BOUNDS ON PERFECT NODE CLASSIFICATION: A CONVEX GRAPH CLUSTERING PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the problem of transductive node classification in graphs where communities align with both node features and labels. We propose a novel convex optimization framework that integrates node-specific information (features and labels) into graph clustering via low-rank matrix estimation. Our analysis reveals a bidirectional interaction between graph structure and node information: not only can features aid clustering, but graph structure can also enhance node classification. In particular, we prove that incorporating suitable node information enables perfect recovery of communities under milder conditions than required by graph clustering alone. To make the framework practical, we develop efficient algorithmic solutions and validate our theory with experiments demonstrating the predicted improvements.

1 INTRODUCTION

Transductive node classification is a fundamental problem in machine learning on graphs: given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where nodes have features and a subset $\mathcal{V}_{\text{train}}$ is labeled, the goal is to predict the labels of the remaining nodes $\mathcal{V}_{\text{test}}$. This setting is ubiquitous in real-world applications such as citation networks, social networks, and recommender systems (Bhagat et al., 2011; Zhu et al., 2003). A central idea of modern graph-based learning is that exploiting graph structure, in addition to node features, can significantly improve classification accuracy. This idea has driven the development of graph neural networks (GNNs) and related methods (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018).

However, despite strong empirical evidence supporting the utility of graph information, there is little formal understanding of *when and why* it actually helps: no existing work establishes rigorous conditions under which incorporating graph structure provably improves classification performance over feature-only methods.

In this paper, we address this fundamental question by formulating transductive node classification as a convex optimization problem, which enables us to derive rigorous theoretical guarantees about when graph structure provably helps classification. We focus on homophilic graphs (McPherson et al., 2001), where nodes of the same class exhibit higher connectivity than nodes of different classes, a property prevalent in many real-world data, including citation networks (Sen et al., 2008; Namata et al., 2012), social networks (Hamilton et al., 2017), and commercial networks (McAuley et al., 2015). In homophilic settings, the graph topology naturally exhibits community structure that correlates with node features and labels.

Our key insight is that under homophily, graph clustering and node classification are fundamentally complementary: graph clustering exploits the graph topology to reveal community structure, while classification leverages features and partial labels. This observation motivates us to develop a unified framework that integrates both tasks. We build on convex methods for graph clustering (Korlakai Vinayak et al., 2014; Chen et al., 2014; Li et al., 2021), which offer theoretical tractability and are closely related to well-understood spectral techniques (Belkin & Niyogi, 2001; Ng et al., 2001; Hajek et al., 2016). In particular, our approach leverages the framework of atomic norms, which generalize the nuclear norm and allow us to define atoms that jointly capture graph structure and node-specific information. This perspective enables us to extend convex graph clustering formulations based on low-rank positive semidefinite representations allowing them to incorporate node features and labels within a single convex optimization framework.

Our contributions are as follows:

- We introduce a novel convex optimization formulation for transductive node classification that generalizes existing graph clustering methods to incorporate node-specific information, including features and partial labels. The framework is inspired by principles of multimodal learning and provides a principled way to combine structural and attribute information.
- We prove that, under suitable structural assumptions, our framework achieves perfect label recovery, i.e., exact classification of all nodes. To the best of our knowledge, this provides the first rigorous theoretical result demonstrating that node features and graph structure can provably interact to improve node classification.
- We develop CADO, a scalable alternating conditional gradient algorithm for solving our optimization problem with a fixed number of atoms. Each step of the algorithm reduces to tractable subproblems with closed-form solutions, making the method both efficient and practical. Through experiments, we show that CADO solves the proposed optimization and validate our theoretical findings.

To the best of our knowledge, no prior work has formulated node classification through the lens of convex graph clustering. Our framework integrates ideas from convex clustering and graph clustering, while directly incorporating both node features and graph structure into a single formulation.

2 Related Work

Convex clustering. Clustering groups data points based solely on their features (Xu & Wunsch, 2005; Jaeger & Banks, 2023; Shalev-Shwartz & Ben-David, 2014; Soltanolkotabi & Candes, 2012). Classical k-means is NP-hard (Aloise et al., 2009), and Lloyd's algorithm is prone to local minima and sensitive to initialization. Convex clustering addresses these issues by adding a sum-of-norms (SON) regularizer and formulating a convex relaxation of k-means (Hocking et al., 2011; Lindsten et al., 2011; Panahi et al., 2017; Tan & Witten, 2015; Sun et al., 2021). Recovery guarantees have been established for two clusters (Zhu et al., 2014), k clusters (Panahi et al., 2017), and weighted variants (Sun et al., 2021), with further contributions by Chiquet et al. (2017); Chi & Steinerberger (2019). However, these works remain tied to k-means. Our framework allows general loss functions and extending recovery guarantees to this broader setting and to node classification.

Graph clustering. Graph clustering (or community detection) seeks to identify clusters using only graph structure (Schaeffer, 2007; Abbe, 2018; Li et al., 2021). The stochastic block model (Holland et al., 1983) is the standard framework, with exact recovery thresholds established for two clusters (Abbe et al., 2015), general k (Wu et al., 2015), and partially-observed graphs (Chen et al., 2014; Korlakai Vinayak et al., 2014). Our work goes beyond SBM by providing recovery guarantees for deterministic graphs and Erdős-Rényi random graphs, thereby broadening the scope of classical graph clustering theory.

Covariate-assisted graph clustering. Many modern graphs include node covariates, motivating methods that leverage both graph struture and node-level information. Heuristic approaches aggregate the adjacency matrix with a Gram or kernel matrix from covariates (Binkiewicz et al., 2017; Yan & Sarkar, 2021; Hu & Wang, 2024; Chunaev, 2020). The contextual SBM (CSBM) (Deshpande et al., 2018) provides a statistical model for this setting, and recent theory shows that covariates can improve graph clustering (Braun et al., 2022; Dreveton et al., 2023; Yang & Fountoulakis, 2023; Braun & Sugiyama, 2024). These works, however, rely on Gaussian or exponential covariates combined within the CSBM and do not capture explicitly the synergy between graph structure and covariates. Unlike these works, our framework establishes a two-way interaction: not only can features support clustering, but graph structure can also enhance node classification. We prove recovery guarantees for this bidirectional setting under static graphs, arbitrary features, and any convex loss functions.

3 Node Classification via Atomic Norms

In this section, we propose a graph clustering formulation that naturally extends to incorporate node-specific information.

We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node $v \in \mathcal{V}$ has feature vector $\boldsymbol{x}_v$. A subset $\mathcal{V}_{\text{train}} \subset \mathcal{V}$ of the nodes has (possibly noisy) labels, and the goal is to predict the labels for the remaining nodes

$\mathcal{V}_{\text{test}}$, referred to as test nodes. We denote by y_v the true label of node v and its noisy label by $\tilde{y}_v$. For convenience, we denote by $\boldsymbol{z}_v$ the information associated with node $v \in \mathcal{V}$, where $\boldsymbol{z}_v = \boldsymbol{x}_v$ for $v \in \mathcal{V}_{\text{test}}$ and $\boldsymbol{z}_v = (\boldsymbol{x}_v, \tilde{y}_v)$ for $v \in \mathcal{V}_{\text{train}}$.

Our starting point is convex optimization techniques for graph clustering, which seek a low-rank positive semi-definite (PSD) approximation of the adjacency matrix $\boldsymbol{A}$ (Korlakai Vinayak et al., 2014),

$$\min_{\boldsymbol{L} \in \mathcal{B}} \ \|\boldsymbol{L} - \boldsymbol{A}\|_1 + \mu_0 \|\boldsymbol{L}\|_*, \tag{1}$$

where $\|\cdot\|_1$ is the ℓ_1-norm, $\|\cdot\|_*$ the *nuclear norm* (sum of the singular values), $\mathcal{B}$ denotes the set of symmetric matrices with entries in $[0, 1]$ and ones on the diagonal, and $\mu_0 \geq 0$ is a regularization parameter.

Since the nuclear norm is a special case of the atomic norm (Bhaskar et al., 2013) (see Appendix A for details), we can equivalently replace $\|\boldsymbol{L}\|_*$ with the atomic norm $\|\boldsymbol{L}\|_{\mathcal{A}}$, where the atomic set is defined as $\mathcal{A} := \{\boldsymbol{e}\boldsymbol{e}^\top : \|\boldsymbol{e}\|_2 = 1\}$, where each atom is a rank-one matrix formed from a unit vector. Using the atomic norm, (1) can be rewritten as

$$\min_{\boldsymbol{L} \in \mathcal{B}} \ \|\boldsymbol{L} - \boldsymbol{A}\|_1 + \mu_0 \|\boldsymbol{L}\|_{\mathcal{A}}. \tag{2}$$

Noting that $\|\boldsymbol{L} - \boldsymbol{A}\|_1$ can be expressed as $-\langle \bar{\boldsymbol{A}}, \boldsymbol{L} \rangle$,[1] (2) can be rewritten as

$$\begin{aligned} \min_{\boldsymbol{L} \in \mathcal{B},\, \{\lambda_i \geq 0,\, \boldsymbol{e}_i \in \mathbb{S}^{n-1}\}_i} \quad & -\langle \bar{\boldsymbol{A}}, \boldsymbol{L} \rangle + \mu_0 \sum_i \lambda_i \\ \text{s.t.} \quad & \boldsymbol{L} = \sum_i \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^\top, \end{aligned} \tag{3}$$

where $\bar{\boldsymbol{A}}$ is the *polarized adjacency matrix* with entries 1 if $\{u, v\} \in \mathcal{E}$ and -1 otherwise. The summations are over the infinite elements as the atomic set contains infinite number of atoms, and this holds for the rest of paper whenever no upper bound is specified.

This optimization serves as the foundation of our framework. Specifically, if $\mathcal{G}$ has r well-separated clusters, the solution of (3) has rank r and admits the decomposition

$$\boldsymbol{L} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^\top, \qquad \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_r), \quad \boldsymbol{E} = [\, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_r \,],$$

The v-th row of $\boldsymbol{E}$, $\boldsymbol{\epsilon}_v$ represents a low-rank embedding of node $v \in \mathcal{V}$. Under some mild conditions on the graph (see Section 4), these embeddings become one-hot vectors that perfectly indicate cluster membership, enabling exact cluster recovery.

Incorporating node-specific information. We extend the convex graph clustering formulation in (3) to incorporate node-specific information. We associate each node v with a model $\boldsymbol{\theta}_v \in \Theta$ and a loss function $f_v(\boldsymbol{z}_v; \boldsymbol{\theta})$ measuring how well $\boldsymbol{\theta}_v$ fits $\boldsymbol{z}_v$. A concrete example will be given in Section 6.1. Based on these components, we formulate the following optimization problem,

$$\begin{aligned} \min_{\substack{\boldsymbol{L} \in \mathcal{B},\, \{\boldsymbol{\theta}_v \in \Theta\}_{v \in \mathcal{V}}, \\ \{\lambda_i \geq 0,\, \boldsymbol{e}_i \in \mathbb{S}^{n-1},\, \boldsymbol{\theta}_i \in \Theta\}_i}} \quad & -\langle \bar{\boldsymbol{A}}, \boldsymbol{L} \rangle + \mu \sum_{v \in \mathcal{V}} f_v(\boldsymbol{z}_v; \boldsymbol{\theta}_v) \\ \text{s.t.} \quad & \boldsymbol{L} = \sum_i \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^\top, \quad \boldsymbol{\theta}_v = \sum_i \lambda_i \epsilon_{i,v}^2 \boldsymbol{\theta}_i, \quad \forall v \in \mathcal{V} \end{aligned} \tag{4}$$

where $\epsilon_{i,v}$ is the v^{th} element of $\boldsymbol{e}_i$, and $\boldsymbol{\theta}_i \in \Theta$ represents the model associated with cluster i. The additional term in the objective function of (4) plays the role of an empirical risk, but with each $\boldsymbol{z}_v$ evaluated against a node-specific model $\boldsymbol{\theta}_v$. The coupling constraint $\boldsymbol{\theta}_v = \sum_i \lambda_i \epsilon_{i,v}^2 \boldsymbol{\theta}_i$ ensures these models are not completely independent. In particular, when the embedding vectors $\boldsymbol{\epsilon}_v = (\epsilon_{i,v})_i$ coincide with the one-hot vectors of the corresponding clusters, we have $\boldsymbol{\theta}_v = \boldsymbol{\theta}_i$ for all v in cluster i, so classification benefits directly from the low-rank clustering mechanism.

Similar to (2)–(3), (4) can be expressed as a regularized atomic norm problem. To this end, we define the joint variable $\boldsymbol{U} := \big(\boldsymbol{L}, \{\boldsymbol{\theta}_v\}_{v \in \mathcal{V}}\big)$, the atoms $\boldsymbol{a}_i := \big(\boldsymbol{e}_i \boldsymbol{e}_i^\top, \{\epsilon_{i,v}^2 \boldsymbol{\theta}_i\}_{v \in \mathcal{V}}\big)$, and the atomic set

$$\mathcal{A} := \Big\{ \big(\boldsymbol{e}\boldsymbol{e}^\top, \{\epsilon_v^2 \boldsymbol{\theta}\}_{v \in \mathcal{V}}\big) \;\Big|\; \boldsymbol{e} = (\epsilon_v)_v,\ \|\boldsymbol{e}\|_2 = 1,\ \boldsymbol{\theta} \in \Theta \Big\}.$$

[1] This follows from the fact that for $a_{ij} \in \{0, 1\}$ and $l_{ij} \in [0, 1]$, we have $|l_{ij} - a_{ij}| = -l_{ij}\bar{a}_{ij} + a_{ij}$.

With this notation, problem (4) can be rewritten compactly as

$$\min_{\boldsymbol{U}\in\mathcal{U},\,\{\lambda_i\geq 0,\,\boldsymbol{a}_i\}_i} \quad \phi(\boldsymbol{U}) + \mu_0 \sum_i \lambda_i \qquad \text{s.t.} \quad \boldsymbol{U} = \sum_i \lambda_i \boldsymbol{a}_i, \tag{5}$$

where $\phi(\boldsymbol{U}) := -\langle \bar{\boldsymbol{A}}, \boldsymbol{L}\rangle + \mu \sum_{v\in\mathcal{V}} f_v(\boldsymbol{z}_v;\boldsymbol{\theta}_v)$ and $\mathcal{U} := \mathcal{B}\times\Theta^{|\mathcal{V}|}$ is the feasible set of all variables $\left(\boldsymbol{L}, \{\boldsymbol{\theta}_v\}_{v\in\mathcal{V}}\right)$ with $\boldsymbol{L}\in\mathcal{B}$ and $\boldsymbol{\theta}_v\in\Theta$.

Regularization by Sum of Norms. The formulation in (5) admits an arbitrary number of atoms, each potentially corresponding to a distinct cluster. This flexibility risks over-parameterization: the node-specific term in ϕ may favor assigning each node to its own cluster. Although the graph-based term in ϕ can counteract this, the resulting balance does not capture the intended synergy between topology and node-specific information.

To address this, we adopt the well-known *sum-of-norms (SON)* regularization (Lindsten et al., 2011; Panahi et al., 2017) defined as

$$\boldsymbol{R}(\boldsymbol{U}) := \sum_{u<v} \|\boldsymbol{\theta}_u - \boldsymbol{\theta}_v\|\,.$$

This penalty encourages node-specific models to coincide, promoting shared representations within clusters. Incorporating it into (5) yields

$$\min_{\boldsymbol{U}\in\mathcal{U},\,\{\lambda_i\geq 0,\,\boldsymbol{a}_i\}_i} \quad \phi(\boldsymbol{U}) + \mu_0 \sum_i \lambda_i + \mu_1 R(\boldsymbol{U}) \qquad \text{s.t.} \quad \boldsymbol{U} = \sum_i \lambda_i \boldsymbol{a}_i \tag{6}$$

This regularization enforces that nodes in the same cluster share identical models, thereby aligning the clustering and classification components.

Equation (6) is our final optimization framework: it extends convex graph clustering by incorporating node-specific information through SON regularization. In the remainder of the paper, we analyze this framework from two complementary perspectives. First, we establish conditions under which it achieves perfect recovery of the underlying clusters. Second, we investigate the computational complexity of solving the problem and introduce CADO, an efficient algorithm for computing its solution.

4 Perfect Recovery

In this section, we establish conditions under which the global solution of (6) achieves perfect recovery of the underlying clusters and, consequently, the class labels. Our analysis proceeds in three steps. First, we characterize the structure of the *ideal solution* that corresponds to perfect recovery. Next, we formalize the probabilistic model governing the graph and node features. Based on this model, we derive explicit conditions on the parameters under which the solution of (6) coincides with the ideal one, thereby achieving perfect recovery. Finally, we present our main theorem, showing that perfect recovery can be achieved from graph structure alone, from node features alone, or from their combination. Crucially, the joint setting admits strictly milder conditions than either source individually, thereby demonstrating the synergistic effect of integrating graph and feature information.

4.1 Ideal Solution

Consider a population $\mathcal{V}$ of size $|\mathcal{V}| = n$, partitioned into K clusters $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K$, where cluster $\mathcal{C}_i$ contains n_i nodes. In the ideal scenario, the solution consists of K atoms (one per cluster) denoted by $\{(\lambda_i^*, \epsilon_{i,v}^*\boldsymbol{\theta}_i^*)\}_{i=1}^K$, with

$$\lambda_i^* = n_i, \qquad \epsilon_{i,v}^* = \begin{cases} \frac{1}{\sqrt{n_i}}, & v\in\mathcal{C}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This yields the ideal partition matrix $\boldsymbol{L}^*$ defined by $L^*_{uv} = 1$ if $u, v \in \mathcal{C}_i$ for some i, and 0 otherwise.

Moreover, each cluster $\mathcal{C}_i$ is associated with a single model $\boldsymbol{\theta}^*_i$. These cluster-level models arise from the following *characteristic optimization problem*, obtained by substituting the ideal coefficients from (7) into (6):

$$\{\boldsymbol{\theta}^*_i\}_i = \min_{\{\boldsymbol{\theta}_i\}_i \in \Theta} \mu \sum_{i=1}^{K} F_i(\boldsymbol{\theta}_i) + \gamma \sum_{i<j} n_j \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|, \tag{8}$$

where $F_i(\boldsymbol{\theta}) := \sum_{v \in \mathcal{C}_i} f_v(\boldsymbol{z}; \boldsymbol{\theta})$ is the aggregate loss function of cluster $\mathcal{C}_i$, and $\gamma := \mu_1/\mu$. The solutions of (8), referred to as *biased centroids*, satisfy the optimality condition $\mathbf{0} \in \boldsymbol{\omega}_i + \partial I_\Theta(\boldsymbol{\theta}^*_i)$, where $\partial I_{\mathcal{U}}(\cdot)$ is the subdifferential of the indicator function $I_{\mathcal{U}}$ of $\mathcal{U}$, and

$$\boldsymbol{\omega}_i := \nabla F_i(\boldsymbol{\theta}^*_i) + \gamma \sum_{j<i} n_j \frac{\boldsymbol{\theta}^*_i - \boldsymbol{\theta}^*_j}{\|\boldsymbol{\theta}^*_i - \boldsymbol{\theta}^*_j\|}. \tag{9}$$

4.2 Our Model

Our optimization framework takes as input both the graph structure and the node features. In what follows, we formalize the statistical assumptions underlying each of these components and introduce the definitions that will be used in our recovery analysis.

4.2.1 Graph

Let $N_{ij} := \sum_{u \in \mathcal{C}_i, v \in \mathcal{C}_j} A_{uv}$ be the connectivity of the different partitions. Naturally, we are interested in the case where N_{ii} is sufficiently larger than N_{ij} for $i \neq j$. For a node v, let $n_{v,j}$ be the number of edges from v to nodes in cluster $\mathcal{C}_j$, $n_{v,j} := \sum_{u \in \mathcal{C}_j} A_{uv}$. We take $n^+_{v,j} = n_j - n_{v,j}$ if $v \in \mathcal{C}_j$, and $n^+_{v,j} = n_{v,j}$ if $v \notin \mathcal{C}_j$. We also define $N^+_{i,j} = \sum_{v \in \mathcal{C}_i} n^+_{v,j}$ and $\rho^+_{i,j} = N^+_{i,j}/n_i n_j$.

These quantities capture the level of *misconnections* between clusters and vanish in the ideal case of perfectly separated clusters. For a fixed $\delta > 0$, we formalize and bound the level of cluster separability using the following assumptions.

Assumption 4.1 ($\delta-$Homogeneity)**.** For any node $v \in \mathcal{C}_i$ and any cluster $\mathcal{C}_j$, it holds that

$$\left| \frac{n^+_{v,j}}{n_j} - \rho^+_{ij} \right| \leq \delta. \tag{10}$$

Note that from Assumption 4.1, the maximum number of mis-connections in each block (i, j), denoted by $d^{\max}_{ij}$, is bounded as $d^{\max}_{ij} \leq (\rho^+_{ij} + \delta)\, n_j$.

Assumption 4.2 ($\delta-$Visibility)**.** For any two clusters $\mathcal{C}_i, \mathcal{C}_j$, it holds that $\rho^+_{ij} < \frac{1}{2} - \delta$.

4.2.2 Node-Specific Vectors

In convex clustering, perfect recovery requires hat inter-cluster separation dominates intra-cluster variability: loosely, the minimum distance between clusters must be significantly larger than the maximum cluster diameter While prior works focus on the k-means loss function, we consider general convex loss functions, which necessitate more general definitions of inter-cluster distance (separability) and cluster diameter. In the following assumptions, we formalize these two measures.

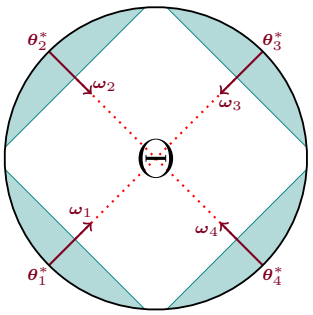

Figure 1: R-separability.

Assumption 4.3 (R-separability)**.** The biased centroids $\boldsymbol{\theta}^*_i$ are distinct. Moreover, for every $\boldsymbol{\theta} \in \Theta$, the relation

$$\langle \boldsymbol{\omega}_i, \boldsymbol{\theta} - \boldsymbol{\theta}^*_i \rangle \leq R$$

holds for at most one index i. In addition, the feasible set Θ is assumed to be bounded, i.e.,

$$\|\boldsymbol{\theta} - \boldsymbol{\theta}^*_i\| \leq \ell, \qquad \forall i.$$

This assumption guarantees that centroids corresponding to different clusters are well-separated, as depicted in Figure 1.

Assumption 4.4 (Gradient variability)**.** For all $u, v \in \mathcal{C}_i$, the gradient variability is bounded by $\|\nabla f_v(\boldsymbol{\theta}_i^*) - \nabla f_u(\boldsymbol{\theta}_i^*)\| \leq \rho$.

This assumption bounds the variability of node-specific information within clusters, i.e., it quantifies the diameter of the clusters. It ensures that within each cluster, the local losses behave similarly around the corresponding centroid $\boldsymbol{\theta}_i^*$ (each node aligns closely with its associated centroid). Furthermore, the parameter ρ can be interpreted as an indirect measure of data noise: larger values of ρ indicate higher within-cluster variability, meaning the data are noisier and less homogeneous.

4.3 Main result

Here, we present our main theoretical results.

Theorem 4.5. *Consider a fixed graph $\mathcal{G}$ with n nodes partitioned into K classes, where class i contains n_i nodes with $n_i \sim n_j$ for all i, j. Let ρ_{ij}^+ denote the relative misconnection rate between classes, and define*

$$\rho^+ := \max_{i,j} \rho_{ij}^+, \qquad p_{\max} = \rho^+ + \delta, \qquad a = \max_{i,j} a_{ij}.$$

Fix $\gamma = \mu_1/\mu$, and suppose Assumptions 4.1–4.4 hold with fixed δ and R, and that the feasible set Θ is bounded. Then:

1. ***Graph-structure-only regime*** *($\mu \to 0$). Perfect recovery is achievable if*

$$p_{\max} \;\leq\; \frac{1}{2 + a \cdot n/n_i}. \tag{11}$$

2. ***Node-information only regime*** *($\mu = \Omega(n \cdot n_i)$). Perfect recovery is achievable if*

$$\rho \;\leq\; \gamma\, n_i. \tag{12}$$

3. ***Synergistic regime*** *($0 < \mu < n \cdot n_i$). Perfect recovery is achievable if*

$$\rho \;\leq\; \left(\frac{1 - (a+2)\, p_{\max}}{\mu \ell} + \gamma \right) n_i \qquad \textit{and} \qquad p_{\max} \;\leq\; \frac{1 - \mu\ell(\rho/n_i - \gamma)}{2 + a}. \tag{13}$$

Proof. The proof is given in Appendix B. □

Theorem 4.5 unifies the recovery conditions across three regimes. In the graph-structure-only regime, the bound $p_{\max} \leq 1/(2 + a \cdot n/n_i)$ shows that recovery becomes increasingly restrictive as the number of classes $K \approx n/n_i$ grows, since more communities require stronger separation. In the node-information-only regime, recovery is governed by the quality of node features, with the admissible noise bounded by $\rho \leq \gamma n_i$, scaling linearly with class size.

The synergistic regime is the most interesting: here both graph structure and node features interact, and a clear trade-off emerges between tolerance to feature noise and tolerance to misconnected edges. As seen in (13), decreasing μ from infinity (the node-information-only setting) introduces an additional term, which enlarges the admissible range of ρ and thus allows for higher noise tolerance. At the same time, from (13), $p_{\max}$ no longer depends on the number of classes K and can be relaxed by either choosing γ close to ρ/n_i or by reducing μ. However, this improvement in $p_{\max}$ comes at the expense of reducing the admissible range for ρ. In other words, there is a tension: one can either tolerate noisier features at the cost of stricter structural requirements, or tolerate higher misconnection rates in the graph at the cost of requiring cleaner features. This trade-off highlights the importance of balancing graph and feature contributions when designing algorithms, as exploiting both sources simultaneously provides the broadest recovery guarantees.

Algorithm 1 CADO Algorithm

Input: Number of atoms r, graph $\bar{\boldsymbol{A}}$, node data $\{\boldsymbol{z}_v\}$, step size sequence $\{\gamma_t = 2/t+2\}$
Initialize $\{\bar{\boldsymbol{e}}_i^{(0)} \in \mathcal{U}\}_{i=1}^r$, $\{\bar{\boldsymbol{\theta}}_i^{(0)} \in \Theta\}_{i=1}^r$
for $t = 0, 1, 2, \ldots$ until convergence **do**
    // Embedding Update via Conditional Gradient
    Compute gradients $\nabla_{\bar{\boldsymbol{e}}_i}\phi$ using current $\bar{\boldsymbol{\theta}}_i^{(t)}$
    Solve LMO (15); let $\tilde{\boldsymbol{E}}^{(t)} = \{\tilde{\boldsymbol{e}}_i^{(t)}\}$ be the solution
    Update: $\bar{\boldsymbol{e}}_i^{(t+1)} = (1-\gamma_t)\bar{\boldsymbol{e}}_i^{(t)} + \gamma_t\tilde{\boldsymbol{e}}_i^{(t)}$
    // Model Update via Conditional Gradient
    Compute gradients $\nabla_{\bar{\boldsymbol{\theta}}_i}\phi$ using updated $\bar{\boldsymbol{e}}_i^{(t+1)}$
    Solve LMO (14); let $\tilde{\boldsymbol{\theta}}_i^{(t)}$ be the solution
    Update: $\bar{\boldsymbol{\theta}}_i^{(t+1)} = (1-\gamma_t)\bar{\boldsymbol{\theta}}_i^{(t)} + \gamma_t\tilde{\boldsymbol{\theta}}_i^{(t)}$
Return: $\{\bar{\boldsymbol{e}}_i^{(T)}\}, \{\bar{\boldsymbol{\theta}}_i^{(T)}\}$

5 Complexity Analysis and Algorithmic Solutions

In this section, we analyze the computational aspects of solving (6). In Appendix C, we show that an iterative procedure exists that achieves polynomial-time convergence to an ϵ-optimal solution. While theoretically appealing, this algorithm is still computationally expensive and does not scale to large graphs. To address this limitation, we propose a practical and scalable method, referred to as *constrained atomic decomposition optimization solver* (CADO), which solves the non-convex formulation in (4) under a fixed number of atoms. This fixed-rank approximation is well motivated by the SON regularization, which naturally promotes sparsity in the active atoms, and in practice we find that CADO consistently converges to the optimal solution.

CADO: An efficient algorithmic solution. To solve (6) with a fixed number of atoms, we propose an alternating conditional gradient algorithm, termed CADO. The algorithm alternates between updating the embedding vectors $\{\bar{\boldsymbol{e}}_i\}$, which define the low-rank matrix $\boldsymbol{L}$, and updating the cluster-level models $\{\bar{\boldsymbol{\theta}}_i\}$, which induce the node-specific models $\{\boldsymbol{\theta}_v\}$. This alternating structure allows CADO to minimize the joint objective efficiently without explicitly enumerating the infinite set of atoms.

Each update step follows a conditional gradient (Frank–Wolfe) approach. At iteration t, a linear minimization oracle (LMO) is solved for both the embeddings and the model parameters, yielding descent directions:

$$\tilde{\boldsymbol{\theta}}_i = \arg\min_{\bar{\boldsymbol{\theta}}_i \in \Theta} \quad \left\langle \nabla_{\bar{\boldsymbol{\theta}}_i}\phi,\, \bar{\boldsymbol{\theta}}_i \right\rangle, \tag{14}$$

$$\tilde{\boldsymbol{E}} = \arg\min_{\{\bar{\boldsymbol{e}}_i\}_{i=1}^r} \quad \sum_{i=1}^r \left\langle \nabla_{\bar{\boldsymbol{e}}_i}\phi,\, \bar{\boldsymbol{e}}_i \right\rangle, \tag{15}$$

$$\text{s.t.} \quad \boldsymbol{L} = \sum_{i=1}^r \bar{\boldsymbol{e}}_i\bar{\boldsymbol{e}}_i^\top \in \mathcal{B}, \quad \boldsymbol{\theta}_v = \sum_{i=1}^r \bar{\epsilon}_{i,v}^2\bar{\boldsymbol{\theta}}_i \in \Theta,$$

where $\bar{\epsilon}_{i,v}$ denotes the v-th entry of $\bar{\boldsymbol{e}}_i$. The exact solutions of these LMOs depend on the loss function f_v and the domain Θ. In the case study presented in Section 6.1, we show that both LMOs admit efficient closed-form solutions. Detailed derivations and alrorithmic steps are provided in Appendix D.

After obtaining the LMOs, the models $\boldsymbol{\theta}_v$ and embeddings $\bar{\boldsymbol{e}}_i$ are updated via a convex combination with the results from previous iterations, ensuring feasibility throughout iterations. While the alternating conditional gradient approach leads to a non-convex problem, in practice CADO converges quickly to stable solutions, as supported by our experimental results. The procedure is summarized in Algorithm 1.

6 EXPERIMENTS

For our experiments, we focus on a particular case study (Section 6.1) that admits closed-form solutions for (14) and (15). The corresponding specialized version of CADO is summarized in Algorithm 2 in Appendix D. We then evaluate our framework and CADO on node classification within this case study. Specifically, we first demonstrate the bidirectional interaction between graph structure and node information, and then empirically assess the effectiveness of CADO. Additional experiments are reported in Appendix E, further supporting our theoretical and algorithmic findings.

6.1 CASE STUDY

Our general framework in (6) accommodates a wide range of settings through different choices of the loss functions f_v. To make the discussion concrete, we focus here on a specific case. We assume that the vectors $\boldsymbol{z}_v$ are statistically independent and, within each class, identically distributed. Furthermore, in the training set, conditional on the true class label y_v, the features and the labels are independent, i.e., $p(\boldsymbol{x}_v, \tilde{y}_v \mid y_v = i) = p(\boldsymbol{x}_v \mid \boldsymbol{R}_i)p(\tilde{y}_v \mid \boldsymbol{\pi}_i)$.

We model the feature vectors as m-dimensional centered Gaussians, $\boldsymbol{x}_v \sim \mathcal{N}(0, \bar{\boldsymbol{R}}_i)$, where $\bar{\boldsymbol{R}}_i$ is the class-dependent covariance matrix. Features in each class are assumed to concentrate near a distinct linear subspace. Specifically, the eigenvalues of $\bar{\boldsymbol{R}}_i$ are divided into two groups: those larger than a fixed positive threshold ρ_+, corresponding to a *signal subspace*, and those smaller than another threshold ρ_-, corresponding to a *noise subspace*. Naturally, we require $\rho_+ > \rho_-$.

In the training set of class i, we assume that the noisy label $\tilde{y}_v = j$ is observed with probability $\bar{\pi}_{ji}$, and define $\bar{\boldsymbol{\pi}}_i = (\bar{\pi}_{ji})_j$. We further assume that $\bar{\pi}_{ii}$ is significantly larger than $\bar{\pi}_{ji}$ for $j \neq i$, i.e., the probability of observing the correct label is significantly higher than that of any incorrect label.

To instantiate (6) in this setting, we choose Θ and f_v as follows. Each model parameter $\boldsymbol{\theta}$ consists of a covariance $\boldsymbol{R}$ and a label distribution $\boldsymbol{\pi}$. The node-level losses are defined by

$$f_{\text{feature}}(\boldsymbol{x}; \boldsymbol{R}) \coloneqq \frac{1}{m}\left(\boldsymbol{x}^\top \boldsymbol{R}^{-1}\boldsymbol{x} + \text{Tr}(\boldsymbol{R})\right), \tag{16}$$

$$f_{\text{label}}(\tilde{y} = j; \boldsymbol{\pi}) \coloneqq -\pi_j. \tag{17}$$

and combined as

$$f_v(\boldsymbol{z}; \boldsymbol{\theta}) = \begin{cases} f_{\text{feature}}(\boldsymbol{x}; \boldsymbol{R}) + \beta f_{\text{label}}(\tilde{y}; \boldsymbol{\pi}) & v \in \mathcal{V}_{\text{train}} \\ f_{\text{feature}}(\boldsymbol{x}; \boldsymbol{R}) & v \in \mathcal{V}_{\text{test}} \end{cases} \tag{18}$$

We constrain $\boldsymbol{R}$ to lie in the set $\mathbb{S}_{\rho_-,\rho_+}$ of symmetric matrices with eigenvalues in $[\rho_-, \rho_+]$, i.e.,

$$\mathbb{S}_{\rho_-,\rho_+} = \left\{\boldsymbol{R} \mid \boldsymbol{R} = \boldsymbol{R}^\top,\ \forall \boldsymbol{x} \in \mathbb{R}^n :\ \rho_-\|\boldsymbol{x}\|_2^2 \leq \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{x} \leq \rho_+\|\boldsymbol{x}\|_2^2\right\}. \tag{19}$$

Similarly, we constrain $\boldsymbol{\pi}$ to lie in the standard simplex $\Delta = \{\boldsymbol{\pi} = (\pi_k) \mid \pi_k \geq 0,\ \sum_k \pi_k = 1\}$, Accordingly, we take $\Theta = \mathbb{S}_{\rho_-,\rho_+} \times \Delta$.

6.2 EXPERIMENTAL SETUP

Data generation model. We generate a synthetic dataset with K clusters of equal size n_0, so that the total numnber of nodes is $n = Kn_0$. The graph is drawn from an SBM, while node features follow a Gaussian mixture model.

Graph structure: Edges are placed independently, with nodes in the same cluster connected with probability p and nodes in different clusters connected with probability q.

Node features: Each cluster generates zero-mean Gaussian feature vectors with a covariance matrix whose eigenvalues encode signal and noise. Specifically, m_ω eigenvalues are set to ω^2 (noise), while the remaining eigenvalues are set to σ^2 (signal). The parameters ω and m_ω jointly control the noise level and hence the degree of separability between the feature subspaces of different clusters.

Node labels: From each cluster, n_{t} nodes are selected as training examples. A training node is assigned the correct label with probability π and an incorrect label chosen uniformly at random otherwise. This design allows us to explore the effect of the fraction of labeled data and the level of label noise.

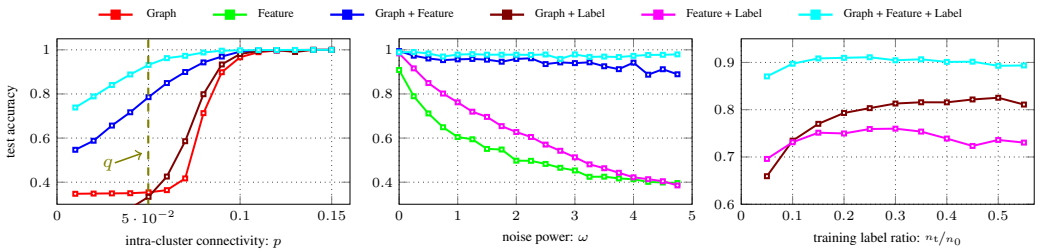

Figure 2: Left: Test accuracy vs. p; Middle: Test accuracy vs. ω; Right: Test accuracy vs. training label ratio. All plots highlight the synergy of combining graph, features, and labels, with the highest accuracy achieved by the full framework (Graph + Feature + Label).

Hyperparameters. A full list of hyperparameters is provided in Appendix E. Unless otherwise specified, we adopt the following defaults: $K = 3$ clusters with $n_0 = 300$ nodes each, and we fix the number of atoms to $r = K$. Edge probabilities are $p = 0.1$ within clusters and $q = 0.05$ across clusters. Node features are 6-dimensional, with $m_\omega = 4$ noisy eigenvalues of magnitude $\omega = 0.04$ in the covariance matrix, yielding well-separated feature subspaces. For training labels, we set the label ratio to $\frac{n_t}{n_0} = 0.2$ and assume no label noise ($\pi = 1$). To simplify the experimental setup, we reparameterize the original scaling setup by introducing effective weights for each term. While our theoretical framework uses a scaling of 1 for the graph term, μ for the feature term, and $\mu\beta$ for the label term, we fix the weights to $\beta_g = 1.0$, $\beta_f = 2.5$, and $\beta_l = 13.0$, corresponding to the graph, feature, and label terms, respectively.

Evaluation settings. To assess the contribution of each information source (graph structure, node features, and node labels) we perform an ablation study by selectively enabling or disabling each term in the objective. Specifically, we evaluate the graph-only setting, the feature-only setting, all pairwise combinations (graph+features, graph+levels, and features+labels), and the full model integrating all three. These ablations are implemented by setting the corresponding regularization weights to zero and solving the resulting problem with CADO, except for the graph-only case, where we apply spectral clustering instead of the ablated CADO variant.

6.3 Our Results

We present three sets of experiments in Figure 2, each isolating the impact of one component while keeping the others fixed. For each configuration, we report accuracy on test nodes.

Graph structure (left figure). We vary the intra-cluster edge probability p while keeping features and labels fixed. Our framework consistently outperforms graph-only and pairwise baselines, especially as the graph becomes less informative (i.e., $p \approx q$).

Feature quality (center figure). We degrade the signal-to-noise ratio by increasing the noise level ω in the feature covariance. Even when features become unreliable, our method remains robust by leveraging the graph structure.

Training label availability (right figure). We vary the ratio of labeled training nodes. Our framework effectively exploits even a small number of noisy labels when integrating graph and feature information.

Detailed discussions of these experiments together with additional experiments are provided in Appendix E, providing further support for our theoretical and algorithmic contributions. Overall, our results confirm that the proposed framework effectively integrates graph, feature, and label information. CADO consistently recovers the true structure when the theoretical conditions are met and remains robust as the problem becomes harder ($p \downarrow, \omega \uparrow$). The first plot illustrates how features and labels enhance spectral clustering when the graph is weak, while the latter two show how graph structure strengthens classification when node-specific information is degraded.

Conclusion. We proposed a novel optimization framework for transductive node classification, formulated through the lens of convex graph clustering. Our approach integrates graph structure, node features, and labels within a unified convex formulation, and we developed an efficient algorithmic solution based on a fixed number of atoms. We established theoretical guarantees for perfect recovery, showing that combining graph and feature information requires strictly milder conditions than using either source alone. Experimental results corroborate our theory, demonstrating that node features enhance graph clustering while graph structure strengthens classification.

REFERENCES

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on information theory*, 62(1):471–487, 2015.

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14, 2001.

Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. *Social network data analytics*, pp. 115–148, 2011.

Badri Narayan Bhaskar, Gongguo Tang, and Benjamin Recht. Atomic norm denoising with applications to line spectral estimation. *IEEE Transactions on Signal Processing*, 61(23):5987–5999, 2013.

Norbert Binkiewicz, Joshua T Vogelstein, and Karl Rohe. Covariate-assisted spectral clustering. *Biometrika*, 104(2):361–377, 2017.

Guillaume Braun and Masashi Sugiyama. Vec-sbm: Optimal community detection with vectorial edges covariates. In *International Conference on Artificial Intelligence and Statistics*, pp. 532–540. PMLR, 2024.

Guillaume Braun, Hemant Tyagi, and Christophe Biernacki. An iterative clustering algorithm for the contextual stochastic block model with optimality guarantees. In *International Conference on Machine Learning*, pp. 2257–2291. PMLR, 2022.

Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *The Journal of Machine Learning Research*, 15(1):2213–2238, 2014.

Eric C Chi and Stefan Steinerberger. Recovering trees with convex clustering. *SIAM Journal on Mathematics of Data Science*, 1(3):383–407, 2019.

Julien Chiquet, Pierre Gutierrez, and Guillem Rigaill. Fast tree inference with weighted fusion penalties. *Journal of Computational and Graphical Statistics*, 26(1):205–216, 2017.

Petr Chunaev. Community detection in node-attributed social networks: a survey. *Computer Science Review*, 37:100286, 2020.

Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual stochastic block models. *Advances in Neural Information Processing Systems*, 31, 2018.

Maximilien Dreveton, Felipe Fernandes, and Daniel Figueiredo. Exact recovery and bregman hard clustering of node-attributed stochastic block model. *Advances in Neural Information Processing Systems*, 36:37827–37848, 2023.

Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory*, 62(5):2788–2797, 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*, pp. 1, 2011.

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

Yaofang Hu and Wanjie Wang. Network-adjusted covariates for community detection. *Biometrika*, 111(4):1221–1240, 2024.

Adam Jaeger and David Banks. Cluster analysis: A modern statistical review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 15(3):e1597, 2023.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Ramya Korlakai Vinayak, Samet Oymak, and Babak Hassibi. Graph clustering with missing data: Convex algorithms and analysis. *Advances in Neural Information Processing Systems*, 27, 2014.

Xiaodong Li, Yudong Chen, and Jiaming Xu. Convex relaxation methods for community detection. *Statistical science*, 36(1):2–15, 2021. ISSN 0883-4237.

Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *2011 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 201–204. IEEE, 2011.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43–52, 2015.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In *International workshop on mining and learning with graphs (MLG)*, volume 8, 2012.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

Ashkan Panahi, Devdatt Dubhashi, Fredrik D Johansson, and Chiranjib Bhattacharyya. Clustering by sum of norms: Stochastic incremental algorithm, convergence and cluster recovery. In *International conference on machine learning*, pp. 2769–2777. PMLR, 2017.

Nikhil Rao, Parikshit Shah, and Stephen Wright. Forward–backward greedy algorithms for atomic norm regularization. *IEEE Transactions on Signal Processing*, 63(21):5798–5811, 2015.

Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3), 2008.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Mahdi Soltanolkotabi and Emmanuel J Candes. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 2012.

Defeng Sun, Kim-Chuan Toh, and Yancheng Yuan. Convex clustering: Model, theoretical guarantee and efficient algorithm. *Journal of Machine Learning Research*, 22(9):1–32, 2021.

Kean Ming Tan and Daniela Witten. Statistical properties of convex clustering. *Electronic journal of statistics*, 9(2):2324, 2015.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

Yihong Wu, Jiaming Xu, and Bruce Hajek. Achieving exact cluster recovery threshold via semidefinite programming under the stochastic block model. In *2015 49th Asilomar Conference on Signals, Systems and Computers*, pp. 1070–1074. IEEE, 2015.

Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

Bowei Yan and Purnamrita Sarkar. Covariate regularized community detection in sparse graphs. *Journal of the American Statistical Association*, 116(534):734–745, 2021.

Shenghao Yang and Kimon Fountoulakis. Weighted flow diffusion for local graph clustering with node attributes: An algorithm and statistical guarantees. In *International Conference on Machine Learning*, pp. 39252–39276. PMLR, 2023.

Changbo Zhu, Huan Xu, Chenlei Leng, and Shuicheng Yan. Convex optimization procedure for clustering: Theoretical revisit. *Advances in Neural Information Processing Systems*, 27, 2014.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *ICML*, 3:912–919, 2003.

MATHEMATICAL NOTATION.

To simplify notation and enhance readability, we use the following conventions throughout the paper. Collections such as $\{\lambda_a\}_{a\in\mathcal{A}}$, $\{L_{vv}\}_{v\in\mathcal{V}}$, $\{L_{uv}\}_{u,v\in\mathcal{V}}$, and $\{\boldsymbol{\theta}_v\}_{v\in\mathcal{V}}$ are often abbreviated to $\{\lambda_a\}$, $\{L_{vv}\}$, $\{L_{uv}\}$, and $\{\boldsymbol{\theta}_v\}$ when the indexing set is clear from context. When two nodes u and v belong to the same cluster, we write $u, v \in \mathcal{C}_i$; otherwise, we denote $u \in \mathcal{C}_i$ and $v \in \mathcal{C}_j$ to indicate distinct clusters.

We denote vectors by bold lowercase letters (e.g., $\boldsymbol{x}$) and matrices by bold uppercase letters (e.g., $\mathbf{W}$). The entry in the v^{th} row and u^{th} column of a matrix $\mathbf{W}$ is denoted by W_{vu}, and its transpose by $\mathbf{W}^\top$. The Euclidean inner product between vectors is written as $\langle\cdot,\cdot\rangle$, and the matrix inner product is defined by $\langle \boldsymbol{A}, \boldsymbol{C}\rangle := \mathrm{tr}(\boldsymbol{A}\boldsymbol{C}^\top)$, where $\mathrm{tr}(\cdot)$ denotes the trace operator.

The indicator function $I_\Theta(\boldsymbol{\theta})$ refers to the indicator of the feasible set Θ, and $\partial I_\Theta(\boldsymbol{\theta})$ denotes the normal cone of Θ at point $\boldsymbol{\theta}$. The cardinality of a set T is denoted by $|T|$. We use $\mathbf{1}_n$ and $\mathbf{0}_n$ to denote the n-dimensional all-one and all-zero vectors, respectively. An $m \times n$ all-zero matrix is denoted by $\boldsymbol{O}_{m\times n}$, with subscripts omitted when dimensions are clear from context.

A ATOMIC NORM REVIEW

In many signal processing and machine learning applications, the objective is to reconstruct a signal that admits a simple or structured representation. The concept of *atomic norms* provides a unifying framework for this purpose by promoting simplicity in the signal representation. Specifically, an atomic norm can be used as a convex regularizer that induces a desired structure, such as sparsity or low-rankness, by relying on a predefined set of fundamental building blocks called *atoms*.

A.1 ATOMS AND ATOMIC NORMS

The basic elements used to represent a signal are referred to as *atoms*. For a signal $\boldsymbol{x}$, the atomic set $\mathcal{A}$ is a collection of these basic elements, allowing the signal to be expressed as a nonnegative linear combination of a small number of atoms from $\mathcal{A}$. The atomic norm, denoted by $\|\boldsymbol{x}\|_{\mathcal{A}}$, quantifies the complexity of the signal in terms of how economically it can be represented using these atoms.

Formally, the atomic norm is defined as:

$$\|\boldsymbol{x}\|_{\mathcal{A}} = \inf\left\{\sum_i c_i \ : \ \boldsymbol{x} = \sum_i c_i \boldsymbol{a}_i,\ \boldsymbol{a}_i \in \mathcal{A},\ c_i \geq 0\right\}. \tag{20}$$

This norm is often employed as a regularizer in optimization problems to encourage solutions that are structurally simple, such as sparse vectors or low-rank matrices. Different choices of the atomic set $\mathcal{A}$ yield different atomic norms. Two important examples are as follows:

- **Sparsity (ℓ_1-norm)**: For sparse signals, the atomic set consists of the signed canonical basis vectors:
$$\mathcal{A}_{\ell_1} = \{\pm \boldsymbol{e}_i \mid i = 1, \ldots, n\},$$
where $\boldsymbol{e}_i$ denotes the i-th canonical basis vector in $\mathbb{R}^n$. The induced atomic norm is:
$$\|\boldsymbol{x}\|_{\mathcal{A}_{\ell_1}} = \sum_{i=1}^{n} |x_i| = \|\boldsymbol{x}\|_1,$$
which is the familiar ℓ_1-norm, widely used to promote sparsity.
- **Low-rank matrices (nuclear norm)**: For low-rank matrix recovery, such as in matrix completion or graph clustering, the atomic set consists of rank-one matrices with unit norm:
$$\mathcal{A}_* = \{\mathbf{u}\mathbf{v}^\top \mid \mathbf{u} \in \mathbb{R}^m,\ \mathbf{v} \in \mathbb{R}^n,\ \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1\}.$$
The induced atomic norm is:
$$\|\mathbf{X}\|_{\mathcal{A}_*} = \sum_{i=1}^{\min(m,n)} \sigma_i(\mathbf{X}) = \|\mathbf{X}\|_*,$$
which is the nuclear norm, equal to the sum of the singular values of $\mathbf{X}$. This norm is the convex surrogate of the rank function and promotes low-rank structure.
For symmetric matrices, the above definitions can be simplified. If $\mathbf{X} = \mathbf{X}^\top$, we can define the symmetric atomic sets
$$\mathcal{A}_{\mathrm{sym},+} \ = \ \{\, \mathbf{u}\mathbf{u}^\top : \|\mathbf{u}\|_2 = 1 \,\}, \qquad \mathcal{A}_{\mathrm{sym},\pm} \ = \ \{\, \pm\, \mathbf{u}\mathbf{u}^\top : \|\mathbf{u}\|_2 = 1 \,\}.$$
Then, with the eigenvalue decomposition $\mathbf{X} = \sum_i \lambda_i\, \mathbf{u}_i \mathbf{u}_i^\top$ (like graph Laplacian matrix):
$$\|\mathbf{X}\|_{\mathcal{A}_{\mathrm{sym},+}} \ = \ \sum_i \lambda_i \ = \ \mathrm{tr}(\mathbf{X}) \quad \text{for } \mathbf{X} \succeq 0,$$
and, for general symmetric $\mathbf{X}$,
$$\|\mathbf{X}\|_{\mathcal{A}_{\mathrm{sym},\pm}} \ = \ \sum_i |\lambda_i| \ = \ \|\mathbf{X}\|_*.$$
In particular, for symmetric matrices the nuclear norm equals the sum of absolute eigenvalues, and for symmetric PSD matrices it reduces to the trace.

B PROOF OF THEOREM 4.5

We begin with an overview of the proof methodology, followed by the problem setup and optimization formulation. We then outline the key elements, focusing on optimality conditions for exact recovery. The main proof is divided into three parts, each covering a different information regime: (i) node-specific, (ii) graph-based, and (iii) combined. For each, we construct a dual certificate via a "guess-and-golfing" approach and establish recovery conditions. We conclude by stating the theorem.

B.1 PROOF OVERVIEW

Our proof follows a standard dual certificate approach. We aim to verify that the ideal solution—comprising one atom per cluster—satisfies the optimality (KKT) conditions of the convex problem (6). The core idea is to construct a dual certificate matrix $\mathbf{Z}$ (and associated subgradients $\{\mathbf{g}_{vu}\}$) such that the ideal solution is locally optimal.

To achieve this, we employ a *guess-and-golfing strategy*. Starting from a natural or trivial guess for $\mathbf{Z}$ (e.g., diagonal or block-structured), we iteratively refine it to meet the necessary conditions. Each refinement step aims to adjust the certificate to satisfy one or more of the following:

- sign constraints ensuring dual feasibility;
- inner product constraints capturing primal-dual consistency;
- and positive semidefiniteness of the gap matrix enforcing inequality optimality.

B.2 PROBLEM SETUP

Optimization Problem Restatement. The optimization problem we analyze, with the goal of establishing conditions for perfect recovery, is our convex framework augmented with a sum-of-norms (SON) regularization term. For clarity and completeness, we restate it below, as in equation (6).

$$\begin{aligned} \min_{\boldsymbol{U}\in\mathcal{U},\,\{\lambda_{\boldsymbol{a}}\geq 0\}_{\boldsymbol{a}\in\mathcal{A}}} \quad & \phi(\boldsymbol{U}) + \mu_0 \sum_{\boldsymbol{a}\in\mathcal{A}} \lambda_{\boldsymbol{a}} + \mu_1 R(\boldsymbol{U}) \\ \text{s.t.} \quad & \boldsymbol{U} = \sum_{\boldsymbol{a}\in\mathcal{A}} \lambda_{\boldsymbol{a}} \boldsymbol{a} \end{aligned} \tag{21}$$

In this formulation, $\boldsymbol{U} := (\boldsymbol{L}, \{\boldsymbol{\theta}_v\}_{v\in\mathcal{V}})$. The feasible set is $\mathcal{U} := \mathcal{B} \times \Theta^{|\mathcal{V}|}$, where $\mathcal{B}$ denotes the set of symmetric matrices with entries in $[0,1]$ and unit diagonal, and Θ is the feasible domain for node-specific models. The objective $\phi(\boldsymbol{U})$ is defined as: $\phi(\boldsymbol{U}) := -\langle \bar{\boldsymbol{A}}, \boldsymbol{L}\rangle + \mu \sum_{v\in\mathcal{V}} f_v(\boldsymbol{z}_v; \boldsymbol{\theta}_v)$, and $R(\boldsymbol{U})$ is the SON regularization term. Each atom $\boldsymbol{a} \in \mathcal{A}$ is of the form $\boldsymbol{a} = \left(\boldsymbol{e}\boldsymbol{e}^\top, \{\epsilon_v^2 \boldsymbol{\theta}\}_{v\in\mathcal{V}}\right)$, and comes from the atomic dictionary $\mathcal{A}$ defined as:

$$\mathcal{A} := \left\{ \left(\boldsymbol{e}\boldsymbol{e}^\top, \{\epsilon_v^2 \boldsymbol{\theta}\}_{v\in\mathcal{V}}\right) \;\middle|\; \boldsymbol{e} = (\epsilon_v)_{v\in\mathcal{V}},\ \|\boldsymbol{e}\| = 1,\ \boldsymbol{\theta} \in \Theta \right\}. \tag{22}$$

Using the definitions above, the optimization problem (21) can be explicitly given as:

$$\begin{aligned} \min_{\substack{\boldsymbol{L}\in\mathcal{B},\,\{\boldsymbol{\theta}_v\in\Theta\}_{v\in\mathcal{V}},\\ \{\lambda_{\boldsymbol{a}}\geq 0\}_{\boldsymbol{a}\in\mathcal{A}}}} \quad & -\langle \bar{\boldsymbol{A}}, \boldsymbol{L}\rangle + \mu \sum_v f_v(\boldsymbol{z}_v, \boldsymbol{\theta}_v) + \mu_0 \sum_{\boldsymbol{a}\in\mathcal{A}} \lambda_{\boldsymbol{a}} + \mu_1 \sum_{u<v} \|\boldsymbol{\theta}_v - \boldsymbol{\theta}_u\| \\ \text{s.t.} \quad & \boldsymbol{L} = \sum_{a\in\mathcal{A}} \lambda_a \boldsymbol{e}_a \boldsymbol{e}_a^\top, \qquad \boldsymbol{\theta}_v = \sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,v}^2 \boldsymbol{\theta}_a \end{aligned} \tag{23}$$

Equivalent Optimization Problem. We simplify the problem in (23) by enforcing $L_{vv} = 1$ for all v, which implies that the sum of coefficients λ_a is fixed, and the conditions $L_{uv} \leq 1$ and $\boldsymbol{\theta}_v \in \Theta$ are automatically satisfied (see Lemma B.2). We also eliminate the auxiliary variables $\boldsymbol{\theta}_v$ by expanding their definition in the objective, leading to the following simplified problem:

$$\begin{aligned} \min_{\substack{\{L_{uv}\geq 0\}_{u,v},\\ \{\lambda_a\geq 0\}_{a\in\mathcal{A}}}} \quad & -\langle \bar{\boldsymbol{A}}, \boldsymbol{L}\rangle + \mu \sum_{v\in\mathcal{V}} f_v\left(\boldsymbol{z}_v,\ \sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,v}^2 \boldsymbol{\theta}_a\right) + \mu_1 \sum_{u<v} \left\| \sum_{a\in\mathcal{A}} \lambda_a \left(\epsilon_{a,u}^2 - \epsilon_{a,v}^2\right) \boldsymbol{\theta}_a \right\| \\ \text{s.t.} \quad & \boldsymbol{L} = \sum_{a\in\mathcal{A}} \lambda_a \boldsymbol{e}_a \boldsymbol{e}_a^\top, \qquad L_{vv} = 1, \quad \forall v \in \mathcal{V}. \end{aligned} \tag{24}$$

B.3 Proof Elements

Ideal Solution. We define the ideal (perfect recovery) solution as one consisting of K atoms—one per cluster—denoted by $\{(\lambda_i^*, \boldsymbol{e}_i^*, \boldsymbol{\theta}_i^*)\}_{i=1}^K$. These ideal atoms take the form:

$$\lambda_i^* = n_i, \qquad \epsilon_{i,v}^* = \begin{cases} \frac{1}{\sqrt{n_i}}, & v \in \mathcal{C}_i \\ 0, & \text{otherwise} \end{cases}, \qquad \boldsymbol{e}_i^* = (\epsilon_{i,v}^*)_{v\in\mathcal{V}}. \tag{25}$$

$$\{\boldsymbol{\theta}_i^*\}_{i=1}^K = \arg\min_{\{\boldsymbol{\theta}_i \in \Theta\}_i} \sum_{v\in\mathcal{V}} h_v(\boldsymbol{\theta}_{y_v}) \tag{26}$$

where $h_v(\boldsymbol{\theta}_i) = f_v(\boldsymbol{z}_v; \boldsymbol{\theta}_i) + \gamma \sum_{i<j} n_j \left\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j^*\right\|$. This yields an ideally partitioned matrix $\boldsymbol{L}$ with:

$$L_{uv}^* = \begin{cases} 1, & u, v \in \mathcal{C}_i \text{ for some } i \\ 0, & \text{otherwise} \end{cases}. \tag{27}$$

and the following *characteristic condition* derives from the first-order optimality condition of (26):

$$\frac{1}{n_i} \sum_{v\in\mathcal{C}_i} \nabla h_v(\boldsymbol{\theta}_{y_v}^*) := \boldsymbol{\omega}_i \in -\partial I_\Theta(\boldsymbol{\theta}_i^*) \tag{28}$$

where $\partial I_\Theta(\boldsymbol{\theta}_i^*)$ is the normal cone of Θ at $\boldsymbol{\theta}_i^*$ defined as

$$\partial I_\Theta(\boldsymbol{\theta}_i^*) := \begin{cases} \{\boldsymbol{\omega} \in \mathbb{R}^d : \langle \boldsymbol{\omega}, \boldsymbol{\theta} - \boldsymbol{\theta}_i^* \rangle \le 0,\ \forall \theta \in \Theta\} & \text{if} \quad \boldsymbol{\theta}_i^* \in \Theta \\ \varnothing & \text{if} \quad \boldsymbol{\theta}_i^* \notin \Theta \end{cases}. \tag{29}$$

Note that,

$$\nabla h_v(\boldsymbol{\theta}_i^*) = \nabla f_v(\boldsymbol{\theta}_i^*) + \gamma \sum_u \mathbf{g}_{vu}\,. \tag{30}$$

with $\gamma := \mu_1/\mu$. The terms $\mathbf{g}_{vu}$ in (30) correspond to subgradients of the SON regularization term, and can be calculated as:

$$\mathbf{g}_{vu} = \begin{cases} \text{any } \mathbf{g}_{vu} \text{ with } \|\mathbf{g}_{vu}\| \le 1, & \text{if } u, v \in \mathcal{C}_i \\ \frac{\boldsymbol{\theta}_i^* - \boldsymbol{\theta}_j^*}{\|\boldsymbol{\theta}_i^* - \boldsymbol{\theta}_j^*\|}, & \text{if } u \in \mathcal{C}_i,\ v \in \mathcal{C}_j,\ i \ne j. \end{cases} \tag{31}$$

By symmetry of the SON term, it is required that $\mathbf{g}_{vu} = -\mathbf{g}_{uv}$. As a results, its easy to show that

$$\boldsymbol{\omega}_i = \nabla F_i(\boldsymbol{\theta}_i^*) + \gamma \sum_{i<j} n_j \frac{\boldsymbol{\theta}_i^* - \boldsymbol{\theta}_j^*}{\|\boldsymbol{\theta}_i^* - \boldsymbol{\theta}_j^*\|} \tag{32}$$

where $\nabla F_i(\boldsymbol{\theta}) := \frac{1}{n_i} \sum_{v\in\mathcal{C}_i} \nabla f_v(\boldsymbol{z}_v; \boldsymbol{\theta})$.

Optimality Conditions. To establish that the ideal solution is optimal for the simplified problem (24), it suffices to verify the first-order optimality (KKT) conditions *at the ideal solution*. Specifically, we aim to construct a dual certificate: a symmetric matrix $\mathbf{Z} \in \mathbb{R}^{n\times n}$ and a collection of subgradients $\{\mathbf{g}_{vu}\}$ such that the following optimality conditions are satisfied:

$$(\mathbf{Z} - \bar{\boldsymbol{A}})_{uv} \le 0, \qquad \text{if } u, v \in \mathcal{C}_i, \tag{33}$$

$$(\mathbf{Z} - \bar{\boldsymbol{A}})_{uv} \ge 0, \qquad \text{if } u \in \mathcal{C}_i,\ v \in \mathcal{C}_j,\ i \ne j, \tag{34}$$

$$\boldsymbol{e}_i^{*\top}\big(\mu \boldsymbol{D}(\boldsymbol{\theta}_i^*) - \mathbf{Z}\big)\boldsymbol{e}_i^* = 0, \qquad \forall i, \tag{35}$$

$$\boldsymbol{e}^\top\big(\mu \boldsymbol{D}(\boldsymbol{\theta}) - \mathbf{Z}\big)\boldsymbol{e} \ge 0, \qquad \forall (\boldsymbol{e}, \boldsymbol{\theta}) \in \mathcal{A}. \tag{36}$$

where $\boldsymbol{D}(\boldsymbol{\theta}) := \operatorname{diag}(d_v(\boldsymbol{\theta}))$ with

$$d_v(\boldsymbol{\theta}) = \left\langle \nabla h_v(\boldsymbol{\theta}_{y_v}^*),\ \boldsymbol{\theta} \right\rangle\,. \tag{37}$$

Conditions (33) and (34) ensures complementary slackness with respect to the matrix variable $\boldsymbol{L}$. Also, importantly, the diagonal elements of $\mathbf{Z}$ are unconstrained and need not satisfy the sign conditions, as they are not involved in the objective due to the fixed diagonal constraint $L_{vv} = 1$.

Inequality (36) must hold for all atoms $(\boldsymbol{e}, \boldsymbol{\theta}) \in \mathcal{A}$, while equality (35) holds for the K ideal atoms $\{(\boldsymbol{e}_i^*, \boldsymbol{\theta}_i^*)\}_{i=1}^K$.

Hence, the KKT conditions reduce to constructing a dual matrix $\mathbf{Z}$ and a consistent set of subgradients $\{\mathbf{g}_{vu}\}$ that jointly satisfy equations (33)–(36). Note that $\mathbf{g}_{vu}$ is undefined when $u, v \in \mathcal{C}_i$ (i.e., both nodes belong to the same cluster). In this case, $\mathbf{g}_{vu}$ acts as a free design parameter and must satisfy the subgradient constraint $\|\mathbf{g}_{vu}\| \leq 1$. We jointly construct the collection $\{\mathbf{g}_{vu}\}_{u,v \in \mathcal{C}_i}$ and the dual certificate $\mathbf{Z}$ such that the full set of optimality conditions (33)–(36) are all satisfied.

B.4 Perfect Recovery with Combined Graph and Node-specific Information

The goal of this section is to demonstrate that by *combining graph structure and node-specific information*, we obtain *looser requirements* on the parameters λ, μ, and ρ for perfect recovery.

B.4.1 Constructing $\mathbf{Z}$

We begin by constructing a candidate for the dual certificate $\mathbf{Z}$ and the subgradients $\{\mathbf{g}_{vu}\}$ that satisfy the optimality conditions (33)- (36). We proceed via a step-by-step design, where we iteratively refine a candidate $\mathbf{Z}$ to satisfy each condition.

Step 1: Structured Guess. We begin with a matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ defined blockwise according to:

$$S_{uv} := \begin{cases} 0 & \text{if } u, v \in \mathcal{C}_i \text{ and } \{u, v\} \in E, \\ -a_{ii} & \text{if } u, v \in \mathcal{C}_i \text{ and } \{u, v\} \notin E, \\ 0 & \text{if } u \in \mathcal{C}_i,\ v \in \mathcal{C}_j,\ \{u, v\} \notin E, \\ a_{ij} & \text{if } u \in \mathcal{C}_i,\ v \in \mathcal{C}_j,\ \{u, v\} \in E, \end{cases}$$

where $a_{ii}, a_{ij} > 0$ are scalar parameters. This form penalizes incorrect intra-cluster disconnections and inter-cluster connections.

Step 2: Projection for Orthogonality. To ensure that $\mathbf{Z}$ is orthogonal to the ideal atoms (i.e., satisfies (35)), we project $\mathbf{S}$ onto the orthogonal complement of the span of ideal atoms:

$$\mathbf{Z}_{\mathrm{s}} := \mathbf{P}^{\perp}(\mathbf{S} - \lambda \mathbf{I})\mathbf{P}^{\perp}, \quad \text{where } \mathbf{P}^{\perp} := \mathbf{I} - \boldsymbol{E}\boldsymbol{E}^{\top},$$

and $\boldsymbol{E} = [\boldsymbol{e}_1^*, \ldots, \boldsymbol{e}_K^*]$ is the matrix of ideal cluster indicators. This construction ensures ${\boldsymbol{e}_i^*}^{\top} \mathbf{Z}_{\mathrm{s}} \boldsymbol{e}_i^* = 0$.

Step 3: Feature Guess. To account the effect of $\mu \boldsymbol{D}(\boldsymbol{\theta}_i^*)$ we will add the term $\mu \bar{\boldsymbol{D}} := \mu \operatorname{diag}\left(d_v(\boldsymbol{\theta}_{y_v}^*)\right)$ to $\mathbf{Z}$ with $d_v(\boldsymbol{\theta})$ is given in equation (37). This ensures that $\left(\boldsymbol{D}(\boldsymbol{\theta}_i^*) - \bar{\boldsymbol{D}}\right)_{vv} = 0 \quad \forall v, i$

The dual certificate defined as:

$$\mathbf{Z} = \mathbf{Z}_{\mathrm{s}} + \mu \bar{\boldsymbol{D}}, \tag{38}$$

B.4.2 Verifying Optimality Condition

The optimality condition (35) holds due to $\bar{\boldsymbol{D}}$'s diagonal structure and the projection $\mathbf{P}^{\perp}$.

B.4.3 Verifying Sign Conditions

Next, we enforce the sign constraints (33)-(34) by analyzing the entrywise form of $\mathbf{Z} - \bar{\boldsymbol{A}}$. Using Assumptions 4.1–4.2, we compute worst-case deviations after projection and derive bounds:

$$\frac{1 + \lambda/n_i}{1 - \rho_{ii} - 2\delta} \leq a_{ii} \leq \frac{1 - \lambda/n_i}{\rho_{ii} + 2\delta},$$
$$\frac{1}{1 - \rho_{ij} - 2\delta} \leq a_{ij} \leq \frac{1}{\rho_{ij} + 2\delta}.$$

To ensure that the sign conditions remain valid, we must enforce $\frac{1+\lambda/n_i}{1-\rho_{ii}-2\delta} \leq \frac{1-\lambda/n_i}{\rho_{ii}+2\delta}$ which lead to

$$\lambda \leq (1 - 2p_{\max})\, n_i \tag{39}$$

where $p_{\max} := \rho^+ + \delta$ with $\rho^+ = \max_{i,j} \rho_{ij}^+$.

B.4.4 Verifying Positive Definiteness.

Finally, to ensure $\mathbf{Z} \succeq 0$ as required by (36), we must choose λ large enough so that $\mathbf{P}^\perp(\lambda \mathbf{I} - \mathbf{S})\mathbf{P}^\perp$ has all non-positive eigenvalues.

Step 1: Design $\mathbf{g}_{vu}$. To enforce condition (36), first notice

$$\begin{aligned} \tilde{d}_v(\boldsymbol{\theta}) := \left(\boldsymbol{D}(\boldsymbol{\theta}) - \bar{\boldsymbol{D}}\right)_{vv} &= (d_v(\boldsymbol{\theta}) - d_v(\boldsymbol{\theta}^*_{y_v})) \\ &= \left\langle \nabla h_v(\boldsymbol{\theta}^*_{y_v}),\ \boldsymbol{\theta} - \boldsymbol{\theta}^*_{y_v} \right\rangle \\ &= \left\langle \nabla f_v(\boldsymbol{\theta}^*_{y_v}) + \gamma \sum_u \mathbf{g}_{vu},\ \boldsymbol{\theta} - \boldsymbol{\theta}^*_{y_v} \right\rangle \end{aligned} \tag{40}$$

By defining $\mathbf{g}_{vu}$ as

$$\mathbf{g}_{vu} = \begin{cases} \dfrac{\nabla f_u(\boldsymbol{\theta}^*_i) - \nabla f_v(\boldsymbol{\theta}^*_i)}{\gamma \cdot \rho}, & \text{if } u, v \in \mathcal{C}_i, \\ \dfrac{\boldsymbol{\theta}^*_i - \boldsymbol{\theta}^*_j}{\|\boldsymbol{\theta}^*_i - \boldsymbol{\theta}^*_j\|}, & \text{otherwise.} \end{cases} \tag{41}$$

we have the following result

$$\tilde{d}_v(\boldsymbol{\theta}) = \left\langle \boldsymbol{\tau_v} + \boldsymbol{\omega}_{y_v},\ \boldsymbol{\theta} - \boldsymbol{\theta}^*_{y_v} \right\rangle \geq 0. \tag{42}$$

where $\boldsymbol{\tau_v} = \left(1 - \frac{\gamma n_i}{\rho}\right)(\nabla f_v(\boldsymbol{\theta}_i) - \nabla F_i(\boldsymbol{\theta}_i))$

Step 2: Bounding Node-specific Term. Substituting the dual certificate $\mathbf{Z}$, we get:

$$\boldsymbol{\epsilon}^\top (\boldsymbol{D}(\boldsymbol{\theta}) - \mathbf{Z})\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^\top \left[\mathbf{Z}_{\mathrm{s}} + \mu \tilde{\boldsymbol{D}}(\boldsymbol{\theta})\right] \boldsymbol{\epsilon} \tag{43}$$

where $\tilde{\boldsymbol{D}}(\boldsymbol{\theta}) = \operatorname{diag}\left(\tilde{d}_v(\boldsymbol{\theta})\right)$ with $\tilde{d}_v(\boldsymbol{\theta}) = \left\langle \boldsymbol{\tau_v} + \boldsymbol{\omega}_{y_v}, \boldsymbol{\theta} - \boldsymbol{\theta}^*_{y_v} \right\rangle$. According to (28) we know that $\left\langle \boldsymbol{\omega}_{y_v}, \boldsymbol{\theta} - \boldsymbol{\theta}^*_{y_v} \right\rangle \geq 0 \quad \forall\, \boldsymbol{\theta} \in \Theta$. Moreover, under assumption 4.3, there exists at most one index i for which

$$\left\langle \boldsymbol{\omega}_{y_v}, \boldsymbol{\theta} - \boldsymbol{\theta}^*_{y_v} \right\rangle \leq R.$$

Without loss of generality, assume this is index $i = 0$. We define the diagonal matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ by:

$$\mathbf{Q} = \operatorname{diag}(r_v) \quad \text{with} \quad r_v = \begin{cases} 0 & v \in \mathcal{C}_i \\ R & v \notin \mathcal{C}_i \end{cases} .$$

Moreover, Based on assumption 4.3, the feasible set Θ is bounded as:

$$\|\boldsymbol{\theta} - \boldsymbol{\theta}_i\| \leq \ell, \quad \forall i,$$

Using the Cauchy-Schwarz inequality, we get:

$$\langle \boldsymbol{\tau_v}, \boldsymbol{\theta} - \boldsymbol{\theta}_i \rangle \geq -\zeta \mathbf{I}. \tag{44}$$

where $\zeta := \ell\,(\rho - \gamma \cdot n_i)$

Thus,

$$\tilde{\boldsymbol{D}}(\boldsymbol{\theta}) \succeq \mathbf{Q} - \zeta \mathbf{I}. \tag{45}$$

Step 3: PSD Conditions. Putting (45) into (43), we obtain:

$$\boldsymbol{\epsilon}^\top (\boldsymbol{D}(\boldsymbol{\theta}) - \mathbf{Z})\boldsymbol{\epsilon} \geq \boldsymbol{\epsilon}^\top (\mu\mathbf{Q} - \mu\zeta\mathbf{I} - \mathbf{Z}_{\mathrm{s}})\, \boldsymbol{\epsilon}. \tag{46}$$

We want the matrix

$$\boldsymbol{\Gamma} := \mathbf{P}^\perp(\lambda\mathbf{I} - \mathbf{S})\mathbf{P}^\perp + \mu\mathbf{Q} - \mu\zeta\mathbf{I}$$

to be positive semi-definite. By invoking Lemma B.3, we obtain sufficient conditions on λ, μ, and ρ that guarantee positive semi-definiteness, and hence, perfect recovery. To apply Lemma B.3, we define the following block components:

$$\boldsymbol{\Gamma}_{\mathbf{11}} = \left(\mathbf{P}^\perp(\lambda\mathbf{I} - \mathbf{S})\mathbf{P}^\perp\right)_{(\mathcal{C}_i, \mathcal{C}_i)} - \mu\zeta\mathbf{I}_{n_i} \tag{47}$$

$$\boldsymbol{\Gamma}_{\mathbf{22}} = \left(\mathbf{P}^\perp(\lambda\mathbf{I} - \mathbf{S})\mathbf{P}^\perp\right)_{(\mathcal{C}_{\neq i}, \mathcal{C}_{\neq i})} + \mu R\mathbf{I}_{n-n_i} - \mu\zeta\mathbf{I}_{n-n_i} \tag{48}$$

$$\boldsymbol{\Gamma}_{\mathbf{12}} = \left(-\mathbf{P}^\perp\mathbf{S}\mathbf{P}^\perp\right)_{(\mathcal{C}_i, \mathcal{C}_{\neq i})} \tag{49}$$

where for a matrix $\mathbf{M}$, the notation $\mathbf{M}_{(\mathcal{A},\mathcal{B})}$ denotes the submatrix with rows indexed by $\mathcal{A}$ and columns indexed by $\mathcal{B}$. Here, $\mathcal{C}_{\neq i} := \mathcal{V} \setminus \mathcal{C}_i$ is the set of nodes not belonging to $\mathcal{C}_i$.

Based on the block definitions in (49) and applying Lemma B.6, we can compute the quantities α, β, and $\sigma^2_{\max}(\boldsymbol{\Gamma}_{\mathbf{12}})$ required in Lemma B.3 as follows:

$$\alpha = \lambda - a \cdot p_{\max} \cdot n_i - \mu\zeta \tag{50}$$

$$\beta = \lambda - a \cdot p_{\max} \cdot (n - n_i) + \mu R - \mu\zeta \tag{51}$$

$$\sigma^2_{\max}(\boldsymbol{\Gamma}_{\mathbf{12}}) = a^2 \cdot p^2_{\max} \cdot n_i \cdot (n - n_i) \tag{52}$$

For positive semidefiniteness, the condition $\sigma^2_{\max}(\boldsymbol{\Gamma}_{\mathbf{12}}) \leq \alpha\beta$ must hold. Substituting the above expressions, this reduces to verifying

$$\lambda \cdot (\lambda - a \cdot p_{\max} \cdot n - \mu\zeta) + \mu \cdot R \cdot (\lambda - a \cdot p_{\max} \cdot n_i - \mu\zeta) \geq 0 \tag{53}$$

B.4.5 Combining Conditions

The sufficient conditions derived earlier can be unified by analyzing three distinct regimes, depending on whether the graph structure dominates, the node-specific information dominates, or both contribute jointly.

Case 1: Graph structure dominates. When $\mu \to 0$, only the graph structure contributes. In this case, the condition for positive semidefiniteness reduces to

$$\lambda \;\geq\; a \cdot p_{\max}\, n. \tag{54}$$

On the other hand, from the spectral bound (39), we also require

$$\lambda \;\leq\; (1 - 2p_{\max})\, n_i. \tag{55}$$

Combining the two yields the feasibility condition

$$p_{\max} \;\leq\; \frac{1}{2 + a \cdot n/n_i}. \tag{56}$$

Thus, recovery is possible only when the graph is sufficiently assortative: dense within clusters and sparse across clusters. The threshold depends on the ratio n/n_i, which can be approximated by the number of classes K.

Case 2: Node-specific information dominates. When $\mu \gg -\langle \bar{\boldsymbol{A}}, \boldsymbol{L} \rangle$, in particular when $\mu = \Omega(n \cdot n_i)$, the node-specific information outweighs the graph structure. In this case we require

$$\lambda \;\geq\; a \cdot p_{\max}\, n_i + \mu\zeta. \tag{57}$$

On the other hand, the spectral bound (39) still enforces

$$\lambda \;\leq\; (1 - 2p_{\max})\, n_i. \tag{58}$$

Combining the two yields the feasibility condition

$$\rho \;\leq\; \left(\frac{1 - (a+2)\, p_{\max}}{\mu\ell} + \gamma\right) n_i \;\approx\; \gamma\, n_i, \tag{59}$$

showing that perfect recovery depends primarily on the feature noise ρ, which can tolerate more with increasing nodes per clusters.

Case 3: Graph and node-specific information are synergistic. When $0 < \mu < n \cdot n_i$, both graph structure and node features contribute jointly. Two scenarios may arise:

1. If $\lambda \geq a \cdot p_{\max} n + \mu\zeta$, then combining with $\lambda \leq (1 - 2p_{\max})\, n_i$ leads to

$$\rho \;\leq\; \left(\frac{1 - (a \cdot n/n_i + 2)\, p_{\max}}{\mu\ell} + \gamma \right) n_i, \tag{60}$$

$$p_{\max} \;\leq\; \frac{1 - \mu\ell(\rho/n_i - \gamma)}{2 + a \cdot n/n_i}. \tag{61}$$

2. If $a \cdot p_{\max} n + \mu\zeta \geq \lambda \geq a \cdot p_{\max} n_i + \mu\zeta$, then combining with $\lambda \leq (1 - 2p_{\max})\, n_i$ gives

$$\left(\frac{1 - (a+2)\, p_{\max}}{\mu\ell} + \gamma \right) n_i \;\geq\; \rho \;\geq\; \left(\frac{1 - (a \cdot n/n_i + 2)\, p_{\max}}{\mu\ell} + \gamma \right) n_i, \tag{62}$$

$$\frac{1 - \mu\ell(\rho/n_i - \gamma)}{2 + a} \;\geq\; p_{\max} \;\geq\; \frac{1 - \mu\ell(\rho/n_i - \gamma)}{2 + a \cdot n/n_i}. \tag{63}$$

Combining both scenarios, the feasible region for perfect recovery can be summarized as

$$\rho \;\leq\; \left(\frac{1 - (a+2)\, p_{\max}}{\mu\ell} + \gamma \right) n_i, \tag{64}$$

$$p_{\max} \;\leq\; \frac{1 - \mu\ell(\rho/n_i - \gamma)}{2 + a}. \tag{65}$$

This highlights the advantage of combining both sources of information: perfect recovery is achievable under sparser graphs (smaller $p_{\max}$) and noisier features (larger ρ) than in the graph-only or node-only regimes.

B.5 Final Theorem

Theorem B.1. *Consider a fixed graph G with n nodes partitioned into K classes, where class i contains n_i nodes with $n_i \sim n_j$ for all i, j. Let ρ_{ij}^{+} denote the relative misconnection rate between classes, and define*

$$\rho^{+} := \max_{i,j} \rho_{ij}^{+}, \qquad p_{\max} = \rho^{+} + \delta, \qquad a = \max_{i,j} a_{ij}.$$

Fix $\gamma = \mu_1/\mu$, and suppose Assumptions 4.1–4.4 hold with fixed δ and R, and that the feasible set Θ is bounded. Then:

1. ***Graph-only regime** ($\mu \to 0$). Perfect recovery is achievable if*

$$p_{\max} \;\leq\; \frac{1}{2 + a \cdot n/n_i}.$$

2. ***Node-only regime** ($\mu = \Omega(n \cdot n_i)$). Perfect recovery is achievable if*

$$\rho \;\leq\; \gamma\, n_i.$$

3. ***Synergistic regime** ($0 < \mu < n \cdot n_i$). Perfect recovery is achievable if*

$$\rho \;\leq\; \left(\frac{1 - (a+2)\, p_{\max}}{\mu\ell} + \gamma \right) n_i, \tag{66}$$

$$p_{\max} \;\leq\; \frac{1 - \mu\ell(\rho/n_i - \gamma)}{2 + a}. \tag{67}$$

B.6 LEMMAS

Lemma B.2. *Let $\boldsymbol{\theta}_v := \sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,v}^2 \boldsymbol{\theta}_a$ and $\boldsymbol{L} := \sum_{a\in\mathcal{A}} \lambda_a \boldsymbol{e}\boldsymbol{e}^\top$, where each $\boldsymbol{\theta}_a \in \Theta$ and $\|\boldsymbol{e}\| = 1$. Assume that for all $v \in \mathcal{V}$, $\sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,v}^2 = 1$. Then, it follows that $\boldsymbol{\theta}_v \in \Theta$, $L_{uv} \le 1$ for all $u, v \in \mathcal{V}$, and $\sum_{a\in\mathcal{A}} \lambda_a = n$.*

Proof. If $\sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,v}^2 = 1$ holds, we have:

- Since each $\boldsymbol{\theta}_a \in \Theta$ and $\sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,v}^2 = 1$, it follows that $\boldsymbol{\theta}_v$ is a convex combination of elements in Θ, and hence $\boldsymbol{\theta}_v \in \Theta$ due to the convexity of Θ.
- For $\boldsymbol{L} = \sum_{a\in\mathcal{A}} \lambda_a \boldsymbol{e}\boldsymbol{e}^\top$, the Cauchy–Schwarz inequality implies

$$L_{uv} = \sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,u}\epsilon_{a,v} \le \sqrt{\sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,u}^2} \cdot \sqrt{\sum_{a\in\mathcal{A}} \lambda_a \epsilon_{a,v}^2} = 1.$$

- Summing the constraint over all v, considering $\|\boldsymbol{e}\|^2 = 1$, gives:

$$\sum_v \sum_a \lambda_a \epsilon_{a,v}^2 = \sum_a \lambda_a \sum_v \epsilon_{a,v}^2 = n.$$

□

Lemma B.3. *Consider the symmetric block matrix*

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma_{11}} & \boldsymbol{\Gamma_{12}} \\ \boldsymbol{\Gamma_{12}}^\top & \boldsymbol{\Gamma_{22}} \end{bmatrix},$$

where $\boldsymbol{\Gamma_{11}}, \boldsymbol{\Gamma_{22}}$ are symmetric matrices satisfying $\boldsymbol{\Gamma_{11}} \succeq \alpha\mathbf{I}$ and $\boldsymbol{\Gamma_{22}} \succeq \beta\mathbf{I}$ for some $\alpha, \beta \ge 0$. Then $\mathbf{D}$ is positive semidefinite if $\sigma^2(\boldsymbol{\Gamma_{12}}) \le \alpha\beta$, where $\sigma(\boldsymbol{\Gamma_{12}})$ denotes the largest singular value of $\mathbf{B}$.

Proof. Take any vector $\mathbf{x} = [\mathbf{x}_1^\top, \mathbf{x}_2^\top]^\top$. Then, the quadratic form associated with $\boldsymbol{\Gamma}$ can be written as:

$$\mathbf{x}^\top \boldsymbol{\Gamma} \mathbf{x} = \mathbf{x}_1^\top \boldsymbol{\Gamma_{11}} \mathbf{x}_1 + 2\mathbf{x}_1^\top \boldsymbol{\Gamma_{12}} \mathbf{x}_2 + \mathbf{x}_2^\top \boldsymbol{\Gamma_{22}} \mathbf{x}_2.$$

Using the given conditions $\mathbf{A} \succeq \alpha\mathbf{I}$ and $\boldsymbol{\Gamma_{22}} \succeq \beta\mathbf{I}$, we have: $\mathbf{x}_1^\top \boldsymbol{\Gamma_{11}} \mathbf{x}_1 \ge \alpha\|\mathbf{x}_1\|^2, \mathbf{x}_2^\top \boldsymbol{\Gamma_{22}} \mathbf{x}_2 \ge \beta\|\mathbf{x}_2\|^2$. Applying these lower bounds and the definition of singular value ($\|\boldsymbol{\Gamma_{11}}\mathbf{x}_2\| \le \sigma(\boldsymbol{\Gamma_{12}})\|\mathbf{x}_2\|$), it follows by the Cauchy–Schwarz inequality that:

$$\mathbf{x}^\top \boldsymbol{\Gamma} \mathbf{x} \ge \alpha\|\mathbf{x}_1\|^2 + \beta\|\mathbf{x}_2\|^2 - 2\sigma(\boldsymbol{\Gamma_{12}})\|\mathbf{x}_1\|\|\mathbf{x}_2\|.$$

Now, complete the square explicitly to factorize the expression clearly:

$$\mathbf{x}^\top \boldsymbol{\Gamma} \mathbf{x} \ge (\sqrt{\alpha}\|\mathbf{x}_1\| - \sqrt{\beta}\|\mathbf{x}_2\|)^2 + 2(\sqrt{\alpha\beta} - \sigma(\boldsymbol{\Gamma_{12}}))\|\mathbf{x}_1\|\|\mathbf{x}_2\|.$$

Since $\sigma^2(\boldsymbol{\Gamma_{12}}) \le \alpha\beta$ (implying $\sigma(\boldsymbol{\Gamma_{12}}) \le \sqrt{\alpha\beta}$), both terms in this expression are nonnegative. Thus:

$$\mathbf{x}^\top \boldsymbol{\Gamma} \mathbf{x} \ge 0, \text{ for all } \mathbf{x}.$$

Hence, $\boldsymbol{\Gamma}$ is positive semi-definite. □

Lemma B.4 (Spectral bound from sparsity and entry size)**.** *Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be symmetric. Suppose each row has at most $d_{\max}$ nonzero entries and $|M_{ij}| \le a$ for all i, j. Then*

$$\lambda_{\max}(\mathbf{M}) \;=\; \|\mathbf{M}\|_2 \;\le\; d_{\max}\, a.$$

Proof. Since $\mathbf{M}$ is symmetric, $\lambda_{\max}(\mathbf{M}) = \|\mathbf{M}\|_2$. Also,

$$\|\mathbf{M}\|_\infty \;=\; \max_i \sum_{j=1}^{n} |M_{ij}| \;\le\; d_{\max}\, a.$$

Because $\mathbf{M}$ is symmetric, $\|\mathbf{M}\|_1 = \|\mathbf{M}\|_\infty$. Hence

$$\|\mathbf{M}\|_2 \;\le\; \sqrt{\|\mathbf{M}\|_1\, \|\mathbf{M}\|_\infty} \;=\; \|\mathbf{M}\|_\infty \;\le\; d_{\max}\, a.$$

This yields the bound $\lambda_{\max}(\mathbf{M}) \le d_{\max} a$. □

Remark B.5. The bound $\lambda_{\max}(\mathbf{M}) \le a\, d_{\max}$ is tight up to constants in general. For example, if $\mathbf{M}$ is the adjacency matrix of a d-regular graph with $a = 1$, then $\lambda_{\max}(\mathbf{M}) = d = d_{\max}$.

Lemma B.6 (Lower bound for a projected principal block)**.** *Let $\mathbf{S} \in \mathbb{R}^{n\times n}$ be symmetric. Fix any index set $\mathcal{B} \subseteq \{1, \dots, n\}$ and let $\mathbf{S}_{\mathcal{B},\mathcal{B}}$ denote the corresponding principal submatrix. Assume:*

1. *Each row of $\mathbf{S}_{\mathcal{B},\mathcal{B}}$ has at most $d_{\max}(\mathcal{B})$ nonzeros;*
2. *$|S_{ij}| \le a$ for all i, j.*

Let $\mathbf{P}^\perp$ be any orthogonal projector ($\mathbf{P}^{\perp 2} = \mathbf{P}^\perp = \mathbf{P}^{\perp\top}$), and define

$$\mathbf{M}_{\mathcal{B}} \;:=\; \big(\mathbf{P}^\perp(\lambda\mathbf{I} - \mathbf{S})\mathbf{P}^\perp\big)_{\mathcal{B},\mathcal{B}}.$$

Then

$$\lambda_{\min}(\mathbf{M}_{\mathcal{B}}) \;\ge\; \min\Big\{0,\; \lambda - a\, d_{\max}(\mathcal{B})\Big\}.$$

Proof. Since $\mathbf{P}^\perp$ is an orthogonal projector, $\|\mathbf{P}^\perp\|_2 = 1$ and

$$\mathbf{M}_{\mathcal{B}} = \big(\lambda\, \mathbf{P}^\perp - \mathbf{P}^\perp\mathbf{S}\mathbf{P}^\perp\big)_{\mathcal{B},\mathcal{B}}.$$

Hence

$$\lambda_{\min}(\mathbf{M}_{\mathcal{B}}) \;\ge\; \min\Big\{0,\; \lambda_{\min}\Big((\lambda\mathbf{I} - \mathbf{S})_{\mathcal{B},\mathcal{B}}\Big)\Big\} \;\ge\; \min\Big\{0,\; \lambda - \lambda_{\max}\Big((\mathbf{S})_{\mathcal{B},\mathcal{B}}\Big)\Big\},$$

where we used $\|\mathbf{P}^\perp\|_2 = 1$ and submultiplicativity to drop $\mathbf{P}^\perp$ in the norm bound on the $\mathcal{B}\times\mathcal{B}$ block.

By Lemma B.4 applied to $\mathbf{S}_{\mathcal{B},\mathcal{B}}$,

$$\lambda_{\max}\Big((\mathbf{S})_{\mathcal{B},\mathcal{B}}\Big) \;\le\; a\, d_{\max}(\mathcal{B}),$$

which yields

$$\lambda_{\min}(\mathbf{M}_{\mathcal{B}}) \;\ge\; \min\Big\{0,\; \lambda - a\, d_{\max}(\mathcal{B})\Big\}.$$

□

C Polynomial-Time Convergence

Our analysis builds on the concept of a *linear minimization oracle (LMO)*, a standard tool in conditional gradient methods. The LMO for a dictionary $\mathcal{A}$ is defined as follows:

Definition C.1 (LMO)**.** The linear minimization oracle (LMO) of a dictionary $\mathcal{A}$ is a map $\mathcal{O}_{\mathcal{A}}(\mathbf{w})$ that, for any vector $\mathbf{w}$, returns an atom $\boldsymbol{a} \in \mathcal{A}$ minimizing the inner product $\boldsymbol{a}^\top \mathbf{w}$.

To establish our convergence result, we make the following assumption:

Assumption C.2. The feasible set $\mathcal{U}$ is closed and convex, the atomic set $\mathcal{A}$ is bounded, and the function $\phi(\boldsymbol{U})$ is convex and lower semi-continuous. Moreover, there exists an optimal solution $(\boldsymbol{U}^*, \{\lambda_i^*\}_i)$ of (5) and vectors $\boldsymbol{z}^*, \mathbf{n}^*, \mathbf{g}^*$ satisfying

$$\mathbf{n}^* \in \partial I_{\mathcal{U}}(\boldsymbol{U}^*), \quad \sup_{\boldsymbol{a}\in\mathcal{A}} \boldsymbol{a}^\top \boldsymbol{z}^* \leq \mu_0, \quad \mathbf{g}^* \in \partial\phi(\boldsymbol{U}^*), \quad \mathbf{0} \in \mathbf{g}^* + \boldsymbol{z}^* + \mathbf{n}^*. \tag{68}$$

Under this assumption, we obtain the following result:

Theorem C.3. *Under Assumption C.2, there exists an algorithm that returns an ϵ-approximate solution of* (5) *in* $\mathrm{poly}(1/\epsilon)$ *number of LMOs, proximal evaluations of ϕ, and orthogonal projections onto $\mathcal{U}$.*

The proof of Theorem C.3 is provided in the following. This result also extends to (6) by replacing ϕ with its composite form $\phi' := \phi + \mu_1 R$. While this establishes the polynomial-time solvability of the problem, the construction is primarily of theoretical interest and is not computationally efficient for large-scale instances. In particular, solving (5) exactly in practice is challenging due to the additional feasibility constraint $\boldsymbol{U} \in \mathcal{U}$. Although algorithms such as CoGEnT (Rao et al., 2015) efficiently handle the unconstrained version of the problem, extending them to the constrained setting remains an open question. To address these practical challenges, we turn to a scalable approach that operates on the non-convex formulation of the problem with fixed number of atoms and provides reliable performance in practice.

C.1 Proof of Theorem C.3

For convenience, we first provide a brief overview of the proof techniques employed.

C.1.1 Proof Overview

We cast the problem in (5) as a constrained atomic norm minimization, which we solve using a tailored ADMM-based algorithm. This method decouples the atomic norm regularization from additional structural constraints via a variable splitting strategy. To establish the complexity result, we demonstrate that our problem meets the conditions required for standard ADMM convergence, yielding an overall rate of $O(1/T)$.

C.1.2 Constrained Atomic Norm Optimization (CANO)

We consider the following general constrained atomic norm optimization problem, referred to as CANO:

$$\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) \quad \text{subject to} \quad \|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau, \quad \boldsymbol{x} \in \mathcal{X}, \tag{69}$$

where $\|\cdot\|_{\mathcal{A}}$ is the atomic norm, and $\mathcal{X} \subseteq \mathbb{R}^d$ is a closed and convex set encoding additional constraints (e.g., feasibility constraints in (5)). This formulation introduces computational challenges due to the coupling of a non-polyhedral norm and convex constraint sets.

C.1.3 CANO-ADMM Algorithm

To solve (69), we apply ADMM by introducing an auxiliary variable $\boldsymbol{z}$ and rewriting the problem as:

$$\min_{\boldsymbol{x},\boldsymbol{z}} \ f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{x} = \boldsymbol{z}, \quad \|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau, \quad \boldsymbol{z} \in \mathcal{X}. \tag{70}$$

The augmented Lagrangian is defined as:

$$\mathcal{L}_\beta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{x} - \boldsymbol{z}\rangle + \frac{\beta}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2, \tag{71}$$

where $\boldsymbol{\lambda}$ is the dual variable, and $\beta > 0$ is the penalty parameter. The ADMM updates proceed as follows:

- **Update $\boldsymbol{x}$:**

$$\boldsymbol{x}_{t+1} = \arg\min_{\|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau} f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}_t, \boldsymbol{x} - \boldsymbol{z}_t \rangle + \frac{\beta}{2} \|\boldsymbol{x} - \boldsymbol{z}_t\|_2^2. \tag{72}$$

  This step is solved using the CoGEnT algorithm (Rao et al., 2015), which is designed for atomic norm-constrained problems.

- **Update $\boldsymbol{z}$:**

$$\boldsymbol{z}_{t+1} = \arg\min_{\boldsymbol{z} \in \mathcal{X}} \langle \boldsymbol{\lambda}_t, \boldsymbol{x}_{t+1} - \boldsymbol{z} \rangle + \frac{\beta}{2} \|\boldsymbol{x}_{t+1} - \boldsymbol{z}\|_2^2. \tag{73}$$

  This corresponds to projecting onto the set $\mathcal{X}$, which is assumed to be tractable via a gradient-projection update:

$$\boldsymbol{z}_{t+1} = \mathsf{P}_{\mathcal{X}} \left(\boldsymbol{z}_t + \alpha(\boldsymbol{\lambda}_t + \beta(\boldsymbol{x}_{t+1} - \boldsymbol{z}_t)) \right),$$

  where α is a step size and $\mathsf{P}_{\mathcal{X}}$ denotes Euclidean projection onto $\mathcal{X}$.

- **Update $\boldsymbol{\lambda}$:**

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \beta(\boldsymbol{x}_{t+1} - \boldsymbol{z}_{t+1}). \tag{74}$$

C.1.4 Convergence Guarantees of CANO-ADMM

The convergence of ADMM with a rate of $O(1/T)$ in objective residuals and constraint violations is guaranteed under the following conditions:

1. $f(\boldsymbol{x})$ is convex and has a Lipschitz-continuous gradient;
2. The constraint sets $\|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau$ and $\mathcal{X}$ are both closed and convex;
3. The subproblems are solvable to sufficient accuracy at each iteration.

Our setup satisfies all these assumptions:

- The function $f(\boldsymbol{x})$ is convex and differentiable (e.g., it is the sum of a linear term and convex node-wise losses);
- The atomic norm ball $\|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau$ is convex by definition, and $\mathcal{X}$ is assumed to be convex and compact;
- The projection and LMO steps in CoGEnT are computationally tractable and converge efficiently.

Hence, CANO$-$ADMM converges at a rate $O(1/T)$ in the number of iterations. As each iteration requires a linear minimization oracle and projection operation, the overall runtime is polynomial in $1/\epsilon$, completing the proof of Theorem C.3.

D Details of the CADO Algorithm

This section provides the full implementation details of the CADO algorithm. We describe how the embedding vectors and model parameters are updated using conditional gradient steps, relying on problem-specific linear minimization oracles (LMOs). Each component is addressed in a separate subsection, followed by the complete algorithm pseudo-code.

D.1 Embedding Update via Conditional Gradient

In this step, we update the embedding vectors $\{\bar{\boldsymbol{e}}_i\}_{i=1}^r$, which define the low-rank matrix $\boldsymbol{L} = \sum_i \bar{\boldsymbol{e}}_i \bar{\boldsymbol{e}}_i^\top$, while keeping the class-wise models $\{\bar{\boldsymbol{\theta}}_i\}$ fixed. The update is performed by solving the following LMO:

$$\begin{aligned} \arg\min_{\{\bar{\boldsymbol{e}}_i\}_{i=1}^r} \quad & \sum_{i=1}^r \langle \nabla_{\bar{\boldsymbol{e}}_i} \phi,\, \bar{\boldsymbol{e}}_i \rangle \\ \text{s.t.} \quad & \boldsymbol{L} = \sum_i \bar{\boldsymbol{e}}_i \bar{\boldsymbol{e}}_i^\top \in \mathcal{B}, \quad \boldsymbol{\theta}_v = \sum_i \bar{\epsilon}_{i,v}^2 \bar{\boldsymbol{\theta}}_i \in \Theta \quad \forall v \in \mathcal{V}, \end{aligned} \tag{75}$$

where ϕ is the objective function from (4):

$$\phi = -\langle \bar{\boldsymbol{A}}, \boldsymbol{L} \rangle + \mu \sum_{v \in \mathcal{V}} f_v\left(\boldsymbol{z}_v;\, \boldsymbol{\theta}_v\right).$$

Constraint simplification. We now simplify the constraints using the structure of $\boldsymbol{L}$ and $\boldsymbol{\theta}_v$. First, the constraint $\boldsymbol{L} \in \mathcal{B}$ requires: $L_{vv} = 1$ for all v, $0 \le L_{uv} \le 1$ for all u, v, and symmetry.

Given $\boldsymbol{L} = \sum_i \bar{\boldsymbol{e}}_i \bar{\boldsymbol{e}}_i^\top$, the diagonal condition $L_{vv} = 1$ becomes:

$$L_{vv} = \sum_i \bar{\epsilon}_{i,v}^2 = 1.$$

This implies that the vector $\bar{\boldsymbol{\epsilon}}_v = (\bar{\epsilon}_{i,v})_i$ lies on the unit sphere, and that:

$$\boldsymbol{\theta}_v = \sum_i \bar{\epsilon}_{i,v}^2 \bar{\boldsymbol{\theta}}_i$$

is a convex combination of class-wise models, hence satisfying $\boldsymbol{\theta}_v \in \Theta$ automatically. Also, under this condition:

$$L_{uv} = \sum_i \bar{\epsilon}_{i,u} \bar{\epsilon}_{i,v} \le \sqrt{\sum_i \bar{\epsilon}_{i,u}^2} \cdot \sqrt{\sum_i \bar{\epsilon}_{i,v}^2} = 1,$$

and the upper bound constraint $L_{uv} \le 1$ is also satisfied. The only remaining constraint is the nonnegativity of L_{uv}, which is equivalent to:

$$L_{uv} = \sum_i \bar{\epsilon}_{i,u} \bar{\epsilon}_{i,v} \ge 0.$$

Reparameterization. To simplify the optimization, we define: $W_{v,i} \coloneqq \bar{\epsilon}_{i,v}^2$. Let $\mathbf{W} \in \mathbb{R}^{n \times r}$ collect these squared embedding values. Under this reparameterization, the low-rank matrix becomes $\boldsymbol{L} = \sum_{i=1}^r \sqrt{W_{:,i}} \sqrt{W_{:,i}}^\top$, and the node-specific models satisfy $\boldsymbol{\theta}_v = \sum_{i=1}^r W_{v,i} \bar{\boldsymbol{\theta}}_i$. The constraint $L_{vv} = \sum_i W_{v,i} = 1$ together with $W_{v,i} \ge 0$ implies that each row of $\mathbf{W}$ lies in the probability simplex.

Thus, the LMO reduces to:

$$\begin{aligned} \arg\min_{\mathbf{W} \in \mathbb{R}^{n \times r}} \quad & -\left\langle \bar{\boldsymbol{A}}, \sum_{i=1}^r \sqrt{W_{:,i}} \sqrt{W_{:,i}}^\top \right\rangle + \mu \sum_{v \in \mathcal{V}} f_v\left(\boldsymbol{z}_v; \sum_{i=1}^r W_{v,i} \bar{\boldsymbol{\theta}}_i\right) \\ \text{s.t.} \quad & \sum_i W_{v,i} = 1, \quad W_{v,i} \ge 0 \quad \forall v \in \mathcal{V}. \end{aligned} \tag{76}$$

We omit the non-negativity constraint $\boldsymbol{L} \ge 0$, as it is empirically satisfied due to the structure of $\bar{\boldsymbol{A}}$. If needed, it can be enforced via ADMM.

Gradient Computation. Let $\boldsymbol{R}_v = \sum_i W_{v,i}\bar{\boldsymbol{R}}_i$, $\boldsymbol{\pi}_v = \sum_i W_{v,i}\bar{\boldsymbol{\pi}}_i$. The gradient of the objective in (76) with respect to $W_{v,i}$ consists of:

- Structural term:
$$\frac{\partial}{\partial W_{v,i}}\left(-\langle \bar{\boldsymbol{A}}, \boldsymbol{L}\rangle\right) = -\sum_{u\in\mathcal{V}} \bar{A}_{u,v}\frac{\sqrt{W_{u,i}}}{\sqrt{W_{v,i}}}$$
- Feature loss:
$$\frac{\partial}{\partial W_{v,i}} f_{\text{feature}}(\boldsymbol{x}_v; \boldsymbol{R}_v) = \frac{1}{m}\left(-\boldsymbol{x}_v^\top \boldsymbol{R}_v^{-1}\bar{\boldsymbol{R}}_i\boldsymbol{R}_v^{-1}\boldsymbol{x}_v + \text{Tr}(\bar{\boldsymbol{R}}_i)\right)$$
- Label loss (training nodes only):
$$\frac{\partial}{\partial W_{v,i}} f_{\text{label}}(\tilde{y}_v; \boldsymbol{\pi}_v) = -\bar{\pi}_{i,\tilde{y}_v}$$

Solution. The optimization in (76) is linear in each row $\mathbf{W}_{v,:}$ and constrained to the simplex. Therefore, the optimal solution is a one-hot vector with 1 at the coordinate corresponding to the minimum gradient value:
$$\tilde{W}_{v,i} = \begin{cases} 1, & i = \arg\min_j \nabla_{W_{v,j}}\phi, \\ 0, & \text{otherwise.} \end{cases}$$
where
$$\nabla_{W_{v,i}}\phi = \frac{\partial}{\partial W_{v,i}}\left(-\langle \bar{\boldsymbol{A}}, \boldsymbol{L}\rangle\right) + \mu\sum_{v\in\mathcal{V}}\frac{\partial}{\partial W_{v,i}} f_{\text{feature}}(\boldsymbol{x}_v; \boldsymbol{R}_v) + \mu\beta\sum_{v\in\mathcal{V}_{\text{train}}}\frac{\partial}{\partial W_{v,i}} f_{\text{label}}(\tilde{y}_v; \boldsymbol{\pi}_v)$$

Once $\mathbf{W}$ is computed, we recover embeddings via the following equations. Taking positive roots preserves the symmetry and ensures non-negativity of $\boldsymbol{L}$.
$$\bar{\epsilon}_{i,v} = \sqrt{W_{v,i}}, \qquad \bar{\boldsymbol{e}}_i = (\bar{\epsilon}_{i,v})_{v=1}^n.$$

D.2 Model Update via Conditional Gradient

Given the updated embedding vectors $\{\bar{\boldsymbol{e}}_i\}_{i=1}^r$, we update the class-wise models $\{\bar{\boldsymbol{\theta}}_i = (\bar{\boldsymbol{R}}_i, \bar{\boldsymbol{\pi}}_i)\}_{i=1}^r$ by solving one LMO over each atom. Specifically, each model parameter is updated via:
$$\arg\min_{\bar{\boldsymbol{\theta}}_i\in\Theta}\left\langle \nabla_{\bar{\boldsymbol{\theta}}_i}\phi,\, \bar{\boldsymbol{\theta}}_i\right\rangle, \qquad \forall i \in \{1,\dots,r\}, \tag{77}$$
where ϕ is the global objective from (4), and the embedding matrix $\mathbf{W}$ (with $W_{v,i} = \bar{\epsilon}_{i,v}^2$) is fixed.

Since each $\bar{\boldsymbol{\theta}}_i$ consists of a feature model $\bar{\boldsymbol{R}}_i \in \mathbb{S}_{\rho_-,\rho_+}$ and a label distribution $\bar{\boldsymbol{\pi}}_i \in \Delta$, the LMO separates into two independent problems:
$$\arg\min_{\bar{\boldsymbol{R}}_i\in\mathbb{S}_{\rho_-,\rho_+}} \left\langle \nabla_{\bar{\boldsymbol{R}}_i}\phi,\, \bar{\boldsymbol{R}}_i\right\rangle, \tag{78}$$
$$\arg\min_{\bar{\boldsymbol{\pi}}_i\in\Delta} \left\langle \nabla_{\bar{\boldsymbol{\pi}}_i}\phi,\, \bar{\boldsymbol{\pi}}_i\right\rangle. \tag{79}$$

Gradient computation. Let $\boldsymbol{R}_v = \sum_{i=1}^r W_{v,i}\bar{\boldsymbol{R}}_i$ and $\boldsymbol{\pi}_v = \sum_{i=1}^r W_{v,i}\bar{\boldsymbol{\pi}}_i$. Then, the gradients are given by:

- Feature loss (all nodes):
$$\nabla_{\bar{\boldsymbol{R}}_i}\phi = \mu\sum_{v\in\mathcal{V}} W_{v,i}\cdot\nabla_{\boldsymbol{R}_v} f_{\text{feature}}(\boldsymbol{x}_v; \boldsymbol{R}_v),$$
where
$$\nabla_{\boldsymbol{R}_v} f_{\text{feature}}(\boldsymbol{x}_v; \boldsymbol{R}_v) = \frac{1}{m}\left(-\boldsymbol{R}_v^{-1}\boldsymbol{x}_v\boldsymbol{x}_v^\top\boldsymbol{R}_v^{-1} + \mathbf{I}\right).$$
- Label loss (training nodes only):
$$\nabla_{\bar{\boldsymbol{\pi}}_i}\phi = \mu\beta\sum_{v\in\mathcal{V}_{\text{train}}} W_{v,i}\cdot\nabla_{\boldsymbol{\pi}_v} f_{\text{label}}(\tilde{y}_v; \boldsymbol{\pi}_v),$$
where
$$\nabla_{\boldsymbol{\pi}_v} f_{\text{label}}(\tilde{y}_v; \boldsymbol{\pi}_v) = -\boldsymbol{e}_{\tilde{y}_v}$$

Closed-form solutions. Both optimization problems admit closed-form solutions:

- Update for $\bar{\boldsymbol{R}}_i$: The optimal solution to the linear minimization problem

$$\arg \min_{\bar{\boldsymbol{R}}_i \in \mathbb{S}_{\rho_-,\rho_+}} \left\langle \nabla_{\bar{\boldsymbol{R}}_i} \phi,\, \bar{\boldsymbol{R}}_i \right\rangle,$$

  is given by

$$\bar{\boldsymbol{R}}_i = U \operatorname{diag}(r_1, \ldots, r_m)\, U^\top,$$

  where $\nabla_{\bar{\boldsymbol{R}}_i} \phi = \boldsymbol{U} \operatorname{diag}(\lambda_1, \ldots, \lambda_m)\, \boldsymbol{U}^\top$ is the eigen-decomposition of the gradient, and

$$r_k = \begin{cases} \rho_- & \text{if } \lambda_k > 0, \\ \rho_+ & \text{if } \lambda_k < 0, \\ \text{any value in } [\rho_-, \rho_+] & \text{if } \lambda_k = 0. \end{cases}$$

- Update for $\bar{\boldsymbol{\pi}}_i$: Since the objective is linear over the simplex, the solution is the vertex corresponding to the smallest coordinate:

$$\tilde{\boldsymbol{\pi}}_i = \boldsymbol{e}_{k^\star}, \quad \text{where} \quad k^\star = \arg\min_k \left[\nabla_{\bar{\boldsymbol{\pi}}_i} \phi\right]_k.$$

These updates define the model step in each iteration of the CADO algorithm.

D.3 The Specialized CADO Algorithm

We now summarize the complete CADO algorithm specialized for the node classification setting studied in this paper. This algorithm solves the constrained atomic decomposition problem in (4) using a conditional gradient approach, alternating between updating the embedding vectors $\{\bar{\boldsymbol{e}}_i\}$ and the class-wise models $\{\bar{\boldsymbol{\theta}}_i = (\bar{\boldsymbol{R}}_i, \bar{\boldsymbol{\pi}}_i)\}$.

Embedding step. In each iteration, the embedding update seeks a direction that reduces the global objective ϕ while maintaining feasibility. To make this step efficient, we reparameterize the squared embedding entries as $W_{v,i} = \bar{\epsilon}_{i,v}^2$, which allows us to enforce both the diagonal constraint on $\boldsymbol{L}$ and the convexity condition on $\boldsymbol{\theta}_v$. The resulting LMO admits a closed-form solution: each row of $\mathbf{W}$ is set to a one-hot vector in the direction of steepest descent.

Model step. Given the updated embeddings, the model parameters $\bar{\boldsymbol{\theta}}_i = (\bar{\boldsymbol{R}}_i, \bar{\boldsymbol{\pi}}_i)$ are updated by solving LMOs over the model space $\Theta = \mathbb{S}_{\rho_-,\rho_+} \times \Delta$. The gradients of the global objective ϕ with respect to both components are derived in closed form based on the structure of the loss functions. These LMOs also admit simple solutions: the covariance matrix $\bar{\boldsymbol{R}}_i$ is updated by projecting the negative gradient onto the spectral box, while the label distribution $\bar{\boldsymbol{\pi}}_i$ is updated by selecting the coordinate with the smallest gradient value.

Alternating optimization. The algorithm alternates between these two steps, using a step size $\gamma_t = \frac{2}{t+2}$ at iteration t to compute convex combinations of the previous and newly computed atoms. This ensures feasibility at all iterations and convergence under standard assumptions. The resulting procedure is efficient, scalable, and compatible with a wide range of feature and label models.

Final algorithm. The complete specialized version of the CADO algorithm is presented in Algorithm 2 below.

Algorithm 2 CADO Algorithm (Specialized for our studied case in section 6.1)

Input: Number of atoms r, graph $\bar{\boldsymbol{A}}$, node data $\{\boldsymbol{z}_v\}$, step size sequence $\{\gamma_t = 2/t+2\}$
Initialize $\{\bar{\boldsymbol{e}}_i^{(0)} \in \mathcal{U}\}_{i=1}^r$, $\{\bar{\boldsymbol{\theta}}_i^{(0)} = (\bar{\boldsymbol{R}}_i^{(0)}, \bar{\boldsymbol{\pi}}_i^{(0)}) \in \Theta\}_{i=1}^r$
for $t = 0, 1, 2, \ldots$ until convergence **do**
    // Embedding Update via Conditional Gradient
    Compute $W_{v,i}^{(t)} = \bar{\epsilon}_{i,v}^{(t)2}$, and evaluate $\boldsymbol{R}_v^{(t)}, \boldsymbol{\pi}_v^{(t)}$
    Compute gradient $\nabla_{W_{v,i}} \phi$
    Solve embedding LMO; set $\tilde{W}_{v,i}^{(t)} = 1$ at minimum coordinate, 0 elsewhere
    Set $\tilde{\epsilon}_{i,v}^{(t)} = \sqrt{\tilde{W}_{v,i}^{(t)}}$, and form $\tilde{\boldsymbol{e}}_i^{(t)} = (\tilde{\epsilon}_{i,v}^{(t)})_v$
    Update: $\bar{\boldsymbol{e}}_i^{(t+1)} = (1-\gamma_t)\bar{\boldsymbol{e}}_i^{(t)} + \gamma_t \tilde{\boldsymbol{e}}_i^{(t)}$
    // Model Update via Conditional Gradient
    Compute $\nabla_{\bar{\boldsymbol{R}}_i} \phi$, $\nabla_{\bar{\boldsymbol{\pi}}_i} \phi$ using formulas in Appendix D.2
    Solve model LMO; set

$$\tilde{\boldsymbol{R}}_i^{(t)} = \mathcal{P}_{\mathbb{S}_{\rho_-,\rho_+}}\left(-\nabla_{\bar{\boldsymbol{R}}_i}\phi\right), \qquad \tilde{\boldsymbol{\pi}}_i^{(t)} = \boldsymbol{e}_{k^\star}, \quad k^\star = \arg\min_k \left[\nabla_{\bar{\boldsymbol{\pi}}_i}\phi\right]_k$$

    Update:

$$\bar{\boldsymbol{R}}_i^{(t+1)} = (1-\gamma_t)\bar{\boldsymbol{R}}_i^{(t)} + \gamma_t \tilde{\boldsymbol{R}}_i^{(t)}, \quad \bar{\boldsymbol{\pi}}_i^{(t+1)} = (1-\gamma_t)\bar{\boldsymbol{\pi}}_i^{(t)} + \gamma_t \tilde{\boldsymbol{\pi}}_i^{(t)}$$

Return: $\{\bar{\boldsymbol{e}}_i^{(T)}\}, \{\bar{\boldsymbol{R}}_i^{(T)}, \bar{\boldsymbol{\pi}}_i^{(T)}\}$

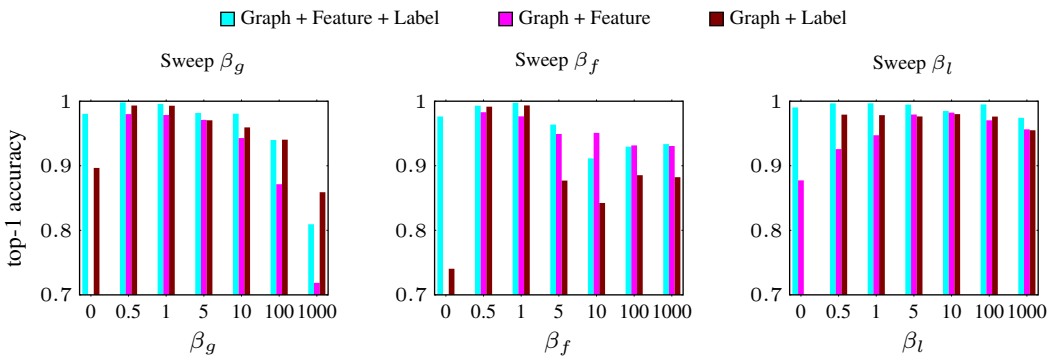

Figure 3: Sensitivity analysis of the proposed method with respect to the weighting parameters β_g, β_f, and β_l, which control the contribution of the graph structure, feature information, and label information, respectively, in the overall optimization objective. We report top-1 classification accuracy for three different configurations: Graph + Feature + Label (cyan), Graph + Feature (magenta), and Graph + Label (brown). Each plot varies one parameter while keeping the others fixed at their default values.

E Extended experiments

E.1 Model Parameter Sensitivity

We study the sensitivity of the model's performance with respect to the weighting parameters β_g, β_f, and β_l, which control the influence of the graph term, feature term, and label term in our unified objective. The results are presented in Figure. 3.

In the left panel, we vary β_g, the weight of the graph term. When $\beta_g = 0$, the model essentially ignores the graph structure, which causes a significant drop in performance in the Graph + Label configuration. However, the full model (Graph + Feature + Label) maintains high accuracy even at $\beta_g = 0$, indicating that feature and label information alone can provide a strong signal. As β_g increases, accuracy improves across all settings, but plateaus after $\beta_g \approx 5$, suggesting diminishing returns from overly amplifying the graph signal.

In the center panel, we vary β_f, the weight of the feature term. When $\beta_f = 0$, the performance of the Graph + Feature configuration drops sharply, as expected. Interestingly, the full model remains relatively robust, highlighting the complementary strength of the graph and label terms. Very large values of β_f lead to performance degradation in some settings, possibly due to overfitting to noisy feature dimensions.

In the right panel, we sweep β_l, the weight of the label term. At $\beta_l = 0$, the Graph + Label configuration performs poorly due to the absence of label supervision. Again, the full model is quite robust and achieves high accuracy even with limited label contribution. Increasing β_l improves performance, but too high values lead to minor declines, likely because the model overemphasizes noisy or limited training labels.

Overall, these results confirm that our model is robust across a wide range of parameter choices and demonstrates strong synergy when all sources of information—graph, features, and labels—are integrated. The default values used in the main experiments strike a good balance across components.

E.2 Data Parameter Sensitivity

Figure 4 investigates how the performance of our method changes as we vary key data generation parameters, specifically the number of nodes n, feature dimension m, and the number of clusters c.

In the left panel, increasing the number of nodes significantly improves test accuracy across all configurations, as larger graphs provide more structure and statistical signal. The full model (Graph + Feature + Label) consistently achieves the highest accuracy and converges quickly, even with a moderate number of nodes. This highlights the data efficiency of our joint framework, particularly when leveraging all available modalities.

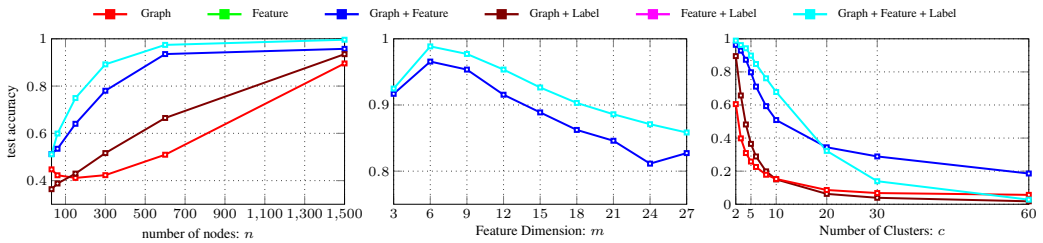

Figure 4: Effect of data parameters on node classification accuracy under different input configurations. Left: Accuracy versus total number of nodes n. Center: Accuracy versus total feature dimension m. Right: Accuracy versus number of clusters c. Each line corresponds to a different combination of information sources used: Graph, Feature, Label, or their combinations.

In the center panel, we vary the total feature dimension m, which includes both signal and noise components. As m increases, performance generally decreases—especially for methods that rely on features (e.g., Graph + Feature)—due to the increasing influence of noisy or uninformative dimensions. However, the full model (cyan) remains relatively stable and outperforms all other combinations, suggesting that combining features with graph structure and labels helps mitigate the curse of dimensionality.

In the right panel, we increase the number of clusters c, making the classification problem more challenging due to finer partitioning and weaker homophily. The performance of all configurations drops, but the full model retains significantly higher accuracy. Notably, using graph-only or label-only configurations fails beyond $c \approx 10$, while feature-based and joint models scale more gracefully. This result supports our theoretical findings: the integration of feature and label information compensates for reduced graph separability as the number of communities increases.

Overall, this experiment confirms the importance of combining multiple modalities. It also demonstrates the robustness of our approach across different data regimes, especially when individual signals become weak or insufficient.