i = 1, 2, \ldots, m \\ & X = xx^\top \end{array}$$

Note that $X = xx^\top$ if and only if

$$\begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \succeq 0 \quad \text{and} \quad \operatorname{rank} \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} = 1. \tag{6}$$

The second step of standard SDP relaxation is to omit the rank constraint, resulting in the following convex optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \operatorname{tr}(A_0 X) + 2b_0^\top x + c_0 \\ \text{s.t.} \quad & \operatorname{tr}(A_i X) + 2b_i^\top x + c_i \leq 0, i = 1, 2, \ldots, m \\ & \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \succeq 0 \end{aligned} \tag{7}$$

A.1 Standard SDP Relaxation

$$\begin{aligned} \min_{\pi \in \mathbb{R}^{m \times n}, P \in \mathbb{R}^{mn \times mn}} \quad & \langle L, P \rangle \\ \text{s.t.} \quad & \begin{pmatrix} P & \operatorname{vec}(\pi) \\ \operatorname{vec}(\pi)^\top & 1 \end{pmatrix} \succeq 0 \\ & \pi \mathbb{1} = \alpha \\ & \pi^\top \mathbb{1} = \beta \\ & \pi \geq 0 \end{aligned} \tag{8}$$

Problem (8) is a tractable convex semi-definite programming, which can be can be efficiently solved in polynomial time, in contrast with the original form (**??**).

Remark A.1. Before studying the performance of SDP relaxation in the GW problem, we highlight several important observations:

1. Since the set $\{\operatorname{vec}(\pi)\operatorname{vec}(\pi)^\top : \pi \in \Pi(\alpha, \beta)\}$ is contained within the feasible region of P in optimization problem (8), the optimal value of (8) provides a lower bound on the optimal value of the original GW problem.

2. The optimal solution $\hat{\pi}$ to (8) inherently satisfies $\hat{\pi} \in \Pi(\alpha, \beta)$; *i.e.* , $\hat{\pi}$ is a feasible transportation map. Consequently, $\operatorname{vec}(\pi)^\top L \operatorname{vec}(\pi)$ yields an upper bound on the optimal value of the GW problem. Moreover, while the idea of relaxing quadratic terms of x to a PSD constraint is quite standard, additional rounding steps are often necessary to obtain a feasible solution [6, 9, 2]. Notably, for the GW problem, such a rounding step is unnecessary.

3. When the upper bound and the lower bound coincide, the relaxation succeeds, that is, the GW problem has been successfully solved to global optimality.

A.2 Tightening the Relaxation

Nevertheless, due to the non-compact nature of the feasible region of P in (8), the optimal value to it can be unbounded. We study this issue in Theorem A.2.

Theorem A.2. *Optimization problem* (8) *is unbounded.*

Proof. Assume $L_{st} > 0$ for some $s, t \in [mn]$. Let $v \in \mathbb{R}^{mn}$ be a vector with all zeros except for the s-th and the t-th entries, which are 1 and -1, respectively. Let $\tilde{\pi} \in \Pi(\alpha, \beta)$. Consider $P_c = \operatorname{vec}(\tilde{\pi})\operatorname{vec}(\tilde{\pi})^\top + cvv^\top$. It is easy to see that $P_c \succeq \operatorname{vec}(\tilde{\pi})\operatorname{vec}(\tilde{\pi})^\top$ for all $c \in \mathbb{R}_+$, and

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

since L is symmetric. Note that $L_{ii} = 0$ for all $i \in [mn]$. Hence, we have $\langle L, P_c \rangle = \langle L, \operatorname{vec}(\tilde{\pi})\operatorname{vec}(\tilde{\pi})^\top \rangle - 2cL_{st}$, which implies $\langle L, P_c \rangle \to -\infty$ as $c \to +\infty$. □

It follows from Theorem A.2 that solving the standard SDP relaxation of the GW problem (optimization instance (8)) leads to meaningless solutions. In fact, we can enhance the performance of the SDP relaxation by incorporating additional constraints that we may freely add to P. Essentially,

any constraints that are valid for $P = \text{vec}(\pi)\text{vec}(\pi)^\top$ with $\pi \in \Pi(\alpha, \beta)$ can be added. In this section, we impose extra linear equality constraints to the SDP relaxation presented in (8).

First, note that $\pi \geq 0$ for all $\pi \in \Pi(\alpha, \beta)$, which enables us to impose the following nonnegative constraint

$$P \geq 0 \tag{Nonneg}$$

Second, denote by $e_i \in \mathbb{R}^{mn}$ the standard unit vector with all zero except its i-th entry, which is one. Let $a_i = \text{vec}(e_i \mathbb{1}_n^\top)$, $b_j = \text{vec}(\mathbb{1}_m e_j^\top)$. Since we can write $\pi \mathbb{1} = \alpha$ and $\pi^\top \mathbb{1} = \beta$ as $a_i^\top \text{vec}(\pi) = \alpha_i, i \in [m]$ and $b_j^\top \text{vec}(\pi) = \beta_j, j \in [n]$, respectively, the following marginal constraints hold for $P = \text{vec}(\pi)\text{vec}(\pi)^\top$:

$$\begin{aligned} Pa_i &= \alpha_i \text{vec}(\pi), i \in [m], \\ Pb_j &= \beta_j \text{vec}(\pi), j \in [n]. \end{aligned} \tag{Margi}$$

By incorporating constraints (Nonneg) and (Margi), we introduce the following tighter convex SDP relaxation of the GW problem, which we refer to as GW-SDP:

$$\begin{aligned} \min_{\pi \in \mathbb{R}^{m \times n}, P \in \mathbb{R}^{mn \times mn}} \quad & \langle L, P \rangle \\ \text{s.t.} \quad & \begin{pmatrix} P & \text{vec}(\pi) \\ \text{vec}(\pi)^\top & 1 \end{pmatrix} \succeq 0 \\ & a_i^\top \text{vec}(\pi) = \alpha_i, i \in [m] \\ & b_j^\top \text{vec}(\pi) = \beta_j, j \in [n] \\ & \text{vec}(\pi) \geq 0 \\ & Pa_i = \alpha_i \text{vec}(\pi), i \in [m] \\ & Pb_j = \beta_j \text{vec}(\pi), j \in [n] \\ & P \geq 0 \end{aligned} \tag{GW-SDP}$$

Note that Remark A.1 also holds for GW-SDP. Intuitively, the nonnegtive constraint (Nonneg) makes the feasible set of (8) compact, thereby guaranteeing a bounded optimal value when solving GW-SDP. And the marginal constraints (Margi) ensure that the optimal solution to GW-SDP has certain desired structure. Moreover, as illustrated in Proposition A.3, the added constraints (Nonneg) and (Margi) do not overlap with the constraints specified in (8). Therefore, adding these constraints are guaranteed to tighten the standard SDP relaxation introduced in Section A.1.

Proposition A.3. *The feasible set of GW-SDP is strictly contained by the feasible set of* (8).

Proof. We start by showing (Nonneg) can not be implied by the feasible set of (8). Let $v \in \mathbb{R}^{mn}$ be a vector with all zeros except for the first two entries, which are 1 and -1, respectively. For any $\pi \in \Pi(\alpha, \beta)$, consider $P = \text{vec}(\pi)\text{vec}(\pi)^\top + (\pi_{11}\pi_{21} + 1)vv^\top$. In this case, since $\pi_{11}, \pi_{21} \geq 0$, we have $P \succeq \text{vec}(\pi)\text{vec}(\pi)^\top$, and hence P is in the feasible set of (8). However, $P_{12} = \pi_{11}\pi_{21} - (\pi_{11}\pi_{21} + 1) = -1 < 0$ implies P doesn't satisfy (Nonneg).

Then we show the feasible set of (8) doesn't imply (Margi). For any $\pi \in \Pi(\alpha, \beta)$, consider $P = \text{vec}(\pi)\text{vec}(\pi)^\top + \mathbb{1}\mathbb{1}^\top$. It is easy to see that $P \succeq \text{vec}(\pi)\text{vec}(\pi)^\top$. However, for any $i \in [m]$ we have

$$Pa_i = \alpha_i \text{vec}(\pi) + n\mathbb{1} \neq \alpha_i \text{vec}(\pi)$$

which implies that P doesn't satisfy (Margi). □

While Proposition A.3 doesn't comprehensively encapsulate this phenomenon, it is noteworthy that in practical scenarios, the GW-SDP typically yields optimal or nearly optimal values, which can be verified by computing the optimality gap.