# Inhomogeneous graph trend filtering via a $\ell_{2,0}$-norm cardinality penalty

Xiaoqing Huang*, Andersen Ang*, Kun Huang, Jie Zhang, and Yijie Wang

*Abstract*—We study estimation of piecewise smooth signals over a graph. We propose a $\ell_{2,0}$-norm penalized Graph Trend Filtering (GTF) model to estimate piecewise smooth graph signals that exhibit inhomogeneous levels of smoothness across the nodes. We prove that the proposed GTF model is simultaneously a k-means clustering on the signal over the nodes and a minimum graph cut on the edges of the graph, where the clustering and the cut share the same assignment matrix. We propose two methods to solve the proposed GTF model: a spectral decomposition method and a method based on simulated annealing. In the experiment on synthetic and real-world datasets, we show that the proposed GTF model has a better performances compared with existing approaches on the tasks of denoising, support recovery and semi-supervised classification. We also show that the proposed GTF model can be solved more efficiently than existing models for the dataset with a large edge set.

*Index Terms*—graph signal processing, graph trend filtering, $\ell_{2,0}$-norm, spectral method, simulated annealing

## I. INTRODUCTION

**E**STIMATING signal from noisy observations is a classical problem in signal processing. This work focuses on Graph Signal Estimation (GSE), which is a powerful tool for analyzing signals that are defined on irregular and complex domains (e.g., social networks, brain networks, and sensor networks) and has a lot of real-world applications [1], [2], [3], [4]. Fig. 1a gives an illustration of a graph signal. In a graph signal, the signal is residing on each node in a graph. The signal could be a scalar-valued signal (as shown in Fig. 1a) or vector-valued signal (as shown Fig. 1c). The edges between nodes in the graph introduce additional information and assumption for the signals residing on the nodes. A typical assumption is that the graph signal is globally smooth with respect to the intrinsic structure of the underlying graph, i.e., the signals on the neighbor nodes tend to be similar, or in other words, the local variation of the signals around each node is low. Such global smoothness of a graph signal is further mathematically defined and quantitatively measured by the $p$-Dirichlet form [5], [6], [7], [8], [4], [9] and the graph Laplacian quadratic form ($p$-Dirichlet form with $p = 2$) [9]. GSE based on such global smoothness assumption has been extensively

investigated in graph signal processing [5], [6], [1], [2], [3], [4] and in the context of Laplacian regularization [7], [10].

Although many existing works [5], [6], [1], [2], [3], [4] assume the global smoothness on the graph signal, many real-world graph signals exhibit inhomogeneous levels of smoothness over the graph [11], [12], [13]. In other words, there is a discontinuity in the first-order behaviour of the signal value across the graph, see Fig. 1(a) for a toy illustration. As shown the signals within the communities are the same (homogeneous) but the signals between two communites are different (inhomogeneous). Such an "intra-community homogeneous, inter-community inhomogeneous" property indicates that the graph signals have large variations between different regions and have a small variation within regions (as shown in Fig. 1a).

In real-world applications, we have many graph signals that possess the "inhomogeneous levels of smoothness" property. In bioinformatics and computational biology, tremendous efforts have been made to build molecular networks for understanding biological processes, identifying disease mechanisms, predicting phenotypic outcomes, etc. These molecular networks include but are not limited to protein-protein interaction networks, gene regulatory networks, metabolic networks, and pathway networks. Many databases provide access to these molecular networks [14], [15], [16]. For example, the database STRING [14] collects functional protein association networks from literature curation, computational predictions, and model organisms based on orthology. The database HumanNet [15] provides human gene networks for disease research. The database Pathway Commons [16] aggregates interactions from pathway and interaction databases. For all these molecular networks, when we add a patient-level or cell-level data (e.g. gene expression, mutation, copy numbers, etc) on top of them, we have generated graph signals that exhibit some levels of inhomogeneousity of smoothness across the graph. For such graph signals, from the system biology point of view [17], a valid assumption is that the signal (e.g., gene expression, mutation, etc) within a graph community tends to be homogeneous (i.e., having similar values or having a smooth varaition across the nodes in the same community), but such signal across the different communities can be highly different (i.e., inhomogeneous). For example, in gene co-expression networks [18], gene expressions within a gene community tend to have similar expression values, but the gene expression across the communities is highly different. GSE on the graph signal of these molecular networks can help disease stratification [17] through denoising the patient-level or the cell-level measurements we put on top of the molecular

X. Huang and K. Huang are with the Dept. of Biostatistics and Health Data Science, Indiana University School of Medicine, Indianapolis, IN 46202, USA.
A. Ang is with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK.
J. Zhang is with the Dept. of Medical and Molecular Genetics, Indiana University, Indianapolis, IN 46202, USA.
Y. Wang is with the Dept. of Computer Science, Indiana University, Bloomington, IN 47408, USA. Email: yijwang@iu.edu
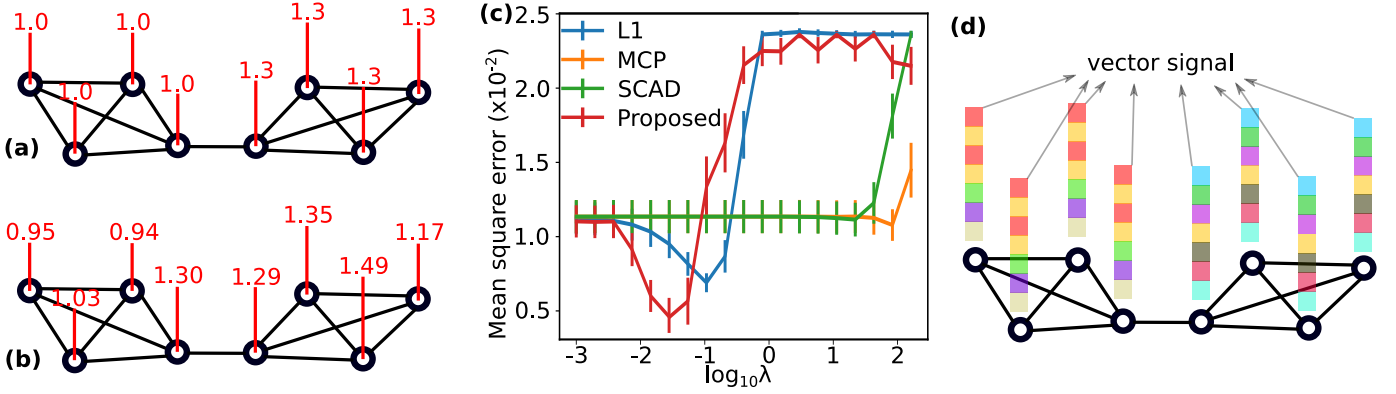*X. Huang and A. Ang are equal contributotrs to this work.

Fig. 1. **Toy examples.** (a) The ground truth of an inhomogeneous scalar-valued graph signal. (b) The same graph signal in (a) with added Gaussian noise $\mathcal{N}(0, 0.1)$. (c) Comparison of the recovery of the ground truth in (a) from 20 different noisy inhomogeneous graph signals as in (b) with different parameter $\lambda$ in (1) and (2). The proposed model (2) achieves the lowest mean square error when screening the parameter $\lambda$. (d) An example of a vector-valued graph signal.

networks.

The existence of real-world piecewise smooth graph signals (especially in computational biology illustrated in the previous paragaph) motivates us to study GSE for such graph signals. Therefore, in this paper, we propose a model: a $\ell_{2,0}$-norm penalized **1st-order Graph Trend Filtering (GTF)** [12], for denoising and extracting patterns of the signal on a graph with localized discontinuities. Before we state explicitly our contribution, below we first review about the GTF model.

### A. GTF for scalar-valued graph signals

Given a graph $G(V, E)$ of $n$ nodes $V = \{1, \ldots, n\}$ and edges $E = \{(i, j) \,|\, i, j \in V$ that are connected$\}$, let $\boldsymbol{y} \in \mathbb{R}^n$ be the observed signal measurement over the nodes, i.e., the element $y_i \in \mathbb{R}$ is the scalar value at node $i$. The goal of the 1st-order GTF for scalar-valued graph signals is to find the estimated graph signal $\boldsymbol{\beta} \in \mathbb{R}^n$, where the $i$th element in $\boldsymbol{\beta}$ denoted by $\beta_i$ (the estimated signal at node $i$), by solving

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^n}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\beta}\|_2^2}_{\text{fitting / denoise}} + \lambda \underbrace{\sum_{(i,j) \in E} \Omega(\beta_i, \beta_j)}_{\text{GTF regularization}}. \tag{1}$$

In (1), $\lambda \geq 0$ is a regularization parameter and $\Omega(\beta_i, \beta_j) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a penalty term that penalizes the difference between the signals on nodes $\{i, j\}$. In other words, $\Omega(\beta_i, \beta_j)$ is a penalization of the neighborhood discrepancy of the graph signal, and it encourages the graph signal value at node $\{i, j\}$ to be similar. Different $\Omega$ have been proposed in the literature for GTF, see Table I.

*1) GTF with the counting norm penalty:* In this paper, we propose the norm $\Omega$ as $\Omega(\beta_i, \beta_j) = \mathbb{I}_{\neq 0}(\beta_i - \beta_j)$, and propose the GTF model as

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\beta}\|_2^2 + \lambda \sum_{(i,j) \in E} \mathbb{I}_{\neq 0}(\beta_i - \beta_j), \tag{2}$$

where $\mathbb{I}_{\neq 0} : \mathbb{R} \to \mathbb{R}$ is an indicator function defined as $\mathbb{I}_{\neq 0}(x) = 0$ if $x = 0$ and $\mathbb{I}_{\neq 0}(x) = 1$ otherwise. In summary, we propose to use a counting norm for $\Omega$, which counts the number of edges that connects nodes with different signal value.

TABLE I
DIFFERENT $\Omega$ PROPOSED IN THE LITERATURE FOR GTF.

| Name | $\Omega(\beta_i, \beta_j)$ | Param. |
|------|---------------------------|--------|
| $\ell_1$-norm [12] | $|\beta_i - \beta_j|$ | - |
| SCAD [11] | $\displaystyle\int_0^{|\beta_i - \beta_j|} \min\left(1, \frac{(\gamma - u/\lambda)_+}{\gamma - 1}\right) du$ | $\gamma \geq 2$ |
| MCP [11] | $\displaystyle\int_0^{|\beta_i - \beta_j|} \left(1 - \frac{u}{\gamma\lambda}\right)_+ du$ | $\gamma$ |

*2) The advantage of using the proposed norm penalty:* The current 1st-order GTF model (1) uses $\ell_1$-norm [12], Smoothly Clipped Absolute Deviation (SCAD) [11], or Minimax Concave Penalty (MCP) [11] to penalize the difference between the signals over connected nodes (as shown in Table I). However, the resulting estimates generated by these penalties are "biased toward zero": the $\ell_1$ penalty enforces the minimization of the term $|\beta_i - \beta_j|$ and, therefore, impose $|\beta_i - \beta_j|$ shrink to zero. As a result, the $\ell_1$ penalty forces $\beta_i$ and $\beta_j$ to be numerically close even if they are not. While the proposed counting norm penalty applies the same penalty to $|\beta_i - \beta_j|$ no matter how different $\beta_i$ and $\beta_j$ are. Therefore, $|\beta_i - \beta_j|$ is not enforced to shrink to zero when $\beta_i$ and $\beta_j$ are different. The same comment also applied to SCAD and MCP penalties. We further illustrate such advantage of the proposed GTF model (2) numerically in Fig. 1c: after screening over the model hyper-parameter ($\lambda$ in (1) and (2), respectively), the GTF model (2) achieves the lowest mean square error to recover the ground-truth signal as shown in Fig. 1(a) from the noisy signal as shown in Fig. 1(b).

### B. GTF for vector-valued graph signals

The previous discussion focused on scalar-valued graph signals. Now we extend the model (2) to vector-valued graph signals (See Fig. 1d for a pictorial illustration): let $\boldsymbol{y}_i \in \mathbb{R}^d$ be the vector-valued signal at node $i$, let $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]^\top \in \mathbb{R}^{n \times d}$

be the data matrix (the rows are the observed vector-valued signals on nodes), we consider the optimization problem

$$\mathcal{P}_0 \;:\; \operatorname*{argmin}_{\boldsymbol{B} \in \mathbb{R}^{n \times d}} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{B}\|_F^2 + \lambda \underbrace{\sum_{(i,j) \in E} \mathbb{I}_{\neq 0}\Big(\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2\Big)}_{=:\mathcal{L}(\boldsymbol{B})}, \quad (3)$$

where $\boldsymbol{B} = [\boldsymbol{b}_1, ..., \boldsymbol{b}_n]^\top \in \mathbb{R}^{n \times d}$ and $\boldsymbol{b}_i \in \mathbb{R}^d$ is the GTF signal estimate at node $i$. Note that (2) is equivalent to (3) when the dimensionality $d = 1$.

*On the term $\mathcal{L}(\boldsymbol{B})$:* We call $\mathcal{L}(\boldsymbol{B})$ in (3) the "$\ell_{2,0}$-norm" in this paper. It can be treated as a "group counting norm", which is the limiting case of the matrix $\ell_{p,q}$-norm with $p = 2$ and $q \to 0$ on a matrix $\boldsymbol{M}$ whose columns are the pairwise difference $\boldsymbol{b}_i - \boldsymbol{b}_j$ of $\boldsymbol{M}$ defined by the edge set $E$, i.e.,

$$\mathcal{L}(\boldsymbol{B}) \;=\; \lim_{q \to 0} \|\boldsymbol{M}\|_{2,q} \;:=\; \lim_{q \to 0} \left( \sum_{j=1}^{|E|} \left( \sqrt{\sum_{i=1}^{d} M_{ij}^2} \right)^q \right)^{\frac{1}{q}}.$$

If $\mathcal{L}(\boldsymbol{B}) = 0$ in $\mathcal{P}_0$, it means that all the node signal estimates $\boldsymbol{b}_i$ share the same value, or equivalently each column of $\boldsymbol{B}$ is a constant vector.

### C. Contributions and paper organization

Grounded on the advantage of using $\mathcal{L}$ (see section I-A2) that it overcomes the problems of using $\ell_1$-norm [12], SCAD [11], and MCP penalties [11] in GTF, in this paper, we propose the 1st-order GTF model $\mathcal{P}_0$.

- We provide theoretical characterizations of the proposed GTF model. In section 2 we prove that $\mathcal{P}_0$ is simultaneously performing a k-means clustering on signals $\boldsymbol{B}$ and a minimum graph cut on $G$ (Theorem 1 and Theorem 2). This means that the model can produce a better result by fusing the information of the node signal with the information of the graph structure to deduce the signal estimate as well as the clustering membership value.
- In section 3 we propose two algorithms to solve $\mathcal{P}_0$. We first show that the proposed GTF problem is combinatorial in nature, see Theorem 1 and Theorem 2. We then propose a method to solve the problem efficiently based on the standard technique of spectral approximation. We finally propose a Hot-bath-based Simulated Annealing algorithm to find the global optimal solution of the proposed GTF problem in case exact solution is required.
- In section 4, we apply Theorem 1 to a semi-supervised learning problem known as the Modified Absorption Problem (MAP) [12], [11], to showcase the analysis of the proposed GTF model is generalizable to other models.
- In section 5 we provide numerical results to support the effectiveness of the algorithms proposed in this work, and to showcase the proposed GTF model outperforms existing GTF models. In particular, we show that the proposed GTF model has a better performances compared with existing approaches on the tasks of denoising, support recovery and semi-supervised classification. We also show that the proposed GTF model can be solved more efficiently than existing models for the dataset with a large edge set.

*Notation:* We use {small italic, capital italic, small bold, capital bold} font to denote {scalar, set, vector, matrix}, respectively. We focus on simple (without multi-edge) connected unweighted undirected graph $G(V, E)$ of $n$ nodes with signal $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]^\top \in \mathbb{R}^{n \times d}$ on $G$. We let $\boldsymbol{A} \in \{0, 1\}^{n \times n}$ be an adjacency matrix to present the connections between nodes in $G$, with $A_{ij} = 1$ indicating nodes $i$ and $j$ are connected and $A_{ij} = 0$ otherwise. We let $\boldsymbol{D}$ be a diagonal matrix with the degree of each node on its diagonal. Let $\boldsymbol{L}$ be the Laplacian of $G$, we have $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. See [19] for more background on graph theory. We let $[n]$ denotes integers from 1 to $n$. Let $\boldsymbol{X} \in \{0, 1\}^{n \times k}$ be the assignment matrix, which can be used to encode the cluster membership of $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$ and the node partition of $G$.

*Remark: digraph vs undigraph* In this work we focus on undirected graph. The proposed approach is applicable to directed graph by transforming a digraph to an undigraph.

## II. EQUIVALENT FORMULATION OF $\mathcal{P}_0$

In this section we provide theoretical characterizations of the proposed GTF model. We show that the proposed GTF model is equivalent to a simultaneous k-means and min-cut (Theorem 1 and Theorem 2), which means that the model is capable of fusing the information of the node signal with the information of the graph structure to deduce the signal estimate as well as the clustering membership.

### A. Theoretical characterizations of the proposed GTF model

Theorem 1 shows that the effect of $\mathcal{L}(\boldsymbol{B}) > 0$ in $\mathcal{P}_0$ is to cut the graph $G$ into $k$ partitions and enforce the node signal in these $k$ partitions to be homogeneous (i.e., they share the same signal value).

**Theorem 1** (The effect of $\mathcal{L}(\boldsymbol{B}) > 0$ in $\mathcal{P}_0$). *Given a data matrix $\boldsymbol{B} \in \mathbb{R}^{n \times d}$ over the nodes $V$ of a simple connected unweighted undigraph graph $G(V, E)$, if $\mathcal{L}(\boldsymbol{B}) > 0$, then it indicates the existence of an edge subset*

$$S_{=1} := \left\{ (i, j) \in E : \mathbb{I}_{\neq 0}\big(\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2\big) = 1 \right\}$$

*that cuts $G$ into $k$ disjoint partitions $C = \{C_1, \ldots, C_k\}$. Furthermore, $\mathcal{L}(\boldsymbol{B}) > 0$ enforces the node signals in those $k$ partitions sharing the same value. Encoding such partition by an assignment matrix $\boldsymbol{X} \in \{0, 1\}^{n \times k}$, then*

$$\mathcal{L}(\boldsymbol{B}) = \frac{1}{2} \, tr(\boldsymbol{X}^\top \boldsymbol{L} \boldsymbol{X}), \quad (4a)$$

$$\boldsymbol{B} = \boldsymbol{X}\boldsymbol{\mu}, \quad (4b)$$

*where $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_k]^\top \in \mathbb{R}^{k \times d}$ with $\boldsymbol{\mu}_i$ is the node signal for all the nodes in the $i$th partition, and $\boldsymbol{L}$ is the graph Laplacian of $G$.*

*Proof.* First we let $E = S_{=1} \cup S_{=0}$ and $S_{=1} \cap S_{=0} = \varnothing$, where

$$S_{=0} := \left\{ (i, j) \in E : \mathbb{I}_{\neq 0}\big(\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2\big) = 0 \right\}.$$

The proof has three parts: (i) show $S_{=1}$ is a graph cut, (ii) prove (4a) and (iii) show (4b).

*$S_{=1}$ is a graph cut:* We show that if we remove all edges in $S_{=1}$ from $G$, then for any edge $(i, j) \in S_{=1}$, the corresponding nodes $\{i, j\}$ are in different disjointed clusters.

- Assume after removing all edges in $S_{=1}$ from $G$ there exists an edge $(h, l) \in S_{=1}$ such that $\mathbb{I}_{\neq 0}(\|\boldsymbol{b}_h - \boldsymbol{b}_l\|_2) = 1$ but the nodes $\{h, l\}$ are in the same partition.
- As $G$ is connected, so nodes $\{h, l\}$ being in the same partition implies there exists a path $p = \{(h, s), \ldots, (t, l)\}$ between $h$ and $l$.
- All the edges along the path $p$ are in $S_{=0}$ because by assumption $S_{=0}$ contains all edges after removing $S_{=1}$ from $G$. This implies $\mathbb{I}_{\neq 0}(\|\boldsymbol{b}_h - \boldsymbol{b}_s\|_2) = \cdots = \mathbb{I}_{\neq 0}(\|\boldsymbol{b}_t - \boldsymbol{b}_l\|_2) = 0$ and thus $\mathbb{I}_{\neq 0}(\|\boldsymbol{b}_h - \boldsymbol{b}_l\|_2) = 0$. A contradiction to $\mathbb{I}_{\neq 0}(\|\boldsymbol{b}_h - \boldsymbol{b}_l\|_2) = 1$.

*Prove (4a):* $S_{=1}$ is a cut and it cuts $G$ into $k \in \mathbb{Z}^+$ ($k \leq n$ is an unknown number) partitions. Let $\boldsymbol{X} \in \{0, 1\}^{n \times k}$ be the assignment matrix (where the $i$th row indicate the cluster membership of the $i$th row of $\boldsymbol{Y}$) that encodes the cut $S_{=1}$, and let $\boldsymbol{L}$ be the Laplacian of $G$, we have the cut size

$$\frac{1}{2} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{L} \boldsymbol{X}) = |S_{=1}| \qquad \text{see [20], [21]}$$

$$= \sum_{(i,j) \in S_{=1}} \mathbb{I}_{\neq 0}(\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2) \qquad \text{by definition of } S_{=1}$$

$$= \sum_{(i,j) \in S_{=1}} \mathbb{I}_{\neq 0}(\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2)$$

$$+ \underbrace{\sum_{(i,j) \in S_{=0}} \mathbb{I}_{\neq 0}(\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2)}_{=0} \qquad \text{by definition of } S_{=0}$$

$$= \sum_{(i,j) \in E} \mathbb{I}_{\neq 0}(\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2).$$

*Prove (4b):* After removing edges in the cut $S_{=1}$ from $G$, the graph $G$ will be separated into $k$ disjoint partitions. Denote such partition as $C = \{C_1, \ldots, C_k\}$.

- For each partition,
  - the nodes within the same partition are connected (because $G$ is connected); and
  - all the edges connecting these nodes are in $S_{=0}$ (by $S_{=0} = E \setminus S_{=1}$).
- By definition of $S_{=0}$, the signal of all the nodes in each partition share the same value, we denote such value by $\boldsymbol{\mu}_i$.
- As $\boldsymbol{\mu}_i$ represents the signal value for all the nodes in partition $C_i$, using the assignment matrix $\boldsymbol{X}$ gives (4b). $\qquad \square$

Now we see that (4b) tells that we assign all the nodes in the same partition with the same signal value stored in $\boldsymbol{\mu}$.

Using Theorem 1, we now show that the solution to $\mathcal{P}_0$ is simultaneously a k-means clustering and minimum graph cut.

**Theorem 2** ($\mathcal{P}_0$ = k-means and min-cut). *$\mathcal{P}_0$ is same as*

$$\mathcal{P}_1 \; : \; \underset{\substack{k \in \mathbb{Z}^+ \\ (\boldsymbol{X}, \boldsymbol{\mu}) \in \Psi}}{argmin} \; \frac{1}{2} \underbrace{\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\mu}\|_F^2}_{\text{k-means}} + \frac{\lambda}{2} \underbrace{tr(\boldsymbol{X}^\top \boldsymbol{L} \boldsymbol{X})}_{\text{graph } k\text{-cut}}, \quad (5)$$

*where $\Psi := \left\{ \boldsymbol{X}, \boldsymbol{\mu} \middle| \boldsymbol{\mu} \in \mathbb{R}^{k \times d}, \, \boldsymbol{X} \in \{0, 1\}^{n \times k}, \, \boldsymbol{X} \boldsymbol{1}_k = \boldsymbol{1}_n \right\}$ and $\boldsymbol{1}_k \in \mathbb{R}^k$ denotes vector of 1s.*

*Proof.* If $\mathcal{L}(\boldsymbol{B}) > 0$, apply (4a) and (4b) in Theorem 2.1 to substitute $\mathcal{L}(\boldsymbol{B})$ and $\boldsymbol{B}$ in $\mathcal{P}_0$. If $\mathcal{L}(\boldsymbol{B}) = 0$, it is trivial $\mathcal{P}_0 = \mathcal{P}_1$ when $k = 1$. $\qquad \square$

### B. Simultaneous k-means and graph cut

The cost function in $\mathcal{P}_1$ has two terms:

1) The first term is in the form of k-means clustering on $\boldsymbol{Y}$ and $\boldsymbol{X}$ is the cluster assignment. This clustering depends solely on the data $\boldsymbol{Y}$ and ignores the graph $G$.
2) The second term is in the form of minimum graph cut on $G$ and $\boldsymbol{X}$ is the graph cut assignment matrix. This cut depends solely on the graph $G$ and ignores the data $\boldsymbol{Y}$.
3) Both k-means and graph cut share the same $\boldsymbol{X}$, indicating $\mathcal{P}_1$ aims to find a $k$-partition that trades off between k-means clustering (using only the signal on $\boldsymbol{Y}$) and minimum graph cut (using only the topology on $G$). This means the information from the signal data and the information of the graph structure are regularizing each other when we solve for $\boldsymbol{B}$, in which this forms the basis why model $\mathcal{P}_0$ is worth studying:
   - if the edge structure is consistent with the node data, we are reinforcing the k-means on the node data using the information of the edge;
   - if the edge structure is inconsistent with the signals on nodes, we are correcting (regularizing) the k-means on the node data using the information of the edge.

The above arguments provide explanations why the proposed model $\mathcal{P}_0$ gave the best recovery in the toy example in Fig. 1. The model $\mathcal{P}_0$ utilizes the two attributes of a graph signal: the graph network structure (collective information) and the data on each node (individual information). The model fuses the two type of information to determine the value of $\boldsymbol{B}$.

Note that k-means and graph cut regularizing each other also improves the stability of the model. For example, in case the graph topology is inaccurate (illustrated by having an inaccurate graph Laplacian $\boldsymbol{L}$), then the model is able to use the signal information to correct the inaccuracy of the topology. Vice versa, if the node signal is highly contaminated (illustrated by the data $\boldsymbol{Y}$ is corrupted by noise), then the model is able to make use of the graph topolgy to correct the inaccuracy of the data.

We now see that model $\mathcal{P}_0$ is basically a mixed-integer program, which is NP-hard to solve:

- We showed that $\mathcal{P}_1$ is equivalent to $\mathcal{P}_0$.
- Theorem 2 shows that $\mathcal{P}_1$ is equivalent to solving k-means. As k-means is an NP-hard problem [22], thus $\mathcal{P}_1$ also contains an NP-hard part in its formulation.

- Furthermore, Theorem 2 also shows that $\mathcal{P}_1$ is equivalent to the minimum $k$-cut, which is polynomial time solvable for fixed $k$ [21]; however, note that $k$ is also a variable in the formulation of $\mathcal{P}_1$, and therefore such graph cut problem is also NP-hard [21].

## III. THE ALGORITHMS TO SOLVE $\mathcal{P}_1$

In the last section we see that problem $\mathcal{P}_0$ is basically a mixed-integer program $\mathcal{P}_1$, where the integer $k \geq 1$ (the number of clusters / partitions) is a unknown in the formulation. This number can also be treated as the factorization rank in the k-means factorization $\boldsymbol{B} = \boldsymbol{X}\boldsymbol{\mu}$. It is generally hard to find the optimal $k$ exactly and thus in principle it is challenging to find optimal exact solution for $\mathcal{P}_1$ (hence $\mathcal{P}_0$). In this section, we develop two algorithms to solve $\mathcal{P}_1$ under two end-user scenarios:

- For an approximate solution: a fast approach using spectral approximation.
- For accurate solution: a slower approach based on hotbath simulated annealing.

### A. Spectral approximation

To solve problem $\mathcal{P}_1$ efficiently, we propose a spectral method based on the work of Mark Newman [23]. Here is the idea of the spectral approximation: first we simplify $\mathcal{P}_1$ to $\mathcal{P}_2$ by Lemma 3 to eliminate $\boldsymbol{\mu}$. Then we find an approximate solution $(k, \boldsymbol{X})$ for $\mathcal{P}_2$ using brute-force on $k$ and spectral method as follows:

- We first consider $\mathcal{P}_2$ with a fixed $k$, denoted as $\mathcal{P}_2^k$.
- We solve $\mathcal{P}_2^k$ to get $(\boldsymbol{X}_k, \boldsymbol{\mu}_k)$ by the approximate spectral method (Theorem 4 and Algorithm 1).
- We screen through different values of $k$ to find the optimal approximate solution $(k_*, \boldsymbol{X}_*)$ for $\mathcal{P}_2$ (Algorithm 2).

Note that brute-force search on $k$ is not scalable, but we argue that in practice $k$ is usually small, and we can always narrow down the search scope of $k$ using prior knowledge from the application.

*On the diagonal of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$:* Before we proceed, we recall a useful fact. For $\boldsymbol{X}$ being a 0-1 assignment matrix, the matrix $\boldsymbol{X}^\top \boldsymbol{X}$ is a diagonal matrix that its $i$th diagonal element is the size of the $i$th cluster, denoted as $|C_i|$. That is

$$\boldsymbol{X}^\top \boldsymbol{X} = \begin{bmatrix} |C_1| & & \\ & \ddots & \\ & & |C_k| \end{bmatrix}. \tag{6}$$

And $\operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) = \sum_{i=1}^k |C_i| = n$.

*1) Eliminate $\boldsymbol{\mu}$:* By observation, the variable $\boldsymbol{\mu}$ in $\mathcal{P}_1$ has no constraint. We simplify $\mathcal{P}_1$ by eliminating $\boldsymbol{\mu}$ using $\boldsymbol{\mu}_* = \boldsymbol{X}^\dagger \boldsymbol{Y}$, where $\boldsymbol{X}^\dagger = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top$ is the left pseudo-inverse of $\boldsymbol{X}$. This gives the following lemma:

**Lemma 3** (Simplifying $\mathcal{P}_1$). *$\mathcal{P}_1$ is equivalent to*

$$\mathcal{P}_2 : \underset{\substack{k \in \mathbb{Z}^+ \\ \boldsymbol{X} \in \Phi}}{argmax} \ q := tr\left(\boldsymbol{Y}\boldsymbol{Y}^\top \boldsymbol{X}\boldsymbol{X}^\dagger\right) + \lambda \, tr\left(\boldsymbol{X}^\top(-\boldsymbol{L})\boldsymbol{X}\right),$$
$$\tag{7}$$

*where* $\Phi = \left\{\boldsymbol{X} | \boldsymbol{X} \in \{0,1\}^{n \times k}, \boldsymbol{X}\boldsymbol{1}_k = \boldsymbol{1}_n\right\}$ *and* $\boldsymbol{X}^\dagger$ *denotes pseudoinverse, i.e.,* $\boldsymbol{X}^\dagger = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top$.

*Proof.* The subproblem on $\boldsymbol{\mu}$ in $\mathcal{P}_1$ has a closed-form solution $\boldsymbol{\mu}^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$. Then,

$$\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\mu}^*\|_F^2 = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{X}^\dagger \boldsymbol{Y}\|_F^2$$

$$= \|\boldsymbol{Y}\|_F^2 - 2\operatorname{tr}\left(\boldsymbol{Y}^\top \boldsymbol{X}\boldsymbol{X}^\dagger \boldsymbol{Y}\right) + \operatorname{tr}\left(\left(\boldsymbol{X}\boldsymbol{X}^\dagger \boldsymbol{Y}\right)^\top \boldsymbol{X}\boldsymbol{X}^\dagger \boldsymbol{Y}\right)$$

$$= \|\boldsymbol{Y}\|_F^2 - 2\operatorname{tr}\left(\boldsymbol{Y}\boldsymbol{Y}^\top \boldsymbol{X}\boldsymbol{X}^\dagger\right) + \operatorname{tr}\left(\boldsymbol{Y}\boldsymbol{Y}^\top (\boldsymbol{X}\boldsymbol{X}^\dagger)^\top \boldsymbol{X}\boldsymbol{X}^\dagger\right)$$

$$= \|\boldsymbol{Y}\|_F^2 - \operatorname{tr}\left(\boldsymbol{Y}\boldsymbol{Y}^\top \boldsymbol{X}\boldsymbol{X}^\dagger\right)$$

where the last equality is by expanding $(\boldsymbol{X}\boldsymbol{X}^\dagger)^\top \boldsymbol{X}\boldsymbol{X}^\dagger$ using $\boldsymbol{X}^\dagger = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top$ and note that $\boldsymbol{X}^\top \boldsymbol{X}$ is symmetric (see (6)). Now $\mathcal{P}_1$ is equivalent to

$$\underset{\substack{k \in \mathbb{Z}^+ \\ \boldsymbol{X} \in \{0,1\}^{n \times k} \\ \boldsymbol{X}\boldsymbol{1}_k = \boldsymbol{1}_n}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{Y}\|_F^2 - \frac{1}{2}\operatorname{tr}\left(\boldsymbol{Y}\boldsymbol{Y}^\top \boldsymbol{X}\boldsymbol{X}^\dagger\right) + \frac{\lambda}{2}\operatorname{tr}\left(\boldsymbol{X}^\top \boldsymbol{L}\boldsymbol{X}\right),$$

Multiply the whole expression by 2, ignore the constant, flip the sign, move the negative sign into $\boldsymbol{L}$ and define $\Phi = \left\{\boldsymbol{X} \mid \boldsymbol{X} \in \{0,1\}^{n \times k}, \boldsymbol{X}\boldsymbol{1}_k = \boldsymbol{1}_n\right\}$ gives $\mathcal{P}_2$. $\square$

*2) $\mathcal{P}_2^k$ is approximately a vector partition problem:* Let $\mathcal{P}_2$ with a fixed $k$ be $\mathcal{P}_2^k$. By inspecting the cost function $q$ of $\mathcal{P}_2^k$, we see that $\boldsymbol{Y}\boldsymbol{Y}^\top$ and $\boldsymbol{L}$ are positive semi-definite, which motivates us to use spectral method to approximate the solution of $\mathcal{P}_2^k$, based on the following arguments:

- in practice, the data matrix $\boldsymbol{Y}$ is often low-rank or approximately low rank [24]; and
- in practice, there are only a few clusters.

Therefore, we can approximate $q$ by only keeping the terms that are associated with the top $k$ largest eigenvalues of $\boldsymbol{Y}\boldsymbol{Y}^\top$ (and the top $k$ smallest eigenvalues of $\boldsymbol{L}$). To proceed, below we define two sets of spectral vectors.

*Two sets of spectral vectors:* Let the eigendecomposition of $\boldsymbol{Y}\boldsymbol{Y}^\top$ and $\boldsymbol{L}$ be $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top$ and $\boldsymbol{V}\boldsymbol{\Gamma}\boldsymbol{V}^\top$, respectively. That is, $(\boldsymbol{u}_i, \sigma_i)$ denotes the $i$th largest (eigenvector, eigenvalue) of $\boldsymbol{Y}\boldsymbol{Y}^\top$, and similarly for $(\boldsymbol{v}_i, \gamma_i)$. Now, given a constant $\alpha \in \mathbb{R}$, define two sets of $k$-dimensional vectors $\{\boldsymbol{r}_i\}_{i=1}^n$, $\{\boldsymbol{t}_i\}_{i=1}^n$ as

$$r_i(j) = \sqrt{\sigma_j} U_{ij}, \quad t_i(j) = \sqrt{\alpha - \gamma_j} V_{ij}, \quad i \in [n] \tag{8}$$

where $r_i(j)$ denotes the $j$th element of $\boldsymbol{r}_i$. Note that we will select $\alpha$ such that $\alpha > \gamma_j$ so that $\boldsymbol{t}_i$ is a real vector. We will discuss how to pick $\alpha$ in section III-A3.

Theorem 4 shows that maximizing $q$ in $\mathcal{P}_2$ is (approximately) equivalent to jointly assigning vectors $\{\boldsymbol{r}_i\}_{i=1}^n$ and $\{\boldsymbol{t}_i\}_{i=1}^n$ into groups so as to maximizing the magnitude of their weighted sum. Such an assignment problem is known as a vector partition problem [23], [25] and it can be effectively solved by any k-means package. We remark that the derivation technique of Theorem 4 is based on the work of Mark Newman [23].

**Theorem 4** ($\mathcal{P}_2^k \approx$ Vector partition). *Given an integer $1 \leq k \leq n$, a constant $\alpha > \gamma_k$, let $|C_i|$ be the size of the partition $C_i$, then $\mathcal{P}_2^k$ is approximately the following vector partition problem*

$$\underset{\boldsymbol{r}_i, \boldsymbol{t}_i}{argmax} \quad \sum_{i=1}^{k} \|\boldsymbol{\xi}_i\|_2^2 + \|\boldsymbol{\zeta}_i\|_2^2 - \lambda \alpha n \tag{VPP}$$
$$s.t. \quad \boldsymbol{\xi}_i = \sum_{i \in C_i} \frac{1}{\sqrt{|C_i|}} \boldsymbol{r}_i, \ \boldsymbol{\zeta}_i = \sum_{i \in C_i} \sqrt{\lambda} \boldsymbol{t}_i.$$

*Proof.* The proof has three parts: (1) rewriting $q$ in $\mathcal{P}_2$ using eigendecomposition, (2) define two sets of spectral vectors and (3) approximate the problem as a vector partition problem.
*Rewriting $q$ using eigendecomposition:* Let the eigendecomposition on $\boldsymbol{Y}\boldsymbol{Y}^\top$ and $\boldsymbol{L}$ be $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top$ and $\boldsymbol{V}\boldsymbol{\Gamma}\boldsymbol{V}^\top$, respectively. That is, $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots]$ stores the eigenvectors of $\boldsymbol{Y}\boldsymbol{Y}^\top$ and $\boldsymbol{\Sigma}$ is a diagonal matrix storing the eigenvalues $\sigma_i$ of $\boldsymbol{Y}\boldsymbol{Y}^\top$, and likewise for $\boldsymbol{V}$, $\boldsymbol{\Gamma}$ for $\boldsymbol{L}$. Now in $\mathcal{P}_2$ we have $q = \mathrm{tr}\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{X}\boldsymbol{X}^\dagger\right) + \lambda \mathrm{tr}\left(\boldsymbol{X}^\top \boldsymbol{V}(-\boldsymbol{\Gamma})\boldsymbol{V}^\top \boldsymbol{X}\right)$. Consider $-\boldsymbol{\Gamma} = \alpha \boldsymbol{I} - \boldsymbol{\Gamma} + \alpha \boldsymbol{I}$ under a predefined $\alpha$, we have

$$q = \begin{aligned} &\mathrm{tr}\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top\right) \\ &+\lambda \mathrm{tr}\left(\boldsymbol{X}^\top \boldsymbol{V}(\alpha \boldsymbol{I} - \boldsymbol{\Gamma})\boldsymbol{V}^\top \boldsymbol{X}\right) - \lambda \alpha n \end{aligned}$$

$$= \begin{aligned} &\mathrm{tr}\left((\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{X}\right) \\ &+\lambda \mathrm{tr}\left((\alpha \boldsymbol{I} - \boldsymbol{\Gamma})\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V}\right) - \lambda \alpha n. \end{aligned}$$

Now, invoke (6),

$$q = \mathrm{tr}\left(\begin{bmatrix} \frac{1}{|C_1|} & & \\ & \ddots & \\ & & \frac{1}{|C_k|} \end{bmatrix} \boldsymbol{X}^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{X} \right.$$
$$\left. + \lambda \begin{bmatrix} \alpha - \gamma_1 & & \\ & \ddots & \\ & & \alpha - \gamma_n \end{bmatrix} \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V}\right) - \lambda \alpha n.$$

Since trace operator focuses on the diagonal, we only need to look at the diagonal of both $\boldsymbol{X}^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{X}$ and $\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V}$. Let $[\boldsymbol{M}]_{ii}$ denotes taking only the diagonal of $\boldsymbol{M}$ and setting the rest to zero, the trace term of $q$ is equivalent to

$$\mathrm{tr}\left(\begin{bmatrix} \frac{1}{|C_1|} & & \\ & \ddots & \\ & & \frac{1}{|C_k|} \end{bmatrix} [\boldsymbol{X}^\top \boldsymbol{U}]_{ii}[\boldsymbol{\Sigma}]_{ii}[\boldsymbol{U}^\top \boldsymbol{X}]_{ii} \right.$$
$$\left. + \lambda \begin{bmatrix} \alpha - \gamma_1 & & \\ & \ddots & \\ & & \alpha - \gamma_n \end{bmatrix} [\boldsymbol{V}^\top \boldsymbol{X}]_{ii}[\boldsymbol{X}^\top \boldsymbol{V}]_{ii}\right),$$

where we note that this expression is the motivation behind the two sets of spectral vector defined in (8).
Now let $h = 1, \dots, k$ be the index for cluster $C_h$, let $j = 1, \dots, n$ be the index of $\sigma_j$ and $\gamma_j$. For the matrix product

$\boldsymbol{C} = \boldsymbol{A}^\top \boldsymbol{B}$, recall that $C_{jh} = \sum_i A_{ij} B_{ih}$ and note that $\boldsymbol{A}^\top \boldsymbol{B}$ and $\boldsymbol{B}^\top \boldsymbol{A}$ share the same diagonal, now $q$ becomes

$$q = \sum_{h=1}^{k} \frac{1}{|C_h|} \sum_{j=1}^{n} \sigma_j \left(\sum_{i=1}^{n} U_{ij} X_{ih}\right)^2$$
$$+ \lambda \sum_{h=1}^{k} \sum_{j=1}^{n} (\alpha - \gamma_j) \left(\sum_{i=1}^{n} V_{ij} X_{ih}\right)^2 - \lambda \alpha n,$$

which gives

$$q = \sum_{h=1}^{k} \frac{1}{|C_h|} \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \sqrt{\sigma_j} U_{ij} X_{ih}\right)^2$$
$$+ \lambda \sum_{h=1}^{k} \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \sqrt{\alpha - \gamma_j} V_{ij} X_{ih}\right)^2 - \lambda \alpha n. \tag{9}$$

*Define two sets of spectral vectors:* Under a fixed $k$, we now approximately maximize $q$ by focusing on the terms in $q$ that are related to first $k$ largest eigenvalues of $\boldsymbol{Y}\boldsymbol{Y}^\top$ and $-\boldsymbol{L}$. First we follows (8) and define two sets of $k$-dimensional vectors $\{\boldsymbol{r}_i\}_{i=1}^n$ and $\{\boldsymbol{t}_i\}_{i=1}^n$. Note that we pick $\alpha > \gamma_k$ (the $k$th largest eigenvalues of $\boldsymbol{L}$) so $\boldsymbol{t}_i$ are real for the first $k$ terms.
*Approximate the problem as a vector partition problem:* Now we apply $\{\boldsymbol{r}_i\}_{i=1}^n$ and $\{\boldsymbol{t}_i\}_{i=1}^n$ on (9): in the first term, we drop the terms related to the smallest $n - k$ eigenvalues of $\boldsymbol{Y}\boldsymbol{Y}^\top$; in the second term, we drop the terms related to the smallest $n - k$ of the factors $\alpha - \gamma_i$, then

$$q \approx \sum_{h=1}^{k} \frac{1}{|C_h|} \sum_{j=1}^{k} \left(\sum_{i=1}^{n} \sqrt{\sigma_j} U_{ij} X_{ih}\right)^2$$
$$+ \lambda \sum_{h=1}^{k} \sum_{j=1}^{k} \left(\sum_{i=1}^{n} \sqrt{\alpha - \gamma_j} V_{ij} X_{ih}\right)^2 - \lambda \alpha n$$

$$= \sum_{h=1}^{k} \frac{1}{|C_h|} \sum_{j=1}^{k} \left(\sum_{i \in C_h} r_{ij}\right)^2$$
$$+ \lambda \sum_{h=1}^{k} \sum_{j=1}^{k} \left(\sum_{i \in C_h} t_{ij}\right)^2 - \lambda \alpha n$$

$$= \sum_{h=1}^{k} \sum_{j=1}^{k} \underbrace{\left(\frac{1}{\sqrt{|C_h|}} \sum_{i \in C_h} r_{ij}\right)^2}_{\|\boldsymbol{\xi}_h\|_2^2}$$
$$+ \sum_{h=1}^{k} \sum_{j=1}^{k} \underbrace{\left(\sqrt{\lambda} \sum_{i \in C_h} t_{ij}\right)^2}_{\|\boldsymbol{\zeta}_h\|_2^2} - \lambda \alpha n$$

$$= \sum_{h=1}^{k} \|\boldsymbol{\xi}_h\|^2 + \|\boldsymbol{\zeta}_h\|^2 - \lambda \alpha n. \tag{10}$$

Changing the index notation $h$ to $i$ in (10) completes the proof. □

We follow [23] to solve VPP by applying k-means on the vectors

$$\boldsymbol{z}_i := \left[\frac{1}{\sqrt{|C_i|}} \boldsymbol{r}_i^\top, \sqrt{\lambda} \boldsymbol{t}_i^\top\right]^\top$$

for $i \in [n]$. This gives an approximate solution of $\mathcal{P}_2^k$. See Algorithm 1.

*When does the spectral approximation work?* The above derivation shows that the estimation of $k$ plays an important role in the whole process. We note that such approach is effective when the true $k$ is small, which is grounded on the fact that most of the information is concentrated in a few eigenvalues of the data covariance matrix $\boldsymbol{Y}\boldsymbol{Y}^\top$ and a few eigenvalues of the Laplacian $\boldsymbol{L}$. A small true $k$ also makes the algorithm effective: the spectral approximation relies on the approximation of a sum with $n$ terms in (9) by the sum with $k$ terms in (10), where $n$ is the number of nodes that is potentially large. In the case $k \ll n$, the spectral approximation dramatically reduces the computational complexity of the whole algorithm and making the brute force search on $k$ possible. See Fig. 2 in the experiment for an example where $k = 4 \ll 2642 = n$.

If it is the casr that the true $k$ is large, spectral approximation no longer works. However, we argue that $k \ll n$ is true most of the time in practice, see discussion in Section III-A2.

---

**Algorithm 1** Spectral method for solving $\mathcal{P}_2^k$

**Input**: $\boldsymbol{Y} \in \mathbb{R}^{n \times d}$, $\boldsymbol{L} \in \mathbb{R}^{n \times n}$, $\lambda \in \mathbb{R}$, and $k$.

**Compute**:

Top $k$ largest eigenvalues $\{\sigma_i\}_{i=1}^k$ and the corresponding eigenvectors $\{\boldsymbol{u}_i\}_{i=1}^k$ of $\boldsymbol{Y}\boldsymbol{Y}^\top$;

Top $k$ smallest eigenvalues $\{\gamma_i\}_{i=1}^k$ and the corresponding eigenvectors $\{\boldsymbol{v}_i\}_{i=1}^k$ of $\boldsymbol{L}$;

$$\alpha = \frac{1}{n-k} \sum_{i=k+1}^n \gamma_i$$

$$\boldsymbol{z}_i := \left[ \frac{1}{\sqrt{|C_i|}} \boldsymbol{r}_i^\top, \sqrt{\lambda} \boldsymbol{t}_i^\top \right]^\top \text{ by (8), } i \in [n];$$

Run k-means on $\{\boldsymbol{z}_i\}_{i=1}^n$ and obtain $\boldsymbol{X}$.

**Return** $\boldsymbol{X}_k = \boldsymbol{X}$.

---

*3) Approximation error and the choice of $\alpha$:* In the approximation we drop the $n-k$ most negative eigenvalues of $-\boldsymbol{L}$ to approximate the matrix $\alpha\boldsymbol{I} - \boldsymbol{L}$ (i.e., the $n-k$ leading eigenvalues of $\boldsymbol{L}$), and, instead of computing $\boldsymbol{V}(\alpha\boldsymbol{I} - \boldsymbol{\Gamma})\boldsymbol{V}^\top$, we compute $\boldsymbol{V}(\alpha\boldsymbol{I}' - \boldsymbol{\Gamma}')\boldsymbol{V}^\top$, where $\boldsymbol{I}'$ and $\boldsymbol{\Gamma}'$ are matrix $\boldsymbol{I}$ and $\boldsymbol{\Gamma}$ with the last $n-k$ diagonal elements set to zero. The approximation error in such a procedure is

$$
\begin{aligned}
\varepsilon &:= \text{tr}\left( \boldsymbol{V}(\alpha\boldsymbol{I} - \boldsymbol{\Gamma})\boldsymbol{V}^\top - \boldsymbol{V}(\alpha\boldsymbol{I}' - \boldsymbol{\Gamma}')\boldsymbol{V}^\top \right) \\
&= \text{tr}\left( (\alpha\boldsymbol{I} - \boldsymbol{\Gamma}) - (\alpha\boldsymbol{I}' - \boldsymbol{\Gamma}') \right) = \sum_{i=k+1}^n (\gamma_i - \alpha)^2.
\end{aligned}
$$

Let $\frac{\partial \varepsilon}{\partial \alpha} = 0$ gives the optimal $\alpha^* = \frac{1}{n-k} \sum_{i=k+1}^n \gamma_i$.

*4) The whole approximate spectral method:* We now propose Algorithm 2 to solve $\mathcal{P}_2$ approximately.

---

**Algorithm 2** Algorithm for solving $\mathcal{P}_2$ approximately

**Input**: $\boldsymbol{Y} \in \mathbb{R}^{n \times d}$, $\boldsymbol{L} \in \mathbb{R}^{n \times n}$, $\lambda \in \mathbb{R}$, and $k_{\max} \in \mathbb{Z}^+$.

**Initialize**: $q^* = +\infty$.

**for all** $k = 1, 2, ..., k_{\max}$ **do**

Using Algorithm 1 to get $\boldsymbol{X}_k$;

Compute $q_k = \text{tr}(\boldsymbol{Y}\boldsymbol{Y}^\top \boldsymbol{X}_k \boldsymbol{X}_k^\dagger) + \lambda \text{tr}\left( \boldsymbol{X}_k^\top (-\boldsymbol{L}) \boldsymbol{X}_k \right)$;

Update $q^* = q_k$ and $\boldsymbol{X}_* = \boldsymbol{X}_k$ if $q_k \leq q^*$.

**end for**

**Return** $\boldsymbol{X}_*$ and $\boldsymbol{\mu} = (\boldsymbol{X}_*^\top \boldsymbol{X}_*)^{-1} \boldsymbol{X}_*^\top \boldsymbol{Y}$.

---

### B. Heat-bath simulated annealing

Theorem 2 indicates that $\mathcal{P}_1$ is a NP-complete problem. In the previoius section, we propose Algorithm 2 based on approximating the eigen-spectrums of $\boldsymbol{Y}\boldsymbol{Y}^\top$ and $\boldsymbol{L}$. In this section, we propose another approach that uses Heat-bath simulated annealing to find the global optimal solution of $\mathcal{P}_1$. Simulated annealing is a probabilistic optimization technique that is used to find global solutions. In simulated annealing, we need a Hamiltonian energy function that represents the objective function we aim to minimize. This energy function maps each solution to a real number representing its "energy". The goal of the algorithm is to find the state with the lowest possible energy. We rewrite $\mathcal{P}_1$ into $\mathcal{P}_3$ as follows and define $\mathcal{H}$ as the Hamiltonian energy function.

$$
\begin{aligned}
\mathcal{P}_3 : \quad &\underset{k, \boldsymbol{\delta}}{\text{argmin}} \quad \mathcal{H}(\boldsymbol{\delta}) := \underbrace{\sum_{i=1}^n \sum_{j=1}^k \mathbb{I}_{=j}(\delta_i) \|\boldsymbol{y}_i - \boldsymbol{c}_j\|_2^2}_{\text{k-means}} \\
&\qquad + \lambda \underbrace{\sum_{i,j} \sum_{s \neq t} A_{ij} \mathbb{I}_{=s}(\delta_i) \mathbb{I}_{=t}(\delta_j)}_{\text{graph k-cut}} \\
&s.t. \quad \boldsymbol{c}_j = \frac{\sum_{l=1}^n \mathbb{I}_{=j}(\delta_l) \boldsymbol{y}_l}{\sum_{l=1}^n \mathbb{I}_{=j}(\delta_l)}, \quad j \in \{1, 2, ..., k\}; \\
&\qquad \delta_i \in \{1, 2, ..., k\}, \quad i = 1, 2, ..., n,
\end{aligned}
\tag{11}
$$

where the vector $\boldsymbol{\delta} = [\delta_1, \delta_2, ..., \delta_n]^\top \in \{1, ..., k\}^n$ is a membership state, with the $i$th element $\delta_i \in [k] := \{1, 2, ..., k\}$ representing the membership of signal $\boldsymbol{y}_i$ and the membership of node $i$, e.g., $\delta_i = h$ means the $\boldsymbol{y}_i$ and node $i$ belong to cluster $h$. The symbols $s, t \in [k]$ are two dummy cluster labels and the indicator function $\mathbb{I}_{=j}(\delta_i) = 1$ if $\delta_i = j$, and $= 0$ otherwise. The matrix $\boldsymbol{A}$ with elements $A_{ij} \in \{0, 1\}$ is the adjacency matrix of the graph $G$, and lastly, $\boldsymbol{c}_j$ is an intermediate variable in $\mathcal{P}_3$.

It is trivial to see the equivalence between $\mathcal{P}_1$ and $\mathcal{P}_3$: in $\mathcal{P}_3$ we uses $\boldsymbol{\delta}$ to replace the assignment matrix $\boldsymbol{X}$ in $\mathcal{P}_1$. The membership states $\{\delta_i\}$ act as a 0-1 on/off switches in both the k-means term and the graph cut term.

*1) Contrasting $\mathcal{P}_1$ and $\mathcal{P}_3$:* Recall that in Algorithm 2 for (approximately) solving $\mathcal{P}_1$, we screen through different values of $k$ in a pre-defined search set $[k_{\max}]$ to find the optimal $k_*$

and we assume $k_* \in [k_{\max}]$. For $\mathcal{P}_3$, we propose to use heat-bath Simulated Annealing (SA) to find $k_*$. We start with an overestimated value $k$ (which is expected to be larger than $k_*$), then SA will find $k_*$ (probabilistically) by a "cooling process."

*2) Heat-bath simulated annealing:* To minimize $\mathcal{H}(\boldsymbol{\delta})$, we use a heat-bath Monte Carlo optimization strategy (abbreviated as heat-bath), which is a strategy similar to the classical SA (abbreviated as cSA) but is shown to have a better performance [26], [27]. While cSA and heat-bath are conceptually similar, cSA is based on the Metropolis approach [28] to compute the transition probabilities without computing partition functions because the partition functions are hard to compute for many applications. However, in the setting of GTF presented in this paper, efficient computation of the partition function is not a challenge, which allows the use of the alternative heat-bath transition probabilities.

*Coordinate-wise update:* In the proposed SA approach to solve $\mathcal{P}_3$, we perform a coordinate update of $\boldsymbol{\delta}$: we update one element $\delta_i$ at a time while keeping the rest of $\boldsymbol{\delta}$ fixed at their most recent value.

*Details of the heat-bath update:* As $\delta_i \in [k]$ represents a cluster label, the update on $\delta_i$ is a switch process. Let $s, t \in [k]$ be cluster labels, where $s$ is the state of $\delta_i$ before the update and $t$ is the state of $\delta_i$ after the update. Such switching update of $\delta_i$ is carried out using single-spin heat-bath update rule [29]: we update the system energy $\mathcal{H}$ when making a membership assignment $\delta_i$ switch from $s$ to $t$, i.e.,

$$\underbrace{\mathcal{H}(\delta_i = t)}_{\text{energy after update}} = \underbrace{\mathcal{H}(\delta_i = s)}_{\text{energy before update}} + \underbrace{\Delta\mathcal{H}(\delta_i = s \to t)}_{\text{change in energy}}.$$

The probability of the membership switch at a temperature $T$ is proportional to the exponential of the corresponding energy change of the entire system, i.e

$$\mathbb{P}(\delta_i = t; T) = \frac{\exp\left\{-\frac{1}{T}\Delta\mathcal{H}(\delta_i = s \to t)\right\}}{\sum\limits_{l=1}^{k} \exp\left\{-\frac{1}{T}\Delta\mathcal{H}(\delta_i = s \to l)\right\}}. \quad (12)$$

*A brief summary of how SA finds global minimum:* From statistical mechanics, at a high enough temperature, the membership state $\delta_i$ follows a discrete uniform distribution, i.e.,

$$\lim_{T \to +\infty} \underbrace{\frac{\exp\left\{-\frac{1}{T}\Delta\mathcal{H}(\delta_i = s \to t)\right\}}{\sum\limits_{l=1}^{k} \exp\left\{-\frac{1}{T}\Delta\mathcal{H}(\delta_i = s \to l)\right\}}}_{\mathbb{P}(\delta_i = t; T)}$$

$$= \frac{\lim\limits_{T \to +\infty} \exp\left\{-\frac{1}{T}\Delta\mathcal{H}(\delta_i = s \to t)\right\}}{\sum\limits_{l=1}^{k} \lim\limits_{T \to +\infty} \exp\left\{-\frac{1}{T}\Delta\mathcal{H}(\delta_i = s \to l)\right\}} = \frac{1}{k},$$

hence at such moment, it is capable of jumping out of a local minimum, or in other words, SA at high temperatures is able to explore all the $k$ possible membership assignments. When the temperature cools down (say $T \to 1$), the state $\delta_i$ obeys

a categorical distribution where $\delta_i$ are more likely to switch to $t$ from the current membership state $s$ if the energy change $\Delta\mathcal{H}(\delta_i = s \to t)$ is negative and small.

*The whole SA algorithm:* Based on the above discussion, we propose the SA Algorithm 3, where scalars $T_{start}$ and $T_{end}$ are the start and end temperatures, $c \in [0, 1)$ is a constant to control the cooling rate, and $K$ is the number of sweep times for each temperature.

---

**Algorithm 3** Hot-bath SA for solving $\mathcal{P}_3$.

---

**Input**: $T_{start} = 100$, $T_{end} = 0.001$, $c = 0.99$, $K$, $\lambda$.

$T = T_{start}$.

**while** $T \geq T_{end}$ **do**

    **for all** $k = 1, 2, ..., K$ **do**

        $R = \text{RandomShuffle}(\{1, 2, \ldots, n\})$;

        **for all** $i = 1, 2, ..., n$ **do**

            Update $\mathcal{H}(\delta_{R[i]})$ at $T$ based on the probability (12) using roulette wheel algorithm [30];

            Update $\mathcal{H}$ after $\delta_{R[i]}$ is assigned;

        **end for**

    **end for**

    $T = T \times c$    % cool down the temperature.

**end while**

**Return** The most recent $\boldsymbol{\delta}$.

---

The major computational cost here comes from computing the term $\Delta\mathcal{H}(\delta_i = s \to t)$, which can be efficiently computed in $\mathcal{O}(n)$ [31], [32]. However, we remark that it can take many iterations for the Hot-bath SA algorithm to converge (cool down).

*C. Section summary*

We recall that the main contribution of this work is the new GTF model $\mathcal{P}_0$. In this section we proposed two algorithmic approaches for solving $\mathcal{P}_0$. In general, compared with the spectral method, the Hot-bath SA algorithm provides a more accurate solution but it also requires more computational time to find the solution. We refer to the discussion in section V for more details.

## IV. EXTENSION TO GRAPH-BASED TRANSDUCTIVE LEARNING

In this section, we show that the Theorem 1 is generalizable to other models. Let $K > 0$ be a given integer. Consider a $K$-class classification problem in a semi-supervised learning setting: given a dataset with $n$ samples, we observe a subset of class labels encoded in a class assignment matrix $\boldsymbol{Y} \in \{0, 1\}^{n \times K}$, and a diagonal indicator matrix $\boldsymbol{M} \in \{0, 1\}^{n \times n}$ that denotes the samples whose labels have been observed. Such classification problem, called the Modified Absorption Problem [33], [34] can be expressed as a 1st-order GTF with $\ell_{2,0}$-norm penalty as

$$\mathcal{Q}: \underset{\boldsymbol{B} \in \mathbb{R}^{n \times K}}{\text{argmin}} \frac{\|\boldsymbol{M}(\boldsymbol{Y} - \boldsymbol{B})\|_F^2}{2} + \lambda\mathcal{L}(\boldsymbol{B}) + \epsilon\|\boldsymbol{R} - \boldsymbol{B}\|_F^2, \quad (13)$$
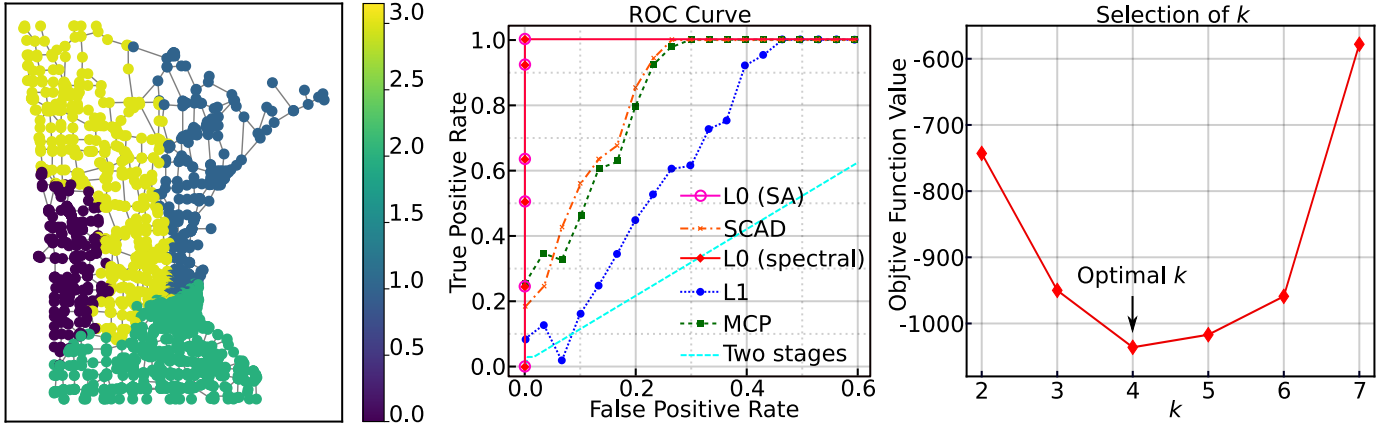
Fig. 2. On identifying boundary edges. Left: the ground truth $\boldsymbol{y}_*$ on the Minnesota road graph. Middle: the ROC curve of the methods. The curves for the spectral method and SA method are overlapped. Here the $\ell_0$ model achieved perfect ROC. Right: the optimal $k_*$ (= 4 clusters) for the $\ell_0$ model $\mathcal{P}_2$.

where $\boldsymbol{R} \in \mathbb{R}^{n \times K}$ encodes a fixed prior belief on the samples, and $\lambda \geq 0, \epsilon \geq 0$ are regularization parameters. The matrix $\boldsymbol{B} \in \mathbb{R}^{n \times K}$ contains the fitted probabilities, with $B_{ij}$ representing the probability that sample $i$ is of class $j$. Using $\boldsymbol{b}_i$ to denote the $i$th row of $\boldsymbol{B}$, the middle term of (13) encourages the probabilities of samples to behave smoothly over a graph $G(V, E)$ constructed based on the sample features.

Now we give a theorem related to $\mathcal{Q}$. The focus here is to demonstrate the generality of Theorem 1, which can be applied to models with the 1st-order GTF with $\ell_{2,0}$-norm penalty.

**Theorem 5.** $\mathcal{Q}$ *is equivalent to*

$$
\mathcal{Q}_1 : \quad \underset{\substack{k \in \mathbb{Z}^+ \\ \boldsymbol{X} \in \{0,1\}^{n \times k} \\ \boldsymbol{X}^\top \mathbf{1}_k = \mathbf{1}_n}}{argmin} \quad \frac{1}{2} \left\| \boldsymbol{M} \left( \boldsymbol{Y} - \boldsymbol{X} \boldsymbol{B}(\boldsymbol{X}) \right) \right\|_F^2 \\
+ \lambda \, tr(\boldsymbol{X}^\top \boldsymbol{L} \boldsymbol{X}) + \epsilon \left\| \boldsymbol{R} - \boldsymbol{X} \boldsymbol{B}(\boldsymbol{X}) \right\|_F^2,
$$

(14)

*with* $\boldsymbol{B}(\boldsymbol{X}) = (\boldsymbol{X}^\top (\boldsymbol{M} + \boldsymbol{I}) \boldsymbol{X})^{-1} (\boldsymbol{X}^\top \boldsymbol{M} \boldsymbol{Y} + \epsilon \boldsymbol{X}^\top \boldsymbol{R})$.

We use the techniques for solving $\mathcal{P}_2$ to solve $\mathcal{Q}_1$ (hence solving $\mathcal{Q}$). Note that $\mathcal{Q}$ and $\mathcal{P}$ have a different optimization subproblem, so we cannot use Algorithm 1 directly to solve the inner minimization of $\mathcal{Q}$. We use the Frank-Wolfe algorithm [35], [36] to solve such inner minimization of $\mathcal{Q}$.

## V. EXPERIMENTAL RESULTS

We now demonstrate the proposed $\ell_{2,0}$-norm penalized 1st-order GTF model (i.e., problem (3)) outperforms three state-of-the-art GTF models (see table II) over two graph learning tasks: support recovery and signal estimation. Then we show the proposed MAP-GTF model outperforms other models in the task of semi-supervised classification. All experiments were run on a computer with 8 cores 3.7GHz Intel CPU, and 32 GB RAM[1].

### A. Support recovery

*1) Dataset:* We test all the methods for identifying the boundaries of a piece-wise constant signal on the Minnesota road graph [37] shown in Fig. 2. The graph $G$ has $n = 2,642$

TABLE II
GTF MODELS WITH DIFFERENT PENALTY

| Penalty | Param. |
|---|---|
| $\ell_{2,0}$-norm (this work) | $k, \lambda$ |
| $\ell_1$-norm [12] | $\lambda$ |
| SCAD [11] | $\gamma, \tau, \lambda$ |
| MCP [11] | $\gamma, \tau, \lambda$ |

nodes and $3,304$ edges. Let $\boldsymbol{y}_* \in \mathbb{R}^n$ be the ground truth scalar-valued signal for each node in $G$, we add Gaussian noise $\boldsymbol{\theta} \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma^2 = 0.05$ to $\boldsymbol{y}_*$ to obtain the simulated observed signal $\boldsymbol{y} = \boldsymbol{y}_* + \boldsymbol{\theta}$. The Signal-to-Noise Ratio (SNR) of $\boldsymbol{y}_*$ is $10 \log_{10} \frac{\|\boldsymbol{y}_*\|}{\sigma^2 nd} \approx 12$.

*2) Support recovery: perfect ROC curve for the proposed method:* The task here is to identify the boundary edges by recovering the true supports of $\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_0, \ \forall(i,j) \in E$. We evaluate the performance of the aforementioned methods by comparing their result with the ground truth boundary edges computed from $\boldsymbol{y}_*$ as shown in the left panel of Fig. 2.

*3) Parameter tuning:* In the test, we report the best results for each model by tuning their hyper-parameters.

- We run Algorithm 2 to use the spectral method to solve the proposed $\ell_{2,0}$ model (3) for the support recovery problem. The algorithm has two hyper-parameters: the number of partitions $k$ and the regularization parameter $\lambda$.
- We run Algorithm 3 to use the SA method to solve the proposed $\ell_{2,0}$ model (11). We set the number of partitions $k = 7$. The algorithm has only one hyper-parameter: the regularization parameter $\lambda$.
- For the $\ell_1$, SCAD, and MCP models, we use the source code provided by [11] to identify supports for this experiment. For SCAD, we set $\gamma = 3.7$ as suggested in [11]. For MCP, we set $\gamma = 1.4$ as suggested in [11].

*4) The results:* We plot the results in Fig. 2. We compare the methods in terms of ROC curve, i.e. the true positive rate versus the false positive rate of identification of a boundary edge correctly. The $\ell_{2,0}$ model solved by both the proposed
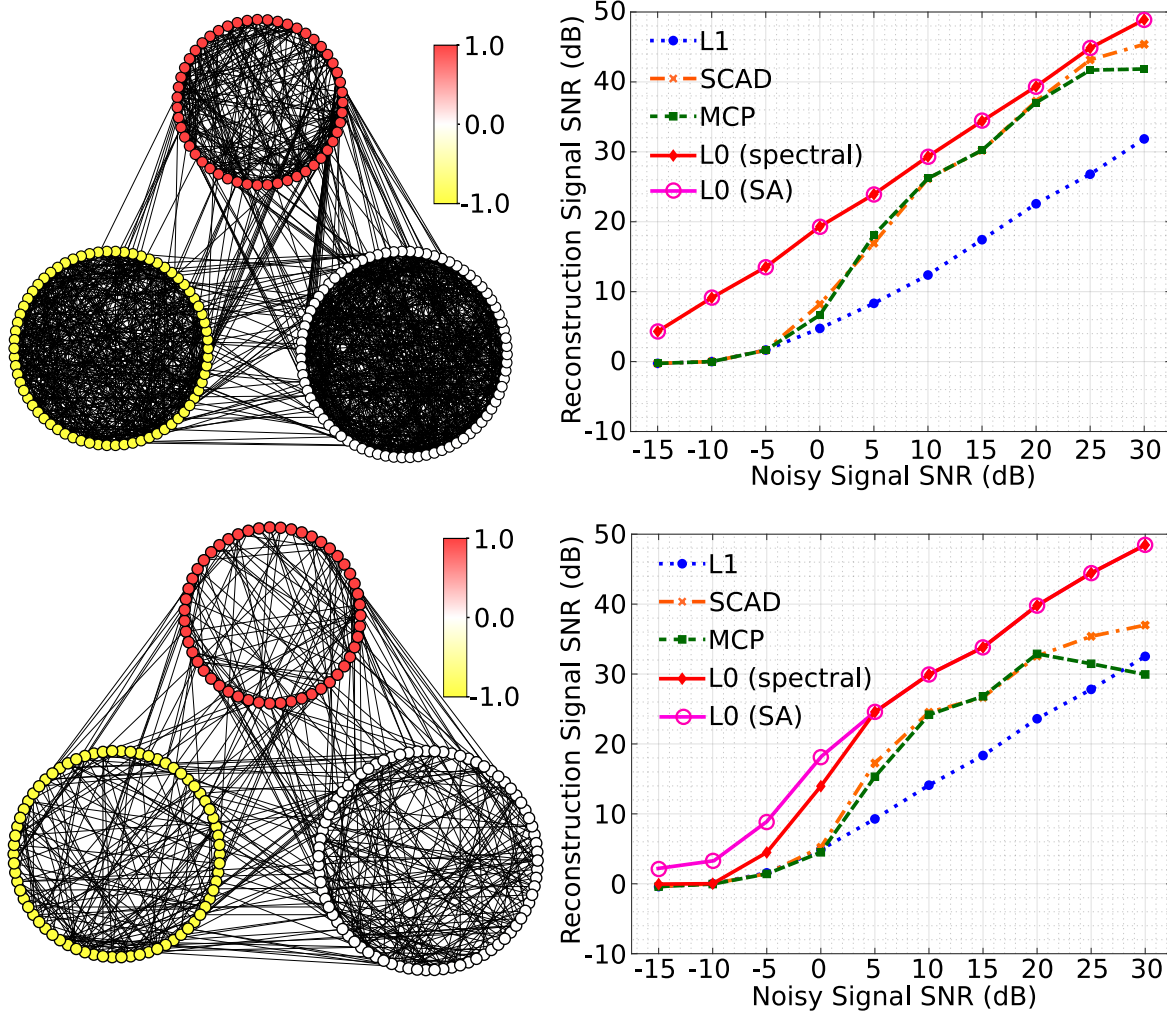
Fig. 3. Left: the graph structure of $G_1$ (top) and $G_2$ (bottom) and the corresponding ground truth signal input to each node in both graphs. Right: the corresponding plots of input signal SNR versus reconstructed signal SNR for the GTF models. Note that in both graphs, the $\ell_{2,0}$ model consistently outperforms other GFT models.

spectral method and the SA method obtains a perfect ROC curve, which significantly outperforms other models. The GTF models with nonconvex penalties (SCAD and MCP) have similar performance, which is better than the $\ell_1$ model.

It is not surprising that the proposed $\ell_{2,0}$ model achieves a significantly better result; see the discussion in Remark 2.3.

In Fig. 2 we also show that we can find the $k_*$ for $\mathcal{P}_2$ by screening different values of $k$ in **Algorithm** 2. As the search space of $k$ is small, hence the proposed method runs very fast. In addition, the SA method can also find the $k_*$ as long as we set $k$ larger than $k_*$.

In Fig. 2, we also showed that a "two-stage approach" for comparison, where the "two-stage approach" refers to first partition the graph and then estimating signals. As shown by the ROC curves, the two-stage approach is not competitive. The reason is that the communities in the graph are inconsistent with the clusters of the signals on the graph. That is, the communities identified by the community detection algorithm ( normalized graph cut algorithm) based on the graph topology do not agree with the clusters identified by the k-means algorithm based on the signal on the nodes. Clearly, in this graph the

clusters of the signal on nodes provide more insights about the location of the boundaries. Identifying communities first in the two-stage approach would miss the cluster structures expressed in the signal on nodes and, therefore, mislead the identification of the boundaries. This in fact showcases the importance of the proposed model, which is able to perform simultaneous graph clustering and k-means. In comparison to the two-stage approach, the proposed approach in this paper balances the information from the community detection on the graph topology and the clustering information on the signal (as shown in Theorem 2). Therefore, the proposed approach is better at detecting the boundaries.

### B. Denoising using GTF

We now compare the performance of the GTF models on denoising.

*1) Dataset: graph generation:* We generate a simple connected undirected unweighted graph $G(V, E)$ with 200 nodes based on a random graph model called Stochastic Block Model [38]. We plant three modules/communities in $G$, which have 50 nodes, 70 nodes, and 80 nodes, respectively. We

TABLE III
TIME COMPARISON (MEAN(STD)) WITH THE NUMBER OF EDGES IN GRAPHS INCREASING.

| | $(p=0.1, q=0.01)$ | | $(p=0.5, q=0.1)$ | | $(p=0.9, q=0.2)$ | |
|---|---|---|---|---|---|---|
| | $|E|$ | Time(s) | $|E|$ | Time(s) | $|E|$ | Time(s) |
| $\ell_1$ | 805(23) | 4.71(1.22) | 4707(69) | 56.74(20.43) | 8750(40) | 134.92(46.53) |
| SCAD | 805(23) | 5.22(1.24) | 4707(69) | 50.32(11.72) | 8750(40) | 106.03(25.34) |
| MCP | 805(23) | 7.71(2.26) | 4707(69) | 42.71(14.01) | 8750(40) | 105.64(33.92) |
| Alg. 2 | 805(23) | 0.01(0.01) | 4707(69) | 0.01(0.01) | 8750(40) | 0.01(0.01) |
| Alg. 3 | 805(23) | 990.54(5.42) | 4707(69) | 993.14(6.41) | 8750(40) | 992.94(8.91) |

introduce two parameters $(p, q)$: $p$ controls the probability of connections between nodes within the modules/communities and $q$ controls the probability of connections between nodes across the modules/communities. Then we generate two graphs $G_1, G_2$ as in Fig.3. For $G_1$, we set $p = 0.2$ and $q = 0.05$ and for $G_2$, we set $p = 0.05$ and $q = 0.01$. Clearly, the modular/community structure of $G_2$ is relatively less clear than that of $G_1$.

*2) Dataset: signal generation:* We generate a ground truth signal $y_* \in \mathbb{R}^{200}$ on the graphs $G_1$ and $G_2$ where on the module/community with 50 nodes has an input signal value of 1, on the module/community with 70 nodes has an input signal value of $-1$, and on the module/community with 80 nodes has input signal value of 0. We simulate $d = 10$ independent noisy realizations of the graph signal and concatenate them to construct a noisy vector-valued graph signal $Y = [y_* + \theta_1, y_* + \theta_2, ...] \in \mathbb{R}^{200 \times 10}$, where all $\theta_i \in \mathbb{R}^{200}$ are independently drew from $\theta_i \sim \mathcal{N}(0, \sigma^2 I)$.

*3) The learning task:* The goal here is to use different GTF models to recover the vector-valued signal under different input signal SNR controlled by $\sigma$. We recall that the reconstructed signal SNR is computed as $10 \log_{10}(\|Y^*\|_F / \|B - Y^*\|_F)$, where $B$ is the reconstructed signal.

*4) Parameter tuning:* We use the same setup of hyperparameters tuning as in the experiment on support recovery, and for each GTF model, we screen through the hyperparameters and report the best result.

*5) The results:* In Fig.3, we plot the SNR of the reconstructed signal on the graphs $G_1$ (top) and $G_2$ (bottom) across different SNR values of the input signal for all the GTF models. The curve of the $\ell_{2,0}$ model solved by the proposed method is above all the other GTF models, which demonstrates that our model can take advantage of the graph structure to recover the true signal. For the reconstruction of signal on the graph $G_2$, which is a harder problem as the modular/community structure of $G_2$ is less clear than $G_1$, the $\ell_{2,0}$ model still has a better reconstruction SNR than all other GTF models, with a consistent amount of at least 5dB better SNR in the high input SNR regime. Furthermore, we find that the SA method has better performance than the spectral method when the noisy signal SNR is smaller for signal recovery on $G_2$.

*C. Computational time*

In this section, we compare the computational time of the competing GTF models.

*1) Dataset:* We follow the procedures described in sections V-B1 and V-B2 to generate graph signals for time comparison. The number of nodes and the community structure of the graphs are the same as described in section V-B1. All generated graphs have 200 nodes and 3 communities with 50, 70, and 80 nodes, respectively. We control the number of edges of the graphs by setting different $(p, q)$ pairs (specified in Table III). We follow the procedure in section V-B2 to generate the signal on graphs.

*2) Parameter:* For each GTF model, we use the best parameters obtained in section V-B4.

*3) The results:* In Tabel III, we show the time comparison for different GTF models on denoising graph signals. For each $(p, q)$ pair, we generate 10 graph signals with 10 different graphs, which have the same community structure but different numbers of edges. We show the average and standard deviation of computational time for each GTF model. We observe that the computational times of previous GTF models ($\ell_1$, SCAD, and MCP) increase with the number of edges in the graphs. However, for the proposed GTF model, the computational times of the spectral and SA algorithms are not sensitive to the edge size of the graph.

*D. Semi-supervised classification*

We now report results on the Modified Absorption Problem (MAP). We compare the proposed MAP with the $\ell_{2,0}$-norm GTF regularization (13) with other MAP-GTF models proposed in [12], [11], which include MAP with $\ell_1$-norm GTF regularization [12], MAP with SCAD penalized GTF regularization [11], and a MAP with MCP penalized GTF regularization [11].

*1) Dataset:* We test the models on 8 popular classification datasets from [39]. For each dataset, we construct a 5-nearest neighbor graph based on the Euclidean distance between provided features. We randomly pick 20% samples from the dataset and treat them as labeled samples. For each dataset, we compute the misclassification rates over 100 repetitions.

TABLE IV
MISCLASSIFICATION RATES AVERAGED OVER 100 TRIALS, WITH THE VARIANCE BELOW THE RATES.

| | heart | wine | iris | breast | car | wine-qual. | ads | yeast |
|---|---|---|---|---|---|---|---|---|
| # of classes | 2 | 3 | 3 | 2 | 4 | 5 | 2 | 10 |
| # of samples | 303 | 178 | 150 | 569 | 1,728 | 1,599 | 3,279 | 1,484 |
| $\ell_1$ | 0.158 | 0.040 | 0.051 | 0.058 | 0.170 | 0.574 | 0.152 | 0.391 |
| | (4e-4) | (2e-4) | (5e-4) | (4e-4) | (6e-5) | (7e-4) | (1e-4) | (4e-4) |
| SCAD | 0.150 | 0.041 | 0.049 | 0.045 | 0.149 | 0.574 | 0.153 | 0.673 |
| | (4e-2) | (5e-4) | (2e-4) | (2e-4) | (3e-5) | (1e-5) | (4e-4) | (3e-5) |
| MCP | 0.148 | 0.041 | 0.048 | 0.041 | 0.148 | 0.572 | 0.150 | 0.507 |
| | (3e-4) | (1e-4) | (3e-4) | (2e-4) | (1e-5) | (1e-6) | (5e-4) | (1e-5) |
| $\ell_{2,0}$ (see section IV) | **0.143** | **0.035** | **0.038** | **0.033** | **0.098** | **0.373** | **0.079** | **0.372** |
| | **(2e-4)** | **(6e-5)** | **(3e-4)** | **(1e-4)** | **(1e-6)** | **(3e-4)** | **(5e-4)** | **(5e-5)** |

*2) Parameter tuning:* For all the MAP-GTF models, we set $\epsilon = 0.01$ (the regularization parameter in (13)) and $R_{ij} = 1/K$. For each model, we screen through the hyper-parameters and report the best results.

*3) The results:* Table IV summarizes the misclassification rate for all the methods over 100 repetitions of randomly selected labeled samples. The proposed MAP-GTF model (13) achieves the smallest misclassification rate. We further do a t-test based on the misclassification rates over 100 repetitions and find that the misclassification rates of the proposed MAP-GTF model (13) for all the datasets are significantly smaller than other methods ($p$-value is less than 0.001).

*4) Concluding remarks:* Lastly, we comment on the comparisons of these GTF regularizers on MAP. The experiment results all show that the $\ell_1$ model while being convex and having the least number of parameters (see Table II), is consistently having the worst performance among all the tested models. For the remaining nonconvex models, the $\ell_{2,0}$ model consistently outperforms SCAD and MCP, and has fewer parameters to tune. We believe that the superior performance of the MAP model with the $\ell_{2,0}$-norm GTF regularization in such a broad range of experiments is convincing evidence for the utility of the $\ell_{2,0}$-norm GTF regularizer.

## VI. CONCLUSION

We propose a first-order GTF model with $\ell_{2,0}$-norm penalty. By converting it into a mixed integer program, we prove that the model is equivalent to a k-means clustering with graph cut regularization (Theorem 2). We propose two methods to solve the problem: approximate spectral method and simulated annealing. Empirically, we demonstrate the performance improvements of the proposed GTF model on both synthetic and real datasets for support recovery, signal estimation, and semi-supervised classification.

We believe that the superior performance of the proposed models in a broad range of experiments is convincing evidence for the utility of $\ell_{2,0}$-norm penalized GTF. For future works, we think it will be interesting to investigate fast algorithms for

solving the GTF model and study the error bound of the model. In addition, we believe that higher-order GTF models (see Fig.3 in [12]) with $\ell_0$-norm penalty also have great potential.

## REFERENCES

[1] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete Signal Processing on Graphs: Sampling Theory," *IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6510–6523, 2015.

[2] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal Recovery on Graphs: Variation Minimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.

[3] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-Based Reconstruction of Graph Signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 764–778, 2017.

[4] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal Discrete Regularization on Weighted Graphs: A Framework for Image and Manifold Processing," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.

[5] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The Emerging Field of Signal Processing on Graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[6] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[7] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and Semi-supervised Learning on Large Graphs," in *Learning Theory*, J. Shawe-Taylor and Y. Singer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 624–638.

[8] S. Bougleux, A. Elmoataz, and M. Melkemi, "Discrete Regularization on Weighted Graphs for Image and Mesh Filtering," in *Scale Space and Variational Methods in Computer Vision*, F. Sgallari, A. Murli, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 128–139.

[9] D. A. Spielman, "Spectral Graph Theory and its Applications," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 2007, pp. 29–38.

[10] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[11] R. Varma, H. Lee, J. Kovačević, and Y. Chi, "Vector-Valued Graph Trend Filtering With Non-Convex Penalties," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 48–62, 2020.

[12] Y. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani, "Trend Filtering on Graphs," *Journal of Machine Learning Research*, vol. 17, no. 105, pp. 1–41, 2016.

[13] G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. G. Bagos, "Using graph theory to analyze biological networks," *BioData Mining*, vol. 4, p. 10, Apr 2011.

[14] D. Szklarczyk, R. Kirsch, M. Koutrouli, K. Nastou, F. Mehryary, R. Hachilif, A. L. Gable, T. Fang, N. T. Doncheva, S. Pyysalo *et al.*, "The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest," *Nucleic Acids Research*, vol. 51, no. D1, pp. D638–D646, 2023.

[15] C. Y. Kim, S. Baek, J. Cha, S. Yang, E. Kim, E. M. Marcotte, T. Hart, and I. Lee, "HumanNet v3: an improved database of human gene networks for disease research," *Nucleic Acids Research*, vol. 50, no. D1, pp. D632–D639, Jan. 2022.

[16] I. Rodchenkov, O. Babur, A. Luna, B. A. Aksoy, J. V. Wong, D. Fong, M. Franz, M. C. Siper, M. Cheung, M. Wrana, H. Mistry, L. Mosier, J. Dlin, Q. Wen, C. O'Callaghan, W. Li, G. Elder, P. T. Smith, C. Dallago, E. Cerami, B. Gross, U. Dogrusoz, E. Demir, G. D. Bader, and C. Sander, "Pathway Commons 2019 Update: integration, analysis and exploration of pathway data," *Nucleic Acids Research*, vol. 48, no. D1, pp. D489–D497, Jan. 2020.

[17] M. Hofree, J. P. Shen, H. Carter, A. Gross, and T. Ideker, "Network-based stratification of tumor mutations," *Nature Methods*, vol. 10, no. 11, pp. 1108–1115, Nov. 2013.

[18] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science*, vol. 302, no. 5643, pp. 249–255, Oct 2003.

[19] F. R. K. Chung, *Spectral Graph Theory*.  American Mathematical Society, 1997.

[20] N. Fan and P. Pardalos, "Multi-way clustering and biclustering by the Ratio cut and Normalized cut in graphs," *Journal of Combinatorial Optimization*, vol. 23, no. 2, pp. 224–251, Sep. 2010.

[21] V. Vazirani, *Approximation Algorithms*.  Springer, 2001.

[22] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is NP-hard," in *WALCOM: Algorithms and Computation*, S. Das and R. Uehara, Eds.  Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 274–285.

[23] M. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, Sep. 2006.

[24] M. Udell and A. Townsend, "Why Are Big Data Matrices Approximately Low Rank?" *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 1, pp. 144–160, 2019.

[25] C. Alpert, A. Kahng, and S.-Z. Yao, "Spectral partitioning with multiple eigenvectors," *Discrete Applied Mathematics*, vol. 90, no. 1-3, pp. 3–26, Jan. 1999.

[26] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical Review E*, vol. 74, p. 016110, Jul 2006.

[27] E. Carlon, "Computational Physics," KU Leuven, Tech. Rep., 2013.

[28] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[29] M. P. Allen and D. J. Tildesley, "Advanced Monte Carlo methods," in *Computer Simulation of Liquids*.  Oxford University Press, 06 2017.

[30] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*.  Oxford University Press, 02 1996.

[31] S. Z. Selim and K. Alsultan, "A simulated annealing algorithm for the clustering problem," *Pattern Recognition*, vol. 24, no. 10, pp. 1003–1008, 1991.

[32] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning," *Operations Research*, vol. 37, no. 6, pp. 865–892, Dec. 1989.

[33] P. Talukdar and K. Crammer, "New Regularized Algorithms for Transductive Learning," in *Machine Learning and Knowledge Discovery in Databases*, 2009, pp. 442–457.

[34] P. Talukdar and F. Pereira, "Experiments in Graph-based Semi-Supervised Learning Methods for Class-Instance Acquisition," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, Jul. 2010, pp. 1473–1481.

[35] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[36] M. Jaggi, "Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization," in *Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol. 28, pp. 17–19.

[37] R. A. Rossi and N. K. Ahmed, "The Network Data Repository with Interactive Graph Analytics and Visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.  AAAI Press, 2015, p. 4292–4293.

[38] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

[39] D. Dua and C. Graff, "UCI Machine Learning Repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml