Generalized Matrix Means for Semi-Supervised Learning with Multilayer Graphs

Pedro Mercado[1], **Francesco Tudisco**[2] **and Matthias Hein**[1]
[1]University of Tübingen, Germany
[2]Gran Sasso Science Institute, Italy

Abstract

We study the task of semi-supervised learning on multilayer graphs by taking into account both labeled and unlabeled observations together with the information encoded by each individual graph layer. We propose a regularizer based on the generalized matrix mean, which is a one-parameter family of matrix means that includes the arithmetic, geometric and harmonic means as particular cases. We analyze it in expectation under a Multilayer Stochastic Block Model and verify numerically that it outperforms state of the art methods. Moreover, we introduce a matrix-free numerical scheme based on contour integral quadratures and Krylov subspace solvers that scales to large sparse multilayer graphs.

1 Introduction

The task of graph-based Semi-Supervised Learning (SSL) is to build a classifier that takes into account both labeled and unlabeled observations, together with the information encoded by a given graph[5, 29]. A common and successful approach is to take a suitable loss function on the labeled nodes and a regularizer which provides information encoded by the graph [2, 17, 33, 35, 38]. Whereas this task is well studied, traditionally these methods assume that the graph is composed by interactions of one single kind, i.e. only one graph is available.

For the case where multiple graphs, or equivalently, multiple layers are available, the challenge is to boost the classification performance by merging the information encoded in each graph. The arguably most popular approach for this task consists of finding some form of convex combination of graph matrices, where more informative graphs receive a larger weight [1, 15, 16, 25, 30, 31, 34, 36].

Note that a convex combination of graph matrices can be seen as a weighted arithmetic mean of graph matrices. In the context of multilayer graph clustering, previous studies [21–23] have shown that weighted arithmetic means are suboptimal under certain benchmark generative graph models, whereas other matrix means, such as the geometric [22] and harmonic means [21], are able to discover clustering structures that the arithmetic means overlook.

In this paper we study the task of semi-supervised learning with multilayer graphs with a novel regularizer based on the power mean Laplacian. The power mean Laplacian is a one-parameter family of Laplacian matrix means that includes as special cases the arithmetic, geometric and harmonic mean of Laplacian matrices.We show that in expectation under a Multilayer Stochastic Block Model, our approach provably correctly classifies unlabeled nodes in settings where state of the art approaches fail. In particular, a limit case of our method is provably robust against noise, yielding good classification performance as long as one layer is informative and remaining layers are potentially just noise. We verify the analysis in expectation with extensive experiments with random graphs, showing that our approach compares favorably with state of the art methods, yielding a good classification performance on several relevant settings where state of the art approaches fail.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

name	minimum	harmonic mean	geometric mean	arithmetic mean	maximum
p	$p \to -\infty$	$p = -1$	$p \to 0$	$p = 1$	$p \to \infty$
$m_p(a,b)$	$\min\{a,b\}$	$2\left(\frac{1}{a}+\frac{1}{b}\right)^{-1}$	$\sqrt{ab}$	$(a+b)/2$	$\max\{a,b\}$

Table 1: Particular cases of scalar power means

Moreover, our approach scales to large datasets: even though the computation of the power mean Laplacian is in general prohibitive for large graphs, we present a matrix-free numerical scheme based on integral quadratures methods and Krylov subspace solvers which allows us to apply the power mean Laplacian regularizer to large sparse graphs. Finally, we perform numerical experiments on real world datasets and verify that our approach is competitive to state of the art approaches.

2 The Power Mean Laplacian

In this section we introduce our multilayer graph regularizer based on the power mean Laplacian. We define a multilayer graph $\mathbb{G}$ with T layers as the set $\mathbb{G} = \{G^{(1)}, \dots, G^{(T)}\}$, with each graph layer defined as $G^{(t)} = (V, W^{(t)})$, where $V = \{v_1, \dots, v_n\}$ is the node set and $W^{(t)} \in \mathbb{R}_+^{n \times n}$ is the corresponding adjacency matrix, which we assume symmetric and nonnegative. We further denote the layers' normalized Laplacians as $L_{\text{sym}}^{(t)} = I - (D^{(t)})^{-1/2} W (D^{(t)})^{-1/2}$, where $D^{(t)}$ is the degree diagonal matrix with $(D^{(t)})_{ii} = \sum_{j=1}^{n} W_{ij}^{(t)}$.

The scalar power mean is a one-parameter family of scalar means defined as

$$m_p(x_1, \dots, x_T) = \left(\tfrac{1}{T} \textstyle\sum_{i=1}^{T} x_i^p\right)^{1/p}$$

where $x_1, \dots, x_T$ are nonnegative scalars and p is a real parameter. Particular choices of p yield specific means such as the arithmetic, geometric and harmonic means, as illustrated in Table 1.

The **Power Mean Laplacian**, introduced in [21], is a matrix extension of the scalar power mean applied to the Laplacians of a multilayer graph and proposed as a more robust way to blend the information encoded across the layers. It is defined as

$$L_p = \left(\tfrac{1}{T} \textstyle\sum_{i=1}^{T} (L_{\text{sym}}^{(i)})^p\right)^{1/p}$$

where $A^{1/p}$ is the unique positive definite solution of the matrix equation $X^p = A$. For the case $p \leq 0$ a small diagonal shift $\varepsilon > 0$ is added to each Laplacian, i.e. we replace $L_{\text{sym}}^{(i)}$ with $L_{\text{sym}}^{(i)} + \varepsilon$, to ensure that L_p is well defined as suggested in [3]. In what follows all the proofs hold for an arbitrary shift. Following [21], we set $\varepsilon = \log_{10}(1 + |p|) + 10^{-6}$ for $p \leq 0$ in the numerical experiments.

3 Multilayer Semi-Supervised Learning with the Power Mean Laplacian

In this paper we consider the following optimization problem for the task of semi-supervised learning in multilayer graphs: Given k classes $r = 1, \dots, k$ and membership vectors $Y^{(r)} \in \mathbb{R}^n$ defined by $Y_i^{(r)} = 1$ if node v_i belongs to class r and $Y_i^{(r)} = 0$ otherwise, we let

$$f^{(r)} = \operatorname*{arg\,min}_{f \in \mathbb{R}^n} \|f - Y^{(r)}\|^2 + \lambda f^T L_p f \,. \tag{1}$$

The final class assignment for an unlabeled node v_i is $y_i = \arg\max\{f_i^{(1)}, \dots, f_i^{(k)}\}$. Note that the solution f of (1), for a particular class r, is such that $(I + \lambda L_p) f = Y^{(r)}$. Equation (1) has two terms: the first term is a loss function based on the labeled nodes whereas the second term is a regularization term based on the power mean Laplacian L_p, which accounts for the multilayer graph structure. It is worth noting that the Local-Global approach of [35] is a particular case of our approach when only one layer ($T = 1$) is considered. Moreover, not that when $p = 1$ we obtain a regularizer term based on the arithmetic mean of Laplacians $L_1 = \frac{1}{T} \sum_{i=1}^{T} L_{\text{sym}}^{(i)}$. In the following section we analyze our proposed approach (1) under the Multilayer Stochastic Block Model.

4 Multilayer Stochastic Block Model

In this section we provide an analysis of semi-supervised learning for multilayer graphs with the power mean Laplacian as a regularizer under the Multilayer Stochastic Block Model (**MSBM**). The MSBM is a generative model for graphs showing certain prescribed clusters/classes structures via a set of membership parameters $p_{\text{in}}^{(t)}$ and $p_{\text{out}}^{(t)}$, $t = 1, \dots, T$. These parameters designate the edge probabilities: given nodes v_i and v_j the probability of observing an edge between them on layer t is $p_{\text{in}}^{(t)}$ (resp. $p_{\text{out}}^{(t)}$), if v_i and v_j belong to the same (resp. different) cluster/class. Note that, unlike the Labeled Stochastic Block Model [13], the MSBM allows multiple edges between the same pairs of nodes across the layers. For SSL with one layer under the SBM we refer the reader to [14, 24, 28].

We present an analysis in expectation. We consider k clusters/classes $\mathcal{C}_1, \dots, \mathcal{C}_k$ of equal size $|\mathcal{C}| = n/k$. We denote with calligraphic letters the layers of a multilayer graph in expectation $E(\mathbb{G}) = \{E(G^{(1)}, \dots, E(G^{(T)})\}$, i.e. $\mathcal{W}^{(t)}$ is the expected adjacency matrix of the t^{th}-layer. We assume that our multilayer graphs are non-weighted, i.e. edges are zero or one, and hence we have $\mathcal{W}_{ij}^{(t)} = p_{\text{in}}^{(t)}$, (resp. $\mathcal{W}_{ij}^{(t)} = p_{\text{out}}^{(t)}$) for nodes v_i, v_j belonging to the same (resp. different) cluster/class.

In order to grasp how different methods classify the nodes in multilayer graphs following the MSBM we analyze two different settings. In the first setting (Section 4.1) all layers have the same class structure and we study the conditions for different regularizers L_p to correctly predict class labels. We further show that our approach is robust against the presence of noise layers, in the sense that it achieves a small classification error when at least one layer is informative and the remaining layers are potentially just noise. In this setting we distinguish the case where each class has the same amount of initial labels and the case where different classes have different number of labels. In the second setting (Section 4.2) we consider the case where each layer taken alone would lead to a large classification error whereas considering all the layers together can lead to a small classification error.

4.1 Complementary Information Layers

A common assumption in multilayer semi-supervised learning is that at least one layer encodes relevant information in the label prediction task. The next theorem discusses the classification error of the expected power mean Laplacian regularizer in this setting.

Theorem 1. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM with k classes $\mathcal{C}_1, \dots, \mathcal{C}_k$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Assume the same number of labeled nodes are available per class. Then, the solution of* (1) *yields zero test error if and only if*

$$m_p(\boldsymbol{\rho}_{\boldsymbol{\epsilon}}) < 1 + \epsilon\,, \tag{2}$$

where $(\boldsymbol{\rho}_{\boldsymbol{\epsilon}})_t = 1 - (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)}) + \epsilon$, and $t = 1, \dots, T$.

This theorem shows that the power mean Laplacian regularizer allows to correctly classify the nodes if p is such that condition (2) holds. In order to better understand how this condition changes when p varies, we analyze in the next corollary the limit cases $p \to \pm\infty$.

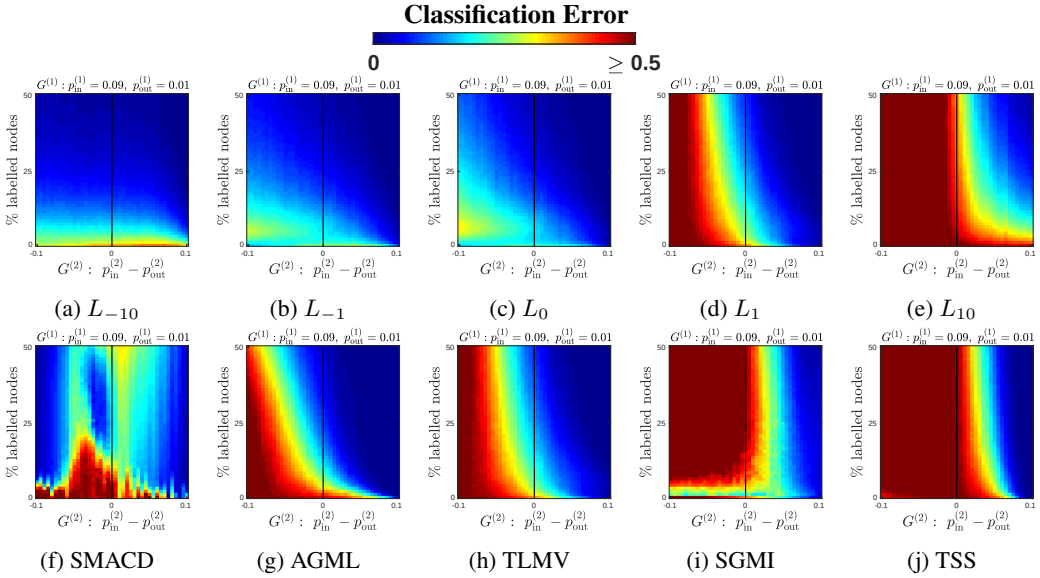
Corollary 1. *Let $E(\mathbb{G})$ be an expected multilayer graph as in Theorem 1. Then,*

- *For $p \to \infty$, the test error is zero if and only if $p_{\text{out}}^{(t)} < p_{\text{in}}^{(t)}$ for all $t = 1, \dots, T$.*
- *For $p \to -\infty$, the test error is zero if and only there exists a $t \in \{1, \dots, T\}$ such that $p_{\text{out}}^{(t)} < p_{\text{in}}^{(t)}$.*

This corollary implies that the limit case $p \to \infty$ requires that *all layers* convey information regarding the clustering/class structure of the multilayer graph, whereas the case $p \to -\infty$ requires that *at least one layer* encodes clustering/class information, and hence it is clear that conditions for the limit $p \to -\infty$ are less restrictive than the conditions for the limit case $p \to \infty$. The next Corollary shows that the smaller the power parameter p is, the less restrictive are the conditions to yield a zero test error.

Corollary 2. *Let $E(\mathbb{G})$ be an expected multilayer graph as in Theorem 1. Let $p \leq q$. If $\mathcal{L}_q$ yields zero test error, then $\mathcal{L}_p$ yields a zero test error.*

The previous results show the effectivity of the power mean Laplacian regularizer in expectation. We now present a numerical evaluation based on Theorem 1 and Corollaries 1 and 2 on random



(a) L_{-10} (b) L_{-1} (c) L_0 (d) L_1 (e) L_{10}

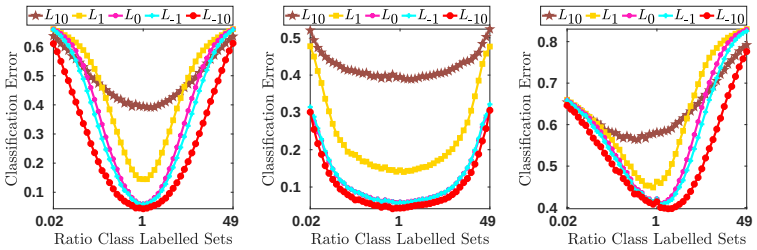
(f) SMACD (g) AGML (h) TLMV (i) SGMI (j) TSS

Figure 1: Average classification error under the Stochastic Block Model computed from 100 runs. **Top Row:** Particular cases with the power mean Laplacian. **Bottom Row:** State of the art models.

graphs sampled from the SBM. The corresponding results are presented in Fig. 1 for classification with regularizers $L_{-10}, L_{-1}, L_0, L_1, L_{10}$ and $\lambda = 1$. We first describe the setting we consider: we generate random multilayer graphs with two layers ($T = 2$) and two classes ($k = 2$) each composed by 100 nodes ($|\mathcal{C}| = 100$). For each parameter configuration $(p_{\text{in}}^{(1)}, p_{\text{out}}^{(1)}, p_{\text{in}}^{(2)}, p_{\text{out}}^{(2)})$ we generate 10 random multilayer graphs and 10 random samples of labeled nodes, yielding a total of 100 runs per parameter configuration, and report the average test error. Our goal is to evaluate the classification performance under different SBM parameters and different amounts of labeled nodes. To this end, we fix the first layer $G^{(1)}$ to be informative of the class structure ($p_{\text{in}}^{(1)} - p_{\text{out}}^{(1)} = 0.08$), i.e. one can achieve a low classification error by taking this layer alone, provided sufficiently many labeled nodes are given. The second layer will go from non-informative (noisy) configurations ($p_{\text{in}}^{(2)} < p_{\text{out}}^{(2)}$, left half of x-axis) to informative configurations ($p_{\text{in}}^{(2)} > p_{\text{out}}^{(2)}$, right half of x-axis), with $p_{\text{in}}^{(t)} + p_{\text{out}}^{(t)} = 0.1$ for both layers. Moreover, we consider different amounts of labeled nodes: going from 1% to 50% (y-axis). The corresponding results are presented in Figs. 1a,1b,1c,1d, and 1e.

In general one can expect a low classification error when both layers $G^{(1)}$ and $G^{(2)}$ are informative (right half of x-axis). We can see that this is the case for all power mean Laplacian regularizers here considered (see top row of Fig. 1). In particular, we can see in Fig. 1e that L_{10} performs well only when **both** layers are informative and completely fails when the second layer is not informative, regardless of the amount of labeled nodes. On the other side we can see in Fig. 1a that L_{-10} achieves in general a low classification error, regardless of the configuration of the second layer $G^{(2)}$, i.e. when $G^{(1)}$ **or** $G^{(2)}$ are informative. Moreover, we can see that overall the areas with low classification error (dark blue) increase when the parameter p decreases, verifying the result from Corollary 2. In the bottom row of Fig. 1 we present the performance of state of the art methods. We can observe that most of them present a classification performance that resembles the one of the power mean Laplacian regularizer L_1. In general their classification performance drops when the level of noise increases, i.e. for non-informative configurations of the second layer $G^{(2)}$, and they are outperformed by the power mean Laplacian regularizer for small values of p.

Unbalanced Class Proportion on Labeled Datasets. In the previous analysis we assumed that we had the same amount of labeled nodes per class. We consider now the case where the number of labeled nodes per class is different. This setting was considered in [38], where the goal was to overcome unbalanced class proportions in labeled nodes. To this end, they propose a Class Mass Normalization (CMN) strategy, whose performance was also tested in [37]. In the following result we show that, provided the ground truth classes have the same size, different amounts of labeled nodes per class affect the conditions in expectation for zero classification error of (1). For simplicity, we consider here only the case of two classes.



(a) Uniform loss (b) Weighted loss (c) CMN

Figure 2: Different class weighted loss strategies. Left to right: uniform loss, weighted loss, and Class Mass Normalization.

Theorem 2. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM with two classes $\mathcal{C}_1, \mathcal{C}_2$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Assume n_1, n_2 nodes from $\mathcal{C}_1, \mathcal{C}_2$ are labeled, respectively. Let $\lambda = 1$. Then* (1) *yields zero test error if*

$$m_p(\boldsymbol{\rho}_{\boldsymbol{\epsilon}}) < \min\left\{\frac{n_1}{n_2}, \frac{n_2}{n_1}\right\} \tag{3}$$

where $(\boldsymbol{\rho}_{\boldsymbol{\epsilon}})_t = 1 - (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)}) + \epsilon$, and $t = 1, \ldots, T$.

Observe that Theorem 2 provides only a sufficient condition. A necessary and sufficient condition for zero test error in terms of p, n_1 and n_2 is given in the supplementary material.

A different objective function can be employed for the case of classes with different number of labels per class. Let C be the diagonal matrix defined by $C_{ii} = n/n_r$, if node v_i has been labeled to belong to class $\mathcal{C}_r$. Consider the following modification of (1)

$$\arg\min_{f \in \mathbb{R}^n} \|f - CY\|^2 + \lambda f^T L_p f \tag{4}$$

The next Theorem shows that using (4) in place of (1) allows us to retrieve the same condition of Theorem 1 for zero test error in expectation in the setting where the number of labeled nodes per class are not equal.

Theorem 3. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM k classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Let $n_1, \ldots, n_k$ be the number of labeled nodes per class. Let $C \in \mathbb{R}^{n \times n}$ be a diagonal matrix with $C_{ii} = n/n_r$ for $v_i \in \mathcal{C}_r$. The solution to* (4) *yields a zero test classification error if and only if*

$$m_p(\boldsymbol{\rho}_{\boldsymbol{\epsilon}}) < 1 + \epsilon\,, \tag{5}$$

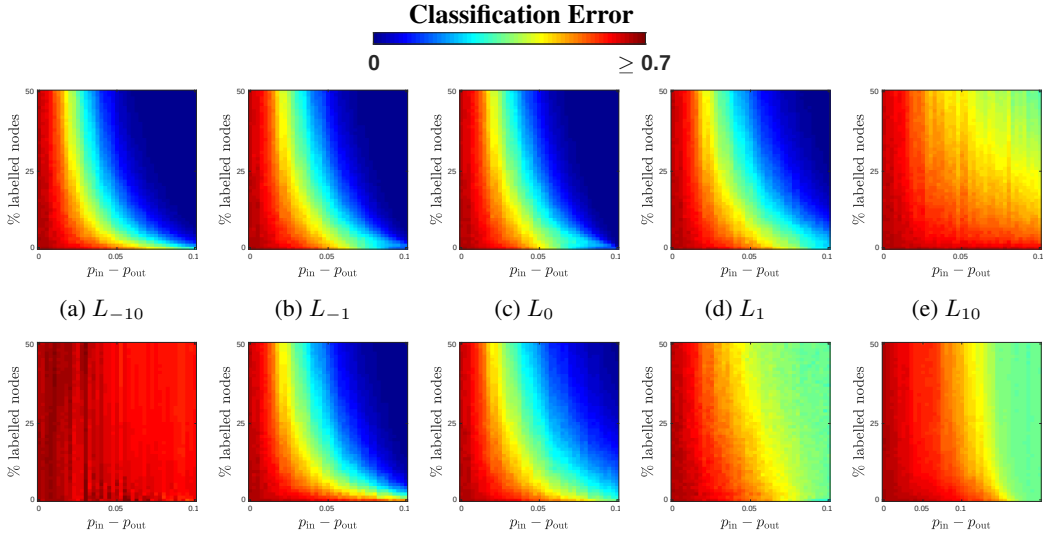
where $(\boldsymbol{\rho}_{\boldsymbol{\epsilon}})_t = 1 - (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)}) + \epsilon$, and $t = 1, \ldots, T$.

In Figs. 2a, 2b, and 2c. we present a numerical experiment with random graphs of our analysis in expectation. We consider the following setting: we generate multilayer graphs with two layers ($T = 2$) and two classes ($k = 2$) each composed by 100 nodes ($|\mathcal{C}| = 100$). We fix $p_{\text{in}}^{(1)} - p_{\text{out}}^{(1)} = 0.08$ and $p_{\text{in}}^{(2)} - p_{\text{out}}^{(2)} = 0$, with $p_{\text{in}}^{(t)} + p_{\text{out}}^{(t)} = 0.1$ for both layers. We fix the total amount of labeled nodes to be $n_1 + n_2 = 50$ and let $n_1, n_2 = 1, \ldots 49$. For each setting we generate 10 multilayer graphs and 10 sets of labeled nodes, yielding a total of 100 runs per setting, and report the average test classification error. In Fig. 2a we can see the performance of the power mean Laplacian regularizer without modifications. We can observe how different proportions of labeled nodes per class affect the performance. In Fig. 2b, we present the performance of the modified approach (4) and observe that it yields a better performance against different class label proportions. Finally in Fig. 2c we present the performance based on Class Mass Normalization [1], where we can see that its effect is slightly skewed to one class and its overall performance is larger than the proposed approach.

4.2 Information-Independent Layers

In the previous section we considered the case where at least one layer had enough information to correctly estimate node class labels. In this section we now consider the case where single layers

[1] We follow the authors' implementation: http://pages.cs.wisc.edu/~jerryzhu/pub/harmonic_function.m



(a) L_{-10} (b) L_{-1} (c) L_0 (d) L_1 (e) L_{10}

(f) SMACD (g) AGML (h) TLMV (i) SGMI (j) TSS

Figure 4: Average test error under the SBM.Multilayer graph with 3 layers and 3 classes.**Top Row:** Particular cases with the power mean Laplacian. **Bottom Row:** State of the art models.

taken alone obtain a large classification error, whereas when all the layers are taken together it is possible to obtain a good classification performance. For this setting we consider multilayer graphs with 3 layers ($T = 3$) and three classes ($k = 3$) $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$, each composed by 100 nodes ($|\mathcal{C}| = 100$) with the following expected adjacency matrix per layer:

$$\mathcal{W}_{i,j}^{(t)} = \begin{cases} p_{\text{in}}, & v_i, v_j \in \mathcal{C}_t \text{ or } v_i, v_j \in \overline{\mathcal{C}_t} \\ p_{\text{out}}, & \text{else} \end{cases} \tag{6}$$

for $t = 1, 2, 3$, i.e. layer $G^{(t)}$ is informative of class $\mathcal{C}_t$ but not of the remaining classes, and hence any classification method using one single layer will provide a poor classification performance. In Fig. 4 we present numerical experiments: for each parameter setting $(p_{\text{in}}, p_{\text{out}})$ we generate 5 multilayer graphs together with 5 samples of labeled nodes yielding a total of 25 runs per setting, and report the average test classification error. Also in this case we observe that the power mean Laplacian regularizer does identify the global class structure and that it leverages the information provided by labeled nodes, particularly for smaller values of p. On the other hand, this is not the case for all other state of the art methods. In fact, we can see that SGMI and TSS performs similarly to L_{10} which has the largest classification error. Moreover, we can see that AGML and TLMV perform similarly to the arithmetic mean of Laplacians L_1, which in turn is outperformed by the power mean Laplacian regularizer L_{-10}. Please see the supplementary material for a more detailed comparison.

5 A Scalable Matrix-free Numerical Method for the System $(I + \lambda L_p)f = Y$

In this section we introduce a matrix-free method for the solution of the system $(I + \lambda L_p)f = Y$ based on contour integrals and Krylov subspace methods. The method exploits the sparsity of the Laplacians of each layer and is matrix-free, in the sense that it requires only to compute the matrix-vector product $L_{\text{sym}}^{(i)} \times vector$, without requiring to store the matrices. Thus, when the layers are sparse, the method scales to large datasets. Observe that this is a critical requirement as L_p is in general a dense matrix, even for very sparse layers, and thus computing and storing L_p is very prohibitive for large multilayer graphs. We present a method for negative integer values $p < 0$, leaving aside the limit case $p \to 0$ as it requires a particular treatment. The following is a brief overview of the proposed approach. Further details are available in the supplementary material.

Let $A_1, \ldots, A_T$ be symmetric positive definite matrices, $\varphi : \mathbb{C} \to \mathbb{C}$ defined by $\varphi(z) = z^{1/p}$ and $L_p = T^{-1/p}\varphi(S_p)$, where $S_p = A_1^p + \cdots + A_T^p$. The proposed method consists of three main steps:

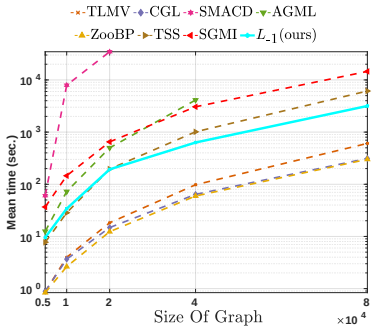


Figure 5: Mean execution time of 10 runs for different methods. L_{-1}(ours) stands for the power mean Laplacian regularizer together with our proposed matrix-free contour integral based method. We generate multilayer graphs with two layers, each with two classes of same size with parameters $p_{\text{in}} = 0.05$ and $p_{\text{in}} = 0.025$ and graphs of of sizes $[0.5, 1, 2, 4, 8] \times 10^4$. Observe that our matrix free approach for L_{-1} (solid blue curve) is competitive to state of the art approaches as TSS[30], outperforming AGML[25], SGMI[15] and SMACD[11]. For TLMV[36] and SGMI we use our own implementation.

1. We solve the system $(I + \lambda L_p)^{-1} Y$ via a Krylov method (e.g. PCG or GMRES) with convergence rate $O((\frac{\kappa^2-1}{\kappa^2})^{h/2})$ [27], where $\kappa = \lambda_{\max}(L_p)/\lambda_{\min}(L_p)$. At iteration h, this method projects the problem onto the Krylov subspace spanned by $\{Y, \lambda L_p Y, (\lambda L_p)^2 Y, \ldots, (\lambda L_p)^h Y\}$, and efficiently solve the projected problem.
2. The previous step requires the matrix-vector product $L_p Y = T^{-1/p} \varphi(S_p) Y$ which we compute by approximating the Cauchy integral form of the function φ with the trapezoidal rule in the complex plane [12]. Taking N suitable contour points and coefficients $\beta_0, \ldots, \beta_N$, we have
$$\varphi_N(S_p) Y = \beta_0 S_p \ \text{Im} \left\{ \sum_{i=1}^{N} \beta_i (z_i^2 I - S_p)^{-1} Y \right\}, \tag{7}$$
which has geometric convergence [12]: $\|\varphi(S_p)Y - \varphi_N(S_p)Y\| = O(e^{-2\pi^2 N/(\ln(M/m)+6)})$, where m, M are such that $M \geq \lambda_{\max}(S_p)$ and $m \leq \lambda_{\min}(S_p)$.
3. The previous step requires to solve linear systems of the form $(zI - S_p)^{-1} Y$. We solve each of these systems via a Krylov subspace method, projecting, at each iteration h, onto the subspace spanned by $\{Y, S_p Y, S_p^2 Y, \ldots, S_p^h Y\}$. Since $S_p = \sum_{i=1}^{T} A_i^{-|p|}$ this problem reduces to computing $|p|$ linear systems with A_i as coefficient matrix, for $i = 1 \ldots, T$. Provided that $A_1, \ldots, A_T$ are sparse matrices, this is done efficiently using pcg with incomplete Cholesky preconditioners.

Notice that the method allows a high level of parallelism. In fact, the N (resp. p) linear systems solvers at step 2 (resp. 3) are independent and can be run in parallel. Moreover, note that the main task of the method is solving linear systems with Laplacian matrices, which can be solved linearly in the number of edges in the corresponding adjacency matrix. Hence, the proposed approach scales to large sparse graphs and is highly parallelizable. A time execution analysis is provided in Fig 5, where we can see that the time execution of our approach is competitive to the state of the art as TSS[30], outperforming AGML[25], SGMI[15] and SMACD[11].

6 Experiments on Real Datasets

In this section we compare the performance of the proposed approach with state of the art methods on real world datasets. We consider the following datasets: *3-sources* [18], which consists of news articles that were covered by news sources BBC, Reuters and Guardian; *BBC*[9] and *BBC Sports*[10] news articles, a dataset of Wikipedia articles with ten different classes [26], the hand written *UCI* digits dataset with six different set of features, and citations datasets *CiteSeer*[19], *Cora*[20] and *WebKB*(Texas)[6]. For each dataset we build the corresponding layer adjacency matrices by taking the symmetric k-nearest neighbour graph using as similarity measure the Pearson linear correlation, (i.e. we take the k neighbours with highest correlation), and take the unweighted version of it. Datasets CiteSeer, Cora and WebKB have only two layers, where the first one is a fixed precomputed citation layer, and the second one is the corresponding k-nearest neighbour graph built from document features.

As **baseline methods** we consider: TSS [30] which identifies an optimal linear combination of graph Laplacians, SGMI [15] which performs label propagation by sparse integration, TLMV [36] which is a weighted arithmetic mean of adjacency matrices, CGL [1] which is a convex combination of the pseudo inverse Laplacian kernel, AGML [25] which is a parameter-free method for optimal graph layer weights, ZooBP [8] which is a fast approximation of Belief Propagation, and SMACD [11] which is a tensor factorization method designed for semi-supervised learning. Finally we set parameters for TSS to ($c = 10, c_0 = 0.4$), SMACD ($\lambda = 0.01$)[2], TLMV ($\lambda = 1$), SGMI

[2]this is the default value in the code released by the authors: https://github.com/egujr001/SMACD

3sources						
	1%	5%	10%	15%	20%	25%
TLMV	29.8	21.5	**20.8**	20.3	15.5	16.5
CGL	50.2	45.5	36.4	30.6	23.8	19.8
SMACD	91.5	91.1	91.2	90.9	90.7	91.3
AGML	**23.9**	26.3	33.9	33.3	26.1	22.0
ZooBP	31.0	21.9	21.3	19.8	15.0	15.3
TSS	29.8	23.9	33.1	34.6	34.8	35.0
SGMI	34.4	26.6	25.4	24.4	19.1	17.9
L_1	33.5	23.9	23.4	20.1	15.6	**14.6**
$L_{\text{-}1}$	28.4	**20.0**	21.8	22.0	17.2	17.9
$L_{\text{-}10}$	40.9	29.1	21.9	**19.3**	**14.8**	14.7

BBC						
	1%	5%	10%	15%	20%	25%
TLMV	**29.0**	19.3	13.2	11.1	9.3	8.8
CGL	72.5	52.3	36.1	27.4	22.0	17.1
SMACD	74.4	73.5	72.8	72.6	72.5	72.4
AGML	60.0	34.2	18.6	13.1	11.0	9.5
ZooBP	31.1	20.1	15.0	12.2	10.0	9.1
TSS	40.4	26.1	20.9	20.1	19.8	19.7
SGMI	37.6	28.9	24.9	22.8	20.7	19.3
L_1	31.3	22.8	17.4	13.5	10.2	8.9
$L_{\text{-}1}$	31.0	**17.0**	**11.5**	**10.5**	**9.2**	**8.7**
$L_{\text{-}10}$	51.6	26.9	16.6	12.8	10.3	9.5

BBCS						
	1%	5%	10%	15%	20%	25%
TLMV	25.6	12.6	10.5	7.5	6.4	5.4
CGL	79.2	51.6	34.9	23.4	16.5	12.7
SMACD	77.8	80.6	82.4	96.4	98.4	98.3
AGML	34.6	17.4	12.1	7.0	6.0	5.4
ZooBP	33.8	13.9	11.3	8.8	7.6	6.2
TSS	23.9	13.2	14.1	12.3	13.1	12.2
SGMI	31.9	19.6	16.6	15.5	14.8	12.1
L_1	29.9	15.0	13.5	10.6	8.7	7.2
$L_{\text{-}1}$	**23.8**	**11.6**	**8.7**	**6.3**	**5.8**	**5.1**
$L_{\text{-}10}$	48.7	22.5	14.2	9.1	7.8	6.1

Wikipedia						
	1%	5%	10%	15%	20%	25%
TLMV	65.7	56.8	46.4	43.1	40.8	39.2
CGL	87.3	83.0	82.5	82.2	83.0	83.0
SMACD	85.4	85.6	85.4	85.3	86.8	90.0
AGML	71.3	66.6	48.1	42.1	38.4	37.3
ZooBP	67.6	58.0	47.0	43.8	41.2	39.8
TSS	87.7	84.7	83.3	81.9	82.3	81.4
SGMI	69.3	84.8	84.5	83.8	83.2	82.8
L_1	68.2	61.1	53.6	48.3	44.1	42.3
$L_{\text{-}1}$	**59.1**	**52.3**	**40.2**	**36.3**	**35.1**	**34.1**
$L_{\text{-}10}$	66.9	57.2	43.2	38.7	36.3	34.9

UCI						
	1%	5%	10%	15%	20%	25%
TLMV	28.9	20.4	16.3	14.4	13.7	12.7
CGL	81.8	64.0	54.6	49.1	46.7	46.7
SMACD	73.6	81.0	90.0	90.0	86.2	81.9
AGML	25.3	17.2	15.2	13.2	12.5	12.0
ZooBP	30.8	21.7	17.6	15.1	14.1	13.0
TSS	**24.0**	17.6	16.6	15.9	15.8	15.6
SGMI	36.0	44.4	50.9	50.4	50.2	48.8
L_1	31.3	23.8	18.7	15.6	14.4	13.2
$L_{\text{-}1}$	30.5	**17.1**	**13.8**	**12.6**	**12.3**	**11.9**
$L_{\text{-}10}$	57.0	33.8	23.7	17.6	15.3	13.4

Citeseer						
	1%	5%	10%	15%	20%	25%
TLMV	51.5	39.4	36.5	33.7	31.6	30.3
CGL	89.3	71.8	58.0	49.8	44.5	40.9
SMACD	90.7	90.4	67.0	65.5	66.8	68.9
AGML	**47.3**	**32.3**	**29.6**	**28.2**	**27.5**	**27.0**
ZooBP	63.6	41.9	38.7	35.8	33.8	32.2
TSS	58.5	49.5	45.9	42.1	39.8	38.4
SGMI	59.4	46.8	44.0	42.3	40.5	39.2
L_1	56.3	44.1	41.2	38.5	36.1	34.7
$L_{\text{-}1}$	52.4	39.0	35.6	32.6	30.9	29.5
$L_{\text{-}10}$	68.6	54.6	48.5	43.0	39.7	37.2

Cora						
	1%	5%	10%	15%	20%	25%
TLMV	46.0	34.1	28.8	25.8	22.5	20.6
CGL	85.5	70.1	56.5	49.1	44.2	40.0
SMACD	75.6	76.7	78.7	78.7	81.0	87.1
AGML	54.7	36.0	25.4	**20.7**	**18.1**	**16.5**
ZooBP	54.7	38.0	32.9	30.2	27.6	26.2
TSS	**38.8**	**27.7**	**24.1**	21.5	20.0	19.1
SGMI	57.3	47.7	43.0	41.8	40.1	38.5
L_1	50.7	38.2	33.4	31.2	28.2	25.6
$L_{\text{-}1}$	43.2	31.8	24.5	21.1	18.8	17.2
$L_{\text{-}10}$	62.0	46.3	35.4	29.4	25.2	22.3

WebKB						
	1%	5%	10%	15%	20%	25%
TLMV	58.6	49.4	45.6	47.2	47.6	48.2
CGL	80.4	82.4	84.4	86.9	82.7	89.2
SMACD	87.3	87.2	87.2	87.4	87.8	87.8
AGML	56.5	50.3	46.8	44.7	47.6	46.8
ZooBP	52.0	45.0	38.7	38.5	**36.4**	**33.5**
TSS	60.9	51.0	50.5	47.3	49.2	48.7
SGMI	**44.9**	**39.7**	41.9	**34.9**	40.3	52.5
L_1	58.5	49.0	44.8	44.3	44.5	44.4
$L_{\text{-}1}$	49.9	45.5	40.7	39.5	39.9	40.3
$L_{\text{-}10}$	52.3	41.9	**38.0**	38.1	36.8	39.5

Table 2: Experiments in real datasets. Notation: **best** performances are marked with bold fonts and gray background and second best performances with only gray background.

($\lambda_1 = 1, \lambda_2 = 10^{-3}$) and $\lambda = 0.1$ for L_1 and $\lambda = 10$ for L_{-1} and L_{-10}. We do not perform cross validation in our experimental setting due to the large execution time in some of the methods here considered. Hence we fix the parameters for each method in all experiments.

We fix nearest neighbourhood size to $k = 10$ and generate 10 samples of labeled nodes, where the percentage of labeled nodes per class is in the range $\{1\%, 5\%, 10\%, 15\%, 20\%, 25\%\}$. The average test errors are presented in table 2, where the **best** (resp. second best) performances are marked with bold fonts and gray background (resp. with only gray background). We can see that the first and second best positions are in general taken by the power mean Laplacian regularizers L_1, L_{-1}, L_{-10}, being clear for all datasets except with 3-sources. Moreover we can see that in 77% of all cases L_{-1} presents either the best or the second best performance, further verifying that our proposed approach based on the power mean Laplacian for semi-supervised learning in multilayer graph is a competitive alternative to state of the art methods[3].

[3]Communications with the authors of [11] could not clarify the bad performance of SMACD.

Acknowledgement P.M and M.H are supported by the DFG Cluster of Excellence "Machine Learning – New Perspectives for Science", EXC 2064/1, project number 390727645

References

[1] A. Argyriou, M. Herbster, and M. Pontil. Combining graph Laplacians for semi–supervised learning. In *NeurIPS*, 2006.

[2] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, 2004.

[3] K. V. Bhagwat and R. Subramanian. Inequalities between means of positive operators. *Mathematical Proceedings of the Cambridge Philosophical Society*, 83(3):393–401, 1978.

[4] P. S. Bullen. *Handbook of means and their inequalities*, volume 560. Springer Science & Business Media, 2013.

[5] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 2010.

[6] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In *AAAI*, 2011.

[7] T. A. Driscoll. Algorithm 843: improvements to the schwarz-christoffel toolbox for matlab. *ACM Transactions on Mathematical Software (TOMS)*, 31(2):239–251, 2005.

[8] D. Eswaran, S. Günnemann, C. Faloutsos, D. Makhija, and M. Kumar. Zoobp: Belief propagation for heterogeneous networks. In *VLDB*, 2017.

[9] D. Greene and P. Cunningham. Producing accurate interpretable clusters from high-dimensional data. In *PKDD*, 2005.

[10] D. Greene and P. Cunningham. A matrix factorization approach for integrating multiple data views. In *ECML PKDD*, 2009.

[11] E. Gujral and E. E. Papalexakis. SMACD: Semi-supervised multi-aspect community detection. In *SDM*, 2018.

[12] N. Hale, N. J. Higham, and L. N. Trefethen. Computing A^{α}, $\log(A)$, and related matrix functions by contour integrals. *SIAM Journal on Numerical Analysis*, 46(5):2505–2523, 2008.

[13] S. Heimlicher, M. Lelarge, and L. Massoulié. Community detection in the labelled stochastic block model. *arXiv:1209.2910*, 2012.

[14] V. Kanade, E. Mossel, and T. Schramm. Global and local information in clustering labeled block models. *IEEE Transactions on Information Theory*, 62(10):5906–5917, 2016.

[15] M. Karasuyama and H. Mamitsuka. Multiple graph label propagation by sparse integration. *IEEE Transactions on Neural Networks and Learning Systems*, 24(12):1999–2012, 2013.

[16] T. Kato, H. Kashima, and M. Sugiyama. Robust label propagation on multiple networks. *Transactions on Neural Networks*, 20(1):35–44, Jan. 2009.

[17] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[18] J. Liu, C. Wang, J. Gao, and J. Han. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*, 2013.

[19] Q. Lu and L. Getoor. Link-based classification. In *ICML*, 2003.

[20] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[21] P. Mercado, A. Gautier, F. Tudisco, and M. Hein. The power mean Laplacian for multilayer graph clustering. In *AISTATS*, 2018.

[22] P. Mercado, F. Tudisco, and M. Hein. Clustering signed networks with the geometric mean of Laplacians. In *NeurIPS*. 2016.

[23] P. Mercado, F. Tudisco, and M. Hein. Spectral clustering of signed graphs via matrix power means. In *ICML*, 2019.

[24] E. Mossel and J. Xu. Local algorithms for block models with side information. In *ITCS*, 2016.

[25] F. Nie, J. Li, and X. Li. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *IJCAI*, 2016.

[26] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM Multimedia*, 2010.

[27] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[28] A. Saade, F. Krzakala, M. Lelarge, and L. Zdeborová. Fast randomized semi-supervised clustering. *Journal of Physics: Conference Series*, 1036:012015, 2018.

[29] A. Subramanya and P. P. Talukdar. *Graph-Based Semi-Supervised Learning*. Morgan & Claypool Publishers, 2014.

[30] K. Tsuda, H. Shin, and B. Schölkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21(2):59–65, 2005.

[31] K. Viswanathan, S. Sachdeva, A. Tomkins, and S. Ravi. Improved semi-supervised learning with multiple graphs. In *AISTATS*, 2019.

[32] J. H. Wilkinson. *The algebraic eigenvalue problem*, volume 662. Oxford Clarendon, 1965.

[33] Z. Yang, W. W. Cohen, and R. Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, 2016.

[34] J. Ye and L. Akoglu. Robust semi-supervised learning on multiple networks with noise. In *PKDD*, 2018.

[35] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NeurIPS*, 2003.

[36] D. Zhou and C. J. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, 2007.

[37] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.

[38] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

Appendices

This section contains all the proofs of results mentioned in the main paper, together with a detailed description of the proposed numerical scheme. It is organized as follows:

- Section A contains auxiliary results,
- Section B contains the proof of Theorem 1,
- Section C contains the proof of Corollary 1,
- Section D contains the proof of Corollary 2,
- Section E contains general version of Theorem 2
- Section F contains the proof of Theorem 2,
- Section G contains the proof of Theorem 3.
- Section H contains a detailed exposition of the numerical scheme presented in Section 5.
- Section I we present a numerical analysis for the case of three layers and three classes as in Sec. 4.2
- Section J presents a numerical analysis of the effect of the regularization parameter λ

A Auxiliary Results

We first present some results that will be useful.

The following theorem states the monotonicity of the scalar power mean.

Theorem 4 ([4], Ch. 3, Thm. 1)**.** *Let $p < q$ then $m_p(a,b) \leq m_q(a,b)$ with equality if and only if $a = b$.*

The following lemma shows the effect of the matrix power mean when matrices have a common eigenvector.

Lemma 1 ([21])**.** *Let $\mathbf{u}$ be an eigenvector of $A_1, \ldots, A_T$ with corresponding eigenvalues $\lambda_1, \ldots, \lambda_T$. Then $\mathbf{u}$ is an eigenvector of $M_p(A_1, \ldots, A_T)$ with eigenvalue $m_p(\lambda_1, \ldots, \lambda_T)$.*

The following Lemma states the eigenvalues and eigenvectors of expected adjacency matrices according to the Stochastic Block Model here considered.

Lemma 2. *Let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ be clusters of equal size $|\mathcal{C}| = n/k$. Let $\mathcal{W} \in \mathbb{R}^{n \times n}$ be defined as*

$$\mathcal{W} = (p_{\text{in}} - p_{\text{out}}) \sum_{i=1}^{k} \mathbf{1}_{\mathcal{C}_i} \mathbf{1}_{\mathcal{C}_i}^T + p_{\text{out}} \mathbf{1}\mathbf{1}^T \tag{8}$$

and let $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k \in \mathbb{R}^n$ be defined as

$$\boldsymbol{\chi}_1 = \mathbf{1}, \qquad \boldsymbol{\chi}_r = \sum_{j=1}^{r} \mathbf{1}_{\mathcal{C}_j} - r\mathbf{1}_{\mathcal{C}_r} \tag{9}$$

for $r = 2, \ldots, k$. Then, $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k$ are orthogonal eigenvectors of $\mathcal{W}$, with eigenvalues

$$\lambda_1 = |\mathcal{C}| \left(p_{\text{in}} + (k-1)p_{\text{out}}\right), \qquad \lambda_r = |\mathcal{C}| \left(p_{\text{in}} - p_{\text{out}}\right) \tag{10}$$

Proof. Please note that from the definition that the matrix $\mathcal{W}$ is equal to p_{in} in the block diagonal and p_{out} elsewhere. We first consider the following matrix vector products that can be easily verified:

$$\mathcal{W}\mathbf{1} = |\mathcal{C}| \left(p_{\text{in}} + (k-1)p_{\text{out}}\right)\mathbf{1} \tag{11}$$

$$\mathcal{W}\mathbf{1}_{\mathcal{C}_i} = |\mathcal{C}| \left(p_{\text{in}}\mathbf{1}_{\mathcal{C}_i} + p_{\text{out}}\mathbf{1}_{\overline{\mathcal{C}_i}}\right) \tag{12}$$

Moreover, we can see that

$$\begin{aligned}
\mathcal{W}\left(\mathbf{1}_{\mathcal{C}_j} - \mathbf{1}_{\mathcal{C}_i}\right) &= |\mathcal{C}| \left(\left(p_{\text{in}}\mathbf{1}_{\mathcal{C}_j} + p_{\text{out}}\mathbf{1}_{\overline{\mathcal{C}}_j}\right) - \left(p_{\text{in}}\mathbf{1}_{\mathcal{C}_i} + p_{\text{out}}\mathbf{1}_{\overline{\mathcal{C}}_i}\right)\right) \\
&= |\mathcal{C}| \left(p_{\text{in}}\left(\mathbf{1}_{\mathcal{C}_j} - \mathbf{1}_{\mathcal{C}_i}\right) + p_{\text{out}}\left(\mathbf{1}_{\overline{\mathcal{C}}_j} - \mathbf{1}_{\overline{\mathcal{C}}_i}\right)\right) \\
&= |\mathcal{C}| \left(p_{\text{in}}\left(\mathbf{1}_{\mathcal{C}_j} - \mathbf{1}_{\mathcal{C}_i}\right) - p_{\text{out}}\left(\mathbf{1}_{\mathcal{C}_i} - \mathbf{1}_{\mathcal{C}_j}\right)\right) \\
&= |\mathcal{C}| \left(p_{\text{in}} - p_{\text{out}}\right)\left(\mathbf{1}_{\mathcal{C}_j} - \mathbf{1}_{\mathcal{C}_i}\right)
\end{aligned}$$

Now we show that $\boldsymbol{\chi}_2, \ldots, \boldsymbol{\chi}_k$ are eigenvectors of $\mathcal{W}$.

$$\begin{aligned}
\mathcal{W}\boldsymbol{\chi}_r &= \mathcal{W}\left(\sum_{j=1}^{r} \mathbf{1}_{\mathcal{C}_j} - r\mathbf{1}_{\mathcal{C}_r}\right) \\
&= \mathcal{W}\sum_{j=1}^{r}(\mathbf{1}_{\mathcal{C}_j} - \mathbf{1}_{\mathcal{C}_r}) \\
&= \sum_{j=1}^{r}\mathcal{W}\left(\mathbf{1}_{\mathcal{C}_j} - \mathbf{1}_{\mathcal{C}_r}\right) \\
&= \sum_{j=1}^{r}|\mathcal{C}|\left(p_{\text{in}} - p_{\text{out}}\right)\left(\mathbf{1}_{\mathcal{C}_j} - \mathbf{1}_{\mathcal{C}_r}\right) \\
&= |\mathcal{C}|\left(p_{\text{in}} - p_{\text{out}}\right)\sum_{j=1}^{r}\left(\mathbf{1}_{\mathcal{C}_j} - \mathbf{1}_{\mathcal{C}_r}\right) \\
&= |\mathcal{C}|\left(p_{\text{in}} - p_{\text{out}}\right)\left(\sum_{j=1}^{r}\mathbf{1}_{\mathcal{C}_j} - r\mathbf{1}_{\mathcal{C}_r}\right) \\
&= |\mathcal{C}|\left(p_{\text{in}} - p_{\text{out}}\right)\boldsymbol{\chi}_r \\
&= \lambda_r\boldsymbol{\chi}_r
\end{aligned}$$

Furthermore, we can see that eigenvectors $\boldsymbol{\chi}_2, \ldots, \boldsymbol{\chi}_k$ are orthogonal. Let $2 \leq r < s \leq k$, then

$$\begin{aligned}
\boldsymbol{\chi}_r^T\boldsymbol{\chi}_s &= \left(\sum_{j_1=1}^{r}\mathbf{1}_{\mathcal{C}_{j_1}} - r\mathbf{1}_{\mathcal{C}_r}\right)^T\left(\sum_{j_2=1}^{s}\mathbf{1}_{\mathcal{C}_{j_2}} - s\mathbf{1}_{\mathcal{C}_s}\right) \\
&= \sum_{j_1=1}^{r}\sum_{j_2=1}^{s}\mathbf{1}_{\mathcal{C}_{j_1}}^T\mathbf{1}_{\mathcal{C}_{j_2}} - s\sum_{j_1=1}^{r}\mathbf{1}_{\mathcal{C}_{j_1}}^T\mathbf{1}_{\mathcal{C}_s} - r\sum_{j_2=1}^{s}\mathbf{1}_{\mathcal{C}_{j_2}}^T\mathbf{1}_{\mathcal{C}_r} + rs\mathbf{1}_{\mathcal{C}_r}^T\mathbf{1}_{\mathcal{C}_s} \\
&= \sum_{j_1=1}^{r}\sum_{j_2=1}^{s}\mathbf{1}_{\mathcal{C}_{j_1}}^T\mathbf{1}_{\mathcal{C}_{j_2}} - r\sum_{j_2=1}^{s}\mathbf{1}_{\mathcal{C}_{j_2}}^T\mathbf{1}_{\mathcal{C}_r} \\
&= \sum_{j_1=1}^{r}\sum_{j_2=1}^{s}\mathbf{1}_{\mathcal{C}_{j_1}}^T\mathbf{1}_{\mathcal{C}_{j_2}} - r\mathbf{1}_{\mathcal{C}_r}^T\mathbf{1}_{\mathcal{C}_r} \\
&= \sum_{j_1=1}^{r}\sum_{j_2=1}^{s}\left(\mathbf{1}_{\mathcal{C}_{j_1}}^T\mathbf{1}_{\mathcal{C}_{j_2}}\right) - r\left|\mathcal{C}\right| \\
&= \sum_{j_1=1}^{r}\left(\mathbf{1}_{\mathcal{C}_{j_1}}^T\mathbf{1}_{\mathcal{C}_{j_1}}\right) - r\left|\mathcal{C}\right| \\
&= \sum_{j_1=1}^{r}\left|\mathcal{C}\right| - r\left|\mathcal{C}\right| \\
&= r\left|\mathcal{C}\right| - r\left|\mathcal{C}\right| \\
&= 0
\end{aligned}$$

where in the third step we used that fact that $\mathbf{1}_{\mathcal{C}_r}^T\mathbf{1}_{\mathcal{C}_s} = 0$ as $r < s$, and $\mathbf{1}_{\mathcal{C}_{j_1}}^T\mathbf{1}_{\mathcal{C}_s} = 0$ as $j_1 < s$. Finally, we can see that for $2 \leq r \leq k$

$$\begin{aligned}
\boldsymbol{\chi}_1^T\boldsymbol{\chi}_r &= \mathbf{1}^T\left(\sum_{j=1}^{r}\mathbf{1}_{\mathcal{C}_j} - r\mathbf{1}_{\mathcal{C}_r}\right) \\
&= \sum_{j=1}^{r}\left(\mathbf{1}^T\mathbf{1}_{\mathcal{C}_j}\right) - r\mathbf{1}^T\mathbf{1}_{\mathcal{C}_r}
\end{aligned}$$

$$= r\,|\mathcal{C}| - r\,|\mathcal{C}|$$
$$= 0$$

and hence $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k$ are orthogonal eigenvectors of the matrix $\mathcal{W}$.

□

The following Lemma shows the eigenvectors and eigenvalues of the power mean Laplacian in expectation under the considered Stochastic Block Model.

Lemma 3. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM with k classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Then the eigenvalues of the power mean Laplacian $\mathcal{L}_p$ are*

$$\lambda_1(\mathcal{L}_p) = \varepsilon, \qquad \lambda_i(\mathcal{L}_p) = m_p(\boldsymbol{\rho}_{\boldsymbol{\epsilon}}), \qquad \lambda_j(\mathcal{L}_p) = 1 + \varepsilon \tag{13}$$

with eigenvectors

$$\boldsymbol{\chi}_1 = \mathbf{1}, \qquad \boldsymbol{\chi}_i = \sum_{j=1}^{i} \mathbf{1}_{\mathcal{C}_j} - i\mathbf{1}_{\mathcal{C}_i}$$

where $(\boldsymbol{\rho}_{\boldsymbol{\epsilon}})_t = 1 - (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)}) + \epsilon$, $t = 1, \ldots, T$, $i = 2, \ldots, k$, and $j = k+1, \ldots, |V|$,

Proof. From Lemma 2 we know that $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_k$ are eigenvectors of $\mathcal{W}^{(1)}, \ldots, \mathcal{W}^{(T)}$. In particular, we have seen that

$$\lambda_1^{(t)} = |\mathcal{C}|\,(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)}), \quad \lambda_i^{(t)} = |\mathcal{C}|\,(p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})$$

for $i = 2, \ldots, k$. Further, as matrices $\mathcal{W}^{(1)}, \ldots, \mathcal{W}^{(T)}$ share all their eigenvectors, they are simultaneously diagonalizable, i.e. there exists a non-singular matrix Σ such that $\Sigma^{-1}\mathcal{W}^{(t)}\Sigma = \Lambda^{(t)}$, where $\Lambda^{(t)}$ are diagonal matrices $\Lambda^{(t)} = \text{diag}(\lambda_1^{(t)}, \ldots, \lambda_k^{(t)}, 0, \ldots, 0)$.

As we assume that all clusters are of the same size $|\mathcal{C}|$, the expected layer graphs are regular graphs with degrees $d^{(1)}, \ldots, d^{(T)}$. Hence, the normalized Laplacians of the expected layer graphs can be expressed as

$$\mathcal{L}_{\text{sym}}^{(t)} = \Sigma(I - \frac{1}{d^{(t)}}\Lambda^{(t)})\Sigma^{-1}$$

Thus, we can observe that

$$\lambda_1^{(t)}(\mathcal{L}_{\text{sym}}^{(t)}) = 0, \qquad \lambda_i^{(t)}(\mathcal{L}_{\text{sym}}^{(t)}) = 1 - \rho^{(t)}, \qquad \lambda_j^{(t)}(\mathcal{L}_{\text{sym}}^{(t)}) = 1,$$

for $i = 2, \ldots, k$, and $j = k+1, \ldots, |V|$, where

$$\rho^{(t)} = (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)})$$

for $t = 1, \ldots, T$. By obtaining the power mean Laplacian on diagonally shifted matrices,

$$\mathcal{L}_p = M_p(\mathcal{L}_{\text{sym}}^{(1)} + \varepsilon I, \ldots, \mathcal{L}_{\text{sym}}^{(1)} + \varepsilon I)$$

we have by Lemma 1

$$\begin{aligned}
\lambda_1(\mathcal{L}_p) &= m_p(\lambda_1^{(1)} + \varepsilon, \ldots, \lambda_1^{(T)} + \varepsilon) = \varepsilon \\
\lambda_i(\mathcal{L}_p) &= m_p(1 - \rho^{(1)} + \varepsilon, \ldots, 1 - \rho^{(T)} + \varepsilon) = m_p(\boldsymbol{\rho}_{\boldsymbol{\epsilon}}) \\
\lambda_j(\mathcal{L}_p) &= m_p(\lambda_j^{(1)} + \varepsilon, \ldots, \lambda_j^{(T)} + \varepsilon) = 1 + \varepsilon
\end{aligned} \tag{14}$$

where $(\boldsymbol{\rho}_{\boldsymbol{\epsilon}})_t = 1 - (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)}) + \epsilon$, and $t = 1, \ldots, T$, $i = 2, \ldots, k$, and $j = k+1, \ldots, |V|$, □

The following Lemma describes the general form of the solution matrix

$$F = (f^{(1)}, \dots, f^{(k)})$$

where the columns of F are obtained from the following optimization problem

$$f^{(r)} = \arg\min_{f \in \mathbb{R}^n} \|f - CY^{(r)}\|^2 + \lambda f^T L_p f$$

Observe that this setting contains as a particular case the problem described in Eq. (1).

Lemma 4. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM with k classes $\mathcal{C}_1, \dots, \mathcal{C}_k$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Let $\boldsymbol{\rho_\epsilon}$ be defined as in Lemma 3. Let $n_1, \dots, n_k$ be the number of labeled nodes per class. Let $C \in \mathbb{R}^{n \times n}$ be a diagonal matrix with $C_{ii} = c_r$ for $v_i \in \mathcal{C}_r$. Let $l(v_i)$ be the label of node v_i, i.e. $l(v_i) = r$ if and only if $v_i \in \mathcal{C}_r$. Let the solution matrix $F = (f^{(1)}, \dots, f^{(k)})$ where*

$$f^{(r)} = \arg\min_{f \in \mathbb{R}^n} \|f - CY^{(r)}\|^2 + \mu f^T \mathcal{L}_p f$$

Then the solution matrix F is such that:

- *If $r < l(v_i)$, then*

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1 - l(v_i)) \frac{1}{\left\|\boldsymbol{\chi}_{l(v_i)}\right\|^2} + \sum_{j=l(v_i)+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

- *If $r > l(v_i)$, then*

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1 - r) \frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

- *If $r = l(v_i)$, then*

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1 - r)^2 \frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

where $\alpha = \frac{1}{1+\mu\epsilon} - \frac{1}{1+\mu(1+\epsilon)}$, and $\beta = \frac{1}{1+\mu m_p(\boldsymbol{\rho_\epsilon})} - \frac{1}{1+\mu(1+\epsilon)}$.

Proof. Let $U \in \mathbb{R}^{n \times n}$ be an orthonormal matrix such that $U = (u_1, u_2, \dots, u_n)$, with $u_i = \boldsymbol{\chi}_i / \|\boldsymbol{\chi}_i\|$ for $i = 1, \dots, k$, where $\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_k$ are eigenvectors of the power mean Laplacian as described in Lemma 3.

The power mean Laplacian $\mathcal{L}_p$ is a symmetric positive semidefinite matrix (see Lemma 3) and hence we can express $\mathcal{L}_p$ as $U\Lambda U^T$ where Λ is a diagonal matrix with entries $\Lambda_{ii} = \lambda_i(\mathcal{L}_p)$, with $i = 1, \dots, n$. Hence, we can see that

$$(I + \mu\mathcal{L}_p)^{-1} = (I + U\Lambda U^T)^{-1} = (U(I + \Lambda)U^T)^{-1} = U(I + \Lambda)^{-1}U^T = U\Omega U^T$$

where Ω is a diagonal matrix with entries $\Omega_{ii} = \frac{1}{1+\mu\lambda_i}$, with $i = 1, \dots, n$.

From Lemma 3 we know that $\lambda_{k+1} = \cdots = \lambda_n = 1 + \epsilon =: \widehat{\omega}$, and hence it follows that $\Omega_{ii} = \frac{1}{1+\mu\widehat{\omega}}$ for $i = k+1, \dots, n$. Moreover, we can express Ω as the sum of two diagonal matrices, i.e.

$$\Omega = \omega I + \Theta$$

where $\omega = \frac{1}{1+\mu\widehat{\omega}}$ and $\Theta = \operatorname{diag}(\Omega_{11} - \omega, \dots, \Omega_{kk} - \omega, 0, \dots, 0)$. Observe that $\Theta_{11} = \Omega_{11} - \omega = \frac{1}{1+\mu\epsilon} - \frac{1}{1+\mu(1+\epsilon)} =: \alpha$ and $\Theta_{jj} = \Omega_{jj} - \omega = \frac{1}{1+\mu m_p(\boldsymbol{\rho_\epsilon})} - \frac{1}{1+\mu(1+\epsilon)} =: \beta$, for $j = 2, \dots, k$. Recall that we are interested in the equation

$$F = (I + \mu\mathcal{L}_p)^{-1}CY = U\Omega U^T CY \in \mathbb{R}^{n \times k},$$

where each column of $Y = [y^{(1)}, \ldots, y^{(k)}]$ is a class indicator of labeled nodes, i.e.

$$y_i^{(j)} = \begin{cases} 1 & \text{if } \; l(v_i) = j \\ 0 & \text{else} \end{cases} \tag{15}$$

Hence, each column of Y can be expressed as

$$y^{(j)} = \sum_{v_i \in V | l(v_i) = j} e_i \tag{16}$$

where $e_i \in \mathbb{R}^n$ and $(e_i)_i = 1$ and zero else. With this in mind, we now study the matrix-vector product $U\Omega U^T e_i$. Recall that $U\Theta U^T$ is a k-rank matrix. Hence we have

$$\begin{aligned} U\Omega U^T e_i &= U(\omega I + \Theta)U^T e_i \\ &= \omega e_i + U\Theta U^T e_i \\ &= \omega e_i + \left(\sum_{j=1}^{k} \Theta_{jj} u_j u_j^T \right) e_i \\ &= \omega e_i + \left(\sum_{j=1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \Theta_{jj} \boldsymbol{\chi}_j \boldsymbol{\chi}_j^T \right) e_i \\ &= \omega e_i + \frac{1}{n}\Theta_{11}\boldsymbol{\chi}_1 + \left(\sum_{j=2}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \Theta_{jj} \boldsymbol{\chi}_j \boldsymbol{\chi}_j^T \right) e_i \\ &= \omega e_i + \frac{1}{n}\alpha\boldsymbol{\chi}_1 + \beta \left(\sum_{j=2}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \boldsymbol{\chi}_j \boldsymbol{\chi}_j^T \right) e_i \end{aligned}$$

where in the last steps we used the fact that $\boldsymbol{\chi}_1^T e_i = \mathbf{1}^T e_i = 1$, and define $\alpha = \Theta_{11}$ and $\beta = \Theta_{jj}$ due to the fact that Θ_{jj} are all equal for $j = 2, \ldots, k$.

The remaining terms $\boldsymbol{\chi}_j \boldsymbol{\chi}_j^T e_i$ depend on the cluster to which the corresponding node v_i belongs to.

We first study the vector product $\boldsymbol{\chi}_r^T e_i$. Observe that

$$\boldsymbol{\chi}_r^T e_i = \left(\sum_{j=1}^{r} \mathbf{1}_{\mathcal{C}_j} - r\mathbf{1}_{\mathcal{C}_r} \right)^T e_i = \sum_{j=1}^{r} \left(\mathbf{1}_{\mathcal{C}_j}^T e_i \right) - r\mathbf{1}_{\mathcal{C}_r}^T e_i$$

Recall that $l(v_i)$ is the label of node v_i, i.e. $l(v_i) = r$ if and only if $v_i \in \mathcal{C}_r$. Then, we have

$$\sum_{j=1}^{r} \left(\mathbf{1}_{\mathcal{C}_j}^T e_i \right) - r\mathbf{1}_{\mathcal{C}_r}^T e_i = \begin{cases} 0 & \text{for} \quad r < l(v_i) \\ 1 - l(v_i) & \text{for} \quad r = l(v_i) \\ 1 & \text{for} \quad r > l(v_i) \end{cases} \tag{17}$$

Therefore,

$$\left(\sum_{j=2}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \boldsymbol{\chi}_j \boldsymbol{\chi}_j^T \right) e_i = (1 - l(v_i)) \frac{\boldsymbol{\chi}_{l(v_i)}}{\left\|\boldsymbol{\chi}_{l(v_i)}\right\|^2} + \sum_{j=l(v_i)+1}^{k} \frac{\boldsymbol{\chi}_j}{\|\boldsymbol{\chi}_j\|^2}$$

All in all we have

$$U\Omega U^T e_i = \omega e_i + \frac{1}{n}\alpha\boldsymbol{\chi}_1 + \beta \left((1 - l(v_i)) \frac{\boldsymbol{\chi}_{l(v_i)}}{\left\|\boldsymbol{\chi}_{l(v_i)}\right\|^2} + \sum_{j=l(v_i)+1}^{k} \frac{\boldsymbol{\chi}_j}{\|\boldsymbol{\chi}_j\|^2} \right)$$

Moreover, the solution matrix F can now be described column-wise as follows

$$
\begin{aligned}
f^{(r)} &= (I + \mu\mathcal{L}_p)^{-1} C y^{(r)} \\
&= c_r \left(\sum_{v_i \in V | l(v_i)=r} U\Omega U^T e_i \right) \\
&= c_r \left(\sum_{v_i \in V | l(v_i)=r} \omega e_i \right) + \frac{1}{n} c_r n_r \alpha \boldsymbol{\chi}_1 + c_r n_r \beta \left((1 - l(v_i)) \frac{\boldsymbol{\chi}_{l(v_i)}}{\left\| \boldsymbol{\chi}_{l(v_i)} \right\|^2} + \sum_{j=l(v_i)+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \right) \\
&= \omega c_r y^{(r)} + c_r n_r \left(\frac{1}{n} \alpha \boldsymbol{\chi}_1 + \beta \left((1 - r) \frac{\boldsymbol{\chi}_r}{\left\| \boldsymbol{\chi}_r \right\|^2} + \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \right) \right)
\end{aligned}
$$

We now study the columns of matrix F. For this, observe that the i^{th} entry of the column corresponding to the class r, is obtained by $f_i^{(r)} = \langle e_i, f^{(r)} \rangle$, and hence have

$$
\begin{aligned}
\langle e_i, f^{(r)} \rangle =& \langle e_i, \omega c_r y^{(r)} + c_r n_r \left(\frac{1}{n} \alpha \boldsymbol{\chi}_1 + \beta \left((1 - r) \frac{\boldsymbol{\chi}_r}{\left\| \boldsymbol{\chi}_r \right\|^2} + \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \right) \right) \rangle \\
=& c_r \frac{n_r}{n} \alpha + c_r n_r \beta \langle e_i, \left((1 - r) \frac{\boldsymbol{\chi}_r}{\left\| \boldsymbol{\chi}_r \right\|^2} c_r + \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \right) \rangle
\end{aligned}
$$

where $\langle e_i, \omega c_r y^{(r)} \rangle = 0$ for unlabeled nodes. Having this, we now proceed to study three different cases of the remaining inner product. We do this by considering the following cases and making use of Eq. (17):

First case: $f_i^{(r)}$ with $r < l(v_i)$. We first analyze the following term

$$
\begin{aligned}
\langle e_i, \left((1 - r) \frac{\boldsymbol{\chi}_r}{\left\| \boldsymbol{\chi}_r \right\|^2} + \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \right) \rangle &= \langle e_i, (1 - r) \frac{\boldsymbol{\chi}_r}{\left\| \boldsymbol{\chi}_r \right\|^2} \rangle + \langle e_i, \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \rangle \\
\text{(by first case of Eq.17)} \qquad &= \langle e_i, \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \rangle \\
\text{(by cases of Eq.17)} \qquad &= (1 - l(v_i)) \frac{1}{\left\| \boldsymbol{\chi}_{l(v_i)} \right\|^2} + \sum_{j=l(v_i)+1}^{k} \frac{1}{\left\| \boldsymbol{\chi}_j \right\|^2}
\end{aligned}
$$

Thus, we have

$$
f_i^{(r)} = c_r \frac{n_r}{n} \alpha + c_r n_r \beta \left((1 - l(v_i)) \frac{1}{\left\| \boldsymbol{\chi}_{l(v_i)} \right\|^2} + \sum_{j=l(v_i)+1}^{k} \frac{1}{\left\| \boldsymbol{\chi}_j \right\|^2} \right)
$$

Second case: $f_i^{(r)}$ with $r > l(v_i)$. We first analyze the following term

$$
\begin{aligned}
\langle e_i, \left((1 - r) \frac{\boldsymbol{\chi}_r}{\left\| \boldsymbol{\chi}_r \right\|^2} + \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \right) \rangle &= \langle e_i, (1 - r) \frac{\boldsymbol{\chi}_r}{\left\| \boldsymbol{\chi}_r \right\|^2} \rangle + \langle e_i, \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\left\| \boldsymbol{\chi}_j \right\|^2} \rangle \\
\text{(by third case of Eq.17)} \qquad &= (1 - r) \frac{1}{\left\| \boldsymbol{\chi}_r \right\|^2} + \sum_{j=r+1}^{k} \frac{1}{\left\| \boldsymbol{\chi}_j \right\|^2}
\end{aligned}
$$

Thus, we have

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1-r)\frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

Third case: $f_i^{(r)}$ **with** $r = l(v_i)$. We first analyze the following term

$$\begin{aligned}\langle e_i, \left((1-r)\frac{\boldsymbol{\chi}_r}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\|\boldsymbol{\chi}_j\|^2} \right) \rangle &= \langle e_i, (1-r)\frac{\boldsymbol{\chi}_r}{\|\boldsymbol{\chi}_r\|^2} \rangle + \langle e_i, \sum_{j=r+1}^{k} \frac{\boldsymbol{\chi}_j}{\|\boldsymbol{\chi}_j\|^2} \rangle \\ \text{(by second case of Eq.17)} \qquad &= (1-r)^2 \frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2}\end{aligned}$$

Thus, we have

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1-r)^2\frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

These three cases are the desired conditions.

□

B Proof Of Theorem 1

Theorem 5. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM with k classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Let the same number of nodes per class be labeled. Then, a zero test classification error is achieved if and only if*

$$m_p(\boldsymbol{\rho_\epsilon}) < 1 + \epsilon\,,$$

where $(\boldsymbol{\rho_\epsilon})_t = 1 - (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)}) + \epsilon$, and $t = 1, \ldots, T$.

Proof. The proof of this theorem builds on top of Lemma 4, where the entries of the solution matrix $F = (f^{(1)}, \ldots, f^{(k)})$ are described, where

$$f^{(r)} = \underset{f \in \mathbb{R}^n}{\arg\min} \|f - CY^{(r)}\|^2 + \mu f^T \mathcal{L}_p f$$

Let $l(v_i)$ be the label of node v_i, i.e. $l(v_i) = r$ if and only if $v_i \in \mathcal{C}_r$. According to Lemma 4 the entries of matrix F for unlabeled nodes are such that

- If $r < l(v_i)$, then

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1-l(v_i))\frac{1}{\left\|\boldsymbol{\chi}_{l(v_i)}\right\|^2} + \sum_{j=l(v_i)+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

- If $r > l(v_i)$, then

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1-r)\frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

- If $r = l(v_i)$, then

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1-r)^2\frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

where $\alpha = \frac{1}{1+\mu\epsilon} - \frac{1}{1+\mu(1+\epsilon)}$, and $\beta = \frac{1}{1+\mu m_p(\boldsymbol{\rho}_\epsilon)} - \frac{1}{1+\mu(1+\epsilon)}$.

Observe that the case here considered corresponds to the case where the amount of labeled data per class is the same, i.e. $n_1 = \cdots = n_k$, and where the matrix C is the identity, i.e. $c_1 = \cdots c_r = 1$.

Moreover, the estimated label assignment for unlabeled nodes goes by the following rule

$$\hat{l}(v_i) = \arg\max\{f_i^{(1)}, \ldots, f_i^{(k)}\}$$

Hence, we need to find conditions so that the following inequality holds

$$f_i^{(j)} < f_i^{(l(v_i))} \qquad \forall\ j \neq l(v_i)$$

Hence, we consider the following two cases:

Case 1: $f_i^{(r)} < f_i^{(l(v_i))}$ for $r > l(v_i)$.

Let $r^* = l(v_i)$, and $r = r^* + \Delta$. Then, we have

$$\begin{aligned}
&f_i^{(r)} < f_i^{(l(v_i))} \Leftrightarrow \\
&f_i^{(r)} < f_i^{(r^*)} \Leftrightarrow \\
&\beta\left((1-r)\frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k}\frac{1}{\|\boldsymbol{\chi}_j\|^2}\right) < \beta\left((1-r^*)^2\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2} + \sum_{j=r^*+1}^{k}\frac{1}{\|\boldsymbol{\chi}_j\|^2}\right) \Leftrightarrow \\
&0 < \beta\left((1-r^*)^2\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2} - (1-r)\frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r^*+1}^{k}\frac{1}{\|\boldsymbol{\chi}_j\|^2} - \sum_{j=r+1}^{k}\frac{1}{\|\boldsymbol{\chi}_j\|^2}\right) \Leftrightarrow \\
&0 < \beta\left((1-r^*)^2\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2} + (r-1)\frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r^*+1}^{k}\frac{1}{\|\boldsymbol{\chi}_j\|^2} - \sum_{j=r^*+\Delta+1}^{k}\frac{1}{\|\boldsymbol{\chi}_j\|^2}\right) \Leftrightarrow \\
&0 < \beta\left((1-r^*)^2\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2} + (r-1)\frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r^*+1}^{r^*+\Delta}\frac{1}{\|\boldsymbol{\chi}_j\|^2}\right) \Leftrightarrow \\
&0 < \beta
\end{aligned}$$

Case 2: $f_i^{(r)} < f_i^{(l(v_i))}$ for $r < l(v_i)$.

Let $r^* = l(v_i)$, and $r^* = r + \Delta$. Then, we have

$$\begin{aligned}
&f_i^{(r)} < f_i^{(l(v_i))} \Leftrightarrow \\
&f_i^{(r)} < f_i^{(r^*)} \Leftrightarrow \\
&\beta\left((1-r^*)\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2} + \sum_{j=r^*+1}^{k}\frac{1}{\|\boldsymbol{\chi}_j\|^2}\right) < \beta\left((1-r^*)^2\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2} + \sum_{j=r^*+1}^{k}\frac{1}{\|\boldsymbol{\chi}_j\|^2}\right) \Leftrightarrow \\
&0 < \beta\left((1-r^*)^2\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2} - (1-r^*)\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2}\right) \\
&0 < \beta\left((1-r^*)^2\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2} + (r^*-1)\frac{1}{\|\boldsymbol{\chi}_{r^*}\|^2}\right) \Leftrightarrow \\
&0 < \beta
\end{aligned}$$

All in all, from the two considered cases we can see that

$$f_i^{(j)} < f_i^{(l(v_i))} \qquad \forall\ j \neq l(v_i) \Longleftrightarrow 0 < \beta$$

In fact,

$$0 < \beta \Leftrightarrow$$

$$
\begin{aligned}
0 < \frac{1}{1+\mu m_p(\boldsymbol{\rho_\epsilon})} - \frac{1}{1+\mu(1+\epsilon)} &\Leftrightarrow \\
\frac{1}{1+\mu(1+\epsilon)} < \frac{1}{1+\mu m_p(\boldsymbol{\rho_\epsilon})} &\Leftrightarrow \\
1+\mu m_p(\boldsymbol{\rho_\epsilon}) < 1+\mu(1+\epsilon) &\Leftrightarrow \\
m_p(\boldsymbol{\rho_\epsilon}) < 1+\epsilon &
\end{aligned}
$$

which is the desired condition. □

C Proof of Corollary 1

Corollary 3. *Let $E(\mathbb{G})$ be an expected multilayer graph as in Theorem 1. Then,*

- *For $p \to \infty$, the classification error is zero if and only if $p_{\text{out}}^{(t)} < p_{\text{in}}^{(t)}$ for all $t = 1, \ldots, T$.*
- *For $p \to -\infty$, the classification error is zero if and only there exists a $t \in \{1, \ldots, T\}$ s.t. $p_{\text{out}}^{(t)} < p_{\text{in}}^{(t)}$.*

Proof. Observe that the limit cases of the scalar power means are

$$
\begin{aligned}
\lim_{p \to -\infty} m_p(x_1, \ldots, x_T) &= \min\{x_1, \ldots, x_T\} \\
\lim_{p \to +\infty} m_p(x_1, \ldots, x_T) &= \max\{x_1, \ldots, x_T\}
\end{aligned}
$$

Applying this to condition

$$
m_p(\boldsymbol{\rho_\epsilon}) < 1+\epsilon\,,
$$

where $(\boldsymbol{\rho_\epsilon})_t = 1 - (p_{\text{in}}^{(t)} - p_{\text{out}}^{(t)})/(p_{\text{in}}^{(t)} + (k-1)p_{\text{out}}^{(t)}) + \epsilon$, and $t = 1, \ldots, T$ yields the desired result.

□

D Proof of Corollary 2

Corollary 4. *Let $E(\mathbb{G})$ be an expected multilayer graph as in Theorem 1. Let $p \leq q$. If $\mathcal{L}_q$ has a zero-classification error, then $\mathcal{L}_p$ has a zero-classification error.*

Proof. By Theorem 4 we have that if $p \leq q$ then $m_p(x_1, \ldots, x_T) \leq m_p(x_1, \ldots, x_T)$. Therefore, applying this to our case we can see that

$$
m_p(\boldsymbol{\rho_\epsilon}) \leq m_q(\boldsymbol{\rho_\epsilon}) < 1+\epsilon
$$

A zero test classification error with parameter q is achieved if and only if $m_q(\boldsymbol{\rho_\epsilon}) < 1+\epsilon$, hence we can see that zero test classification error with parameter p is achieved if it is achieved with parameter q and $p \leq q$.

□

E General version of Theorem 2

Theorem 6. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM with two classes $\mathcal{C}_1, \mathcal{C}_2$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Let n_1, n_2 nodes from classes $\mathcal{C}_1, \mathcal{C}_2$ be labeled, respectively. Let $\mu = 1$. Then, a zero test classification error is achieved if and only if*

$$
m_p(\boldsymbol{\rho_\epsilon}) < \min\left\{ \frac{(n_1+n_2)((1+\epsilon)^2+1) - 2n_2}{2n_2 + (n_1+n_2)\epsilon}, \frac{(n_1+n_2)((1+\epsilon)^2+1) - 2n_1}{2n_1 + (n_1+n_2)\epsilon} \right\}
$$

where $(\boldsymbol{\rho_\epsilon})_l = 1 - (p_{\text{in}}^{(l)} - p_{\text{out}}^{(l)})/(p_{\text{in}}^{(l)} + (k-1)p_{\text{out}}^{(l)}) + \epsilon$, and $l = 1, \ldots, T$.

Proof. The proof of this theorem builds on top of Lemma 4, where the entries of the solution matrix $F = (f^{(1)}, \ldots, f^{(k)})$ are described, where

$$f^{(r)} = \underset{f \in \mathbb{R}^n}{\arg\min} \|f - CY^{(r)}\|^2 + \mu f^T \mathcal{L}_p f$$

Let $l(v_i)$ be the label of node v_i, i.e. $l(v_i) = r$ if and only if $v_i \in \mathcal{C}_r$. According to Lemma 4 the entries of matrix F for unlabeled nodes are such that

- If $r < l(v_i)$, then

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1 - l(v_i)) \frac{1}{\left\|\boldsymbol{\chi}_{l(v_i)}\right\|^2} + \sum_{j=l(v_i)+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

- If $r > l(v_i)$, then

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1 - r) \frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

- If $r = l(v_i)$, then

$$f_i^{(r)} = c_r \frac{n_r}{n}\alpha + c_r n_r \beta \left((1 - r)^2 \frac{1}{\|\boldsymbol{\chi}_r\|^2} + \sum_{j=r+1}^{k} \frac{1}{\|\boldsymbol{\chi}_j\|^2} \right)$$

where $\alpha = \frac{1}{1+\mu\epsilon} - \frac{1}{1+\mu(1+\epsilon)}$, and $\beta = \frac{1}{1+\mu m_p(\boldsymbol{\rho}_\epsilon)} - \frac{1}{1+\mu(1+\epsilon)}$.

Observe that the case here considered corresponds to the case with two classes, i.e. $k = 2$ with equal size classes $\mathcal{C}_1$ and $\mathcal{C}_2$ where the amount of labeled data per class is n_1 and n_2, respectively, with the matrix C as the identity, i.e. $c_1 = c_2 = 1$, and regularization parameter $\mu = 1$.

Moreover, the estimated label assignment for unlabeled nodes goes by the following rule

$$\hat{l}(v_i) = \arg\max\{f_i^{(1)}, f_i^{(2)}\}$$

Hence, we need to find conditions so that the following inequality holds

$$f_i^{(j)} < f_i^{(l(v_i))} \qquad \forall\ j \neq l(v_i)$$

Let $l(v_i) = 1 \Leftrightarrow v_i \in \mathcal{C}_1$, and $l(v_i) = 2 \Leftrightarrow v_i \in \mathcal{C}_2$. A quick computation following Lemma 4 yields

- $f_i^{(1)} = \frac{n_1}{n}\alpha + n_1\beta(\frac{1}{\|\boldsymbol{\chi}_2\|^2})$ for $v_i \in \mathcal{C}_1$, i.e. $l(v_i) = 1$
- $f_i^{(1)} = \frac{n_1}{n}\alpha - n_1\beta(\frac{1}{\|\boldsymbol{\chi}_2\|^2})$ for $v_i \in \mathcal{C}_2$, i.e. $l(v_i) = 2$
- $f_i^{(2)} = \frac{n_2}{n}\alpha - n_2\beta(\frac{1}{\|\boldsymbol{\chi}_2\|^2})$ for $v_i \in \mathcal{C}_1$, i.e. $l(v_i) = 1$
- $f_i^{(2)} = \frac{n_2}{n}\alpha + n_2\beta(\frac{1}{\|\boldsymbol{\chi}_2\|^2})$ for $v_i \in \mathcal{C}_2$, i.e. $l(v_i) = 2$

Observing that $\|\boldsymbol{\chi}_2\|^2 = n$ these conditions can be rephrase as follows

$$f^{(1)} = \frac{n_1}{n}\left((\alpha + \beta)\,\mathbf{1}_{\mathcal{C}} + (\alpha - \beta)\,\mathbf{1}_{\overline{\mathcal{C}}}\right)$$
$$f^{(2)} = \frac{n_2}{n}\left((\alpha - \beta)\,\mathbf{1}_{\mathcal{C}} + (\alpha + \beta)\,\mathbf{1}_{\overline{\mathcal{C}}}\right)$$

Hence, the conditions for correct label assignment of unlabeled nodes are

$$n_1\,(\alpha + \beta) > n_2\,(\alpha - \beta) \text{ and } n_2\,(\alpha + \beta) > n_1\,(\alpha - \beta)$$

Let $\Omega_{11} = \frac{1}{1+\epsilon}, \Omega_{22} = \frac{1}{1+m_p(\boldsymbol{\rho}_\epsilon)}$, and $\omega = \frac{1}{1+(1+\epsilon)}$. Then, $\alpha = \Omega_{11} - \omega$, and $\beta = \Omega_{22} - \omega$.

By studying the first condition we observe

$$
\begin{aligned}
n_1\left(\alpha+\beta\right) &> n_2\left(\alpha-\beta\right) \Leftrightarrow \\
n_1\left(\Omega_{11}-\omega+\Omega_{22}-\omega\right) &> n_2\left(\Omega_{11}-\omega-\left(\Omega_{22}-\omega\right)\right) \Leftrightarrow \\
n_1\left(\Omega_{11}+\Omega_{22}-2\omega\right) &> n_2(\Omega_{11}-\Omega_{22}) \Leftrightarrow \\
(n_1-n_2)\Omega_{11}+(n_1+n_2)\Omega_{22} &> 2n_1\omega \Leftrightarrow \\
\Omega_{22} &> \frac{1}{n_1+n_2}\left(2n_1\omega-(n_1-n_2)\Omega_{11}\right) \Leftrightarrow \\
\frac{1}{1+m_p(\boldsymbol{\rho_\epsilon})} &> \frac{1}{n_1+n_2}\left(2n_1\frac{1}{1+(1+\epsilon)}-(n_1-n_2)\Omega_{11}\right) \Leftrightarrow \\
\frac{1}{1+m_p(\boldsymbol{\rho_\epsilon})} &> \frac{1}{n_1+n_2}\left(2n_1\frac{1}{2+\epsilon}-(n_1-n_2)\frac{1}{1+\epsilon}\right) \Leftrightarrow \\
\frac{1}{1+m_p(\boldsymbol{\rho_\epsilon})} &> \frac{1}{n_1+n_2}\left(\frac{2n_2+(n_1+n_2)\epsilon}{(2+\epsilon)(1+\epsilon)}\right) \Leftrightarrow \\
1+m_p(\boldsymbol{\rho_\epsilon}) &< (n_1+n_2)\left(\frac{(2+\epsilon)(1+\epsilon)}{2n_2+(n_1+n_2)\epsilon}\right) \Leftrightarrow \\
m_p(\boldsymbol{\rho_\epsilon}) &< (n_1+n_2)\left(\frac{(2+\epsilon)(1+\epsilon)}{2n_2+(n_1+n_2)\epsilon}\right)-1 \Leftrightarrow \\
m_p(\boldsymbol{\rho_\epsilon}) &< (n_1+n_2)\left(\frac{(2+\epsilon)(1+\epsilon)-(2n_2+(n_1+n_2)\epsilon)}{2n_2+(n_1+n_2)\epsilon}\right) \Leftrightarrow \\
m_p(\boldsymbol{\rho_\epsilon}) &< \frac{(n_1+n_2)((2+\epsilon)(1+\epsilon)-\epsilon)-2n_2}{2n_2+(n_1+n_2)\epsilon} \Leftrightarrow \\
m_p(\boldsymbol{\rho_\epsilon}) &< \frac{(n_1+n_2)((1+\epsilon)^2+1)-2n_2}{2n_2+(n_1+n_2)\epsilon} \Leftrightarrow
\end{aligned}
$$

The corresponding condition for $\mathcal{C}_2$ can be obtained in a similar way, yielding

$$
m_p(\boldsymbol{\rho_\epsilon}) < \frac{(n_1+n_2)((1+\epsilon)^2+1)-2n_1}{2n_1+(n_1+n_2)\epsilon}
$$

Hence, both conditions hold if and only if

$$
m_p(\boldsymbol{\rho_\epsilon}) = m_p(\boldsymbol{\rho_\epsilon}) < \min\left\{\frac{(n_1+n_2)((1+\epsilon)^2+1)-2n_2}{2n_2+(n_1+n_2)\epsilon}, \frac{(n_1+n_2)((1+\epsilon)^2+1)-2n_1}{2n_1+(n_1+n_2)\epsilon}\right\}
$$

□

F Proof of Theorem 2

Theorem 7. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM with two classes $\mathcal{C}_1, \mathcal{C}_2$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Let n_1, n_2 nodes from clusters $\mathcal{C}_1, \mathcal{C}_2$ be labeled, respectively. Let $\mu = 1$. Then, a zero test classification error is achieved if*

$$
m_p(\boldsymbol{\rho_\epsilon}) < \min\left\{\frac{n_1}{n_2}, \frac{n_2}{n_1}\right\}
$$

where $(\boldsymbol{\rho_\epsilon})_l = 1-(p_{\text{in}}^{(l)}-p_{\text{out}}^{(l)})/(p_{\text{in}}^{(l)}+(k-1)p_{\text{out}}^{(l)})+\epsilon$, and $l = 1, \ldots, T$.

Proof. We first analyze the first condition of the right hand side of Theorem 6. Let $g(\epsilon) = \frac{(n_1+n_2)((1+\epsilon)^2+1)-2n_2}{2n_2+(n_1+n_2)\epsilon}$. Then,

$$
g(0) = \frac{2(n_1+n_2)-2n_2}{2n_2} = \frac{n_1}{n_2}
$$

Moreover, it is clear that g is monotone, as it is quadratic on ϵ on the numerator and linear on the denominator, and hence $g(0) < g(\epsilon)$.

A similar procedure with the second condition of the right hand side of Theorem 6 leads to the condition $\frac{n_2}{n_1}$, leading to the desired result. □

G Proof of Theorem 3

Theorem 8. *Let $E(\mathbb{G})$ be the expected multilayer graph with T layers following the multilayer SBM with k classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$ of equal size and parameters $\left(p_{\text{in}}^{(t)}, p_{\text{out}}^{(t)}\right)_{t=1}^{T}$. Let $n_1, \ldots, n_k$ be the number of nodes per class be labeled. Let $C \in \mathbb{R}^n$ be a cost vector where with $C_i = n/n_r$ for $v_i \in \mathcal{C}_r$. Then, a zero test classification error is achieved if and only if*

$$m_p(\boldsymbol{\rho_\epsilon}) < 1 + \epsilon\,,$$

where $(\boldsymbol{\rho_\epsilon})_l = 1 - (p_{\text{in}}^{(l)} - p_{\text{out}}^{(l)})/(p_{\text{in}}^{(l)} + (k-1)p_{\text{out}}^{(l)}) + \epsilon$, and $l = 1, \ldots, T$.

Proof. The proof is similar to the one of Theorem 1 (see Section B). The only change is in the terms $c_r \frac{n_r}{n}$. Since we have by definition that $c_r = \frac{n}{n_r}$ we have that $c_r \frac{n_r}{n} = 1$, leading to the conditions obtained by Theorem 1. □

H A scalable matrix-free method for the linear system $(I + \lambda L_p)f = Y$

Computing the generalized matrix mean of T positive definite matrices $A_1, \ldots, A_T$ requires to compute $T + 1$ matrix functions: $A_1^p, \ldots, A_T^p$ and $(\sum_i A_i^p)^{1/p}$. Typically, the matrices A_i^p are full even though each A_i is a sparse matrix and so, computing L_p explicitly is unfeasible if the A_i's have large dimensions. Given a vector $\mathbf{y}$ and a negative integer p, here we propose a matrix-free method for solving the linear system $(I + \lambda L_p)^{-1}\mathbf{y}$. The method exploits the sparsity of the Laplacians of each layer and is matrix-free in the sense that it requires only to compute the matrix-vector product $A_i \times vector$, without requiring to store the matrices A_i themselves nor to compute any matrix function A_i^p explicitly. Thus, when the layers are sparse, the method scales to large datasets. Below we give further details about the method presented in the short version of the paper. We present the method for a general set of positive definite matrices $A_1, \ldots, A_T$, and for a general vector $\mathbf{y}$, for the sake of generality.

Let $S_p = A_1^p + \cdots + A_T^p$, $\varphi : \mathbb{C} \to \mathbb{C}$ be the complex function $\varphi(z) = z^{1/p}$ and let L_p be the matrix function $L_p = T^{-1/p}\varphi(S_p)$. The proposed method essentially transforms the original problem into a series of subproblems which thus allow us to solve the linear system $(I + \lambda L_p)^{-1}\mathbf{y}$ by solving several different linear systems with A_i as coefficient matrices. The method consists of three main nested inner–steps which we present below.

1. First, we solve the linear system $(I + \lambda L_p)^{-1}\mathbf{y}$ by a Krylov method (GMRES in our case [27]). At each iteration, this method projects the problem into the Krylov subspace spanned by $\{\mathbf{y}, \lambda L_p\mathbf{y}, (\lambda L_p)^2\mathbf{y}, \ldots, (\lambda L_p)^h\mathbf{y}\}$. If $\kappa = \lambda_{\max}(L_p)/\lambda_{\min}(L_p)$, then the method converges as

$$O\left(\left(\frac{\kappa^2 - 1}{\kappa^2}\right)^{h/2}\right).$$

Thus, if L_p is well conditioned, a relatively small h is required. In order to build the appropriate Krylov subspace, at each iteration we need to efficiently perform one matrix–vector product $L_p\mathbf{y}$.

2. Second, in order to compute $L_p\mathbf{y} = T^{-1/p}\varphi(S_p)\mathbf{y}$ we use the Cauchy integral form of the function φ, transformed via a conformal map, to approximate $\varphi(S_p)$ via the trapezoidal rule, as proposed in [12]. Let $m, M > 0$ be such that the interval $[m, M]$ contains the whole spectrum of S_p and let $t_1, \ldots, t_N$ be N equally spaced contour points to be used in the trapezoidal rule. As φ has a singularity at $z = 0$ but just a brunch cut on $(-\infty, 0)$, we can approximate $\varphi(S_p)\mathbf{y}$ via [12]

$$\varphi_N(S_p)\mathbf{y} = \frac{-8K(mM)^{1/4}}{\pi N k} S_p \ \text{Im}\left\{\sum_{i=1}^{N} \frac{\varphi(z_i^2)c_i d_i}{z_i(k^{-1} - s_i)^2}(z_i^2 I - S_p)^{-1}\mathbf{y}\right\}$$

where Im denotes the imaginary part, $k = \big((M/m)^{1/4} - 1\big)/\big((M/m)^{1/4} + 1\big)$, K is the value of the complete elliptic integral of the first kind, evaluated at ke^2, $s_i = \mathrm{sn}(t_i)$ is the Jacobi elliptic sine function evaluated on the i-th contour point t_i, and

$$z_i = (mM)^{1/4} \left(\frac{k^{-1} + s_i}{k^{-1} - s_i} \right), \quad c_i = \sqrt{1 - s_i^2}, \quad d_i = \sqrt{1 - k^2 s_i^2},$$

for $i = 1, \ldots, N$. This approximation converges geometrically as the number of points increases. Precisely, it holds

$$\|\varphi(S_p)\mathbf{y} - \varphi_N(S_p)\mathbf{y}\| = O(e^{-2\pi^2 N/(\ln(M/m)+6)}) .$$

Thus, the computation of $\varphi(S_p)\mathbf{y}$ is reduced to N linear systems $(z_i^2 I - S_p)^{-1}\mathbf{y}$. Note that these systems are independent and thus they can be solved in parallel.

3. Finally, in order to solve the linear system $(zI - S_p)^{-1}\mathbf{y}$ we employ again a Krylov method. In order to build the Krylov space for $(zI - S_p)$ and $\mathbf{y}$ we need to efficiently perform one multiplication S_p times a vector per iteration. As $S_p = \sum_{i=1}^{T} A_i^p = \sum_{i=1}^{T} (A_i^{-1})^{|p|}$, this problem reduces to solving q linear systems with A_i as coefficient matrix, for $i = 1, \ldots, T$. As the matrices A_i are assumed sparse and positive definite, we can very efficiently solve each of these systems via the Preconditioned Conjugate Gradient method with an incomplete Cholesky preconditioner.

The pseudocode for the proposed algorithm is presented in Algorithms 1–3.

Input: $A_1, \ldots, A_T, p, \mathbf{y}, \lambda$
1 Compute preconditioners $P_1, \ldots, P_T$ for $A_1, \ldots, A_T$
2 Compute estimates for m and M such that $\mathrm{eigenvalues}(S_p) \subseteq [m, M]$
3 Choose number of contour points N
4 Compute contour coefficients z_i, s_i, K, k
5 Solve $(I + \lambda L_p)^{-1}\mathbf{y}$ with GMRES, using Alg.2 as subroutine
Output: $\mathbf{u} = (I + \lambda L_p)^{-1}\mathbf{y}$

Algorithm 1: Solve $(I + \lambda L_p)^{-1}\mathbf{y}$

Input: $A_1, \ldots, A_T, p, \mathbf{y}, N, m, M$, contour coefficients z_i, s_i, c_i, d_i, k, K
1 $\mathbf{u} \leftarrow S_p\mathbf{y}$, using Alg.3
2 **for** $i = 1, \ldots, N$ **do**
3 $\quad \mathbf{u} \leftarrow \mathrm{solve}(z_i I - S_p, \mathbf{y})$ with GMRES, using Alg.3 as subroutine
4 $\quad \mathbf{u} \leftarrow \frac{(z_i^2)^{1/p} c_i d_i}{z_i (k^{-1} - s_i)^2}\mathbf{u}$
5 $\quad \mathbf{u}_{k+1} = \|\mathbf{v}_{k+1}\|_q^{1-q} |\mathbf{v}_{k+1}|^{q-2}\mathbf{v}_{k+1}$
6 **end**
7 $\mathbf{u} \leftarrow \frac{1}{T^{1/p}} \frac{-8K(mM)^{1/4}}{\pi N k} \mathrm{Im}(\mathbf{u})$
Output: $\mathbf{u} = L_p\mathbf{y}$

Algorithm 2: Multiply L_p times a vector

Input: $A_1, \ldots, A_T, P_1, \ldots, P_T, \mathbf{y}$
1 **for** $k = 1, \ldots, T$ **do**
2 $\quad \mathbf{u} \leftarrow \mathbf{u} + \mathrm{solve}(A_i^{|p|}, \mathbf{y})$ using CG preconditioned with P_i
3 **end**
Output: $\mathbf{u} = S_p\mathbf{y}$

Algorithm 3: Multiply S_p times a vector

H.1 Implementation details and computational complexity

Few implementation details are in order:

The preconditioners P_i can be computed using an incomplete Cholesky factorization. In our test we observe that a `1e-4` threshold is enough to ensure convergence of Alg.3 to `1e-8` precision in just 2 or 3 iterations. As in our case the A_i are Laplacians, another excellent preconditioner can be obtained using a Combinatorial Multi Grid method (CMG). In our experiments, the CMG preconditioner performed similarly (but slightly worse) than the incomplete Cholesky.

A precise estimate of M in Alg.1 step 2 can be obtained using a Krylov eigensolver with Alg.3 as subroutine. As for m, since each A_i^p is positive definite and p is a negative integer, a good estimate

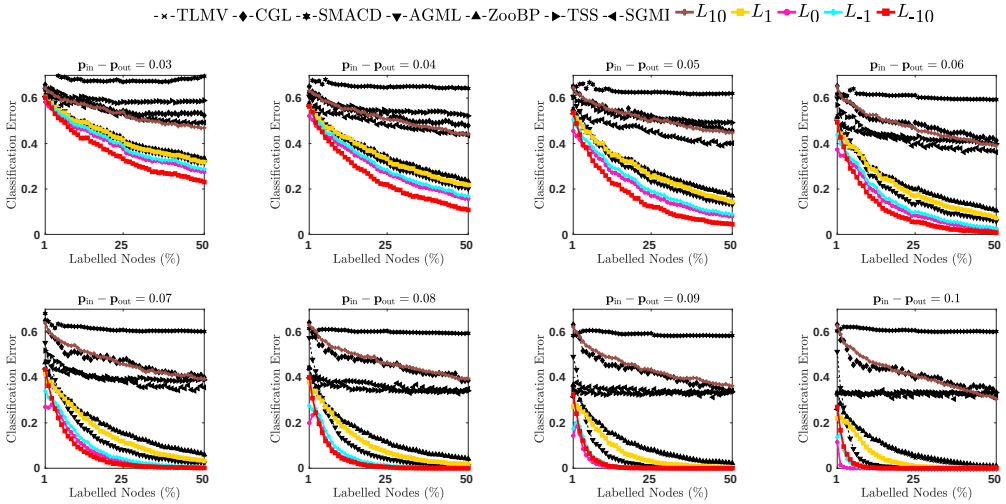


Figure 6: Experiments with three Layers and three classes as in Section 4.2 and Figure.4.

can be obtained by exploiting the Weyl's inequality (see e.g. [32])

$$m = \lambda_{\max}(A_1)^p + \cdots + \lambda_{\max}(A_T)^p \leq \lambda_{\min}(S_p)\,.$$

The number of contour points N can be chosen using the geometric convergence of φ_N. In our experiments, we chose a precision $\tau =$`1e-8` and we set

$$N = \left|(\ln(M/m) + 6)\ln(\tau)/2\pi^2\right|.$$

The contour points have been calculated using the code from [7].

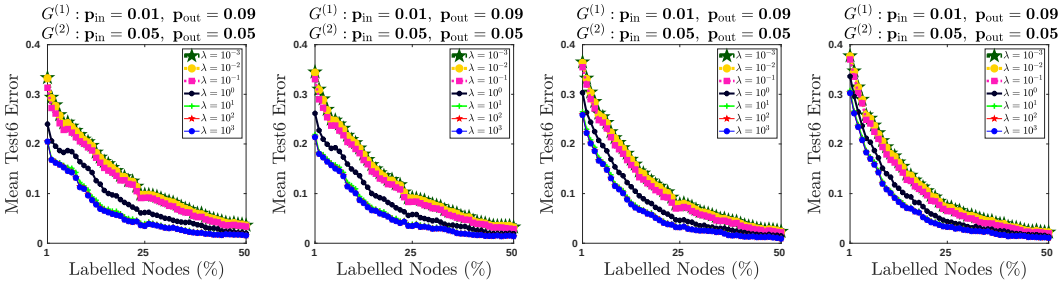
Concerning the computational cost of the method, the following analysis shows that it is proportional to the number of edges in each layer, i.e. Alg.1 scales to large sparse datasets. Let $c(A_i)$ be the cost of multiplying $c(A_i)$ times a vector (which is proportional to the number of nonzeros in A_i, i.e. the number of edges in the layer i when A_i is the normalized Laplacian of the i-th layer). Let K_1, K_2, K_3 be the number of iterations of GMRES,GMRES and PCG in lines 5, 3 and 2 of Algorithms 1, 2 and 3, respectively. Each instance of $\mathrm{solve}(A_i^{|p|}, \mathbf{y})$ in Alg.3 requires $K_3 p\, c(A_i)$ operations per step. So The cost of Alg.3 is roughly $pK_3 \sum_{i=1}^{T} c(A_i)$. This implies that the cost of Alg.2 is $NK_2K_3p \sum_{i=1}^{T} c(A_i)$. Therefore, the cost of solving the linear system $(I + \lambda L_p)^{-1}\mathbf{y}$ with Alg.1 is

$$K_1 N K_2 K_3 p\big(c(A_1) + \cdots + c(A_T)\big)\,,$$

showing that the method scales as the number of nonzeros in each layer, as claimed. It is important to notice that the Algorithm allows for a high level of parallelism. In fact, the computation of the preconditioners P_i at step 1 of Alg.1, the **for** at step 2 of Alg.2 and the **for** at step 1 of Alg.3 can all be run in parallel.

I Analysis on Three Layers with Three Classes

In this section we give a more detailed exposition of experiments presented in Section 4.2. We consider the cases where $p_{\text{in}} - p_{\text{out}} \in \{0.03, 0.04, \ldots, 0.1\}$ which are depicted in Fig.6. In the x-axis we have the amount of labeled nodes and in the y- we have the classification error. We can see that in general there is a trend between the performance of our proposed method (colorful curves) and state of the art methods (black curves). We can see that the larger the gap $p_{\text{in}} - p_{\text{out}}$ the larger the difference is between our proposed method and state of the art methods. Moreover, one can see that the smaller the value of p the better the performance of our proposed method. Moreover, there is a set of state of the art methods that do not improve their performance with larger amounts of labeled nodes. Yet, one can observe that there are three methods from the state of the art that perform close to our methods: TLMV, ZooBP and AGML, which performs similarly to our method L_1 (i.e. the arithmetic mean of Laplacians).



(a) L_{-1} (b) L_{-2} (c) L_{-5} (d) L_{-10}

Figure 7: Mean test classification error under MSBM for different values of λ. Details in Sec. J.

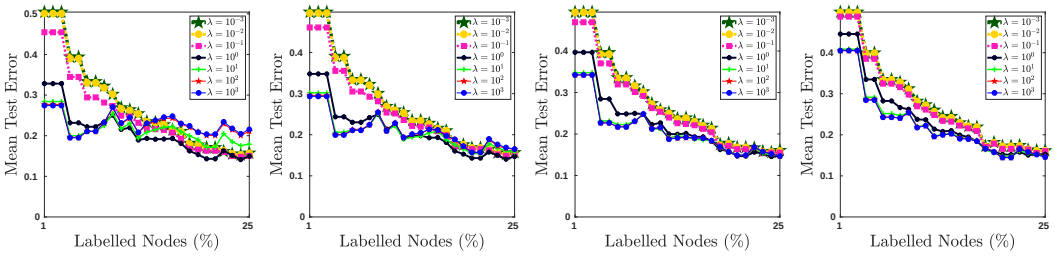
J Analysis on Effect of Regularization Parameter

In this section we present a numerical evaluation on the effect of the regularization parameter λ under the multilayer stochastic block model and on real world datasets. The corresponding results are depicted in Fig. 7 and Fig. .

Experiments under Multilayer Stochastic Block Model. We analyze the effect of the regularization parameter λ under the Multilayer Stochastic Block Model. The experimental setting is as follows: We fix the parameters of the first layer $G^{(1)}$ and second layer $G^{(2)}$ to $p_{\text{in}}^{(1)} = 0.09, p_{\text{out}}^{(1)} = 0.01, p_{\text{in}}^{(2)} = 0.05, p_{\text{out}}^{(2)} = 0.05$. We consider values of $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$, different amount of labeled nodes $\{1\%, \ldots, 50\%\}$. We sample five random multilayer graphs with the corresponding parameters and 5 random samples of labeled nodes with a fixed percentage, and present the average classification error. In Fig. 7 we can see that in general the larger the value of λ the smaller the classification error. In particular we can see that the performance does not present any relevant changes with $\lambda \leq 10^{-1}$.

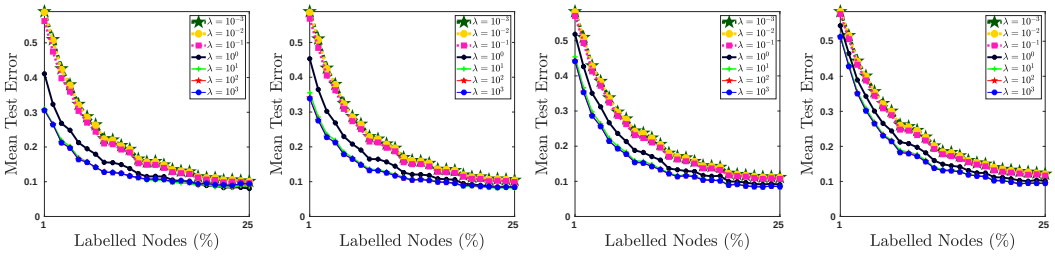
Experiments with real world datasets. We analyze the effect of the regularization parameter λ with real world datasets considered in Section 6. For each dataset we build the corresponding layer adjacency matrices by the taking symmetric k-nearest neighbour graph and take as similarity measure the Pearson linear correlation, (i.e. we take the k neighbours with highest correlation), and take the unweighted version of it.

We fix nearest neighbourhood size to $k = 10$ and generate 10 samples of labeled nodes, where the percentage of labeled nodes per class is in the range $\{1\%, 2\%, \ldots, 25\%\}$. The average test errors are presented in Fig. 8, for power mean Laplacian regularizers L_{-1}, L_{-2}, L_{-5}, and L_{-10}. We can see that in general the best performance, i.e. smallest mean test classificaton error corresponds to values of $\lambda = 10, 10^2, 10^3$, verifying the choice of $\lambda = 10$ presented in Section 6. Moreover, we can see that the mean test error in general decreases with larger amounts of labeled data, which verifies our previous experiments on multilayer graphs following the Multilayer Stochastic Block Model.



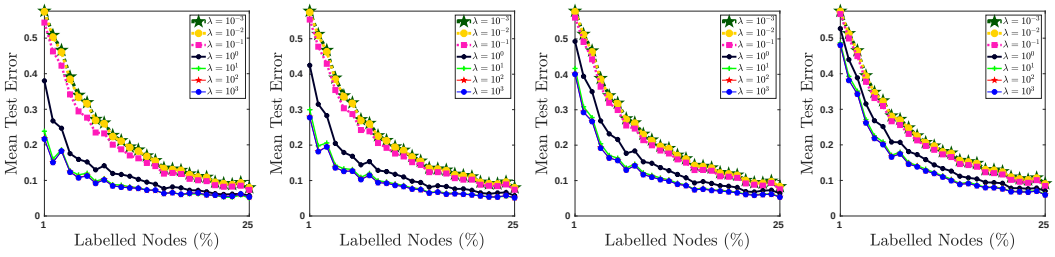
(a1) L_{-1} (a2) L_{-2} (a3) L_{-5} (a4) L_{-10}

(a) Dataset: 3sources



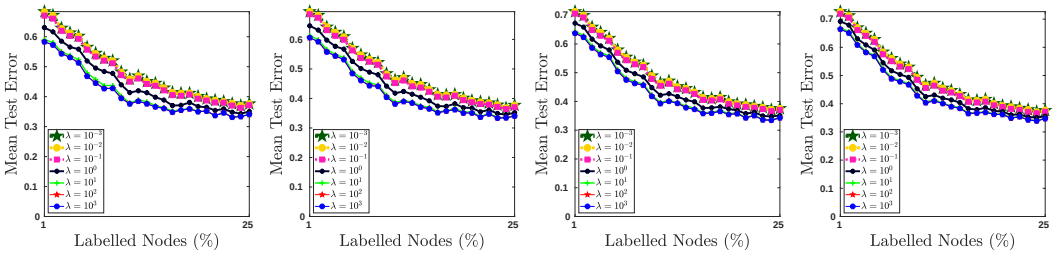
(b1) L_{-1} (b2) L_{-2} (b3) L_{-5} (b4) L_{-10}

(b) Dataset: BBC



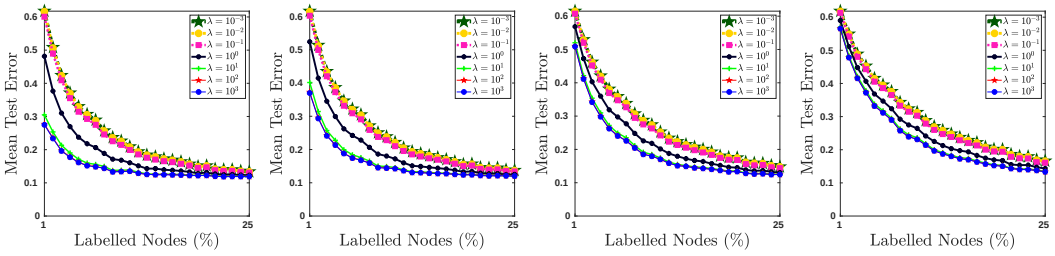
(c1) L_{-1} (c2) L_{-2} (c3) L_{-5} (c4) L_{-10}

(c) Dataset: BBCS



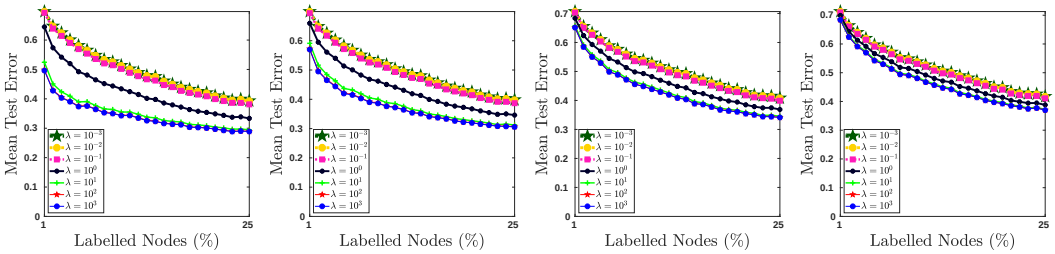
(d1) L_{-1} (d2) L_{-2} (d3) L_{-5} (d4) L_{-10}

(d) Dataset: Wikipedia



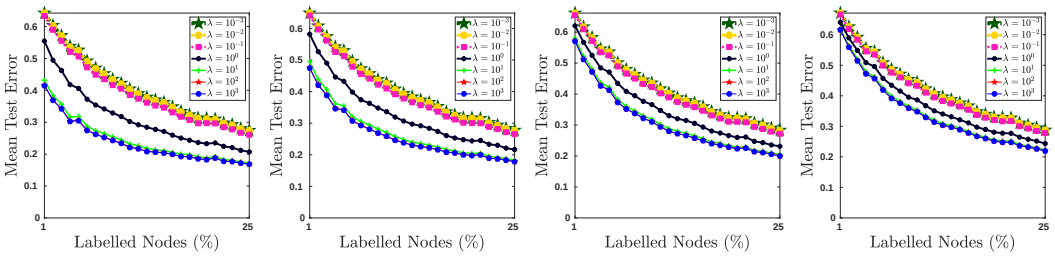
(e1) L_{-1} (e2) L_{-2} (e3) L_{-5} (e4) L_{-10}

(e) Dataset: UCI



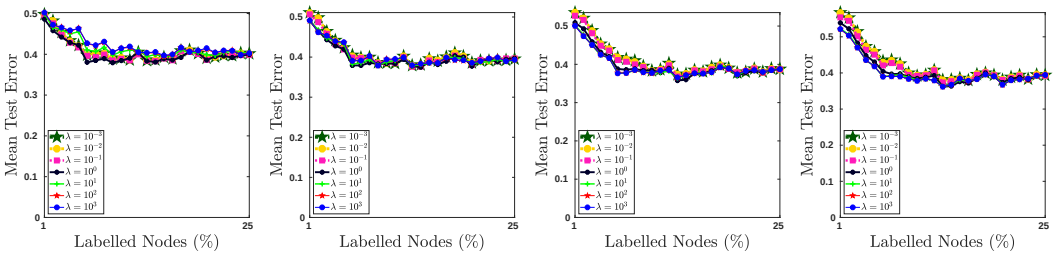
(f1) L_{-1} (f2) L_{-2} (f3) L_{-5} (f4) L_{-10}

(f) Dataset: Citeseer



(g1) L_{-1} (g2) L_{-2} (g3) L_{-5} (g4) L_{-10}

(g) Dataset: Cora



(h1) L_{-1} (h2) L_{-2} (h3) L_{-5} (h4) L_{-10}

(h) Dataset: WebKB

Figure 8: Mean test classification error on real world datasets for different values of λ. Details in Sec. J.