Generalized Spectral Clustering via Gromov-Wasserstein Learning

Abstract

We establish a bridge between spectral clustering and Gromov-Wasserstein Learning (GWL), a recent optimal transport-based approach to graph partitioning. This connection both explains and improves upon the state-of-the-art performance of GWL. The Gromov-Wasserstein framework provides probabilistic correspondences between nodes of source and target graphs via a quadratic programming relaxation of the node matching problem. Our results utilize and connect the observations that the GW geometric structure remains valid for any rank-2 tensor, in particular the adjacency, distance, and various kernel matrices on graphs, and that the heat kernel outperforms the adjacency matrix in producing stable and informative node correspondences. Using the heat kernel in the GWL framework provides new multiscale graph comparisons without compromising theoretical guarantees, while immediately yielding improved empirical results. A key insight of the GWL framework toward graph partitioning was to compute GW correspondences from a source graph to a template graph with isolated, self-connected nodes. We show that when comparing against a two-node template graph using the heat kernel at the infinite time limit, the resulting partition agrees with the partition produced by the Fiedler vector. This in turn yields a new insight into the $k$-cut graph partitioning problem through the lens of optimal transport. Our experiments on a range of real-world networks achieve comparable results to, and in many cases outperform, the state-of-the-art achieved by GWL.

1 Introduction

The Gromov-Wasserstein (GW) problem is a nonconvex quadratic program \cite{SPKS16} whose solution yields the GW distance, a generalization of Wasserstein distance from classical optimal transport which is able to compare distributions defined on different metric spaces. This is accomplished by replacing classical Wasserstein loss with a loss function defined in terms of relational information coming from metric data. Because it is able to compare distributions defined on a priori incomparable spaces, GW distance is increasingly finding applications for learning problems on irregular domains such as graphs \cite{Hen16, TMK+18, VCTF19, XLZD19, Xu20}. In this context, a graph can be considered as a metric space by endowing it with geodesic distance. A soft matching between nodes of two different graphs is obtained by choosing distributions on each graph’s nodes (e.g., uniform distributions) and computing the GW optimal transport plan between them.

Applications of GW distance have been bolstered by the observation that the GW problem does not fundamentally require a metric to operate; that is, the definition of the GW loss function extends to other forms of relational data. Xu, Luo, and Carin used this observation to produce the state-of-the-art Scalable Gromov-Wasserstein Learning (S-GWL) framework for graph matching and partitioning.
The Wasserstein distance, introduced by Mémoli in the context of Riemannian manifolds [Mém11b], is a metric that captures information about a graph. An example is the geodesic distance function derived from the adjacency function. Let $G = (V, E)$ be a finite, unweighted, possibly directed graph. We refer to $V$ as the set of nodes and $E$ as the set of edges. Let $A : V \times V \to \{0, 1\}$ and $D : V \to \mathbb{Z}$ denote the adjacency and degree functions defined as $A(v, w) := 1$ if $(v, w) \in E$, 0 otherwise, and $D(v) := \{|(w : (v, w) \in E)|\}$. Given an ordering on $V$, these functions can be represented as matrices in $\mathbb{R}^{V \times V}$, and we will switch between both the function and matrix forms. In addition to the adjacency function, which fully encodes a graph, there are numerous derived representations which capture information about a graph. An example is the geodesic distance function $d$ which contains all the shortest path lengths in the graph, and is derived from the adjacency function.

In this paper we study the GW graph optimal transport problem by representing graphs via heat kernels rather than adjacency matrices. This amounts to finding soft correspondences between the nodes of two graphs by comparing spectral, rather than adjacency, information. We refer to this as the SpecGWL framework, reserving S-GWL to refer to the adjacency-based GW framework.

Numerical experiments demonstrate that SpecGWL outperforms S-GWL in graph partitioning tasks. Moreover, a main goal of this paper is to introduce tools for studying the GW problem in a more rigorous manner. We use a Markov Chain sampling technique to explore the energy landscape of the GW loss function for adjacency and heat kernel graph representations, showing empirically that SpecGWL loss has fewer spurious local minima and a 10x acceleration in convergence of gradient descent over S-GWL loss. We also introduce a visualization technique which allows one to ascertain the quality of soft node matchings obtained by any GW method. Examples of this technique intuitively demonstrate the idea that heat kernel-based matchings more faithfully preserve global graph structure than adjacency-based matchings. Finally, we establish theoretical results on the sparsity of optimal couplings in the SpecGWL framework and on the precise relationship of the SpecGWL graph partitioning algorithm to classical spectral clustering. The latter result creates a novel connection between spectral clustering and optimal transport.

Related literature. Gromov-Wasserstein distance was originally introduced as a theoretical tool used in metric geometry to study the convergence of sequences of metric measure spaces [Stu06, Mém11a, Stu12]. The idea of using heat kernels for GW matching goes back to the Spectral Gromov-Wasserstein distance introduced by Mémoli in the context of Riemannian manifolds [Mém11b]. The Riemannian heat kernel has been celebrated for its multiscale and informative properties [SOG09] — the former refers to the observation that the heat kernel defines a family of Gaussian filters that get progressively shorter and wider as $t \to \infty$, and the latter refers to the classical lemma of Varadhan showing that at the small time limit, the log-heat kernel approximates the geodesic distance on a Riemannian manifold. A surprising result in this direction is that under mild conditions, the collection of traces of the heat kernel forms an isometry invariant of a manifold [SOG09] despite giving up most of the information contained in the heat kernel. Heat kernel traces have recently been used for graph comparison in [TMK18], where it was shown that the desirable properties of the heat kernel for Riemannian manifolds have natural and informative analogues in the setting of graphs. There is a deep literature on spectral graph comparison, and a few other reference include [PK12, BG13, HRG14, NVM+18, DS20].

GW distance has surged in popularity in recent years, starting from its early applications in computer vision [Mém07, Mém11a, SS13] to alignment of word embedding spaces [AM18], learning of generative models across different domains [BAMKJ19], graph factorization [XLZD19], and learning autoencoders [XLH+20]. Other related topics include fused GW [VCF+18], sliced GW [VFC+19] and the Gromov-Monge problem [MN18]. Theoretical study of the GW distance and its geometry, including the structure of geodesics and gradient flows, was performed in [Stu12]. Recently these techniques have been utilized to create a Riemannian framework for performing averaging and tangent PCA across different graph-structured domains [CN19].

## 2 Spectral Gromov-Wasserstein Distances

### Gromov-Wasserstein Distance.

Let $G = (V, E)$ be a finite, unweighted, possibly directed graph. We refer to $V$ as the set of nodes and $E$ as the set of edges. Let $A : V \times V \to \{0, 1\}$ and $D : V \to \mathbb{Z}$ denote the adjacency and degree functions defined as $A(v, w) := 1$ if $(v, w) \in E$, 0 otherwise, and $D(v) := \{|(w : (v, w) \in E)|\}$. Given an ordering on $V$, these functions can be represented as matrices in $\mathbb{R}^{V \times V}$, and we will switch between both the function and matrix forms. In addition to the adjacency function, which fully encodes a graph, there are numerous derived representations which capture information about a graph. An example is the geodesic distance function $d$ which contains all the shortest path lengths in the graph, and is derived from the adjacency function.
The graph geodesic distance representation has previously been used [Hen16] to solve graph matching problems via the Gromov-Wasserstein (GW) distance [Mem11a]. This is a distance between metric measure (mm) spaces. Throughout this work, we will only deal with finite mm spaces and write them in matrix-vector notation. A finite mm space \((X, d, p)\) consists of a finite set of points \(X\), a metric function \(d\) written as a matrix \(d = (d_{ij}) \in \mathbb{R}^{|X| \times |X|}\), and a probability distribution \(p\) on \(X\) written as a vector \(p = (p_i) \in [0, 1]^{|X|}\). As a generalized form of optimal transport, GW distance relies on the notion of a coupling of \((X, d, p)\) with \((Y, d, q)\) on a set \(X\) and \(Y\). A coupling of \(p\) and \(q\) is a joint probability measure \(C\) on \(X \times Y\) with marginals \(p\) and \(q\); i.e., \(C = (C_{ij}) \in \mathbb{R}^{\{X\} \times |Y|}\) satisfies equality constraints \(C_{ij}p_i = q_j\) and \(C_{ij}1^{n \times 1} = q (1^m \times 1\) denoting the matrix of all ones), and entrywise inequality constraints \(0 \leq C \leq 1\). The set of all couplings of \(p\) and \(q\) is denoted \(\text{C}(p, q)\)—this is a convex polytope in \(\mathbb{R}^{\{X\} \times |Y|}\). Given mm spaces \((X, d^X, p), (Y, d^Y, q)\), the Gromov-Wasserstein distance is defined as:

\[
d_{GW}(X, Y) := \frac{1}{2} \min_{C \in \text{C}(p, q)} \left( \sum_{i,k} \sum_{j,l} (d_{ik}^X - d_{jl}^Y)^2 C_{ij} C_{kl} \right)^{1/2}.\]

The GW problem is a nonconvex quadratic program over a convex domain for which approximate solutions may be obtained via projected gradient descent [PCS16] or Sinkhorn iterations with an entropy [SPKS16] or KL divergence regularizer [XLC19]. Computational implementations can be found in the Python Optimal Transport library [FC17].

**Gromov-Wasserstein Learning.** Peyré, Cuturi, and Solomon observed in [PCST16] that solving the GW problem with input matrices that are not strictly distances—in particular kernel matrices—still leads to a discrepancy that can be informative when comparing matrices of different sizes and arising in incompatible domains. This idea was pushed further by the theoretical work in [CM19], which shows that any square matrix representation of a graph, including the adjacency matrix, can be used in the GW problem to obtain a bona fide distance (actually a pseudometric). This observation was used heavily in [XLC19], where the authors create a unified Gromov-Wasserstein Learning (GWL) framework for unsupervised learning tasks on graphs such as finding node correspondences between unlabeled graphs and graph partitioning, where it achieves state-of-the-art performance. One of the key ideas in the GWL framework is to use of adjacency matrices in the GW problem. We now formulate the GWL framework more precisely.

Let \(G\) and \(H\) be graphs with distributions \(p\) and \(q\) on their nodes and let \(A^G\) and \(A^H\) denote their adjacency matrices. The GWL framework considers the loss

\[
C(p, q) \ni \text{Adj}(C) \mapsto \text{Adj}_G,C,p,H,q(C) := \sum_{i,k} \sum_{j,l} (A^G_{ik} - A^H_{jl})^2 C_{ij} C_{kl},
\]

which we refer to as adjacency loss. The minimum of \(\text{Adj}(C)^{1/2}\) defines a distance between the pairs \((G, p)\) and \((H, q)\)—we refer to such pairs as measure graphs and assume for convenience that distributions are fully supported. If \(p\) and \(q\) are themselves derived from adjacency data then the minimizer \(C\) of (1) provides a natural soft correspondence between the nodes of \(G\) and \(H\). For example, [XLC19] considers the following family of node distributions:

\[
p = (p_1, \ldots, p_n)^T, \quad p_j = \frac{p_j}{\sum_{k=1}^n p_k}, \quad \overline{p}_j = (\deg(v_j) + a)^b,
\]

where \(p_j = p(v_j)\) for \(v_j\) a vertex of \(G\), \(a \geq 0\) is used to enforce the full support condition and the exponent \(b \in [0, 1]\) allows interpolation between the uniform distribution and the degree distribution.

**Heat Kernels.** Given an undirected graph \(G = (V, E)\), let \(L^2(V)\) denote the linear space of functions \(f : V \to \mathbb{R}\). The Laplacian of \(G\) is the operator \(L : L^2(V) \to L^2(V)\) defined by

\[
L(\phi)(v) := D(v)\phi(v) - \sum_{(v,w) \in E} \phi(w).
\]

After fixing an ordering on \(V\), we can use matrix-vector notation to write \(\mathbb{R}^{|V| \times |V|} \ni L = D - A\) and \(\phi \in \mathbb{R}^{|V| \times 1}\). The Laplacian is symmetric positive semidefinite, so the spectral theorem guarantees an eigendecomposition \(L = \Phi \Lambda \Phi^T\) with real, nonnegative eigenvalues \(\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n\), arranged
on the diagonal of $\Lambda$. The corresponding eigenvectors $\phi_1, \phi_2, \ldots, \phi_n$ are arranged as the columns of $\Phi$. There are several variants of the Laplacian with analogous properties, including the normalized Laplacian given by $I - D^{-1/2}AD^{-1/2}$. This is the version that we will typically use for experiments.

For a (strongly connected) directed graph $G = (V,E)$, we use the normalized Laplacian defined by Chung [Chu05] via the language of random walks. Consider the transition probability matrix $P$ defined by writing $P_{ij} = 1/D_i$ if $(i,j) \in E$, 0 otherwise. By Perron-Frobenius theory, there is a unique left eigenvector $\psi$ with all entries positive such that $\psi^T P = \psi^T$. The directed graph Laplacian is defined as $L := I - (\psi^{1/2} P \psi^{-1/2} + \psi^{-1/2} P^T \psi^{1/2})/2$, where $\Psi := \text{diag}(\psi)$.

In what follows, we use $L$ generically to refer to any of the Laplacians defined above. The heat equation on a graph $G$ (either undirected or directed) is then given as $du/dt = -Lu$, where $u \in L^2(V \times \mathbb{R}_{\geq 0})$. If $u$ represents a time-dependent heat distribution on the nodes of $G$, the heat equation describes heat diffusion according to Newton’s Law. The heat kernel is the fundamental solution to this heat equation, given in closed form as $K_t = \exp(-tL) = \Phi \exp(-t\Lambda)\Phi^T$.

**Spectral GW Distance.** Given measure graphs $(G,p)$ and $(H,q)$, we consider the spectral loss

$$C(p,q) \ni C \mapsto \text{Spec}^t(C) = \text{Spec}^t_{G,p,H,q}(C) := \sum_{i,k} \sum_{j,l} (K_{G,t}^{ik} - K_{H,t}^{kl})^2 C_{ij} C_{kl}$$

for each $t > 0$, where $K_{G,t}^{ik}$ and $K_{H,t}^{kl}$ are the heat kernels of $G$ and $H$, written in matrix form. We then obtain a one parameter family of pseudometrics

$$d^\text{spec}_{GW}[t]((G,p), (H,q)) := \min_{C \in \mathbb{C}(p,q)} \text{Spec}^t(C)^{1/2}$$

on the space of measure graphs. As in the GWL framework, choosing node distributions $p$ and $q$ from the family (2) yields minimizing couplings $C$ which give meaningful correspondences between the nodes of $G$ and $H$. As $t$ varies, one obtains multiscale couplings between nodes, with small $t$ encoding local and large $t$ encoding global structure (this is made precise in Section 3). One can further define an overall pseudometric as, say, $d^\text{spec}_{GW} := \sup_{t>0} d^\text{spec}_{GW}[t]$ [Mem11b], but our work will focus on the minimizing couplings at various $t$-values.

**Properties of SpecGWL.** For measure graphs $(G,p)$ and $(H,q)$, fix $t > 0$ and write $J := K_{G,t}^{ik}$, $K := K_{H,t}^{kl}$. After expanding the square and invoking the marginalization constraints (see [SPKS16] for an explicit derivation), one sees that minimizing (3) is equivalent to maximizing $(JC, CK)$ subject to $C \in \mathbb{C}(p,q)$, where $\langle \cdot,\cdot \rangle$ denotes the Frobenius inner product. Because the graph heat kernel is symmetric positive definite, we can take Cholesky decompositions $J = U^T U$, $K = V^T V$ to write

$$\langle JC, CK \rangle = \text{tr}(JC^T CK) = \text{tr}(C^T U^T UCV^T V) = \text{tr}(V C^T U^T UCV^T) = \|UCV^T\|^2.$$  

The map $C \mapsto \|UCV^T\|^2$ is convex. We record this as the following lemma:

**Lemma 1.** For each $t > 0$, spectral loss (3) is minimized over the convex polytope $\mathbb{C}(p,q)$ by a maximizer of the convex function $C \mapsto \langle JC, CK \rangle$.

Optimization problems of this type are not tractable to solve deterministically, but we demonstrate experimentally that approximation via gradient descent enjoys faster convergence and fewer spurious local minima than adjacency loss (1). It has been empirically observed that (local) minimizers of adjacency loss tend to become sparse [XLZD19], and this is utilized in [CNI19] to give a gradient descent-based algorithm for averaging networks. This empirical observation can be proved in the spectral setting (the proof is provided in Supplementary Materials).

**Theorem 2.** Let $(G,p)$ and $(H,q)$ be measure graphs of comparable sizes, say on $n$ nodes. Then for any $t > 0$, there is a minimizer of spectral loss with $o(n)$ nonzero entries.

**Complexity of SpecGWL.** Assuming graphs of comparable size $n$, the eigendecomposition incurs a time complexity of $O(n^3)$ and memory complexity of $\Theta(n^2)$. Computing the GW loss using gradient descent involves computing $\nabla \langle JC, CK \rangle = J^T CK + JCK^T$, which also incurs a time complexity of $O(n^3)$ and memory complexity of $O(n^2)$. Regularized methods with Sinkhorn iterations still require paying a cost for matrix multiplication [PCS16]. However, accelerations
have already been proposed: [TMK+18] suggests methods for approximating heat kernels in $O(n^2)$ operations and [XLC19] proposes a recursive divide-and-conquer approach to reduce the complexity of the GW comparison to $O(n^2 \log n)$. Note that because heat kernel matrices are dense, we cannot get the advantages of sparse matrix operations. We propose a multiscale approach to large-scale graph analysis using the compatible S-GWL and SpecGWL frameworks: scalable adjacency-based methods can be employed to break graphs into manageable chunks via the recursive partitioning method of [XLC19], at which point our spectral methods (which we show below are faster and better performing than adjacency-based methods on smaller networks) can be employed.

3 Graph Partitioning

Graph Partitioning Method. Graph partitioning is a crucial unsupervised learning task used for community detection in social and biological networks [GN02]. The goal is to partition the vertices of a graph into some number of clusters $m$ in accordance with the maximum modularity principle—edges within clusters are dense, while edges between clusters are sparse. In [XLC19], a GW-based approach to graph partitioning is proposed, where an $m$-way partition of a measured graph $(G, p)$ is obtained by minimizing the following variant of adjacency loss (1):

$$C(p, q) \equiv C \mapsto \sum_{i,k} \sum_{j,l} (A_{ik} - Q_{jl})^2 C_{ij} C_{kl}, \quad (5)$$

where $A$ is the adjacency matrix of $G$, $Q = \text{diag}(q)$ and $q$ is a distribution estimated by sorting the weights of $p$, sampling $m$ values via linear interpolation and renormalizing. Intuitively, $Q$ is the weighted adjacency matrix of a graph on $m$ nodes with only self-loops—an ideally clustered template graph. A minimizer $C$ of (5) defines an $m$-way partition of $G$: each node $v_i$ of $G$ is assigned a label in $\{1, \ldots, m\}$ according to the column index of the maximum weight in row $i$ of $C$.

Soft-matching nodes of the target graph to an ideally clustered template is intuitively appealing and it is shown in [XLDZ19] that this method achieves state-of-the-art performance. We propose a variant of the algorithm: letting $K^t$ denote the heat kernel for $G$ at some $t > 0$, we minimize

$$C(p, q) \equiv C \mapsto \sum_{i,k} \sum_{j,l} (K_{ik}^t - Q_{jl})^2 C_{ij} C_{kl}, \quad (6)$$

with $Q$ defined as above. For each $t > 0$, we obtain an optimal coupling which can be used to partition $G$ as described above. Experimental results in Section 4 show that this change to the algorithm gives a significant performance boost over the adjacency-based version.

Connection to Spectral Clustering. Let $G$ be an undirected, connected graph with graph Laplacian $L$. The connectivity of $G$ implies that $L$ has exactly one zero eigenvalue with constant eigenvector. Assume for simplicity that the multiplicity of the smallest positive eigenvalue of $L$ is one. A fundamental concept in spectral graph theory is that the corresponding eigenvector—the Fiedler vector of $G$—can be used to give a 2-way partitioning of $G$ with good theoretical properties: nodes of $G$ are partitioned according to the sign of their entry in the Fiedler vector [Pie73]. We refer to this as the Fiedler partitioning of $G$. The proof of the following is provided in the Supplementary Materials.

Theorem 3. Let $G$ be a connected graph whose first positive eigenvalue has multiplicity one, endowed with the uniform node probability distribution $p$. For sufficiently large $t$, the 2-way partition of $G$ derived from a minimizer of (4) agrees with the Fiedler partitioning.

The theorem demonstrates a novel connection between optimal transport and classical spectral clustering. On the other hand, the heat kernel of $G$ has Taylor expansion $K^t = I_n + tL + O(t^2)$, where $I_n$ is the $n \times n$ identity matrix. It follows that for low values of $t$, spectral GW partitioning is driven by matchings of graph Laplacians, which contain local adjacency information.

4 Experiments

We present several numerical experiments demonstrating the boost in performance obtained by using heat kernels in the GW problem rather than adjacency matrices. In experiments with undirected graphs, we used the normalized graph Laplacian to construct heat kernels—results were qualitatively
similar using heat kernels of the standard Laplacian, but we found some boost in quantitative performance in graph partitioning when using the normalized version. Experiments with directed graphs use Chung’s normalized Laplacian. Open source code and reproducible experiments are publicly available at: https://github.com/trneedham/Spectral-Gromov-Wasserstein.

**Table 1: Results of Energy Landscape Experiment.**

| Loss Function | Time/Iter. (s) | Worst Error (%) | Product Error (%) |
|---------------|---------------|-----------------|-------------------|
| Adj           | .0230         | 24.21           | 4.81              |
| Spec, $t = 5$ | .0014         | 22.35           | 6.00              |
| Spec, $t = 10$| .0013         | 3.86            | 1.34              |
| Spec, $t = 20$| .0010         | 0.05            | 0.02              |

Energy Landscapes and Convergence Rates. Adjacency loss (1) is highly nonconvex, while Lemma 1 shows that minimizing spectral loss (3) is equivalent to maximizing a convex function over a convex polytope. While the latter optimization problem is still intractable, we claim that its approximation via projected gradient descent is well-behaved. To test this claim, we perform the following experiment. In each trial, two random graphs from the IMDB-Binary [YV15] actor collaboration graph dataset (1000 graphs with 19.77 nodes and 96.53 edges on average) are selected. Probability distributions $p$ and $q$ for the nodes of the two graphs are selected; uniform distributions were used in the reported results, but qualitatively similar results were obtained when using other distributions from the family [2]. Next an ensemble of couplings between these measures is generated by running a custom Markov Chain Monte Carlo (MCMC) hit-and-run sampler [Smi84] on the coupling polytope $C(p, q)$ (details in the Supplementary Materials). We sampled 100 points in the polytope by running 100,000 MCMC steps and subsampling uniformly. Using each coupling in the ensemble as an initialization, we run projected gradient descent on adjacency loss (1) and spectral loss (3) with $t = 5, 10$ and $20$ and record the loss at the local minimum from each initialization. This process is repeated 100 times (100 choices of pairs of IMDB graphs).

Statistics for the experiment are reported in Table 1. For each method, we report the mean time for convergence of each gradient descent. For each trial, we obtain a distribution of losses from the 100 initializations in the ensemble, with the minimum loss treated as the putative global minimum. The “Worst Error” for each trial is $(\text{max loss} - \text{min loss})/\text{min loss}$. We report the mean Worst Error over 100 trials. In available packages, gradient descent for GW matching is by default initialized with the product coupling $C = pq^T$, so we also report the “Product Error” $(\text{product loss} - \text{min loss})/\text{min loss}$, averaged over all trials. We see that heat kernel representations provide an order of magnitude speed up of convergence, with decreasing error as the $t$ parameter increases; e.g., for $t = 20$, all initializations converge to a coupling with less than 0.1% error. Moreover, when using adjacency loss (1), approximately 10% of trials incurred > 20% Product Error (Figure 2 in Supplementary Materials), which is significant enough to lead to potential issues in applications. We also ran the experiment using the proximal gradient descent method introduced in [XLC19] to minimize Adj. For the size of the involved networks, it did not provide any speedup over standard gradient descent and produced worse error rates than the reported results.

Graph Matching and Averaging. Let $(G, p)$ and $(H, q)$ be measure graphs. Minimizers of the loss functions (1) and (3) are couplings $C$ which can be understood as soft node correspondences between $G$ and $H$. To assess the intuitive meaning of such a coupling, it is useful to visualize this node correspondence at the graph level. Using ideas developed in [CN19], we produce such visualizations, as shown in Figure 1. The figure shows six separate examples. For any particular example, we display an interpolation between graph $G$ (on the left) and graph $H$ (on the right). The optimal coupling $C$ is used to interpolate node positions from $G$ to $H$, with new edges phasing in during the interpolation—see Supplementary Materials for the details of this algorithm.

Three pairs of graphs are considered in Figure 1 with node distributions in each case taken to be uniform, for the sake of simplicity. For each of the three pairs of graphs, we produce an optimal coupling using “local” data (either a minimizer for adjacency loss (1) or for spectral loss (3) with low $t$-value) and using “global” data (spectral loss with high $t$-value). Visualizations of both couplings are shown. These visualizations make it intuitively clear that the couplings obtained from Spec...
Figure 1: Visualizations of GW graph matchings. **Top:** Matchings between binary trees using adjacency loss (1) (left) and heat kernel loss (3) with \( t = 20 \) (right). **Middle:** Matchings between circular graphs using HK loss, \( t = 2 \) (left) and HK loss, \( t = 10 \) (right). **Bottom:** Matchings between IMDB graphs using adjacency loss (left) and HK loss, \( t = 20 \) (right).

Table 2: Node Correctness Scores, Mean ± St. Dev. (Time).

| Loss   | Proteins | Enzymes | Reddit | Collab |
|--------|----------|---------|--------|--------|
| Adj    | .68 ± .22 (31.9) | .70 ± .18 (8.9) | .29 ± .21 (3941.7) | .50 ± .27 (4.3) |
| Spec\(^{10}\) | **.78 ± .22 (5.1)** | **.79 ± .17 (1.4)** | **.50 ± .11 (206.1)** | **.50 ± .27 (5.6)** |

with high \( t \)-value produce node correspondences which respect large scale graph structure much more faithfully than the couplings obtained Adj or Spec\(^{t}\) with low \( t \)-value. The midpoint of each interpolation can be understood as a structural average of graphs \( G \) and \( H \) — observe that the averages of the interpolations in the global data regime (right column) display a more natural blend of features from the endpoint graphs. Another experiment illustrating improved stability of the graph averaging algorithm of [PCS16] when using spectral loss is presented in the Supplementary Materials.

We also assess the quality of node correspondences quantitatively. In this experiment, we consider two biological graph databases Proteins [BOS\(^+\)05] (1113 graphs with 39.06 nodes and 72.82 edges on average) and Enzymes [DD03; SCE\(^+\)04] (600 graphs with 32.63 nodes and 62.14 edges on average), and two social graph databases Reddit (subset of 500 graphs with 375.9 nodes and 449.3 edges on average) and Collab (subset of 1000 graphs with 63.5 nodes and 855.6 edges on average), both from [YV15]. All processed datasets were downloaded from [KKM\(^+\)16]. For each graph \( G = (V,E) \), we assign a node distribution \( p \) from (2) (parameters tuned overall for the best performance in each method). A “new” measure graph \((H,q)\) is created by randomly permuting node labels of \( G \). There is a ground truth node correspondence between \( G \) and \( H \) and the goal is to measure the ability of GWL and SpecGWL to recover it. Given a coupling \( C \) of \((G,p)\) and \((H,q)\), we measure its performance by the node correctness score \( |S \cap S_{GT}|/|S| \), where \( S = \{(i,j)\mid C_{ij} > \epsilon\}, \epsilon > 0 \) a small threshold parameter, is the set of node correspondences from \( C \) and \( S_{GT} \) is the set of ground truth node correspondences. It was shown in [XLC19] that the adjacency-based GWL framework achieves state-of-the-art graph matching performance with respect to this metric.

For each graph \( G \) and permuted version \( H \), we compute couplings minimizing adjacency loss (1) and spectral loss (3) with \( t = 10 \) and compute their node correctness scores. Mean scores for each dataset are provided in Table 2 where we see that spectral loss outperforms adjacency, except on the dense Collab graph where results agree. Minimizers of spectral loss in SpecGWL were computed via standard gradient descent. Minimizers of adjacency loss in GWL were computed via both gradient descent and the regularized proximal gradient descent method of [XLC19], with the best results we obtained reported in Table 2 (proximal for the biological graphs, standard for the social graphs).

**Graph Partitioning.** We compare the performance of SpecGWL on a graph partitioning task against the following methods: Fluid [PGV\(^+\)17], FastGreedy [CNM04], Louvain [BGLL08], Infomap [RB08], and the partitioning module of S-GWL [XLC19]. Our experiments are carried out on four real-world datasets. The first is a directed Wikipedia hyperlink network [LK14] that we preprocessed by choosing 15 webpage categories and extracting their induced subgraphs. The resulting digraph had 1998 nodes and 2700 edges. The second was obtained from an Amazon product
Table 3: Comparison of adjusted mutual information scores across a variety of datasets.

| Dataset      | Fluid | FastGreedy | Louvain | Infomap | GWL   | SpecGWL |
|--------------|-------|------------|---------|---------|-------|---------|
| Wikipedia    |       | 0.382      | 0.377   | 0.332   | 0.312 | 0.442*  |
|              | sym, raw | —         |         |         |       |         |
|              | sym, noisy | 0.341    | 0.329   | 0.329   | 0.285 | 0.395   |
|              | asym, raw | —         | —       | 0.332   | 0.178 | 0.376   |
|              | asym, noisy | —      | —       | —       | 0.329 | 0.170   | 0.307  |
| EU-email     |       | 0.312      | 0.447   | 0.374   | 0.451 | 0.487   |
|              | sym, raw | —         |         |         |       |         |
|              | sym, noisy | 0.251    | 0.382   | 0.379   | 0.404 | 0.425   |
|              | asym, raw | —         | —       | —       | 0.443 | 0.420   | 0.437  |
|              | asym, noisy | —      | —       | —       | 0.356 | 0.422   | 0.377  |
| Amazon       |       | 0.637      | 0.622   | 0.940   | 0.443*| 0.692   |
|              | raw | —         |         |         |       |         |
|              | noisy | 0.347      | 0.573   | 0.584   | 0.463 | 0.352   | 0.441  |
| Village      |       | 0.881      | 0.881   | 0.881   | 0.606*| 0.801*  |
|              | raw | —         |         |         |       |         |
|              | noisy | 0.778      | 0.827   | 0.827   | 0.190 | 0.560   | 0.758  |

*Slight improvements possible with proximal gradient, but overall performance rankings are preserved.

network \([LK14]\) by taking the subgraph induced by the top 12 product categories. The resulting graph had 1501 nodes and 4626 edges. The third dataset was a digraph of email interactions between 42 departments of a European research institute \(\text{(EU-email)}\), and it comprised 1005 nodes and 25571 edges. The final dataset was a real-world network of interactions (8423 edges) among 1991 residents of 12 Indian villages \([\text{BCDJ13}]\), which we refer to as the Village dataset. We created noisy versions of each graph by adding up to 10% additional edges, and also created symmetrized versions of the Wikipedia and EU-email graphs by adding reciprocal edges.

The quality of each graph partition was measured by computing the adjusted mutual information (AMI) score against the ground-truth partition. The results are provided in Table 3. Reported results use standard gradient descent to compute GWL and SpecGWL scores. Despite issues with numerical instability, we also computed scores via the regularized proximal gradient method of \([XLC19]\) where possible. Slight score improvements are possible in some cases, but overall score rankings between methods are unchanged—details are provided in the Supplementary Materials. We find that SpecGWL consistently produces improved results compared to GWL, and moreover is the most consistent leader across all the methods. Its performance on directed graphs is especially relevant, considering that its closest competitor Infomap is a state-of-the-art method for digraph partitioning.

An important observation regarding runtime is that SpecGWL contains an additional scale parameter \(t\). In our experiments we tuned this parameter, starting by sampling \(\sim 10\) values in log space and following up in some cases by sampling \(\sim 20\) additional values in linear space. Given training data, choosing an optimal \(t\) parameter can be phrased as a separate learning task that can be carried out in a preprocessing step. Thus the correct runtime comparison between GWL and SpecGWL would be between the runtime of GWL and the average runtime of SpecGWL across \(t\) parameters. The average runtimes for all clustering experiments were \(8.5 \pm 5.7\) seconds for GWL and \(1.6 \pm 0.7\) seconds for SpecGWL, both using gradient descent optimization. Regularized proximal gradient descent provides a speedup for GWL; for example its 8.6s runtime on the Amazon dataset is reduced to 1.2s, on par with the average 1.4s runtime of SpecGWL.

5 Discussion

We have introduced a spectral notion of GW distance for graph comparison problems based on comparing heat kernels rather than adjacency matrices. This spectral variant is shown qualitatively and quantitatively to improve performance in graph matching and partitioning tasks. The techniques introduced here should be useful for studying further variants of GW distances. While spectral GW is faster than its adjacency counterpart on smaller graphs, it does not enjoy the same scalability properties. A significant direction of future work will be to construct multiscale approaches to analyze large scale graphs through both adjacency and spectral methods using divide-and-conquer techniques introduced in \([XLC19]\). On the theoretical front, Theorems 2 and 5 suggest that it is tractable to study...
spectral GW rigorously, and developing Theorem into a larger theory should illuminate further connections between optimal transport and spectral graph theory.

Acknowledgement

We would like to thank Facundo Mémoli for introducing us to the Gromov-Wasserstein problem and for many conversations on the topic.

References

[AMJ18] David Alvarez-Melis and Tommi Jaakkola. Gromov-Wasserstein alignment of word embedding spaces. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1881–1890, 2018.

[BAMKJ19] 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, 2019.

[BCDJ13] Abhijit Banerjee, Arun Chandrasekhar, Esther Duflo, and Matthew O Jackson. The diffusion of microfinance. Science, 341(6144):1236498, 2013.

[BG13] Michael M Bronstein and Klaus Glashoff. Heat kernel coupling for multiple graph analysis. arXiv preprint arXiv:1312.3035, 2013.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10):P10008, 2008.

[BOS+05] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. Bioinformatics, 21(suppl_1):i47–i56, 2005.

[Chi17] Lénaïc Chizat. Transport optimal de mesures positives: modèles, méthodes numériques, applications. PhD thesis, Université Paris-Dauphine, 2017.

[Chu05] Fan Chung. Laplacians and the Cheeger inequality for directed graphs. Annals of Combinatorics, 9(1):1–19, 2005.

[CM19] Samir Chowdhury and Facundo Mémoli. The Gromov–Wasserstein distance between networks and stable network invariants. Information and Inference: A Journal of the IMA, 8(4):757–787, 2019.

[CN19] Samir Chowdhury and Tom Needham. Gromov-Wasserstein averaging in a Riemannian framework. arXiv preprint arXiv:1910.04308, 2019.

[CNM04] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. Physical review E, 70(6):066111, 2004.

[DD03] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. Journal of molecular biology, 330(4):771–783, 2003.

[DS20] Yihe Dong and Will Sawin. Copt: Coordinated optimal transport on graphs. arXiv preprint arXiv:2003.03892, 2020.

[FC17] Rémi Flamary and Nicolas Courty. POT: Python Optimal Transport library. 2017.

[Fie73] Miroslav Fiedler. Algebraic connectivity of graphs. Czechoslovak mathematical journal, 23(2):298–305, 1973.

[GN02] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 99(12):7821–7826, 2002.

[Hen16] Reigo Hendrikson. Using Gromov-Wasserstein distance to explore sets of networks. Master’s thesis, University of Tartu, 2016.

[HRG14] Nan Hu, Raif M Rustamov, and Leonidas Guibas. Stable and informative spectral signatures for graph matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2305–2312, 2014.

[KKM+16] Kristian Kersting, Nils M. Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann. Benchmark data sets for graph kernels, 2016.

[LK14] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data June 2014.
[Mém07] Facundo Mémoli. On the use of Gromov-Hausdorff distances for shape comparison. The Eurographics Association, 2007.

[Mém11a] Facundo Mémoli. Gromov–Wasserstein distances and the metric approach to object matching. Foundations of Computational Mathematics, 11(4):417–487, 2011.

[Mém11b] Facundo Mémoli. A spectral notion of Gromov–Wasserstein distance and related methods. Applied and Computational Harmonic Analysis, 30(3):363–401, 2011.

[MN18] Facundo Mémoli and Tom Needham. Gromov-Monge quasi-metrics and distance distributions. arXiv preprint arXiv:1810.09646, 2018.

[NVM+18] Huda Nassar, Nate Veldt, Shahin Mohammadi, Ananth Grama, and David F Gleich. Low rank spectral network alignment. In Proceedings of the 2018 World Wide Web Conference, pages 619–628, 2018.

[PCS16] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. In International Conference on Machine Learning, pages 2664–2672, 2016.

[PGV+17] Ferran Parés, Dario García Gasulla, Armand Vilalta, Jonatan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. Fluid communities: a competitive, scalable and diverse community detection algorithm. In International Conference on Complex Networks and their Applications, pages 229–240. Springer, 2017.

[PK12] Rob Patro and Carl Kingsford. Global network alignment using multiscale spectral signatures. Bioinformatics, 28(23):3105–3114, 2012.

[RB08] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4):1118–1123, 2008.

[SCE+04] Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Brenda, the enzyme database: updates and major new developments. Nucleic acids research, 32(suppl_1):D431–D433, 2004.

[Smi84] Robert L Smith. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. Operations Research, 32(6):1296–1308, 1984.

[SOG09] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In Computer graphics forum, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[SPKS16] Justin Solomon, Gabriel Peyré, Vladimir G Kim, and Suvrit Sra. Entropic metric alignment for correspondence problems. ACM Transactions on Graphics (TOG), 35(4):72, 2016.

[SS13] Bernhard Schmitzer and Christoph Schnörr. Modelling convex shape priors and matching based on the Gromov-Wasserstein distance. Journal of mathematical imaging and vision, 46(1):143–159, 2013.

[Stu06] Karl-Theodor Sturm. On the geometry of metric measure spaces. Acta mathematica, 196(1):65–131, 2006.

[Stu12] Karl-Theodor Sturm. The space of spaces: curvature bounds and gradient flows on the space of metric measure spaces. arXiv preprint arXiv:1208.0434, 2012.

[TMK+18] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. NetLSD: hearing the shape of a graph. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2347–2356. ACM, 2018.

[VCF+18] Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties. arXiv preprint arXiv:1811.02834, 2018.

[VCTF19] Titouan Vayer, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In International Conference on Machine Learning, pages 6275–6284, 2019.

[VFC+19] Titouan Vayer, Rémi Flamary, Nicolas Courty, Romain Tavenard, and Laetitia Chapel. Sliced Gromov-Wasserstein. In Advances in Neural Information Processing Systems, pages 14726–14736, 2019.

[XLC19] Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable Gromov-Wasserstein learning for graph partitioning and matching. In Advances in Neural Information Processing Systems, pages 3046–3056, 2019.

[XLH+20] Hongteng Xu, Dixin Luo, Ricardo Henao, Svtí Shah, and Lawrence Carin. Learning autoencoders with relational regularization. arXiv preprint arXiv:2002.02913, 2020.
A Proofs of Theorems

A.1 Theorem 2

Proof. Suppose \( G = (V^G, E^G, p) \) and \( H = (V^H, E^H, q) \) have \( m \), \( n \) nodes, respectively. Let \( C \in \mathcal{C}(p, q) \). Then \( C \in [0, 1]^{m \times n} \) and satisfies \((m + n - 1)\) linear equality constraints coming from the row and column sums. By Lemma 1, spectral loss (6) is minimized at an extreme point of the convex polytope \( \mathcal{C}(p, q) \). This polytope lies in an \((mn - (m + n - 1))\)-dimensional affine subspace of \( mn \)-dimensional space. The equality constraints automatically ensure that each \( C_{ij} < 1 \), where the strict inequality holds because the graphs are fully supported and thus each \( p_i, q_j < 1 \). Therefore, estimating the number of zero entries is equivalent to estimating the number \( k \) of active nonnegativity constraints. An extreme point corresponds to the intersection of \( k \) hyperplanes in general position with this affine subspace, and this intersection has dimension \( mn - (m + n - 1) - k \). Because the extreme point has dimension 0, we have \( k = mn - (m + n - 1) \). If the hyperplanes are not in general position, then the number of active nonnegativity constraints, i.e. the number of zeros, is \( \geq mn - (m + n - 1) \). Next suppose \( m \sim n \). Then the ratio of nonzero entries to total entries of \( C \) is roughly \( \frac{n^2 - k}{n^2} = \frac{2n - 1}{n^2} \), and this term tends to 0 as \( n \to \infty \). \( \square \)

A.2 Theorem 3

Proof. Let \( G = (V, E) \), with \(|V| = n\), be a graph satisfying the assumptions, endowed with uniform vertex distribution \( p \) and let \( K^T \) denote the heat kernel of \( G \). By definition,

\[
K^T = \Phi e^{-t \Lambda} \Phi^T = \sum_{j=1}^{n} e^{-t \lambda_j} \phi_j \phi_j^T,
\]

where \( \Phi \) is a matrix whose columns are the orthonormal eigenvectors \( \phi_1, \ldots, \phi_n \) of the graph Laplacian \( L \) of \( G \) and \( \Lambda \) is the diagonal matrix of sorted eigenvalues \( 0 = \lambda_1 < \lambda_2 < \lambda_3 \leq \cdots \leq \lambda_n \) of \( L \). Let \( Q \) be the 2-way partitioning template from [5]. Since \( p \) is uniform, the estimated distribution \( q \) is also uniform and \( Q = \frac{1}{2} I_2 \), where \( I_2 \) is the \( 2 \times 2 \) identity matrix.

The 2-way spectral GW partitioning of \( G \) is obtained from a coupling minimizing the spectral partitioning loss (6). Using Lemma 1 we see that this optimization task is equivalent to maximizing

\[
C \mapsto \langle K^T C, CQ \rangle = \frac{1}{2} \langle K^T C, C \rangle
\]

coupling the polytope \( \mathcal{C}(p, q) \). Since the factor of \( \frac{1}{2} \) does not effect the optimization, we suppress it and further simplify the objective function as

\[
\langle K^T C, C \rangle = \text{tr} \left( (K^T C)^T C \right) = \text{tr} \left( C C^T \sum_{j=1}^{n} e^{-t \lambda_j} \phi_j \phi_j^T \right) = \sum_{j=1}^{n} e^{-t \lambda_j} \text{tr} \left( C C^T \phi_j \phi_j^T \right).
\]

Since the leading eigenvector is \( \phi_1 = \frac{1}{\sqrt{n}} \) (the normalized vector of all ones), it is easy to check that the term \( \text{tr}(CC^T \phi_1 \phi_1^T) \) is constant for all \( C \in \mathcal{C}(p, q) \). The objective therefore becomes to maximize over \( C \in \mathcal{C}(p, q) \) the quantity

\[
\sum_{j=2}^{n} e^{-t \lambda_j} \text{tr} \left( CC^T \phi_j \phi_j^T \right) = e^{-t \lambda_2} \left( \text{tr} \left( CC^T \phi_2 \phi_2^T \right) + \sum_{j=3}^{n} e^{-t(\lambda_j - \lambda_2)} \text{tr} \left( CC^T \phi_j \phi_j^T \right) \right),
\]

which is in turn equivalent to maximizing

\[
C \mapsto \text{tr} \left( CC^T \phi_2 \phi_2^T \right) + \sum_{j=3}^{n} e^{-t(\lambda_j - \lambda_2)} \text{tr} \left( CC^T \phi_j \phi_j^T \right).
\]

(7)
We further simplify the objective function (8) as
\[ C \mapsto \text{tr} \left( C C^T \phi_2 \phi_2^T \right) \] over \( C(p,q) \). It then remains to study the structure of maximizers of (8).

We further simplify the objective function (8) as
\[ \text{tr} \left( C C^T \phi_2 \phi_2^T \right) = \text{tr} \left( (C^T \phi_2)^T (C^T \phi_2) \right) = ||C^T \phi_2||^2, \]
where the norm in the last line is the Frobenius norm. We denote the column vectors of \( C \) by \( C_1, C_2 \in \mathbb{R}^{1 \times n} \), so that
\[ ||C^T \phi_2||^2 = \left\| \begin{pmatrix} C_1 \cdot \phi_2 \\ C_2 \cdot \phi_2 \end{pmatrix} \right\|^2, \]
where the norm on the right is the Euclidean norm. Since \( C \in \mathcal{C}(p,q) \) and \( p \) is uniform, we have \( C_2 = \frac{1}{n} \mathbf{1}_{n \times 1} - C_1 \), whence
\[ C_2 \cdot \phi_2 = \left( \frac{1}{n} \mathbf{1}_{n \times 1} - C_1 \right) \cdot \phi_2 = -C_1 \cdot \phi_2, \]
since \( \phi_2 \) is orthogonal to \( \mathbf{1}_{n \times 1} = \sqrt{n} \phi_1 \). The objective (8) is finally reduced to
\[ C \mapsto 2(C_1 \cdot \phi_2)^2. \] (9)

Let \( \phi_2^+ \) be the vector of positive entries of \( \phi_2 \) with all negative entries thresholded to zero and likewise define \( \phi_2^- \) to be the vector of negative entries of \( \phi_2 \). Assume without loss of generality that \( ||\phi_2^+|| \geq ||\phi_2^-|| \) (the other case follows entirely similarly). Then in order to maximize (9), one should set each entry of \( C_1 \) to be nonzero if and only if the corresponding entry of \( \phi_2 \) is positive. The spectral GW partitioning therefore agrees with the Fiedler partitioning, and the proof is complete. \( \square \)

B  An MCMC Sampler for Couplings.

Both the adjacency (1) and spectral (3) loss functions are nonconvex, and solving such problems effectively often relies on a clever choice of initialization. A limitation of the current practice is that this initialization is often chosen to be the product coupling \( pq^T \), which we empirically find to be sub-optimal in even simple cases. This is accomplished by running gradient descent from each point in an ensemble of initializations generated by a Markov Chain Monte Carlo Hit-And-Run sampler [Smi84]. This algorithm is well-known, but we describe it below for the convenience of the reader. Our code includes a lean Python implementation written specifically for sampling the coupling polytope; we hope such an implementation will be useful to the broader optimal transport community.

### Algorithm 1 Markov chain sampler

1: function MARKOVSTEP(\( A, p, q, C \))
2:     // \( A \): matrix of linear constraints
3:     // \( p, q : m \times 1, n \times 1 \) probability vectors
4:     // \( C : m \times n \) initial coupling matrix
5:     \( V \leftarrow \) random \( m \times n \) matrix as direction
6:     \( Q \leftarrow \) o.n. basis for row space of \( A \)
7:     \( V \leftarrow V - QQ^T V \)  \( \triangleright \) project \( V \) to correct subspace
8:     \( \text{pos} \leftarrow \) indices where \( V > 0 \)
9:     \( \text{neg} \leftarrow \) indices where \( V < 0 \)
10:    \( \alpha \leftarrow \max(-C[\text{pos}]^T/V[\text{pos}]) \)
11:    \( \beta \leftarrow \min(-C[\text{neg}]^T/V[\text{neg}]) \)  \( \triangleright [\alpha, \beta] \) is maximal range of step sizes
12:    \( \gamma \leftarrow \) random element of \( [\alpha, \beta] \)
13:    \[ C \leftarrow C + \gamma V \]  \( \triangleright \) new coupling matrix
14: return \( C \)
15: end function

C  Additional experiments and implementation details

C.1  Additional Landscape Results

Figure 2 gives a more detailed view of the results reported in Table 1. For each plot, the \( x \)-axis is (Worst or Product) error percentage. The \( y \)-axis shows the percentage of samples whose error was above the relative error.
We see that a significant number of samples have high error rates for adjacency loss (1) and spectral loss (3) with $t = 5$. For spectral loss with $t = 10$ or 20, these error rates are greatly decreased. In particular, spectral loss with $t = 20$ has essentially zero samples with error rate above 2%.

C.2 Visualizing Graph Matchings

Here we describe how the interpolations used to visualize coupling quality in Figure 1 were produced. Let $(G, p)$ and $(H, q)$ be measure graphs and $C \in C(p, q)$ a coupling. To produce an interpolation, we first “blow up” $C$ so that it has the form of a weighted permutation matrix. This is done by first scanning across rows; any row with more than a single nonzero entry is split into “dummy” copies, each of which contains a single nonzero entry from the original row. The splits allow us to split nodes of $G$ into dummy copies, with weights given by entries in the corresponding row of $C$. The same procedure is applied to split columns of $C$ and to split nodes of $H$. The result is a pair of expanded measure graphs $(G', p')$ and $(H', q')$ together with an expanded coupling $C'$ which provides a bijective correspondence between the nodes of $G'$ and $H'$. Once such a bijective correspondence is obtained, we position each graph $G'$ and $H'$ in the plane using a common embedding modality and then performing Procrustes alignment of the resulting embeddings. To interpolate the graphs, we simply interpolate positions of the bijectively matched nodes, while phasing in new edges that are formed. This visualization method has strong theoretical justification: building on work of [Stu12], it is shown in [CN19] that this process represents a geodesic (in the metric geometry sense) in the space of edge-weighted measure graphs. We observe that the conclusion of Lemma 1 is useful here, since the theoretical guarantee on the sparsity of $C$ implies that $C$ will not get too large in the “blow up” phase of the algorithm.

To produce each example in Figure 1, we sampled 100 couplings from the coupling polytope via the MCMC algorithm (1000 MCMC steps between each coupling) as initializations. We then computed an optimal coupling between the graphs by optimizing the relevant loss function from each initialization and keeping the coupling with the lowest loss from the resulting ensemble.

C.3 Averaging

We use the observations regarding the energy landscape and the quality of matchings to show that in the GW averaging problem, using the heat kernel leads to 10x faster convergence than the adjacency matrix, and moreover, the heat kernel yields a more “unique” barycenter. Specifically, given measure network representations $X_1, X_2, \ldots, X_n$, a Fréchet mean is an element of $\arg \min_X \sum_i d_{GW}(X, X_i)^2$. The objective of the GW averaging problem is to compute this barycenter, i.e. an average representation. In the Python OT package [FC17], this barycenter is computed iteratively from a random initialization (cf. the gromov_barycenter function). As a proxy for the “uniqueness” of the barycenter, we compute the barycenter for multiple random initializations, and then take the variance of the distribution of Fréchet losses achieved by the barycenters.

We demonstrate this claim on the Village dataset. We ran a bootstrapping procedure to sample 10 sets of 30 nodes, and took the induced subgraphs to obtain 10 subgraphs. To keep the samples from being too sparse, we first sorted the nodes in order of decreasing betweenness centrality, and then selected 30 nodes (for each iteration) from the top 40 nodes with the highest centrality. Next we computed both adjacency and heat kernel representations (for $t = 3, 7, 11$) of these subgraphs. Then we used the gromov_barycenter function to compute averages of the adjacency and heat kernel representations. Each call to gromov_barycenter uses a random initialization. Using this randomness as a source of stochasticity, we repeated the set of barycenter computations 10 times to obtain four distribution of Fréchet losses. After mean-centering the distributions, the variance of the adjacency distribution was found to be two orders of magnitude higher than any of the heat kernel distributions, and each of the three comparisons was found to be statistically significant by computing
Figure 3: **Left:** Differences in Fréchet loss of the GW average across representations. **Center:** Mean-centered Fréchet loss, indicating the greater variance and sensitivity to initialization for the adjacency representation. **Right:** Distribution of runtimes shows 10x speedup for the heat kernel.

Table 4: Comparison between runtime of GWL and average runtime of SpecGWL across $t$ parameters. “-Prox” rows use regularized proximal gradient with Sinkhorn iterations as in [XLC19], other rows use vanilla gradient descent.

| Method       | Wikipedia (sym) | Wikipedia (asym) | EU-email (sym) | EU-email (asym) | Amazon | Village |
|--------------|-----------------|------------------|---------------|----------------|--------|---------|
|              | raw noisy       | raw noisy        | raw noisy     | raw noisy      | raw noisy | raw noisy | raw noisy |
| GWL          | 14.1 16.1 16.0 14.2 | 1.5 6.7 0.9 1.6 | 8.6 13.1 3.3 5.4 |
| SpecGWL      | 1.8 2.3 2.4 2.4 | 1.0 0.9 0.9 1.0 | 1.4 0.9 1.8 2.7 |
| GWL-Prox     | — — — —         | 0.9 0.8 — —     | 1.2 1.0 2.2 2.2 |
| SpecGWL-Prox | 2.9 2.6 2.9 2.9 | 0.9 1.0 1.0 1.0 | 1.3 1.8 1.8 2.0 |

Bartlett tests for unequal variance ($p < 10^{-6}$ for all, adjusted for multiple comparison via Bonferroni correction). Boxplots of the results are shown in Figure 3.

C.4 Graph Partitioning

Runtimes for GWL and SpecGWL on the graph partitioning experiment are reported in Table 4. For SpecGWL, the times are averaged over several values of $t$, with the idea that finding the correct $t$-value is a preprocessing hyperparameter tuning step. For both GWL and SpecGWL, partitionings were obtained using standard projected gradient descent. Speedups are obtained for GWL via the regularized proximal gradient method, but we were not able to obtain results on all datasets with this method due to numerical issues (see below). Runtimes for this method are also reported as GWL-Prox. We observe that spectral loss provides up to 10x acceleration in convergence rate for standard gradient descent and even outperforms the proximal gradient in compute time.

When employing the regularized proximal gradient method, we found that the results were sensitive to the choice of regularization parameter $\beta$ (as is also observed in [XLC19]), leading to numerical blowups if not chosen carefully. In reporting each of the results below, we hand-tuned $\beta$ after testing in the $10^{-1}, 10^{-2}, 10^{-3}, \ldots, 10^{-9}$ regimes. For **Wikipedia**, we used $\beta = 2 \cdot 10^{-5}$ for SpecGWL, but were unable to find a $\beta$ that provided stable results for GWL. For **EU-email**, we used $2 \cdot 10^{-7}$ for GWL and $3 \cdot 10^{-8}$ for SpecGWL. For **Amazon**, we used $\beta = 4 \cdot 10^{-5}$ for GWL and $1.5 \cdot 10^{-6}$ SpecGWL. Finally, for **Village** we used $\beta = 5 \cdot 10^{-6}$ for SpecGWL. This $\beta$ led to numerical instability for GWL, but $\beta = 5 \cdot 10^{-5}$ worked and yielded the results we report below. In summary, it appears that the structure of the graph has a significant effect on the optimal choice of regularization parameter (e.g. the Wikipedia graph is relatively very sparse). Because the numerical instability issues are very sensitive to the regularization, one avenue for future work could be to incorporate the strategies described in the PhD thesis [Chi17] (e.g. “absorption into the log domain”) to stabilize the regularized proximal gradient method.
Table 5: Performance of GWL and SpecGWL using regularized proximal gradient descent and Sinkhorn iterations as in [XLC19]. ‘—’ denotes that an AMI score could not be calculated due to numerical instability.

| Method         | Wikipedia | EU-email | Amazon | Village |
|----------------|-----------|----------|--------|---------|
|                | asym      | sym      | asym   |         |
|                | raw       | noisy    | raw    | noisy   |
| GWL-Prox       | —         | —        | —      | —       | 0.49 0.39 0.72* 0.58 |
| SpecGWL-Prox   | 0.51 0.39 | 0.39 0.29 | 0.01 0.01 | 0.03 0.03 | **0.66** 0.43 **0.84** **0.72** |

*The code provided with [XLC19] included a representation matrix as `database['cost']`, and this yielded the score of 0.72. However, this matrix was asymmetric and not equal to the symmetrized adjacency matrix that was used in experiments with other benchmarks. When using a symmetrized adjacency matrix, the score drops from 0.72 to 0.66.