Generalized Spectral Clustering via Gromov-Wasserstein Learning: Supplementary Materials

6 Proofs of Theorems

6.1 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, at least one of the minimizers of spectral loss (5) is located 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 greater than or equal to $mn - (m + n - 1)$. 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 $n \to \infty$. □

6.2 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 Φ is a matrix whose columns are the orthonormal eigenvectors $\phi_1, \ldots, \phi_n$ of the graph Laplacian L of G and Λ is the diagonal matrix of sorted eigenvalues $0 = \lambda_1 < \lambda_2 < \lambda_3 \leq \lambda_4 \leq \cdots \leq \lambda_n$ of L. Let Q be the 2-way partitioning template from (7). Since p is uniform, the estimated distribution q is also uniform and $Q = \frac{1}{2}I_2$, where I_2 is the 2×2 identity matrix.

The 2-way spectral GW partitioning of G is obtained from a coupling minimizing the spectral partitioning loss (8). Using Lemma 1, we see that this optimization task is equivalent to maximizing

$$C \mapsto \langle K^tC, CQ\rangle = \frac{1}{2}\langle K^tC, C\rangle$$

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

$$\langle K^tC, C\rangle = \operatorname{tr}\left((K^tC)^TC\right) = \operatorname{tr}\left(CC^T\sum_{j=1}^{n} e^{-t\lambda_j}\phi_j\phi_j^T\right) = \sum_{j=1}^{n} e^{-t\lambda_j}\operatorname{tr}\left(CC^T\phi_j\phi_j^T\right).$$

Since the leading eigenvector is $\phi_1 = \frac{1}{\sqrt{n}}1^{n\times 1}$ (the normalized vector of all ones), it is easy to check that the term $\operatorname{tr}(CC^T\phi_1\phi_1^T)$ 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

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

which is in turn equivalent to maximizing

$$C \mapsto \operatorname{tr}\left(CC^T\phi_2\phi_2^T\right) + \sum_{j=3}^{n} e^{-t(\lambda_j-\lambda_2)}\operatorname{tr}\left(CC^T\phi_j\phi_j^T\right). \tag{9}$$

Observe that the summation term goes to zero as $t \to \infty$ (using $\lambda_j > \lambda_2$ for all $j \geq 3$), while the first term is independent of t. It follows that, for sufficiently large t, maximization of (9) is equivalent to maximizing

$$C \mapsto \operatorname{tr}\left(CC^T\phi_2\phi_2^T\right) \tag{10}$$

over $\mathcal{C}(p,q)$. It then remains to study the structure of maximizers of (10).

We further simplify the objective function (10) as

$$\operatorname{tr}\left(CC^T\phi_2\phi_2^T\right) = \operatorname{tr}\left((C^T\phi_2)^T(C^T\phi_2)\right) = \|C^T\phi_2\|^2,$$

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 $C_2 = \frac{1}{n}1^{n\times 1} - C_1$, whence

$$C_2\cdot\phi_2 = \left(\frac{1}{n}1^{n\times 1} - C_1\right)\cdot\phi_2 = -C_1\cdot\phi_2,$$

since ϕ_2 is orthogonal to $1^{n\times 1} = \sqrt{n}\phi_1$. The objective (10) is finally reduced to

$$C \mapsto 2(C_1\cdot\phi_2)^2. \tag{11}$$

Let ϕ_2^+ be the vector of positive entries of ϕ_2 with all negative entries thresholded to zero and likewise define ϕ_2^- to be the vector of negative entries of ϕ_2. Assume without loss of generality that $\|\phi_2^+\| \geq \|\phi_2^-\|$ (the other case follows entirely similarly). Then in order to maximize (11), one should set each entry of C_1 to be nonzero if and only if the corresponding entry of ϕ_2 is positive. The spectral GW partitioning therefore agrees with the Fiedler partitioning, and the proof is complete. □

7 An MCMC Sampler for Couplings.

Both the adjacency (3) and spectral (5) 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 (Smith, 1984). 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.

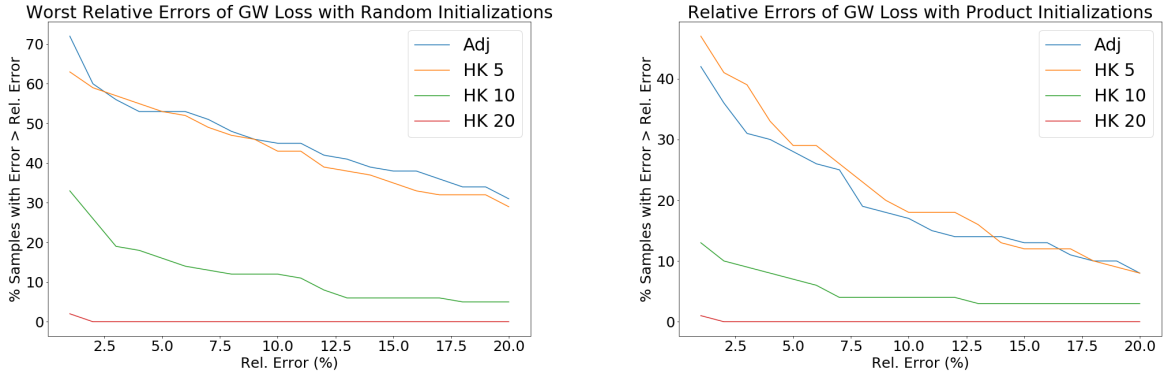

Figure 3: Results of energy landscape experiments.

Algorithm 1 Markov chain sampler

function MARKOVSTEP(A, p, q, C)
  // A: matrix of linear constraints
  // $p, q : m \times 1, n \times 1$ probability vectors
  // $C : m \times n$ initial coupling matrix

  $V \leftarrow$ random $m \times n$ matrix as direction
  $Q \leftarrow$ o.n. basis for row space of A
  $V \leftarrow V - QQ^TV$ ▷ project V to correct subspace
  pos $\leftarrow$ indices where $V > 0$
  neg $\leftarrow$ indices where $V < 0$
  $\alpha \leftarrow \max(-C[\text{pos}]/V[\text{pos}])$
  $\beta \leftarrow \min(-C[\text{neg}]/V[\text{neg}])$ ▷ $[\alpha, \beta]$ is maximal range of step sizes
  $\gamma \leftarrow$ random element of $[\alpha, \beta]$
return $C + \gamma V$ ▷ new coupling matrix
end function

8 Additional experiments and implementation details

8.1 Additional Landscape Results

Figure 3 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 rate. We see that a significant number of samples have high error rates for adjacency loss (3) and spectral loss (5) 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%.

8.2 Additional figures

Here we add some figures that help better understand some of the quantitative results presented in the main text. Figure 4 shows that the improvement in the graph matching experiment obtained via SpecGWL remains stable across a wide range of scale parameters. Specifically, we computed SpecGWL loss for $t \in \{10, 20, \ldots, 90\}$ for each of the four datasets. The **Collab** dataset was the only one where there was no appreciable improvement from using SpecGWL, but there was no significant decrease in performance either for scales in the range $t \in \{10, 20, 30\}$.

Figure 5 shows the 10 stochastic block model networks used in the synthetic graph partitioning experiment. Each network has 5 blocks, and the block sizes were chosen uniformly at random from the range $[20, 50]$ at the beginning of the experiment. Within-block edge densities were fixed at 0.5, and across-block edge densities were

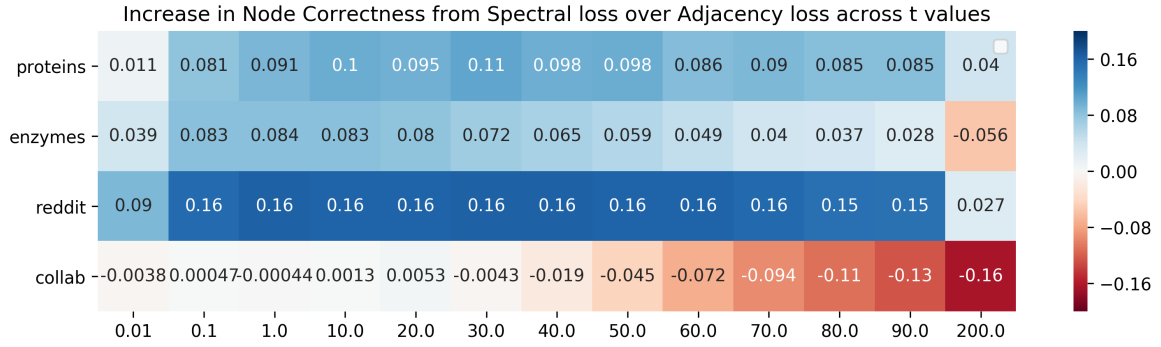

Figure 4: Improvement (blue cells) obtained by using spectral loss (5) instead of adjacency loss (3) across a range of t values (x axis).

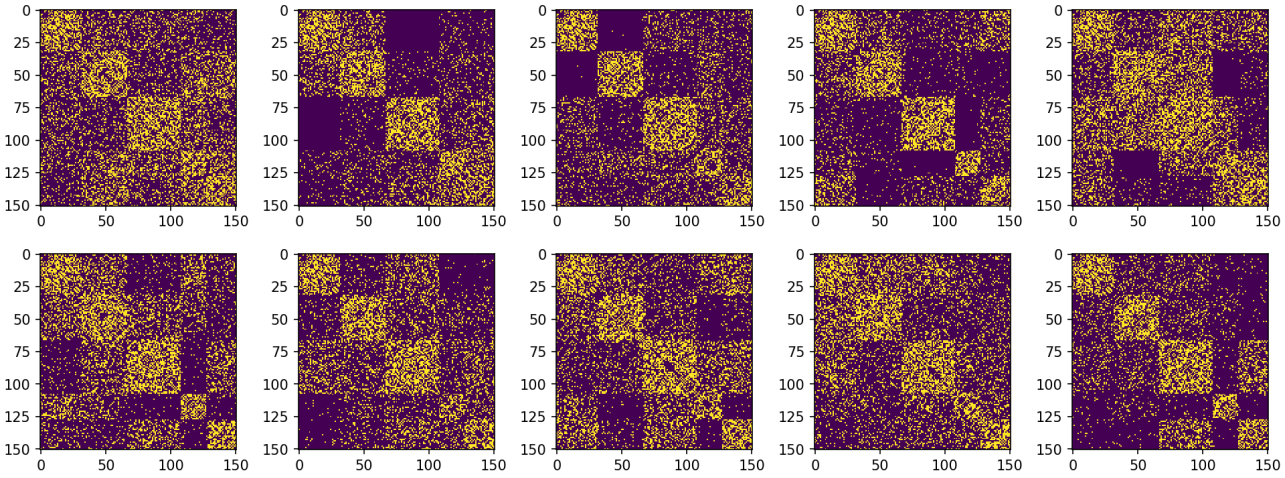

Figure 5: Stochastic block models used in the supervised partitioning task with cross-validation

chosen uniformly at random in the range $[0, 0.3]$.

8.3 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 $C \in \mathcal{C}(p, q)$ 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 Sturm (2012), it is shown by Chowdhury and Needham (2020) 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

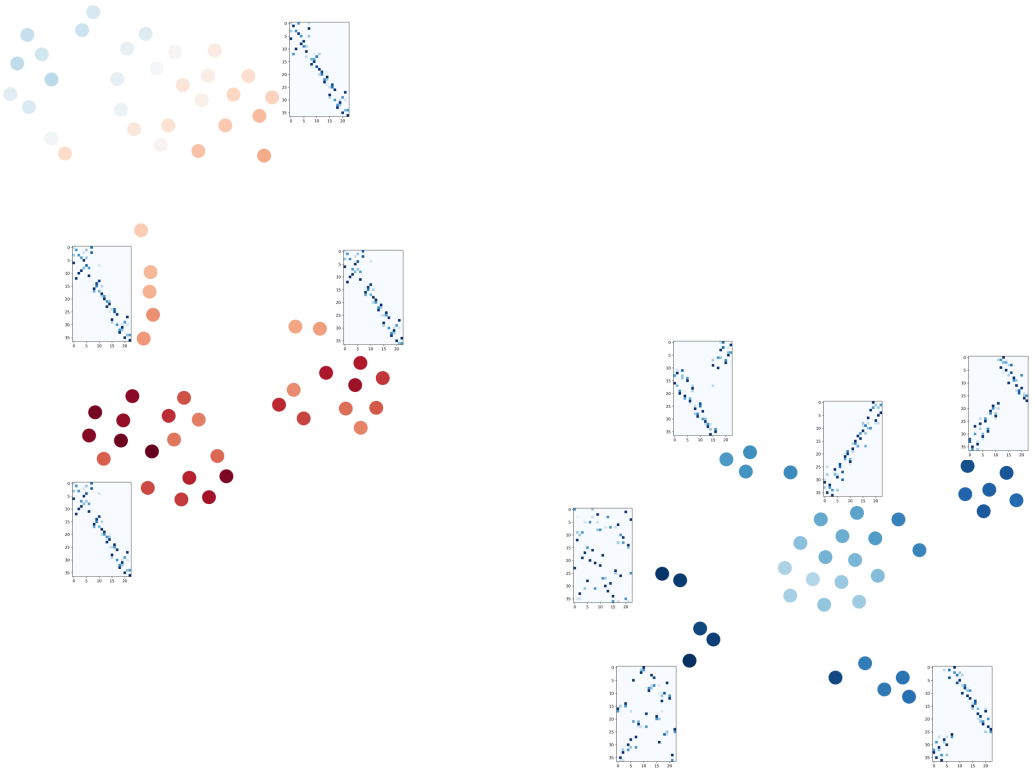

Figure 6: t-SNE embedding of optimal couplings between two graphs from the **Enzymes** dataset obtained via spectral loss (5) using 100 equally spaced t values in the range $[0, 50]$. The ground metric is the l^2 norm between vectorized representations of the couplings. The range $[0, 50]$ is mapped linearly from the blue to red color scheme.

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.

8.4 Dependence of Matchings on t-Value

To understand the landscape of optimal couplings that occur for different scale parameters, we took two graphs from the **Enzymes** dataset and computed optimal couplings for spectral loss using 100 linearly spaced t values in the range $[0, 50]$. Figure 6 shows a t-SNE embedding (with perplexity $= 15$) obtained after flattening these couplings into vectors in Euclidean space. The inset coupling matrices are representatives of the points in each significant cluster. Interpolation visualizations for some of the coupling matrices from different clusters are provided in Figure 7.

8.5 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 $\arg\min_X \sum_i d_{\mathrm{GW}}(X, X_i)^2$. The objective of the GW averaging problem is to compute this barycenter, i.e. an average representation. In the Python OT package (Flamary and Courty, 2017), 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,

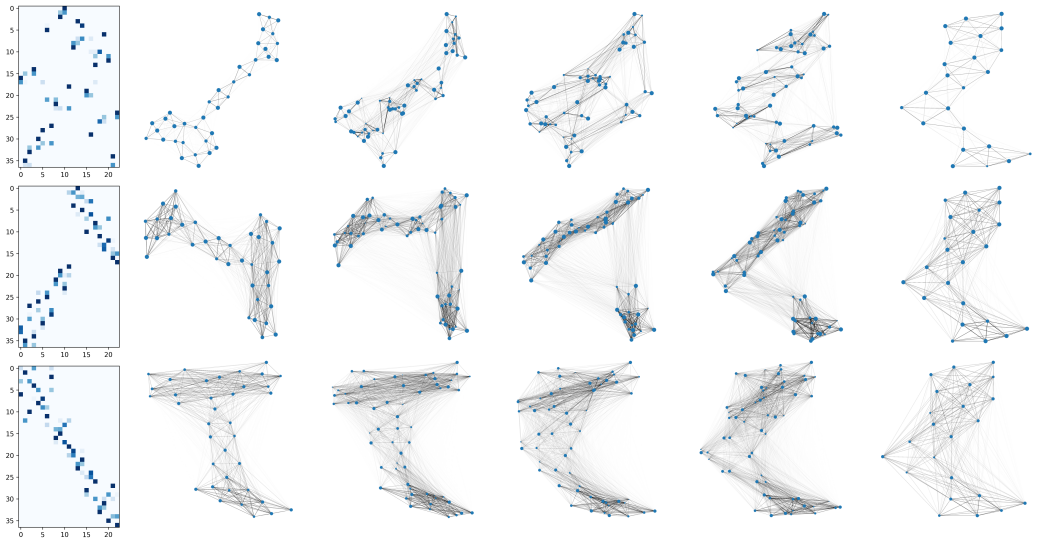

Figure 7: Interpolation visualizations obtained via some of the representative couplings in Figure 6 arranged top to bottom with increasing t. Note that the interpolation in the third row, corresponding to the largest t value, represents global structure more faithfully.

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 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 8.

Figure 8: **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.

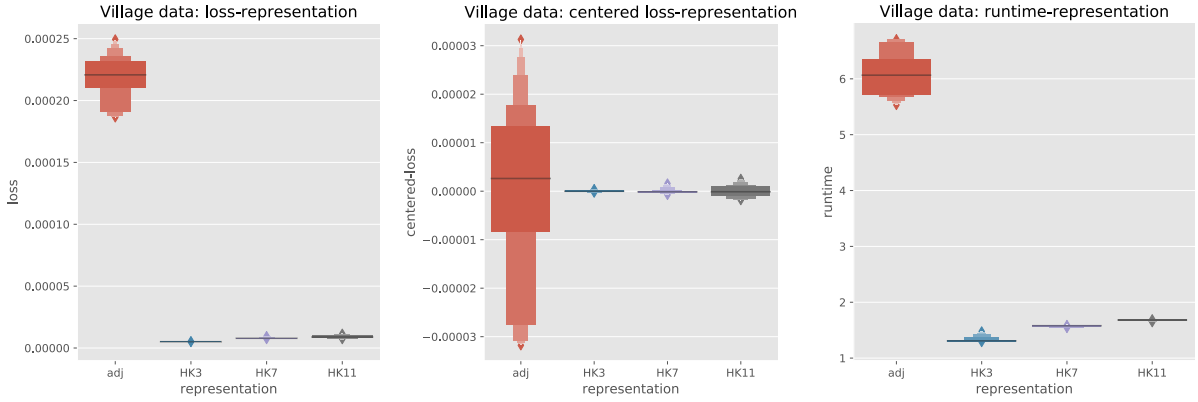

Table 5: Comparison between runtime of GWL and average runtime of SpecGWL across t parameters. "-Prox" rows use regularized proximal gradient with Sinkhorn iterations as used by Xu et al. (2019a). Other rows use vanilla gradient descent.

Method	Wikipedia				EU-email				Amazon		Village	
	sym		asym		sym		asym					
	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**

Table 6: Performance of GWL and SpecGWL using regularized proximal gradient descent and Sinkhorn iterations as in Xu et al. (2019a). '—' 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	raw	noisy	raw	noisy	raw	noisy	raw	noisy
GWL-Prox	—	—	—	—	**0.45**	**0.40**	—	—	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 by Xu et al. (2019a) 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.

8.6 Graph Partitioning

Runtimes for GWL and SpecGWL on the graph partitioning experiment are reported in Table 5. 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 β (as is also observed by Xu et al. (2019a)), leading to numerical blowups if not chosen carefully. In reporting each of the results below, we hand-tuned β 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 β 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 $\beta = 4 \cdot 10^{-3}$ for GWL and $1.5 \cdot 10^{-6}$ SpecGWL. Finally, for **Village** we used $\beta = 5 \cdot 10^{-6}$ for SpecGWL. This β 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 of Chizat (2017) (e.g. "absorption into the log domain") to stabilize the regularized proximal gradient method.