A Linearly Convergent Optimization Framework for Learning Graphs from Smooth Signals

Xiaolu Wang, Chaorui Yao, and Anthony Man-Cho So

***Abstract*—Learning graph structures from a collection of smooth graph signals is a fundamental problem in data analysis and has attracted much interest in recent years. Although various optimization formulations of the problem have been proposed in the literature, existing methods for solving them either are not practically efficient or lack strong convergence guarantees. In this paper, we consider a unified graph learning formulation that captures a wide range of static and time-varying graph learning models and develop a first-order method for solving it. By showing that the set of Karush-Kuhn-Tucker points of the formulation possesses a so-called *error bound property*, we establish the linear convergence of our proposed method. Moreover, through extensive numerical experiments on both synthetic and real data, we show that our method exhibits sharp linear convergence and can be substantially faster than a host of other existing methods. To the best of our knowledge, our work is the first to develop a first-order method that not only is practically efficient but also enjoys a linear convergence guarantee when applied to a large class of graph learning models.**

***Index Terms*—Graph learning, graph signal processing, proximal ADMM, error bound, linear convergence**

I. Introduction

Graph is a fundamental mathematical object that has long been used to model structural relationships among different entities. Motivated by various types of real-world data from, e.g., social networks, brain signal analysis, and urban traffic flows, there has been much interest in recent years to take additional attributes of the entities into account and model them as signals that reside on the graph [1], [2]. This gives birth to the field of graph signal processing [3], which aims to develop signal processing techniques to better understand the interplay between graph topology and graph signals. Numerous methods in signal processing and machine learning, such as sampling [4], filtering [5], and classification [6], have been developed to deal with graph-structured data. Nevertheless, the concrete graph topology is often not known a priori, which hinders further analysis and processing of the data. In some applications, such as brain networks [7], the graph connectivity itself is exactly what we want to find. Therefore, it is crucial to learn the topology of the underlying graph from a given set of graph signals.

To formally describe the graph learning problem, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with $\mathcal{V} = [m]$ being the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ being the set of edges.[1] We assume that $(i,i) \notin \mathcal{E}$ for all $i \in [m]$, which means that there is no self-loop in the graph. The graph structure is characterized by a symmetric and non-negative weight matrix $\boldsymbol{W} \in \mathbb{R}^{m\times m}$, where $W_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}$. A graph signal $\boldsymbol{x}$ is usually represented by a column vector in $\mathbb{R}^m$, whose i-th coordinate x_i is the signal value on node i. In many applications, the weight matrix $\boldsymbol{W}$ is not known. Thus, our goal is to learn the underlying graph structure $\boldsymbol{W}$ from a collection of $\bar{n}$ graph signals $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{\bar{n}} \in \mathbb{R}^m$.

Obviously, some prior knowledge about the relationship between the graph structure and the graph signal is required to achieve the goal of graph learning. One common assumption is that the graph signal varies smoothly across the graph [8]–[10]. Intuitively, a signal is smooth on the graph if any two nodes that are connected by an edge with a large weight have similar signal values on them. A standard measure of the smoothness of a signal $\boldsymbol{x}$ on the graph $\mathcal{G}$ is the Laplacian quadratic form, which employs the Laplacian matrix $\boldsymbol{L} := \mathrm{Diag}(\boldsymbol{W}\boldsymbol{1}) - \boldsymbol{W}$ of $\mathcal{G}$ and is defined as

$$\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} W_{ij}(x_i - x_j)^2.$$

In particular, for the graph signals $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{\bar{n}} \in \mathbb{R}^m$ that reside on the same underlying graph $\mathcal{G}$ with Laplacian matrix $\boldsymbol{L}$, the following quantity, also known as the Dirichlet energy, is used to measure their overall smoothness:

$$\sum_{k=1}^{\bar{n}} \boldsymbol{x}_k^\top \boldsymbol{L}\boldsymbol{x}_k = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} W_{ij}\left\|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\right\|_2^2 = \frac{1}{2}\|\boldsymbol{W}\odot\boldsymbol{D}\|_{1,1}. \tag{1}$$

Here, $\tilde{\boldsymbol{x}}_i := [(\boldsymbol{x}_1)_i, \dots, (\boldsymbol{x}_{\bar{n}})_i]^\top$ is the data vector associated with the i-th node and $D_{ij} := \left\|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\right\|_2^2$ is the squared pairwise distance between the node vectors $\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j$.

A. Models for Learning Graphs from Smooth Signals

Based on the signal smoothness prior, we consider a general setting for graph learning, where the underlying graph structure may vary over time. Suppose that the graph signals $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^m$ are sequentially collected from T non-overlapping time slots. Within the t-th time slot, where $t = 1, \dots, T$, the weight matrix $\boldsymbol{W}^{(t)} \in \mathbb{R}^{m\times m}$ of the underlying graph remains static. The n graph signals are partitioned into T disjoint groups and the t-th group $\{\boldsymbol{x}_1^{(t)}, \dots, \boldsymbol{x}_{n_t}^{(t)}\}$ consists of signals collected in the t-th time

Xiaolu Wang and Anthony Man-Cho So are with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, HKSAR, China (e-mails: {xlwang, manchoso}@se.cuhk.edu.hk). Chaorui Yao is with the Department of Electrical and Computer Engineering, University of California, Los Angeles, USA (e-mail: chaorui@ucla.edu).

[1] A summary of the notation used in this paper can be found in Section I-D.

slot. It follows that $\sum_{t=1}^{T} n_t = n$. For $t = 1, \dots, T$, let $\boldsymbol{D}^{(t)} \in \mathbb{R}^{m \times m}$ be given by $(\boldsymbol{D}^{(t)})_{ij} := \|\tilde{\boldsymbol{x}}_i^{(t)} - \tilde{\boldsymbol{x}}_j^{(t)}\|_2^2$ with $\tilde{\boldsymbol{x}}_i^{(t)} := [(\boldsymbol{x}_1^{(t)})_i, \dots, (\boldsymbol{x}_{n_t}^{(t)})_i]^\top$ and $\boldsymbol{D} := [\boldsymbol{D}^{(1)}, \dots, \boldsymbol{D}^{(T)}]$. The goal then is to infer the (possibly time-varying) graphs represented by $\Xi_T := [\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(T)}]$. In this paper, we consider the following formulation for this task:

$$\begin{aligned} \min_{\Xi_T} \ & \mathcal{F}_T(\Xi_T) + \mathcal{H}_T(\Xi_T) + \mathcal{R}_T(\Xi_T) \\ \text{s.t.} \ & \boldsymbol{W}^{(t)} \geq \boldsymbol{0}, \ \boldsymbol{W}^{(t)} = (\boldsymbol{W}^{(t)})^\top, \ \operatorname{diag}(\boldsymbol{W}^{(t)}) = \boldsymbol{0}, \\ & \text{for } t = 1, \dots, T. \end{aligned} \tag{2}$$

Here,

$$\begin{aligned} \mathcal{F}_T(\Xi_T) &:= \sum_{t=1}^{T} \|\boldsymbol{W}^{(t)} \odot \boldsymbol{D}^{(t)}\|_{1,1}, \\ \mathcal{H}_T(\Xi_T) &:= \sum_{t=1}^{T} -\alpha \boldsymbol{1}^\top \log(\boldsymbol{W}^{(t)}\boldsymbol{1}) + \frac{\beta}{2}\|\boldsymbol{W}^{(t)}\|_F^2, \end{aligned}$$

and $\alpha, \beta > 0$ are given parameters. The first term $\mathcal{F}_T$ promotes the overall signal smoothness on the learned graphs $\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(T)}$. If we take $T = 1$, then $\mathcal{F}_1$ reduces to the Dirichlet energy in (1). The second term $\mathcal{H}_T$ induces extra structural properties in each time slot. Specifically, it combines the logarithmic barrier with the squared Frobenius norm to control both connectivity and density of the learned graphs [9], [11]. The logarithmic barrier term, albeit not having a Lipschitz continuous gradient, is crucial to the formulation, as it rules out both the trivial all-zero solution to (2) and isolated nodes in the learned graphs [9]. The third term $\mathcal{R}_T$ is used to impose certain temporal variation prior on the time-varying graphs for $T \geq 2$. There are several common choices of $\mathcal{R}_T$. Given a parameter $\gamma > 0$, we can consider the Tikhonov regularization [12], [13]

$$\mathcal{R}_T(\Xi_T) = \frac{\gamma}{2} \sum_{t=1}^{T-1} \|\boldsymbol{W}^{(t+1)} - \boldsymbol{W}^{(t)}\|_F^2, \tag{3}$$

which aims to promote graphs whose edges change smoothly over time; the $L_{1,1}$-norm [11], [13]

$$\mathcal{R}_T(\Xi_T) = \frac{\gamma}{2} \sum_{t=1}^{T-1} \|\boldsymbol{W}^{(t+1)} - \boldsymbol{W}^{(t)}\|_{1,1}, \tag{4}$$

which aims to induce sparsity in the temporal variation of the graphs (i.e., most edges remain unchanged between successive time slots); and the structured temporal variation regularizer [14]

$$\mathcal{R}_T(\Xi_T) = \frac{\gamma}{2} \sum_{(u,v) \in \mathcal{M}} \|\boldsymbol{W}^{(u)} - \boldsymbol{W}^{(v)}\|_{1,1} \tag{5}$$

with $\mathcal{M} \subseteq [T] \times [T]$ being the edge set of the so-called *temporal graph*, which generalizes the $L_{1,1}$-norm (by taking the temporal graph to be the chain $\mathcal{M} = \{(1,2), (2,3), \dots, (T-1, T)\}$) and provides a way to capture other sparsity patterns in the temporal variation.

The formulation (2) is very general and subsumes many effective graph learning formulations in the literature. In particular, by taking $T = 1$ and $\mathcal{R}_T \equiv 0$, we obtain the classic static graph learning model [9], [15]

$$\begin{aligned} \min_{\boldsymbol{W} \in \mathbb{R}^{m \times m}} \ & \|\boldsymbol{W} \odot \boldsymbol{D}\|_{1,1} - \alpha \boldsymbol{1}^\top \log(\boldsymbol{W}\boldsymbol{1}) + \frac{\beta}{2}\|\boldsymbol{W}\|_F^2 \\ \text{s.t.} \ & \boldsymbol{W} \geq \boldsymbol{0}, \ \boldsymbol{W} = \boldsymbol{W}^\top, \ \operatorname{diag}(\boldsymbol{W}) = \boldsymbol{0}. \end{aligned} \tag{6}$$

For $T \geq 2$ and different choices of the temporal regularization function $\mathcal{R}_T$, we obtain different time-varying graph learning models [11]–[14], [16].

B. Existing Optimization Methods

A commonly adopted strategy for solving graph learning formulations of the form (2) is to use primal–dual methods (see [17] for an overview of these methods). Indeed, such a strategy has been pursued in the cases where (i) $T = 1$ and $\mathcal{R}_T \equiv 0$ [9], (ii) $T \geq 2$ and $\mathcal{R}_T$ is given by (3) [12], and (iii) $T \geq 2$ and $\mathcal{R}_T$ is given by (4) [11], [16]. However, none of the mentioned works provides a rigorous convergence analysis of the primal–dual method used. More critically, none of those primal–dual methods are well defined, as the iterates generated by them are not guaranteed to lie in the domain of the logarithmic barrier in $\mathcal{H}_T$. In practice, we observe that the primal–dual methods for solving static and time-varying graph learning models tend to converge rather slowly.

Recently, several alternative approaches that are more efficient than the primal–dual one have been proposed for solving the static graph learning model (6). One approach, which is first developed in [18], is based on the alternating direction method of multipliers (ADMM). The method is shown to be globally convergent and achieves a better performance than the primal–dual method proposed in [9]. Another approach is to apply the proximal gradient method to the dual of (6) [19]. By adapting the convergence result of FISTA [20], the method, which is termed fast dual proximal gradient (FDPG) in [19], is shown to converge at a sublinear rate when solving the dual of (6). Lastly, an approach based on the majorization-minimization (MM) method is developed in [21], which solves a majorizing surrogate problem in each step. The MM method is shown to converge to the optimal solution to Problem (6) but its convergence rate is not known. All these three algorithms have the same per-iteration computational complexity.

From the above discussion, we see that existing optimization methods for graph learning under the smoothness prior are designed only for particular graph learning models. Moreover, they lack either computational efficiency in practice or strong convergence guarantees in theory. Although Problem (2) is a convex program when $\mathcal{R}_T$ is a convex function, it does not possess properties such as strong convexity or Lipschitz continuity of the gradient. Thus, it is difficult to prove that existing methods enjoy fast (e.g., linear) convergence rates, even though some of them (e.g., the ADMM in [18]) do exhibit fast convergence empirically.

C. Our Contributions

In this paper, we develop a unified convex optimization framework for learning both static and time-varying graphs from smooth signals that can overcome the aforementioned limitations. We summarize our main contributions as follows:

- We show that Problem (2) can be reformulated as a nonsmooth convex program with linear equality constraints, whose structure can be effectively exploited by the proximal ADMM (pADMM) [22]. Based on this, we develop a pADMM-based optimization framework, which we refer to as pADMM-GL, and show that it can be applied to efficiently solve different instantiations of (2).
- We show that our pADMM-GL globally converges to an optimal solution to Problem (2). Moreover, we establish the linear convergence rate of pADMM-GL. Note that our linear convergence result does not follow from standard convergence analyses of the ADMM in the literature (such as [23]), as they require the objective function to have a term that is strongly convex and has Lipschitz continuous gradient—a requirement that is not satisfied by Problem (2). Instead, we need to show that the set of Karush-Kuhn-Tucker (KKT) points of the aforementioned reformulation of Problem (2) possesses a so-called *error bound property*, which is non-trivial and can be of independent interest. To the best of our knowledge, our proposed pADMM-GL is the first provably linearly convergent first-order method for graph learning from smooth signals.
- We show via extensive numerical experiments that our proposed pADMM-GL exhibits sharp linear convergence on both synthetic and real data. Moreover, its convergence performance and computation time are superior to those of other state-of-the-art algorithms.

We also remark that this paper extends its earlier conference version [18] in four major aspects. First, our conference paper [18] only considers the static graph learning model (6), while this paper considers both static and time-varying graph learning models (with different temporal regularizers for the latter) and casts them into the unified framework (2). Second, this paper explains in detail the theoretical and computational considerations behind the choices of the proximal terms (see (18a) and (18b)) in our proposed pADMM-GL; see Section II-B. This fills a missing piece in our conference paper [18]. Third, our conference paper [18] only shows that the ADMM proposed therein converges to an optimal solution to the static graph learning model (6). By contrast, this paper establishes not only the convergence but also the rate of convergence of the proposed pADMM-GL to an optimal solution to the unified graph learning formulation (2). This is achieved by a much more involved analysis of the optimization problem at hand than that in [18]; see Section III-B. Fourth, this paper presents a much richer set of numerical results than our conference paper [18]. In particular, it includes numerical comparisons of our proposed pADMM-GL with the recently proposed FDPG [19] and MM [21] methods for static graph learning and with the primal–dual methods in [11], [12], [16] for time-varying graph learning.

Before we leave this subsection, let us briefly comment on the work [24]. Although our work may seem similar to [24] in that both propose ADMM-type methods for graph learning, they actually focus on rather different settings. Indeed, the work [24] considers the setting where the graph topology is known and the graph signal is assumed to be generated from a Gaussian Markov random field model, and the goal is to perform statistical estimation of the weight matrix under the given graph topology constraints. By contrast, our work considers the setting where the graph topology is *not* known and the graph signal is assumed to vary smoothly across the graph, and the goal is to find a weight matrix that minimizes certain regularized form of the Dirichlet energy. In particular, the graph signal is not assumed to follow any generative model. Even from an algorithmic point of view, the results obtained in our work are different from and go substantially beyond those in [24]. Specifically, our work establishes the convergence rate of the proposed pADMM-GL, which requires the development of new techniques. By contrast, the work [24] simply invokes existing results in the literature to conclude the convergence of the proposed ADMM-type methods.

D. Notation

Given a positive integer m, we define $[m] \coloneqq \{1, 2, \dots, m\}$. We use $\mathbf{1}$ (resp. $\mathbf{0}$) to denote the all-one (resp. all-zero) matrix whose dimension will be clear from the context.

Given vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, we use a_i, $\|\boldsymbol{a}\|_2$, $\log(\boldsymbol{a})$, $\boldsymbol{a}^2$, and $\sqrt{\boldsymbol{a}}$ to denote the i-th coordinate, ℓ_2-norm, element-wise logarithm, element-wise square, and element-wise square root of $\boldsymbol{a}$, respectively; $\mathrm{Diag}(\boldsymbol{a})$ to denote the diagonal matrix with $\boldsymbol{a}$ on its diagonal; $\boldsymbol{a}/\boldsymbol{b}$ to denote the element-wise division of $\boldsymbol{a}$ and $\boldsymbol{b}$.

Given matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we use A_{ij}, $\|\boldsymbol{A}\|_F$, $\|\boldsymbol{A}\|_{1,1}$, and $\|\boldsymbol{A}\|_2$ to denote the (i, j)-th element, Frobenius norm, $L_{1,1}$-norm, and spectral norm of $\boldsymbol{A}$, respectively; $\mathrm{diag}(\boldsymbol{A})$ to denote the vector formed using the diagonal entries of $\boldsymbol{A}$; $\boldsymbol{A} \odot \boldsymbol{B}$ and $\boldsymbol{A} \otimes \boldsymbol{B}$ to denote the Hadamard product and Kronecker product of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively; $[\boldsymbol{A}; \boldsymbol{B}] \coloneqq \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix}$ to denote the block column matrix generated by $\boldsymbol{A}$ and $\boldsymbol{B}$.

Given a vector $\boldsymbol{a}$, a set $\mathcal{A}$ in the same space, and a positive semidefinite matrix $\boldsymbol{A}$, we use

$$\boldsymbol{a} \mapsto \mathbb{1}_{\mathcal{A}}(\boldsymbol{a}) \coloneqq \begin{cases} 0, & \boldsymbol{a} \in \mathcal{A} \\ +\infty, & \boldsymbol{a} \notin \mathcal{A} \end{cases}$$

to denote the indicator function associated with $\mathcal{A}$,

$$\mathcal{C}_{\mathcal{A}}(\boldsymbol{a}) \coloneqq \left\{ \boldsymbol{v} \mid \boldsymbol{v}^\top (\boldsymbol{b} - \boldsymbol{a}) \leq 0 \text{ for all } \boldsymbol{b} \in \mathcal{A} \right\}$$

to denote the normal cone of $\mathcal{A}$ at $\boldsymbol{a}$, $\|\boldsymbol{a}\|_{\boldsymbol{A}} \coloneqq (\boldsymbol{a}^\top \boldsymbol{A} \boldsymbol{a})^{1/2}$ to denote the $\boldsymbol{A}$-norm of $\boldsymbol{a}$, and

$$\mathrm{dist}(\boldsymbol{a}, \mathcal{A}) \coloneqq \inf \left\{ \|\boldsymbol{a} - \boldsymbol{b}\|_2 \mid \boldsymbol{b} \in \mathcal{A} \right\},$$
$$\mathrm{dist}_{\boldsymbol{A}}(\boldsymbol{a}, \mathcal{A}) \coloneqq \inf \left\{ \|\boldsymbol{a} - \boldsymbol{b}\|_{\boldsymbol{A}} \mid \boldsymbol{b} \in \mathcal{A} \right\}$$

to denote the distance between $\boldsymbol{a}$ and $\mathcal{A}$ under the Euclidean norm and $\boldsymbol{A}$-norm, respectively.

Given a proper extended real-valued function f, we use $\mathrm{dom}(f) \coloneqq \{\boldsymbol{a} \mid f(\boldsymbol{a}) < +\infty\}$ to denote its domain,

$$\mathrm{prox}_f(\boldsymbol{a}) \coloneqq \arg\min_{\boldsymbol{b}} \left\{ f(\boldsymbol{b}) + \frac{1}{2} \|\boldsymbol{b} - \boldsymbol{a}\|_2^2 \right\}$$

to denote the proximal mapping of f evaluated at $\boldsymbol{a} \in \mathrm{dom}(f)$, and

$$\partial f(\boldsymbol{a}) := \big\{\boldsymbol{c} \mid f(\boldsymbol{b}) \geq f(\boldsymbol{a}) + \boldsymbol{c}^\top(\boldsymbol{b} - \boldsymbol{a}) \text{ for all } \boldsymbol{b}\big\}$$

to denote the subdifferential of f at $\boldsymbol{a} \in \mathrm{dom}(f)$.

Given a set-valued mapping $\Psi : \mathbb{R}^{n_1} \rightrightarrows \mathbb{R}^{n_2}$, which assigns a subset of $\mathbb{R}^{n_2}$ to a point in $\mathbb{R}^{n_1}$, we use

$$\mathrm{gph}(\Psi) := \{(\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid \boldsymbol{b} \in \Psi(\boldsymbol{a})\}$$

to denote its graph and $\Psi^{-1} : \mathbb{R}^{n_2} \rightrightarrows \mathbb{R}^{n_1}$ given by

$$\Psi^{-1}(\boldsymbol{b}) := \{\boldsymbol{a} \in \mathbb{R}^{n_1} \mid \boldsymbol{b} \in \Phi(\boldsymbol{a})\}$$

to denote its inverse mapping. If $\Psi : \mathbb{R}^{n_1} \rightrightarrows \mathbb{R}^{n_2}$ assigns a singleton of $\mathbb{R}^{n_2}$ to a point in $\mathbb{R}^{n_1}$, then Ψ is called a single-valued mapping.

E. Paper Organization

The rest of this paper is organized as follows. In Section II, we give a reformulation of the unified graph learning model (2) and develop a pADMM-based framework to tackle it. We then establish in Section III the global convergence and local linear convergence of our proposed method. Section IV presents a comparison of the numerical performance of our proposed method with that of existing ones. Lastly, we give some concluding remarks in Section V.

II. Proposed Optimization Framework

A. Problem Reformulation

Let us begin with a reformulation of Problem (2), which exposes the underlying structure of the problem and facilitates the design of an optimization framework for solving it. To fix ideas, let us take $\mathcal{R}_T$ to be the $L_{1,1}$-norm regularizer (4) and consider, for any $T \geq 1$, $\alpha, \beta > 0$, and $\gamma \geq 0$, the following instance of Problem (2):

$$\begin{aligned} \min_{\boldsymbol{\Xi}_T} \ & \mathcal{F}_T(\boldsymbol{\Xi}_T) + \mathcal{H}_T(\boldsymbol{\Xi}_T) + \frac{\gamma}{2}\sum_{t=1}^{T-1} \|\boldsymbol{W}^{(t+1)} - \boldsymbol{W}^{(t)}\|_{1,1} \\ \text{s.t. } \ & \boldsymbol{W}^{(t)} \geq \boldsymbol{0},\ \boldsymbol{W}^{(t)} = (\boldsymbol{W}^{(t)})^\top,\ \mathrm{diag}(\boldsymbol{W}^{(t)}) = \boldsymbol{0}, \\ & \text{for } t = 1, \dots, T. \end{aligned} \tag{7}$$

Let $\boldsymbol{w}^{(t)}$ (resp. $\boldsymbol{d}^{(t)}$) be the vector formed by stacking the entries above the main diagonal of $\boldsymbol{W}^{(t)}$ (resp. $\boldsymbol{D}^{(t)}$) in a column and set $\boldsymbol{w} := \big[\boldsymbol{w}^{(1)}; \boldsymbol{w}^{(2)}; \dots; \boldsymbol{w}^{(T)}\big] \in \mathbb{R}^{Tp}$ and $\boldsymbol{d} := \big[\boldsymbol{d}^{(1)}; \boldsymbol{d}^{(2)}; \dots; \boldsymbol{d}^{(T)}\big] \in \mathbb{R}^{Tp}$ with $p := m(m-1)/2$. We can then rewrite (7) as

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^{Tp}} \ & 2\boldsymbol{d}^\top \boldsymbol{w} - \alpha \boldsymbol{1}^\top \log(\boldsymbol{B}\boldsymbol{w}) + \beta\|\boldsymbol{w}\|_2^2 + \gamma\|\boldsymbol{B}'\boldsymbol{w}\|_1 \\ \text{s.t.} \ \ & \boldsymbol{w} \geq 0, \end{aligned} \tag{8}$$

where $\boldsymbol{B} \in \{0,1\}^{Tm \times Tp}$ and $\boldsymbol{B}' \in \{0, \pm 1\}^{Tp \times Tp}$ satisfy

$$\boldsymbol{B}\boldsymbol{w} = [\boldsymbol{W}^{(1)}\boldsymbol{1}; \dots; \boldsymbol{W}^{(T)}\boldsymbol{1}], \tag{9}$$

$$\boldsymbol{B}'\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}' \tag{10}$$

with $\boldsymbol{w}' := \big[\boldsymbol{w}^{(1)}; \boldsymbol{w}^{(1)}; \dots; \boldsymbol{w}^{(T-1)}\big] \in \mathbb{R}^{Tp}$. From the defining equations (9) and (10), it is not hard to deduce that $\boldsymbol{B}$ is given by

$$\boldsymbol{B} = \boldsymbol{I}_T \otimes \boldsymbol{S}, \tag{11}$$

where $\boldsymbol{S} \in \{0,1\}^{m \times p}$ is a matrix that has exactly $m-1$ ones in each row and satisfies $\boldsymbol{S}\boldsymbol{w}^{(t)} = \boldsymbol{W}^{(t)}\boldsymbol{1}$ for $t = 1, \dots, T$, and $\boldsymbol{B}'$ is given by

$$\boldsymbol{B}' = \left[\begin{array}{ccccc} & & \boldsymbol{0}_{p \times Tp} & & \\ \hdashline -\boldsymbol{I}_p & \boldsymbol{I}_p & & & \\ & -\boldsymbol{I}_p & \boldsymbol{I}_p & & \\ & & \ddots & \ddots & \\ & & & -\boldsymbol{I}_p & \boldsymbol{I}_p \end{array}\right]. \tag{12}$$

By introducing the new variables $\boldsymbol{v} := [\boldsymbol{v}_1; \boldsymbol{v}_2]$ with $\boldsymbol{v}_1 \in \mathbb{R}^{Tm}$, $\boldsymbol{v}_2 \in \mathbb{R}^{Tp}$ and setting $\boldsymbol{B}\boldsymbol{w} = \boldsymbol{v}_1$, $\boldsymbol{B}'\boldsymbol{w} = \boldsymbol{v}_2$, we obtain the following reformulation of Problem (8):

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^{Tp}, \boldsymbol{v} \in \mathbb{R}^{T(m+p)}} \ & f_T(\boldsymbol{w}) + g_T(\boldsymbol{v}) \\ \text{s.t.} \qquad & \boldsymbol{C}\boldsymbol{w} - \boldsymbol{v} = \boldsymbol{0}. \end{aligned} \tag{13}$$

Here,

$$\boldsymbol{C} := [\boldsymbol{B}; \boldsymbol{B}'], \tag{14}$$

$$f_T(\boldsymbol{w}) := 2\boldsymbol{d}^\top \boldsymbol{w} + \beta\|\boldsymbol{w}\|_2^2 + \mathbb{1}_{\mathbb{R}_+^{Tp}}(\boldsymbol{w}), \tag{15}$$

$$g_T(\boldsymbol{v}) := g_T^1(\boldsymbol{v}_1) + g_T^2(\boldsymbol{v}_2) \tag{16}$$

with $g_T^1(\boldsymbol{v}_1) := -\alpha \boldsymbol{1}^\top \log(\boldsymbol{v}_1)$ and $g_T^2(\boldsymbol{v}_2) := \gamma\|\boldsymbol{v}_2\|_1$. Note that Problem (13) always has an optimal solution (this follows from a simple coercivity argument). Also, note that when $T = 1$, the function g_1^2 is vacuous. In this case, we may simply write $f_1(\boldsymbol{w}) = 2\boldsymbol{d}^\top \boldsymbol{w} + \beta\|\boldsymbol{w}\|_2^2 + \mathbb{1}_{\mathbb{R}_+^p}(\boldsymbol{w})$ and $g_1(\boldsymbol{v}) = -\alpha \boldsymbol{1}^\top \log(\boldsymbol{v})$.

As will be elaborated in the next subsection, the structure of Problem (13) can be effectively exploited by ADMM-type methods. Before we proceed, let us remark that the reformulation techniques used above can be applied to handle other regularizers as well. For instance, if we consider the Tikhonov regularizer (3), then we can simply replace the ℓ_1-norm in g_T^2 with the squared ℓ_2-norm. If we consider the structured temporal variation regularizer (5), then we can replace the matrix $\boldsymbol{B}'$ in (12) with one whose identity blocks reflect the edge structure in the temporal graph $\mathcal{M}$. We also note that when $T = 1$ and $\gamma = 0$, we have $\boldsymbol{C} = \boldsymbol{S}$ and Problem (13) becomes

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^{p}, \boldsymbol{v} \in \mathbb{R}^{m}} \ & 2\boldsymbol{d}^\top \boldsymbol{w} + \beta\|\boldsymbol{w}\|_2^2 + \mathbb{1}_{\mathbb{R}_+^p}(\boldsymbol{w}) - \alpha \boldsymbol{1}^\top \log(\boldsymbol{v}) \\ \text{s.t.} \qquad & \boldsymbol{S}\boldsymbol{w} = \boldsymbol{v}, \end{aligned} \tag{17}$$

which is equivalent to the static graph learning model (6); cf. [18]. In a nutshell, Problem (13) provides a unified formulation of various static and time-varying graph learning models.

B. Algorithmic Development

In this subsection, we present our optimization framework pADMM-GL for solving the unified graph learning formulation (13). To begin, let $\boldsymbol{\lambda} \in \mathbb{R}^{T(m+p)}$ be the dual variable associated with the constraint $\boldsymbol{C}\boldsymbol{w} - \boldsymbol{v} = \boldsymbol{0}$ in Problem (13). The augmented Lagrangian function with penalty parameter $\rho > 0$ associated with Problem (13) is given by

$$\begin{aligned} & \mathcal{L}_\rho(\boldsymbol{w}, \boldsymbol{v}; \boldsymbol{\lambda}) \\ & := f_T(\boldsymbol{w}) + g_T(\boldsymbol{v}) - \boldsymbol{\lambda}^\top(\boldsymbol{C}\boldsymbol{w} - \boldsymbol{v}) + \frac{\rho}{2}\|\boldsymbol{C}\boldsymbol{w} - \boldsymbol{v}\|_2^2. \end{aligned}$$

In the k-th iteration (where $k \geq 0$), our pADMM-GL proceeds with the updates

$$\boldsymbol{w}^{k+1} = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}^{Tp}} \mathcal{L}_\rho\left(\boldsymbol{w}, \boldsymbol{v}^k, \boldsymbol{\lambda}^k\right) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}^k\|_{\boldsymbol{G}}^2, \tag{18a}$$

$$\boldsymbol{v}^{k+1} = \operatorname*{arg\,min}_{\boldsymbol{v}\in\mathbb{R}^{T(m+p)}} \mathcal{L}_\rho\left(\boldsymbol{w}^{k+1}, \boldsymbol{v}, \boldsymbol{\lambda}^k\right) + \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{v}^k\|_{\boldsymbol{H}}^2, \tag{18b}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \rho\left(\boldsymbol{C}\boldsymbol{w}^{k+1} - \boldsymbol{v}^{k+1}\right), \tag{18c}$$

where $\boldsymbol{G} \in \mathbb{R}^{Tp\times Tp}$ and $\boldsymbol{H} \in \mathbb{R}^{T(m+p)\times T(m+p)}$ can be chosen as any positive semidefinite matrices. Such choices ensure that the proximal terms $\|\boldsymbol{w} - \boldsymbol{w}^k\|_{\boldsymbol{G}}^2/2$ and $\|\boldsymbol{v} - \boldsymbol{v}^k\|_{\boldsymbol{H}}^2/2$ are non-negative.

Note that when $\boldsymbol{G} = \boldsymbol{0}$ and $\boldsymbol{H} = \boldsymbol{0}$, the proximal terms become vacuous and the updates (18a)–(18c) reduce to those of the classic ADMM (cADMM) (see, e.g., [25, Section 3.1]). However, the sub-problems of the cADMM do not always have simple closed-form solutions. Specifically, when $\boldsymbol{G} = \boldsymbol{0}$, the $\boldsymbol{w}$-update (18a) becomes

$$\boldsymbol{w}^{k+1} = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}_+^{Tp}} 2\boldsymbol{d}^\top\boldsymbol{w} + \beta\|\boldsymbol{w}\|_2^2 - \boldsymbol{\lambda}^\top\boldsymbol{C}\boldsymbol{w} + \frac{\rho}{2}\|\boldsymbol{C}\boldsymbol{w} - \boldsymbol{v}\|_2^2,$$

which is a non-negative least squares problem and is usually solved by iterative methods [26]. As we shall see, by adding the proximal terms with properly chosen $\boldsymbol{G}$ and $\boldsymbol{H}$, both sub-problems (18a) and (18b) admit closed-form solutions that are easy to compute.

Another important motivation for introducing non-vacuous proximal terms is to weaken the conditions for fast convergence. For cADMM, either f_T or g_T needs to be strongly convex and have Lipschitz continuous gradient in order to guarantee the linear convergence of the sequence $\{[\boldsymbol{w}^k; \boldsymbol{v}^k; \boldsymbol{\lambda}^k]\}_{k\geq 0}$ (see [23] for a summary). However, neither f_T nor g_T in our formulation (13) satisfies such a requirement: The function f_T does not have a Lipschitz continuous gradient due to the presence of the indicator, while the function g_T does not have a Lipschitz continuous gradient due to the logarithmic term. Nevertheless, we note that our updates (18a)–(18c) fit into the pADMM framework in [22], [27], where the linear convergence rate of the sequence $\{[\boldsymbol{w}^k; \boldsymbol{v}^k; \boldsymbol{\lambda}^k]\}_{k\geq 0}$ can be established with properly chosen $\boldsymbol{G}$ and $\boldsymbol{H}$ and under certain regularity conditions (these will be elaborated in Section III). This indicates that pADMM can better exploit the structure of (13) than cADMM in the presence of suitable regularity conditions.

To proceed, let $\boldsymbol{G} = \boldsymbol{I}/\tau_1 - \rho\boldsymbol{C}^\top\boldsymbol{C}$ with $0 < \tau_1 < 1/\rho\|\boldsymbol{C}\|_2^2$ so that $\boldsymbol{G}$ is positive definite. Then, we can write the update (18a) as

$$\begin{aligned}\boldsymbol{w}^{k+1} &= \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}^{Tp}} \Big\{ 2\boldsymbol{d}^\top\boldsymbol{w} + \beta\|\boldsymbol{w}\|_2^2 + \mathbb{1}_{\mathbb{R}_+^{Tp}}(\boldsymbol{w}) - \boldsymbol{\lambda}^\top\boldsymbol{C}\boldsymbol{w} \\ &\qquad + \frac{\rho}{2}\|\boldsymbol{C}\boldsymbol{w} - \boldsymbol{v}\|_2^2 + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}^k\|_{\boldsymbol{G}}^2 \Big\} \\ &= \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}^{Tp}} \Big\{ 2\boldsymbol{d}^\top\boldsymbol{w} + \beta\|\boldsymbol{w}\|_2^2 + \mathbb{1}_{\mathbb{R}_+^{Tp}}(\boldsymbol{w}) \\ &\qquad + \rho(\boldsymbol{w} - \boldsymbol{w}^k)^\top\boldsymbol{C}^\top\left(\boldsymbol{C}\boldsymbol{w}^k - \boldsymbol{v}^k - \frac{\boldsymbol{\lambda}^k}{\rho}\right) \\ &\qquad + \frac{1}{2\tau_1}\|\boldsymbol{w} - \boldsymbol{w}^k\|_2^2 \Big\} \\ &= \operatorname{prox}_{\tau_1 f_T}\left[\boldsymbol{w}^k - \tau_1\boldsymbol{C}^\top\left(\boldsymbol{C}\boldsymbol{w}^k - \boldsymbol{v}^k - \frac{\boldsymbol{\lambda}^k}{\rho}\right)\right]. \end{aligned} \tag{19}$$

It is worth noting that the update (19) coincides with that obtained by applying one proximal gradient step to $\mathcal{L}_\rho(\boldsymbol{w}, \boldsymbol{v}^k; \boldsymbol{\lambda}^k)$ at $\boldsymbol{w}^k$ with step size τ_1. Similarly, by letting $\boldsymbol{H} = (1/\tau_2 - \rho)\,\boldsymbol{I}$ with $0 < \tau_2 < 1/\rho$ so that $\boldsymbol{H}$ is positive definite, the update (18b) can be written as

$$\boldsymbol{v}^{k+1} = \operatorname{prox}_{\tau_2 g_T}\left[\boldsymbol{v}^k + \tau_2\rho\left(\boldsymbol{C}\boldsymbol{w}^{k+1} - \boldsymbol{v}^k - \frac{\boldsymbol{\lambda}^k}{\rho}\right)\right], \tag{20}$$

which coincides with that obtained by applying one proximal gradient step to $\mathcal{L}_\rho(\boldsymbol{w}^{k+1}, \boldsymbol{v}; \boldsymbol{\lambda}^k)$ at $\boldsymbol{v}^k$ with step size τ_2. The update formulas (19) and (20) involve the proximal mappings of f_T and g_T. As the following propositions show, both of them admit simple closed forms:

Proposition 1. *If $f_T(\boldsymbol{w}) = 2\boldsymbol{d}^\top\boldsymbol{w} + \beta\|\boldsymbol{w}\|_2^2 + \mathbb{1}_{\mathbb{R}_+^{Tp}}(\boldsymbol{w})$ for $\boldsymbol{w} \in \mathbb{R}^{Tp}$, then given any $\tau > 0$, we have*

$$\operatorname{prox}_{\tau f_T}(\boldsymbol{w}) = \max\left\{\frac{\boldsymbol{w} - 2\tau\boldsymbol{d}}{2\tau\beta + 1}, \boldsymbol{0}\right\}.$$

Proposition 2. *If $g_T(\boldsymbol{v}) = -\alpha\boldsymbol{1}^\top\log(\boldsymbol{v}_1) + \gamma\|\boldsymbol{v}_2\|_1$ for $\boldsymbol{v} = [\boldsymbol{v}_1; \boldsymbol{v}_2]$ with $\boldsymbol{v}_1 \in \mathbb{R}^{Tm}$ and $\boldsymbol{v}_2 \in \mathbb{R}^{Tp}$, then given any $\tau > 0$, we have*

$$\operatorname{prox}_{\tau g_T}(\boldsymbol{v}) = \begin{bmatrix} \frac{1}{2}\left(\boldsymbol{v}_1 + \sqrt{\boldsymbol{v}_1^2 + 4\alpha\tau\boldsymbol{1}}\right) \\ \mathcal{S}_{\tau,\gamma}(\boldsymbol{v}_2) \end{bmatrix},$$

where $\mathcal{S}_{\tau,\gamma}$ is the soft thresholding operator [28] given by

$$\left(\mathcal{S}_{\tau,\gamma}(\boldsymbol{v}_2)\right)_i = \begin{cases} 0, & |(\boldsymbol{v}_2)_i| \leq \tau\gamma, \\ \operatorname{sgn}((\boldsymbol{v}_2)_i)\left(|(\boldsymbol{v}_2)_i| - \tau\gamma\right), & |(\boldsymbol{v}_2)_i| > \tau\gamma \end{cases}$$

for $i = 1, \ldots, Tp$.

Proposition 1 is proved in Appendix A and Proposition 2 is adapted from [28, Sections 6.5.2 and 6.7.5]. Applying Propositions 1 and 2 to (19) and (20), respectively, we can rewrite the pADMM-GL updates (18a)–(18c) as

$$\boldsymbol{w}^{k+1} = \max\left\{\tilde{\boldsymbol{w}}^{k+1}, \boldsymbol{0}\right\}, \tag{21a}$$

$$\boldsymbol{v}^{k+1} = \begin{bmatrix} \frac{1}{2}\left(\tilde{\boldsymbol{v}}_1^{k+1} + \sqrt{\left(\tilde{\boldsymbol{v}}_1^{k+1}\right)^2 + 4\alpha\tau_2\boldsymbol{1}}\right) \\ \mathcal{S}_{\tau_2,\gamma}(\tilde{\boldsymbol{v}}_2^{k+1}) \end{bmatrix}, \tag{21b}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \rho\left(\boldsymbol{C}\boldsymbol{w}^{k+1} - \boldsymbol{v}^{k+1}\right), \tag{21c}$$

where

$$\tilde{\boldsymbol{w}}^{k+1} := \frac{\boldsymbol{w}^k - \tau_1\boldsymbol{C}^\top\left(\boldsymbol{C}\boldsymbol{w}^k - \boldsymbol{v}^k - \frac{\boldsymbol{\lambda}^k}{\rho}\right) - 2\tau_1\boldsymbol{d}}{2\tau_1\beta + 1},$$

$$\tilde{\boldsymbol{v}}^{k+1} := (1 - \tau_2\rho)\boldsymbol{v}^k + \tau_2\rho\boldsymbol{C}\boldsymbol{w}^{k+1} - \tau_2\boldsymbol{\lambda}^k.$$

In particular, by substituting $\boldsymbol{C} = \boldsymbol{S}$, the pADMM-GL updates (21a)–(21c) can be used to solve the static graph learning model (17).

The overall description of our pADMM-GL is given in Algorithm 1. The stopping criterion is that the primal residual $r_{\mathrm{p}} := \left\|\boldsymbol{C}\boldsymbol{w}^k - \boldsymbol{v}^k\right\|_2$ and the dual residual $r_{\mathrm{d}} =$

Algorithm 1 pADMM-GL for Problem (13)

```
Input: model parameters $\alpha, \beta > 0$, $T \geq 1$, and $\gamma \geq 0$; penalty parameter $\rho > 0$; step sizes $\tau_1, \tau_2 > 0$; tolerances $\varepsilon_{\mathrm{p}}, \varepsilon_{\mathrm{d}} > 0$;
Initialize: $k = 0$, randomly pick $\boldsymbol{w}^0 \in \mathbb{R}^{Tp}$, $\boldsymbol{v}^0 \in \mathbb{R}^{T(m+p)}$ and $\boldsymbol{\lambda}^0 \in \mathbb{R}^{T(m+p)}$, and pick sufficiently large $r_{\mathrm{p}}$, $r_{\mathrm{d}}$;
while $r_{\mathrm{p}} \geq \varepsilon_{\mathrm{p}}$ or $r_{\mathrm{d}} \geq \varepsilon_{\mathrm{d}}$ do
    update $\boldsymbol{w}$ according to (21a);
    update $\boldsymbol{v}$ according to (21b);
    update $\boldsymbol{\lambda}$ according to (21c);
    set primal residual $r_{\mathrm{p}} \leftarrow \left\|\boldsymbol{C}\boldsymbol{w}^k - \boldsymbol{v}^k\right\|_2$;
    set dual residual $r_{\mathrm{d}} \leftarrow \rho \left\|\boldsymbol{C}^\top \left(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\right)\right\|_2$;
    $k \leftarrow k + 1$;
end while
```

$\rho \left\|\boldsymbol{C}^\top \left(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\right)\right\|_2$ are less than the prescribed tolerances ε_{p} and ε_{d}, respectively. The per-iteration computational cost of our pADMM-GL is $\mathcal{O}(Tp)$, which is comparable to the state-of-the-art methods [9], [11], [19].

III. Convergence Analysis of pADMM-GL

Observe that (13) is a linearly constrained convex optimization problem with at least one optimal solution. Thus, its KKT conditions, which are given by

$$\begin{aligned} \boldsymbol{0} &\in \partial f_T(\boldsymbol{w}) - \boldsymbol{C}^\top \boldsymbol{\lambda}, \\ \boldsymbol{0} &\in \partial g_T(\boldsymbol{v}) + \boldsymbol{\lambda}, \\ \boldsymbol{0} &= \boldsymbol{C}\boldsymbol{w} - \boldsymbol{v}, \end{aligned} \tag{22}$$

are both necessary and sufficient for optimality. Let $\mathbb{O}^*$ denote the set of KKT points of Problem (13), i.e., $[\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}] \in \mathbb{O}^*$ if and only if $(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda})$ satisfies (22). Since pADMM-GL performs both primal and dual updates to generate the iterates $\{[\boldsymbol{w}^k; \boldsymbol{v}^k; \boldsymbol{\lambda}^k]\}_{k \geq 0}$, it is natural to ask whether these iterates will converge to a KKT point of Problem (13), and if so, at what rate. We shall address these questions in this section.

A. Global Convergence

As mentioned in Section II-B, our proposed pADMM-GL fits into the pADMM framework in [22], [27]. Using our choice of $\boldsymbol{G}$, $\boldsymbol{H}$ and adapting, e.g., the proof of [22, Theorem B.1], we immediately obtain the following global convergence result for pADMM-GL:

Theorem 1. *Suppose that the step sizes in Algorithm 1 satisfy $\tau_1 < 1/\rho\|\boldsymbol{C}\|_2^2$ and $\tau_2 < 1/\rho$. Then, the sequence $\{[\boldsymbol{w}^k; \boldsymbol{v}^k; \boldsymbol{\lambda}^k]\}_{k \geq 0}$ generated by Algorithm 1 converges to some point $[\boldsymbol{w}^*; \boldsymbol{v}^*; \boldsymbol{\lambda}^*] \in \mathbb{O}^*$.*

As Theorem 1 indicates, the largest singular value of $\boldsymbol{C}$ determines the maximum τ_1 that guarantees the global convergence of Algorithm 1. To guide the choice of step sizes, we have the following proposition, whose proof is deferred to Appendix B.

Proposition 3. *The largest singular values of $\boldsymbol{S}$ and $\boldsymbol{C}$ satisfy (1) $\|\boldsymbol{S}\|_2 = \sqrt{2(m-1)}$; (2) $\|\boldsymbol{C}\|_2 \leq \sqrt{2(m-1)} + 2$.*

Remark 1. *Theorem 1 guarantees the global convergence of Algorithm 1 with an arbitrary initial point given proper step sizes. In view of Proposition 3(1), we should let $\tau_1 < 1/(2\rho(m-1))$ and $\tau_2 < 1/\rho$ when solving the static graph learning model* (17)*. Moreover, Proposition 3(2) suggests that we should choose $\tau_1 < 1/\rho(\sqrt{2(m-1)} + 2)^2$ and $\tau_2 < 1/\rho$ when solving the time-varying graph learning model* (13) *with $T \geq 2$ and $\gamma > 0$.*

B. Local Linear Convergence

To determine the convergence rate of pADMM-GL when solving the graph learning formulation (13), one natural idea is to estimate how the distance between the iterate $[\boldsymbol{w}^k; \boldsymbol{v}^k; \boldsymbol{\lambda}^k]$ generated by the method and the set $\mathbb{O}^*$ of KKT points of Problem (13) changes as the iteration counter k increases. To implement this idea, we first use the definition of the proximal mapping given in Section I-D to rewrite the KKT conditions (22) as

$$\boldsymbol{w} - \mathrm{prox}_{f_T}(\boldsymbol{w} + \boldsymbol{C}^\top \boldsymbol{\lambda}) = \boldsymbol{0}, \tag{23a}$$

$$\boldsymbol{v} - \mathrm{prox}_{g_T}(\boldsymbol{v} - \boldsymbol{\lambda}) = \boldsymbol{0}, \tag{23b}$$

$$\boldsymbol{C}\boldsymbol{w} - \boldsymbol{v} = \boldsymbol{0}. \tag{23c}$$

Next, we define the (single-valued) proximal KKT mapping $\Pi_{\mathrm{KKT}}^{\mathrm{p}} : \mathbb{R}^\ell \to \mathbb{R}^\ell$, where $\ell := T(2m + 3p)$, associated with Problem (13) as

$$\Pi_{\mathrm{KKT}}^{\mathrm{p}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda}) := \begin{bmatrix} \boldsymbol{w} - \mathrm{prox}_{f_T}(\boldsymbol{w} + \boldsymbol{C}^\top \boldsymbol{\lambda}) \\ \boldsymbol{v} - \mathrm{prox}_{g_T}(\boldsymbol{v} - \boldsymbol{\lambda}) \\ \boldsymbol{C}\boldsymbol{w} - \boldsymbol{v} \end{bmatrix}, \tag{24}$$

where $\boldsymbol{w} \in \mathbb{R}^{Tp}$ and $\boldsymbol{v}, \boldsymbol{\lambda} \in \mathbb{R}^{T(m+p)}$.

The motivation for considering the proximal KKT mapping $\Pi_{\mathrm{KKT}}^{\mathrm{p}}$ is twofold. First, using (23) and (24), the set of KKT points of Problem (13) can be expressed as

$$\begin{aligned} \mathbb{O}^* &= (\Pi_{\mathrm{KKT}}^{\mathrm{p}})^{-1}(\boldsymbol{0}) = \left\{[\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}] \in \mathbb{R}^\ell \mid \Pi_{\mathrm{KKT}}^{\mathrm{p}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda}) = \boldsymbol{0}\right\} \\ &= \left\{ \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} \in \mathbb{R}^\ell \;\middle|\; \begin{array}{l} \boldsymbol{w} - \mathrm{prox}_{f_T}(\boldsymbol{w} + \boldsymbol{C}^\top \boldsymbol{\lambda}) = \boldsymbol{0}, \\ \boldsymbol{v}_1 - \mathrm{prox}_{g_T^1}(\boldsymbol{v}_1 - \boldsymbol{\lambda}_1) = \boldsymbol{0}, \\ \boldsymbol{v}_2 - \mathrm{prox}_{g_T^2}(\boldsymbol{v}_2 - \boldsymbol{\lambda}_2) = \boldsymbol{0}, \\ \boldsymbol{C}\boldsymbol{w} - [\boldsymbol{v}_1; \boldsymbol{v}_2] = \boldsymbol{0} \end{array} \right\}, \end{aligned} \tag{25}$$

where $\boldsymbol{v} = [\boldsymbol{v}_1; \boldsymbol{v}_2]$ and $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1; \boldsymbol{\lambda}_2]$ with $\boldsymbol{v}_1, \boldsymbol{\lambda}_1 \in \mathbb{R}^{Tm}$ and $\boldsymbol{v}_2, \boldsymbol{\lambda}_2 \in \mathbb{R}^{Tp}$. Second, as it turns out, for any $[\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}] \in \mathbb{R}^\ell$ that is sufficiently close to $\mathbb{O}^*$, the norm $\|\Pi_{\mathrm{KKT}}^{\mathrm{p}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda})\|_2$ can be used as a surrogate of the distance $\mathrm{dist}([\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}], \mathbb{O}^*)$. Specifically, we have the following result, which is the main technical contribution of this section:

Theorem 2. *The mapping $\Pi_{\mathrm{KKT}}^{\mathrm{p}}$ is* metrically subregular *at any KKT point of Problem* (13)*. Specifically, there exist a constant $\zeta > 0$ and a neighborhood $\mathcal{U} \subseteq \mathbb{R}^\ell$ with $\mathbb{O}^* \subseteq \mathcal{U}$ such that whenever $[\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}] \in \mathcal{U}$, we have*

$$\begin{aligned} \mathrm{dist}([\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}], \mathbb{O}^*) &= \mathrm{dist}\left([\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}], (\Pi_{\mathrm{KKT}}^{\mathrm{p}})^{-1}(\boldsymbol{0})\right) \\ &\leq \zeta \cdot \mathrm{dist}\left(\boldsymbol{0}, \Pi_{\mathrm{KKT}}^{\mathrm{p}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda})\right) = \zeta \|\Pi_{\mathrm{KKT}}^{\mathrm{p}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda})\|_2. \end{aligned} \tag{26}$$

The inequality (26) is commonly known in the optimization literature as an *error bound* for the set $\mathbb{O}^*$ with residual function

$(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda}) \mapsto \|\Pi^{\mathrm{p}}_{\mathrm{KKT}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda})\|_2$. It is well known that error bounds furnish a powerful tool for establishing convergence rates of various iterative methods; see, e.g., [27], [29]–[34] and the references therein. In particular, the following result, which shows that pADMM-GL enjoys a local linear convergence rate when applied to Problem (13), is a direct consequence of Theorem 1, Theorem 2, and [27, Theorem 2]:

Theorem 3. *Suppose that the step sizes in Algorithm 1 satisfy* $\tau_1 < 1/\rho\|\boldsymbol{C}\|_2^2$ *and* $\tau_2 < 1/\rho$. *Let* $\boldsymbol{M} := \mathrm{Diag}\left(\frac{1}{\tau_1}\boldsymbol{I}_{Tp} - \frac{3\rho}{4}\boldsymbol{C}^\top\boldsymbol{C}, \left(\frac{1}{\tau_2} + \frac{\rho}{4}\right)\boldsymbol{I}_{T(m+p)}, \frac{1}{\rho}\boldsymbol{I}_{T(m+p)}\right)$. *Then, there exist* $K > 0$ *and* $\mu \in (0, 1)$ *such that for all* $k > K$, *we have*

$$\begin{aligned}&\mathrm{dist}^2_{\boldsymbol{M}}([\boldsymbol{w}^{k+1}; \boldsymbol{v}^{k+1}; \boldsymbol{\lambda}^{k+1}], \mathbb{O}^*) + (1/\tau_2 - \rho)\|\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\|_2^2 \\ &\le \mu\left(\mathrm{dist}^2_{\boldsymbol{M}}([\boldsymbol{w}^k; \boldsymbol{v}^k; \boldsymbol{\lambda}^k], \mathbb{O}^*) + (1/\tau_2 - \rho)\|\boldsymbol{v}^k - \boldsymbol{v}^{k-1}\|_2^2\right).\end{aligned}$$

Theorem 3 shows that the *iterate sequence* generated by pADMM-GL converges to some point in $\mathbb{O}^*$ at the *linear* rate of $\mathcal{O}(\mu^k)$. It is worth noting that when compared with the corresponding result in the recent work [19], which shows that the *function value sequence* generated by FDPG converges at the *sublinear* rate of $\mathcal{O}(1/k^2)$, Theorem 3 guarantees a faster convergence rate (i.e., linear vs. sublinear) for a stronger mode of convergence (i.e., iterate sequence convergence vs. function value sequence convergence). In fact, to the best of our knowledge, Theorems 1 and 3 yield the best convergence guarantee known to date for solving the unified graph learning formulation (13).

To prove Theorem 2, we first recall that $g_T^1(\boldsymbol{v}_1) = -\alpha\boldsymbol{1}^\top\log(\boldsymbol{v}_1)$, which implies that g_T^1 is strictly convex on $\mathbb{R}^{Tm}_{++}$. This suggests that every point $[\boldsymbol{w}; \boldsymbol{v}_1; \boldsymbol{v}_2; \boldsymbol{\lambda}_1; \boldsymbol{\lambda}_2] \in \mathbb{O}^*$ has the same $\boldsymbol{v}_1$-component. Using this and (23b), we can obtain the following alternative characterization of $\mathbb{O}^*$:

Lemma 1. *There exists a vector* $\boldsymbol{v}_1^* \in \mathbb{R}^{Tm}_{++}$ *such that the set of KKT points of Problem* (13) *can be written as*

$$\mathbb{O}^* = \left\{ \begin{bmatrix}\boldsymbol{w}\\ \boldsymbol{v}_1\\ \boldsymbol{v}_2\\ \boldsymbol{\lambda}_1\\ \boldsymbol{\lambda}_2\end{bmatrix} \in \mathbb{R}^\ell \;\middle|\; \begin{array}{l}\boldsymbol{0} \in \partial f_T(\boldsymbol{w}) - \boldsymbol{C}^\top\boldsymbol{\lambda},\\ \boldsymbol{0} = \boldsymbol{\lambda}_1 - \alpha\boldsymbol{1}/\boldsymbol{v}_1^*\\ \boldsymbol{0} = \boldsymbol{v}_1 - \boldsymbol{v}_1^*,\\ \boldsymbol{0} \in \partial g_T^2(\boldsymbol{v}_2) + \boldsymbol{\lambda}_2,\\ \boldsymbol{0} = \boldsymbol{C}\boldsymbol{w} - [\boldsymbol{v}_1; \boldsymbol{v}_2]\end{array} \right\}. \tag{27}$$

The proof of Lemma 1 can be found in Appendix C.

Now, we define the set-valued mapping $\Gamma_{\mathrm{KKT}} : \mathbb{R}^\ell \rightrightarrows \mathbb{R}^{\ell+Tm}$ as

$$\Gamma_{\mathrm{KKT}}(\boldsymbol{w}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) := \begin{bmatrix}\partial f_T(\boldsymbol{w}) - \boldsymbol{C}^\top\boldsymbol{\lambda}\\ \boldsymbol{\lambda}_1 - \alpha\boldsymbol{1}/\boldsymbol{v}_1^*\\ \boldsymbol{v}_1 - \boldsymbol{v}_1^*\\ \partial g_T^2(\boldsymbol{v}_2) + \boldsymbol{\lambda}_2\\ \boldsymbol{C}\boldsymbol{w} - [\boldsymbol{v}_1; \boldsymbol{v}_2]\end{bmatrix}, \tag{28}$$

where $\boldsymbol{w} \in \mathbb{R}^{Tp}$; $\boldsymbol{v}_1, \boldsymbol{\lambda}_1 \in \mathbb{R}^{Tm}$; and $\boldsymbol{v}_2, \boldsymbol{\lambda}_2 \in \mathbb{R}^{Tp}$. In view of (27), we have

$$\mathbb{O}^* = (\Gamma_{\mathrm{KKT}})^{-1}(\boldsymbol{0}).$$

The advantage of considering the mapping Γ_{KKT} is made evident in the following result:

Lemma 2. *The set-valued mapping* Γ_{KKT} *is piecewise polyhedral.*[2] *Consequently, it is* metrically subregular *at any KKT point of Problem* (13). *Specifically, there exist constants* $\epsilon, \eta > 0$ *such that whenever* $[\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}] \in \mathbb{R}^\ell$ *satisfies* $\mathrm{dist}(\boldsymbol{0}, \Gamma_{\mathrm{KKT}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda})) \le \epsilon$, *we have*

$$\begin{aligned}\mathrm{dist}([\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}], \mathbb{O}^*) &= \mathrm{dist}\left([\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}], (\Gamma_{\mathrm{KKT}})^{-1}(\boldsymbol{0})\right)\\ &\le \eta \cdot \mathrm{dist}\left(\boldsymbol{0}, \Gamma_{\mathrm{KKT}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda})\right).\end{aligned} \tag{29}$$

The proof of Lemma 2 can be found in Appendix D.

In view of Lemma 2, we can complete the proof of Theorem 2 by establishing the following link between the metric subregularity properties of $\Pi^{\mathrm{p}}_{\mathrm{KKT}}$ and Γ_{KKT}:

Proposition 4. *If* Γ_{KKT} *is metrically subregular at any KKT point of Problem* (13) *(i.e., the error bound* (29) *holds for some constants* $\epsilon, \eta > 0$*), then so is* $\Pi^{\mathrm{p}}_{\mathrm{KKT}}$ *(i.e., the error bound* (26) *holds for some constant* $\zeta > 0$ *and neighborhood* $\mathcal{U} \subseteq \mathbb{R}^\ell$ *with* $\mathbb{O}^* \subseteq \mathcal{U}$*).*

The proof of Proposition 4 can be found in Appendix E.

To summarize, we have shown that when applied to the graph learning formulation (13), the iterates generated by our proposed pADMM-GL will converge to a KKT point of the formulation at a local linear rate. It is worth pointing out that our convergence results remain valid if we replace the $L_{1,1}$-norm regularizer (4) in Problem (13) by either the Tikhonov regularizer (3) or the structured temporal variation regularizer (5). Indeed, the former results in a convex quadratic g_T^2, while the latter yields a piecewise linear g_T^2 (see (16)). As such, the arguments used to prove Theorems 1 and 2 still apply. To the best of our knowledge, our work is the first to develop a first-order method that applies to the large class of static and time-varying graph learning formulations encapsulated in (2) and comes with a linear convergence guarantee.

IV. Numerical Experiments

A. Static Graph Learning

In this subsection, we present the numerical results of our pADMM-GL, the primal–dual method [9], the FDPG method [19], and the MM method [21] when solving the static graph learning model (6). All algorithms are implemented in MATLAB.[3] To test the primal–dual method, we use the code provided in the *Graph Signal Processing toolbox*[4] [36] and incorporate the scaling trick given in [15, Proposition 1] to accelerate the convergence. To test FDPG, we use the code provided by [19].[5] To test the MM method, we use the code provided by [21][6] but remove the data normalization step to ensure that the implementation is consistent with the other algorithms. The parameters α and β in Problem (6) are best-tuned so that the learned graphs have the highest quality in terms of the F-measure [8], [37]. Moreover, the parameters

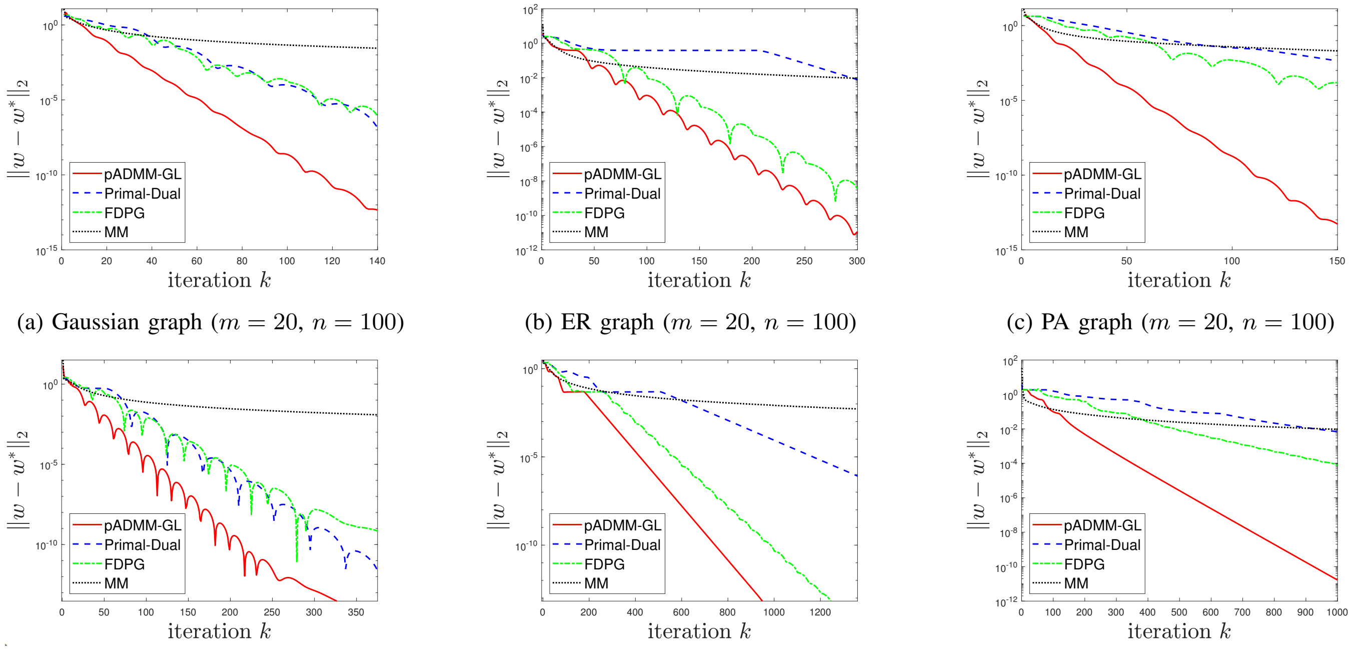
[2] A set-valued mapping Φ is called *piecewise polyhedral* if $\mathrm{gph}(\Phi)$ can be expressed as the union of finitely many polyhedral sets; see [35, Example 9.57].

[3] Our code is available at https://github.com/xwangcu/padmm-gl.

[4] https://epfl-lts2.github.io/gspbox-html/doc/demos/gsp_demo_learn_graph_large.html

[5] http://www.ece.rochester.edu/~gmateosb/code/FDPG.zip

[6] https://github.com/ghaniafatima/GLMM



(a) Gaussian graph ($m = 20$, $n = 100$)

(b) ER graph ($m = 20$, $n = 100$)

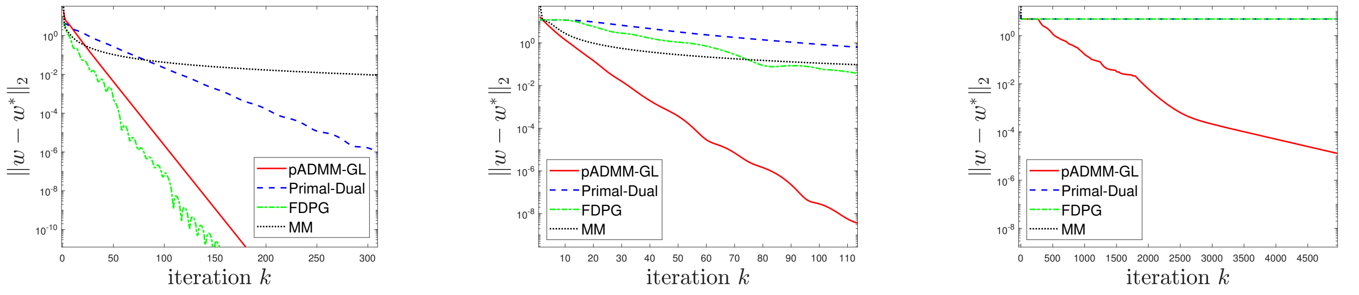
(c) PA graph ($m = 20$, $n = 100$)

(d) Gaussian graph ($m = 50$, $n = 400$)

(e) ER graph ($m = 50$, $n = 400$)

(f) PA graph ($m = 50$, $n = 400$)

Fig. 1: Convergence performance of algorithms for static graph learning on synthetic graphs



(a) mesh1e1 graph ($m = 48$, $n = 100$)

(b) bcspwr graph ($m = 118$, $n = 100$)

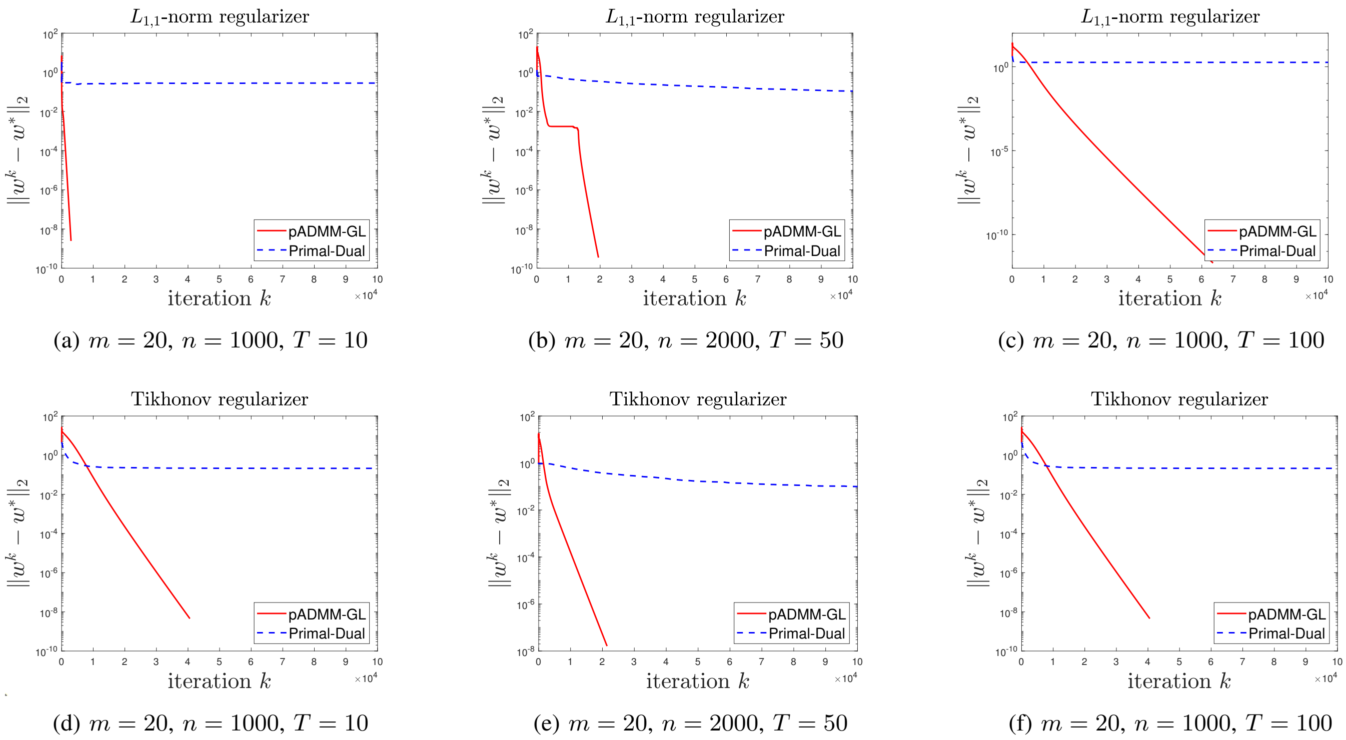
(c) lshp graph ($m = 1561$, $n = 2000$)

Fig. 2: Convergence performance of algorithms for static graph learning on real-world graphs

ρ, τ_1, τ_2 in pADMM-GL and the step sizes in the primal–dual method are also best-tuned to achieve sharp convergence rates. We generate the graph signals according to the factor analysis model proposed in [8]. Specifically, suppose that the Laplacian matrix of the ground-truth graph is $\boldsymbol{L} = \mathrm{Diag}(\boldsymbol{W}\mathbf{1}) - \boldsymbol{W}$ and admits the eigen-decomposition $\boldsymbol{L} = \boldsymbol{\chi}\boldsymbol{\Lambda}\boldsymbol{\chi}^\top$. Then, the smooth graph signal is generated as $\boldsymbol{x} = \boldsymbol{\chi}\boldsymbol{h} + \boldsymbol{\delta}$, where $\boldsymbol{h} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}^\dagger\right)$ is the latent variable that follows the Gaussian distribution with mean equal to $\mathbf{0}$ and covariance equal to the Moore-Penrose inverse $\boldsymbol{\Lambda}^\dagger$ of $\boldsymbol{\Lambda}$ and $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \varepsilon\boldsymbol{I})$ is the Gaussian noise with noise level ε.

1) Synthetic Graphs: We first carry out experiments on three types of synthetic graphs, namely, the Gaussian graph, the Erdős-Rényi (ER) graph, and the preferential attachment (PA) graph. The Gaussian graph is generated as follows: First, the nodes are placed uniformly at random in a unit square. Then, an edge is placed between nodes i and j if the weight determined by the radial basis function $\exp(-d(i,j)^2/2\xi^2)$, where $d(i,j)$ is the Euclidean distance between nodes i and j and $\xi = 0.5$ is the kernel width parameter, is at least 0.75. The ER graph is generated by placing an edge between each pair of nodes independently with probability 0.2. The PA graph is generated by having 2 connected nodes initially and then adding new nodes one at a time, where each new node is connected to exactly 1 previous node that is randomly chosen with a probability that is proportional to its degree at the time. The edges in the Gaussian graph have weights given by the radial basis function, while those in the ER and PA graphs are set to 1. We generate different sets of graph signals with the same noise level $\varepsilon = 0.5$.

Since the algorithms we consider have the same $\mathcal{O}(m^2)$ per-iteration computational cost with m being the dimension of graph signals, we evaluate their performance through the suboptimality gap $\|\boldsymbol{w}^k - \boldsymbol{w}^*\|_2$ for different values of m and n (number of graph signals). The results are shown in Fig. 1, from which we observe that pADMM-GL always exhibits substantially sharper convergence rates than the other three algorithms. In some cases, e.g., Fig. 1c, the primal–dual



(a) $m = 20$, $n = 1000$, $T = 10$

(b) $m = 20$, $n = 2000$, $T = 50$

(c) $m = 20$, $n = 1000$, $T = 100$

(d) $m = 20$, $n = 1000$, $T = 10$

(e) $m = 20$, $n = 2000$, $T = 50$

(f) $m = 20$, $n = 1000$, $T = 100$

Fig. 3: Convergence performance of algorithms for time-varying graph learning on synthetic graphs

method and FDPG converge rather slowly, while pADMM-GL still performs quite well. It is reported in [21] that the MM method is faster than other algorithms (including the preliminary version of our pADMM-GL [18]) in the very initial stage. Our experiments show that the suboptimality gap of the MM method decreases fast in the first few iterations, which confirms the empirical observation in [21]. However, the MM method appears to converge only at a sublinear rate and does not attain high-precision solutions in all the experiments.

2) Real-World Graphs: We also test the numerical performance of the algorithms on several real-world graphs from the *SuiteSparse Matrix Collection*[7] [38]. In particular, we select the mesh1e1 network with $m = 48$, the bcspwr power network with $m = 118$, and the lshp thermal network with $m = 1561$. The numerical results are shown in Fig. 2. For the mesh1e1 network, pADMM-GL is comparable with FDPG, while they are both faster than the primal–dual method. For the bcspwr and lshp graphs, pADMM-GL converges much faster than the primal–dual method and FDPG. Even for the relatively large lshp graph, pADMM-GL can still achieve 10^{-5} precision, while the primal–dual method and FDPG can hardly obtain a desirable suboptimal solution. The MM method even fails to converge to an acceptable solution for this large graph.

	m	CVX	primal–dual	FDPG	pADMM-GL
Gaussian	20	1.9	0.026	0.0023	**0.0017**
	50	13.0	0.038	0.0084	**0.0072**
ER	20	3.6	0.077	0.0031	**0.0026**
	50	12.4	0.278	0.0215	**0.0209**
PA	20	2.0	0.056	0.0026	**0.0012**
	50	12.0	0.634	0.0483	**0.0239**

(a) Runtime (in seconds) on synthetic graphs

	m	CVX	primal–dual	FDPG	pADMM-GL
mesh1e1	48	10.76	0.20	**0.005**	0.006
bcspwr	118	635.9	0.37	0.07	**0.03**

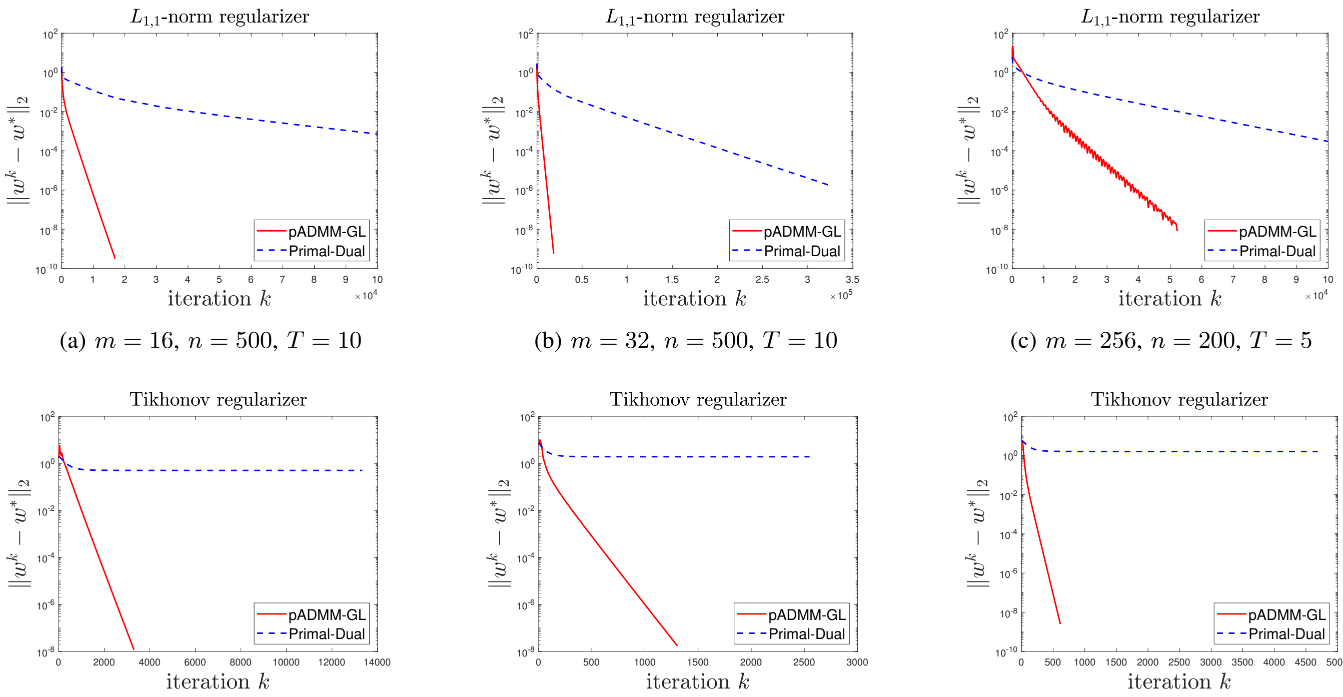
(b) Runtime (in seconds) on real-world graphs

TABLE I: Runtime of algorithms for static graph learning

B. Runtime Comparison

Next, we compare the CPU runtime of pADMM-GL, FDPG, and the primal–dual method by stopping them when the suboptimality gap $\|\boldsymbol{w}^k - \boldsymbol{w}^*\|_2$ is less than 10^{-8}. We do not report the CPU runtime of the MM method, since it converges rather slowly and takes considerably more time than other algorithms to achieve the required precision. Since Problem (6) is convex, we solve it using the convex optimization package CVX [39] with SDPT3 as the solver and the precision set to highest. The runtime of CVX is provided as a baseline. Table I(a) reports the average runtime over 10 independent runs under the same experiment settings as those for Fig. 1. It can be seen that our pADMM-GL consumes moderately less runtime than FDPG for Gaussian and ER graphs and requires less than half the runtime of FDPG for PA graphs. In all cases, pADMM-GL consumes substantially less time than the

[7] https://sparse.tamu.edu/



(a) $m = 16$, $n = 500$, $T = 10$

(b) $m = 32$, $n = 500$, $T = 10$

(c) $m = 256$, $n = 200$, $T = 5$

(d) $m = 16$, $n = 500$, $T = 10$

(e) $m = 32$, $n = 500$, $T = 50$

(f) $m = 256$, $n = 500$, $T = 5$

Fig. 4: Convergence performance of algorithms for time-varying graph learning on point cloud data

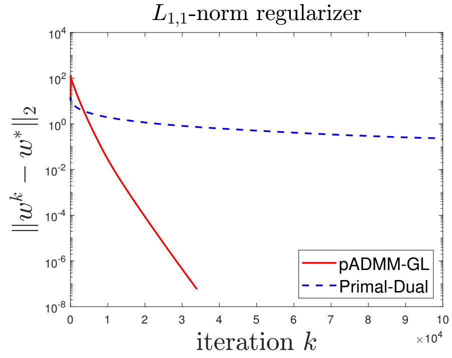


Fig. 5: Convergence performance of algorithms for time-varying graph learning on Ireland temperature data ($m = 25$, $n = 38712$, $T = 1613$)

primal–dual method. Table I(b) reports the average runtime over 10 independent runs under the same experiment settings as those for Fig. 2. For the medium-sized mesh1e1 and bcspwr graphs, pADMM-GL and MM exhibit comparable runtime, and they both consume significantly less runtime than the primal–dual method and CVX. We do not provide the runtime comparison for the largest lshp graph, as all the compared methods except our pADMM-GL converge exceedingly slowly (as shown in Fig. 2(c)).

C. Time-Varying Graph Learning

In this subsection, we present the numerical results of our pADMM-GL and the primal–dual methods when solving the time-varying graph learning model (2) with $T \geq 2$ and the temporal regularizer being either the Tikhonov regularizer (3) or the $L_{1,1}$-norm regularizer (4). When $\mathcal{R}_T$ is given by (3), the vectorized reformulation of (2) is the same as (13) except that $g_T^2(\boldsymbol{v}_2) = \gamma\|\boldsymbol{v}_2\|_2^2$. Observe that for any $\boldsymbol{v}_2 \in \mathbb{R}^{Tp}$ and $\tau_2 > 0$, we have

$$\begin{aligned}\operatorname{prox}_{\tau_2 g_T^2}(\boldsymbol{v}_2) &= \arg\min_{\boldsymbol{u}\in\mathbb{R}^{Tp}} \left\{ \tau_2\gamma\|\boldsymbol{u}\|_2^2 + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}_2\|_2^2 \right\} \\ &= \boldsymbol{v}_2/(1+2\tau_2\gamma).\end{aligned}$$

Thus, in order to apply Algorithm 1 to solve Problem (2) with the Tikhonov regularizer, it suffices to modify the update formula (21b) as

$$\boldsymbol{v}^{k+1} = \begin{bmatrix} \frac{1}{2}\left(\tilde{\boldsymbol{v}}_1^{k+1} + \sqrt{\left(\tilde{\boldsymbol{v}}_1^{k+1}\right)^2 + 4\alpha\tau_2\mathbf{1}}\right) \\ \tilde{\boldsymbol{v}}_2^{k+1}/(1+2\tau_2\gamma) \end{bmatrix}.$$

We use the primal–dual methods in [12] and [11] to solve the Tikhonov and $L_{1,1}$-norm regularized time-varying graph learning models, respectively. In each experiment, we fix a set of α, β, and γ in Problem (2) and best-tune the algorithmic parameters ρ, τ_1, τ_2 in pADMM-GL and the step sizes in the primal–dual methods for fast convergence. The algorithms are also implemented in MATLAB.

1) Synthetic Graphs: We follow the approach in [16] to generate synthetic time-varying graphs and the associated synthetic graph signals. Suppose that there are T time slots. We generate T static graphs as follows: First, we construct an initial static ER graph in $\boldsymbol{W}^{(1)}$ with m nodes, where the connecting probability of each edge is fixed to be $p = 0.05$. The edge weights in $\boldsymbol{W}^{(1)}$ are uniformly distributed within the interval $[0, 1]$. Then, for $t = 2, \ldots, T$, we construct the graph in the t-th time slot $\boldsymbol{W}^{(t)}$ by randomly resampling 5% of the edges of the previous graph $\boldsymbol{W}^{(t-1)}$. Due to this rule, most edges and their corresponding weights remain unchanged within a short period. In each experiment, we generate n graph signals in total. In the t-th time slot, we independently generate n/T graph signals that reside on the t-th graph $\boldsymbol{W}^{(t)}$ in the same way as in Section IV-A.

Since both pADMM-GL and the primal–dual methods have the same $\mathcal{O}(Tm^2)$ per-iteration computational cost, we evaluate their performance through the suboptimality gap $\|\boldsymbol{w}^k - \boldsymbol{w}^*\|_2$ for different values of m, n, and T. The results are shown in Fig. 3. It can be observed that pADMM-GL always exhibits substantially sharper convergence rates than the primal–dual methods. In contrast to the efficient pADMM-GL, the primal–dual methods can hardly converge to a satisfactory precision in the considered cases.

2) Point Cloud Data: Analogous to [12] and [16], we apply the time-varying graph learning model (2) to tackle the time-varying point cloud denoising problem. A key step in the problem is to learn the time-varying Laplacian matrices $\boldsymbol{L}^{(1)}, \dots, \boldsymbol{L}^{(T)}$ from a noisy point cloud dataset $\boldsymbol{X} = [\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(T)}]$. In our experiments, we use the motion-capture point cloud data[8] [40] as the noiseless graph signals. In particular, we evolve the "*dancer*" point cloud data over n frames. For each frame, we down-sample the point cloud to m points, which form an m-dimensional graph signal. We then add Gaussian noise to the noiseless data with a signal-to-noise ratio of 2dB to generate the noisy data $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathbb{R}^{m\times n}$. To enhance the numerical performance, we normalize the data matrix $\boldsymbol{X}$ by

$$\boldsymbol{x}_i \leftarrow \frac{\boldsymbol{x}_i - \min_{k\in[m]}\{(\boldsymbol{x}_i)_k\}}{\max_{k\in[m]}\{(\boldsymbol{x}_i)_k\} - \min_{k\in[m]}\{(\boldsymbol{x}_i)_k\}}$$

for all $i \in [n]$. We divide the n frames into T time slots, so that each time slot contains n/T graph signals. We use both our pADMM-GL and the primal–dual methods to solve Problem (2) to learn a sequence of graphs from $\boldsymbol{X}$.

The convergence performance in terms of the suboptimality gap $\|\boldsymbol{w}^k - \boldsymbol{w}^*\|_2$ for different choices of m, n, and T are presented in Fig. 4. The numerical results on the real point cloud data indicate that pADMM-GL still exhibits sharp linear convergence and is always much faster than the primal–dual methods.

3) Temperature Data: Analogous to [41], we consider the realistic weather data provided by the Irish Meteorological Service. The dataset contains $n = 38712$ hourly temperature measurements (in Celsius) from January 2016 to May 2020, which are acquired by 25 stations across Ireland. Each temperature measurement can be regarded as a 25-dimensional graph signal, and each time slot consists of 24 graph signals. Then, we apply pADMM-GL and the primal–dual method in [11] to solve the $L_{1,1}$-norm regularized time-varying graph learning model (7) with $T = 1613$. The convergence performance is presented in Fig. 5, which again shows the superiority of our proposed method.

V. Conclusion

In this paper, we have developed an efficient and flexible optimization method for solving a unified formulation of various static and time-varying graph learning tasks. We have shown that the iterates generated by our method will converge linearly to an optimal solution to the formulation. This was achieved by showing that the set of KKT points of the formulation possesses an error bound property, which can be of independent interest. Furthermore, we have shown via extensive numerical experiments on both synthetic and real data that the convergence performance and computation time of our proposed method outperform those of other state-of-the-art methods. A natural future direction is to extend our algorithmic framework to tackle other emerging graph learning models.

Appendix

A. *Proof of Proposition 1*

Note that $f_T(\boldsymbol{w}) = \sum_{i=1}^{Tp} f^{(i)}(w_i)$, where $f^{(i)}(w_i) := 2d_iw_i + \beta w_i^2 + \mathbb{1}_{\mathbb{R}_+}(w_i)$, and

$$\begin{aligned}
\operatorname{prox}_{\tau f^{(i)}}(w_i) &= \operatorname*{arg\,min}_{u_i\in\mathbb{R}} \left\{ f^{(i)}(u_i) + \frac{1}{2\tau}(u_i - w_i)^2 \right\} \\
&= \operatorname*{arg\,min}_{u_i\in\mathbb{R}} \left\{ 2d_iu_i + \beta u_i^2 + \mathbb{1}_{\mathbb{R}_+}(u_i) + \frac{1}{2\tau}(u_i - w_i)^2 \right\} \\
&= \max\left\{ \frac{w_i - 2\tau d_i}{2\tau\beta + 1}, 0 \right\}
\end{aligned}$$

for $i = 1, \dots, Tp$. The desired result follows immediately.

B. *Proof of Proposition 3*

It is shown in [42, Lemma 1] that $\|\boldsymbol{S}\|_2 = \sqrt{2(m-1)}$. Using (11), we have

$$\|\boldsymbol{B}\|_2 = \|\boldsymbol{I}_T \otimes \boldsymbol{S}\|_2 = \sqrt{2(m-1)}. \tag{30}$$

To bound $\|\boldsymbol{B}'\|_2$, we use (12) to write

$$\boldsymbol{B}'\boldsymbol{B}'^{\top} = \begin{bmatrix} \boldsymbol{0}_p & & & & & \\ & 2\boldsymbol{I}_p & -\boldsymbol{I}_p & & & \\ & -\boldsymbol{I}_p & 2\boldsymbol{I}_p & -\boldsymbol{I}_p & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\boldsymbol{I}_p & 2\boldsymbol{I}_p & -\boldsymbol{I}_p \\ & & & & -\boldsymbol{I}_p & 2\boldsymbol{I}_p \end{bmatrix}.$$

By the Gershgorin circle theorem (see, e.g., [43, Theorem 7.2.1]), the region

$$\mathbb{G} := \{\lambda \in \mathbb{R} \mid \lambda = 0 \text{ or } |\lambda - 2| \le 1 \text{ or } |\lambda - 2| \le 2\}$$

contains all the eigenvalues of $\boldsymbol{B}'\boldsymbol{B}'^{\top}$. It follows that

$$\|\boldsymbol{B}'\|_2 \le \sqrt{\max_{\lambda\in\mathbb{G}} \lambda} = 2. \tag{31}$$

Combining (14), (30), and (31) gives

$$\|\boldsymbol{C}\|_2 \le \|\boldsymbol{B}\|_2 + \|\boldsymbol{B}'\|_2 \le \sqrt{2(m-1)} + 2.$$

[8] http://pages.iai.uni-bonn.de/gall_juergen/projects/skelsurf/skelsurf.html

C. Proof of Lemma 1

Let $[\boldsymbol{w}'; \boldsymbol{v}_1'; \boldsymbol{v}_2'; \boldsymbol{\lambda}_1', \boldsymbol{\lambda}_2'], [\boldsymbol{w}''; \boldsymbol{v}_1''; \boldsymbol{v}_2''; \boldsymbol{\lambda}_1''; \boldsymbol{\lambda}_2''] \in \mathbb{O}^*$ be two KKT points of Problem (13). Then, both $(\boldsymbol{w}', [\boldsymbol{v}_1'; \boldsymbol{v}_2'])$ and $(\boldsymbol{w}'', [\boldsymbol{v}_1''; \boldsymbol{v}_2''])$ are optimal solutions to Problem (13) with optimal value, say, θ. We claim that $\boldsymbol{v}_1' = \boldsymbol{v}_1''$. Suppose that this is not the case. Since $\nabla^2 g_T^1(\boldsymbol{v}_1) = \mathrm{Diag}(\alpha \mathbf{1}/\boldsymbol{v}_1^2)$ is positive definite for any $\boldsymbol{v}_1 \in \mathbb{R}^{Tm}_{++}$, we see that g_T^1 is strictly convex on $\mathbb{R}^{Tm}_{++}$. Now, set

$$\bar{\boldsymbol{w}} = \frac{\boldsymbol{w}' + \boldsymbol{w}''}{2}, \quad \bar{\boldsymbol{v}}_1 = \frac{\boldsymbol{v}_1' + \boldsymbol{v}_1''}{2}, \quad \bar{\boldsymbol{v}}_2 = \frac{\boldsymbol{v}_2' + \boldsymbol{v}_2''}{2}.$$

Note that $(\bar{\boldsymbol{w}}, [\bar{\boldsymbol{v}}_1; \bar{\boldsymbol{v}}_2])$ is feasible for Problem (13). Moreover, since $\boldsymbol{v}_1' \neq \boldsymbol{v}_1''$ and g_T^1 is strictly convex, we have $g_T^1(\bar{\boldsymbol{v}}_1) < (g_T^1(\boldsymbol{v}_1') + g_T^1(\boldsymbol{v}_1''))/2$. It follows that

$$\begin{aligned} \theta &\leq f_T(\bar{\boldsymbol{w}}) + g_T^1(\bar{\boldsymbol{v}}_1) + g_T^2(\bar{\boldsymbol{v}}_2) \\ &< \frac{1}{2}\big(f_T(\boldsymbol{w}') + f_T(\boldsymbol{w}'') + g_T^1(\boldsymbol{v}_1') + g_T^1(\boldsymbol{v}_1'') \\ &\qquad + g_T^2(\boldsymbol{v}_2') + g_T^2(\boldsymbol{v}_2'')\big) \\ &= \theta, \end{aligned}$$

which is a contradiction. Thus, there exists a vector $\boldsymbol{v}_1^* \in \mathbb{R}^{Tm}_{++}$ such that for any $[\boldsymbol{w}; \boldsymbol{v}_1; \boldsymbol{v}_2; \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2] \in \mathbb{O}^*$, we have $\boldsymbol{v}_1 = \boldsymbol{v}_1^*$.

Using Proposition 2, we deduce that the equation $\boldsymbol{v}_1 - \mathrm{prox}_{g_T^1}(\boldsymbol{v}_1 - \boldsymbol{\lambda}_1) = \mathbf{0}$ in (25) is equivalent to

$$\boldsymbol{v}_1 - \frac{1}{2}(\boldsymbol{v}_1 - \boldsymbol{\lambda}_1 + \sqrt{(\boldsymbol{v}_1 - \boldsymbol{\lambda}_1)^2 + 4\alpha\mathbf{1}}) = \mathbf{0}.$$

Further simplifying gives $\boldsymbol{\lambda}_1 - \alpha\mathbf{1}/\boldsymbol{v}_1 = \mathbf{0}$. This, together with the equation $\boldsymbol{v}_1 - \boldsymbol{v}_1^* = \mathbf{0}$ we obtained in the preceding paragraph, yields the desired characterization of $\mathbb{O}^*$.

D. Proof of Lemma 2

In view of the definition of Γ_{KKT} in (28), we define the mappings $\Omega_i : \mathbb{R}^\ell \rightrightarrows \mathbb{R}^{\ell+Tm}$ for $i = 1, \ldots, 5$ as follows:

$$\begin{aligned} \Omega_1(\boldsymbol{w}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) &:= \partial f_T(\boldsymbol{w}) - \boldsymbol{C}^\top \boldsymbol{\lambda}, \\ \Omega_2(\boldsymbol{w}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) &:= \boldsymbol{\lambda}_1 - \alpha\mathbf{1}/\boldsymbol{v}_1^*, \\ \Omega_3(\boldsymbol{w}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) &:= \boldsymbol{v}_1 - \boldsymbol{v}_1^*, \\ \Omega_4(\boldsymbol{w}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) &:= \partial g_T^2(\boldsymbol{v}_2) + \boldsymbol{\lambda}_2, \\ \Omega_5(\boldsymbol{w}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) &:= \boldsymbol{C}\boldsymbol{w} - [\boldsymbol{v}_1; \boldsymbol{v}_2]. \end{aligned}$$

Then, we can express the graph of Γ_{KKT} as

$$\begin{aligned} \mathrm{gph}(\Gamma_{\mathrm{KKT}}) &= \{(\boldsymbol{u}, \boldsymbol{p}) \mid \boldsymbol{p} \in \Gamma_{\mathrm{KKT}}(\boldsymbol{u})\} \\ &= \{(\boldsymbol{u}, \boldsymbol{p}) \mid \boldsymbol{p}_i \in \Omega_i(\boldsymbol{u}) \text{ for } i = 1, \ldots, 5\} \\ &= \{(\boldsymbol{u}, \boldsymbol{p}) \mid (\boldsymbol{u}, \boldsymbol{p}_i) \in \mathrm{gph}(\Omega_i) \text{ for } i = 1, \ldots, 5\}, \end{aligned} \tag{32}$$

where $\boldsymbol{u} = [\boldsymbol{w}; \boldsymbol{v}_1; \boldsymbol{v}_2; \boldsymbol{\lambda}_1; \boldsymbol{\lambda}_2] \in \mathbb{R}^\ell$ and $\boldsymbol{p} = [\boldsymbol{p}_1; \boldsymbol{p}_2; \boldsymbol{p}_3; \boldsymbol{p}_4; \boldsymbol{p}_5] \in \mathbb{R}^{\ell+Tm}$.

Now, observe that since both f_T and g_T^2 are piecewise linear-quadratic functions,[9] the mappings ∂f_T and ∂g_T^2 are piecewise polyhedral [35, Proposition 12.30]. Consequently, the graphs $\mathrm{gph}(\Omega_1)$ and $\mathrm{gph}(\Omega_4)$ can be expressed as unions of finitely many polyhedral sets. It is obvious that the graphs $\mathrm{gph}(\Omega_2)$, $\mathrm{gph}(\Omega_3)$, and $\mathrm{gph}(\Omega_5)$ are polyhedral. It follows from (32) that $\mathrm{gph}(\Gamma_{\mathrm{KKT}})$ can be expressed as the union of finitely many polyhedral sets, which implies that Γ_{KKT} is piecewise polyhedral.

To complete the proof, we note that the piecewise polyhedrality of Γ_{KKT} implies that $\mathrm{gph}((\Gamma_{\mathrm{KKT}})^{-1})$ can be expressed as the union of finitely many polyhedral sets [35, page 489]. It then follows from [44, Theorem 3D.1 and Exercise 3D.7] that Γ_{KKT} is metrically subregular at any KKT point of Problem (13).

E. Proof of Proposition 4

The proof consists of two steps. First, we consider the set-valued mapping $\Pi_{\mathrm{KKT}} : \mathbb{R}^\ell \rightrightarrows \mathbb{R}^\ell$, which is defined as

$$\Pi_{\mathrm{KKT}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda}) := \begin{bmatrix} \partial f_T(\boldsymbol{w}) - \boldsymbol{C}^\top \boldsymbol{\lambda} \\ \partial g_T(\boldsymbol{v}) + \boldsymbol{\lambda} \\ \boldsymbol{C}\boldsymbol{w} - \boldsymbol{v} \end{bmatrix} \tag{33}$$

for $\boldsymbol{w} \in \mathbb{R}^{Tp}$ and $\boldsymbol{v}, \boldsymbol{\lambda} \in \mathbb{R}^{T(m+p)}$, and establish a link between the metric subregularity properties of Γ_{KKT} and Π_{KKT}. Then, based on this link, we establish the metric subregularity of $\Pi_{\mathrm{KKT}}^{\mathrm{p}}$.

1) Step 1: Suppose that Γ_{KKT} is metrically subregular at any KKT point of Problem (13), i.e., the error bound (29) holds for some constants $\epsilon, \eta > 0$. Consider the neighborhood

$$\mathcal{U} = \left\{[\boldsymbol{w}'; \boldsymbol{v}'; \boldsymbol{\lambda}'] \in \mathbb{R}^\ell \mid \mathrm{dist}\left(\mathbf{0}, \Gamma_{\mathrm{KKT}}(\boldsymbol{w}', \boldsymbol{v}', \boldsymbol{\lambda}')\right) \leq \epsilon\right\}.$$

It is clear that $\mathbb{O}^* \subseteq \mathcal{U}$, as $\mathbf{0} \in \Gamma_{\mathrm{KKT}}(\boldsymbol{w}', \boldsymbol{v}', \boldsymbol{\lambda}')$ for any $[\boldsymbol{w}'; \boldsymbol{v}'; \boldsymbol{\lambda}'] \in \mathbb{O}^*$. Let $[\boldsymbol{w}; \boldsymbol{v}; \boldsymbol{\lambda}] \in \mathcal{U}$ be arbitrary. To prepare for our subsequent development, let us collect some basic facts.

Let

$$[\boldsymbol{a}; \boldsymbol{b}; \boldsymbol{\phi}] \in \Pi_{\mathrm{KKT}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda}), \tag{34}$$

where $\boldsymbol{a} \in \mathbb{R}^{Tp}$ and $\boldsymbol{b}, \boldsymbol{\phi} \in \mathbb{R}^{T(m+p)}$. We write $\boldsymbol{v} = [\boldsymbol{v}_1; \boldsymbol{v}_2]$, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1; \boldsymbol{\lambda}_2]$, and $\boldsymbol{b} = [\boldsymbol{b}_1; \boldsymbol{b}_2]$ with $\boldsymbol{v}_1, \boldsymbol{\lambda}_1, \boldsymbol{b}_1 \in \mathbb{R}^{Tm}$ and $\boldsymbol{v}_2, \boldsymbol{\lambda}_2, \boldsymbol{b}_2 \in \mathbb{R}^{Tp}$. From the definition of Π_{KKT} in (33), we have

$$\boldsymbol{a} \in \partial f_T(\boldsymbol{w}) - \boldsymbol{C}^\top \boldsymbol{\lambda}, \tag{35a}$$

$$\boldsymbol{b}_1 = \nabla g_T^1(\boldsymbol{v}_1) + \boldsymbol{\lambda}_1, \tag{35b}$$

$$\boldsymbol{b}_2 \in \partial g_T^2(\boldsymbol{v}_2) + \boldsymbol{\lambda}_2 \tag{35c}$$

$$\boldsymbol{\phi} = \boldsymbol{C}\boldsymbol{w} - [\boldsymbol{v}_1; \boldsymbol{v}_2]. \tag{35d}$$

This, together with the definition of Γ_{KKT} in (28), yields

$$\left[\boldsymbol{a}; \boldsymbol{\lambda}_1 - \alpha\mathbf{1}/\boldsymbol{v}_1^*; \boldsymbol{v}_1 - \boldsymbol{v}_1^*; \boldsymbol{b}_2; \boldsymbol{\phi}\right] \in \Gamma_{\mathrm{KKT}}(\boldsymbol{w}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2). \tag{36}$$

Since $\mathrm{dist}\left(\mathbf{0}, \Gamma_{\mathrm{KKT}}(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{\lambda})\right) \leq \epsilon$, we see from (28) that $\boldsymbol{v}_1 \in \mathbb{B}(\boldsymbol{v}_1^*, \epsilon) := \{\boldsymbol{\pi} \in \mathbb{R}^{Tm} \mid \|\boldsymbol{\pi} - \boldsymbol{v}_1^*\|_2 \leq \epsilon\}$. In view of the fact that $\boldsymbol{v}_1^* \in \mathbb{R}^{Tm}_{++}$, by shrinking $\epsilon > 0$ if necessary, we may assume that $\mathbb{B}(\boldsymbol{v}_1^*, \epsilon) \subseteq \mathbb{R}^{Tm}_{++}$. Using the fact that $\nabla g_T^1(\boldsymbol{v}_1) = -\alpha\mathbf{1}/\boldsymbol{v}_1$ and $\nabla^2 g_T^1(\boldsymbol{v}_1) = \mathrm{Diag}(\alpha\mathbf{1}/\boldsymbol{v}_1^2)$, we deduce that g_T^1 is ν-strongly convex and ∇g_T^1 is $\hbar$-Lipschitz continuous on $\mathbb{B}(\boldsymbol{v}_1^*, \epsilon)$ for some constants $\nu, \hbar > 0$.

[9] A proper extended real-valued function f is called *piecewise linear-quadratic* if $\mathrm{dom}(f)$ can be represented as the union of finitely many polyhedral sets, relative to each of which $f(\boldsymbol{x})$ is given by an expression of the form $\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} + c$ for some scalar c, vector $\boldsymbol{b}$, and symmetric matrix $\boldsymbol{A}$; see [35, Definition 10.20].

Since

$$\mathrm{dist}\left(\mathbf{0},\Pi_{\mathrm{KKT}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})\right) = \inf_{\substack{[\boldsymbol{a};\boldsymbol{b};\boldsymbol{\phi}]\\ \in\Pi_{\mathrm{KKT}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})}} \|[\boldsymbol{a};\boldsymbol{b};\boldsymbol{\phi}]\|_2\,, \qquad (37)$$

we have

$$\begin{aligned}
&\mathrm{dist}([\boldsymbol{w};\boldsymbol{v};\boldsymbol{\lambda}],\mathbb{O}^*) \le \eta\cdot\mathrm{dist}\left(\mathbf{0},\Gamma_{\mathrm{KKT}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})\right)\\
&\le \eta\left(\|\boldsymbol{a}\|_2 + \left\|\boldsymbol{\lambda}_1 - \frac{\alpha\mathbf{1}}{\boldsymbol{v}_1^*}\right\|_2 + \|\boldsymbol{v}_1-\boldsymbol{v}_1^*\|_2 + \|\boldsymbol{b}_2\|_2 + \|\boldsymbol{\phi}\|_2\right)\\
&\le \eta\Bigg(\|\boldsymbol{a}\|_2 + \|\boldsymbol{b}_1\|_2 + \|\nabla g_T^1(\boldsymbol{v}_1) - \nabla g_T^1(\boldsymbol{v}_1^*)\|_2\\
&\qquad\qquad + \|\boldsymbol{v}_1-\boldsymbol{v}_1^*\|_2 + \|\boldsymbol{b}_2\|_2 + \|\boldsymbol{\phi}\|_2\Bigg)\\
&\le \eta\Bigg(\|\boldsymbol{a}\|_2 + \|\boldsymbol{b}_1\|_2 + \hbar\,\|\boldsymbol{v}_1-\boldsymbol{v}_1^*\|_2\\
&\qquad\qquad + \|\boldsymbol{v}_1-\boldsymbol{v}_1^*\|_2 + \|\boldsymbol{b}_2\|_2 + \|\boldsymbol{\phi}\|_2\Bigg)\\
&\le \eta\left(2\,\|[\boldsymbol{a};\boldsymbol{b}_1;\boldsymbol{b}_2;\boldsymbol{\phi}]\|_2 + (1+\hbar)\,\|\boldsymbol{v}_1-\boldsymbol{v}_1^*\|_2\right), \qquad (38)
\end{aligned}$$

where the first inequality follows from (29) and the assumption that $\mathrm{dist}\left(\mathbf{0},\Gamma_{\mathrm{KKT}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})\right)\le\epsilon$, the second inequality follows from (36), the third inequality follows from (35b) and the fact that $-\alpha\mathbf{1}/\boldsymbol{v}_1^* = \nabla g_T^2(\boldsymbol{v}^*)$, and the fourth inequality is due to the $\hbar$-Lipschitz continuity of ∇g_T^1 on $\mathbb{B}(\boldsymbol{v}_1^*,\epsilon)$.

Next, we bound $\|\boldsymbol{v}_1-\boldsymbol{v}_1^*\|_2$. Let $(\bar{\boldsymbol{w}},\bar{\boldsymbol{v}},\bar{\boldsymbol{\lambda}})$ be the projection of $(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})$ on $\mathbb{O}^*$. Such a projection is well defined, as it can be easily verified that $\mathbb{O}^*$ is a closed convex set. Let $\bar{\boldsymbol{v}} = [\bar{\boldsymbol{v}}_1;\bar{\boldsymbol{v}}_2]$ and $\bar{\boldsymbol{\lambda}} = [\bar{\boldsymbol{\lambda}}_1;\bar{\boldsymbol{\lambda}}_2]$ with $\bar{\boldsymbol{v}}_1,\bar{\boldsymbol{\lambda}}_1\in\mathbb{R}^{Tm}$ and $\bar{\boldsymbol{v}}_2,\bar{\boldsymbol{\lambda}}_2\in\mathbb{R}^{Tp}$. Note that we have $\bar{\boldsymbol{v}}_1 = \boldsymbol{v}_1^*$ by (27) and

$$\begin{aligned}
\mathbf{0} &\in \partial f_T(\bar{\boldsymbol{w}}) - \boldsymbol{C}^\top\bar{\boldsymbol{\lambda}}, &(39a)\\
\mathbf{0} &= \nabla g_T^1(\bar{\boldsymbol{v}}_1) + \bar{\boldsymbol{\lambda}}_1, &(39b)\\
\mathbf{0} &\in \partial g_T^2(\bar{\boldsymbol{v}}_2) + \bar{\boldsymbol{\lambda}}_2, &(39c)\\
\mathbf{0} &= \boldsymbol{C}\bar{\boldsymbol{w}} - [\bar{\boldsymbol{v}}_1;\bar{\boldsymbol{v}}_2] &(39d)
\end{aligned}$$

by the KKT conditions of Problem (13). Using (35a), (39a), and the property of ∂f_T, we have

$$\left(\boldsymbol{C}^\top\boldsymbol{\lambda} + \boldsymbol{a} - \boldsymbol{C}^\top\bar{\boldsymbol{\lambda}}\right)^\top(\boldsymbol{w}-\bar{\boldsymbol{w}}) \ge 0.$$

This, together with (35d) and (39d), gives

$$\boldsymbol{a}^\top(\boldsymbol{w}-\bar{\boldsymbol{w}}) \ge -(\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}})^\top\boldsymbol{C}(\boldsymbol{w}-\bar{\boldsymbol{w}}) = -(\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}})^\top(\boldsymbol{\phi}+\boldsymbol{v}-\bar{\boldsymbol{v}}),$$

or equivalently,

$$\boldsymbol{a}^\top(\boldsymbol{w}-\bar{\boldsymbol{w}}) + \boldsymbol{\phi}^\top(\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}}) \ge -(\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}})^\top(\boldsymbol{v}-\bar{\boldsymbol{v}}). \qquad (40)$$

Since g_T^1 is ν-strongly convex on $\mathbb{B}(\boldsymbol{v}_1^*,\epsilon)$, by using (35b) and (39b), we have

$$\begin{aligned}
\nu\|\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1\|_2^2 &\le (\nabla g_T^1(\boldsymbol{v}_1) - \nabla g_T^1(\bar{\boldsymbol{v}}_1))^\top(\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1)\\
&= \left(\boldsymbol{b}_1 - \boldsymbol{\lambda}_1 + \bar{\boldsymbol{\lambda}}_1\right)^\top(\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1),
\end{aligned}$$

which is equivalent to

$$\boldsymbol{b}_1^\top(\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1) \ge (\boldsymbol{\lambda}_1-\bar{\boldsymbol{\lambda}}_1)^\top(\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1) + \nu\|\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1\|_2^2. \quad (41)$$

Furthermore, using (35c), (39c), and the property of ∂g_T^2, we have

$$\left(\boldsymbol{b}_2 - \boldsymbol{\lambda}_2 + \bar{\boldsymbol{\lambda}}_2\right)^\top(\boldsymbol{v}_2-\bar{\boldsymbol{v}}_2) \ge 0,$$

or equivalently,

$$\boldsymbol{b}_2^\top(\boldsymbol{v}_2-\bar{\boldsymbol{v}}_2) \ge (\boldsymbol{\lambda}_2-\bar{\boldsymbol{\lambda}}_2)^\top(\boldsymbol{v}_2-\bar{\boldsymbol{v}}_2). \qquad (42)$$

Now, summing (40)–(42) and noting that

$$(\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}})^\top(\boldsymbol{v}-\bar{\boldsymbol{v}}) = (\boldsymbol{\lambda}_1-\bar{\boldsymbol{\lambda}}_1)^\top(\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1) + (\boldsymbol{\lambda}_2-\bar{\boldsymbol{\lambda}}_2)^\top(\boldsymbol{v}_2-\bar{\boldsymbol{v}}_2),$$

we obtain

$$\begin{aligned}
&\boldsymbol{a}^\top(\boldsymbol{w}-\bar{\boldsymbol{w}}) + \boldsymbol{b}_1^\top(\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1) + \boldsymbol{b}_2^\top(\boldsymbol{v}_2-\bar{\boldsymbol{v}}_2) + \boldsymbol{\phi}^\top(\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}})\\
&\ge \nu\|\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1\|_2^2.
\end{aligned}$$

It follows that

$$\begin{aligned}
&\nu\|\boldsymbol{v}_1-\boldsymbol{v}_1^*\|_2^2 = \nu\|\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1\|_2^2\\
&\le \boldsymbol{a}^\top(\boldsymbol{w}-\bar{\boldsymbol{w}}) + \boldsymbol{b}_1^\top(\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1) + \boldsymbol{b}_2^\top(\boldsymbol{v}_2-\bar{\boldsymbol{v}}_2) + \boldsymbol{\phi}^\top(\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}})\\
&\le \|\boldsymbol{a}\|_2\|\boldsymbol{w}-\bar{\boldsymbol{w}}\|_2 + \|\boldsymbol{b}_1\|_2\|\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1\|_2 + \|\boldsymbol{b}_2\|_2\|\boldsymbol{v}_2-\bar{\boldsymbol{v}}_2\|_2\\
&\qquad + \|\boldsymbol{\phi}\|_2\|\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}}\|_2\\
&\le \sqrt{\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2\|_2^2 + \|\boldsymbol{\phi}\|_2^2}\\
&\quad\times\sqrt{\|\boldsymbol{w}-\bar{\boldsymbol{w}}\|_2^2 + \|\boldsymbol{v}_1-\bar{\boldsymbol{v}}_1\|_2^2 + \|\boldsymbol{v}_2-\bar{\boldsymbol{v}}_2\|_2^2 + \|\boldsymbol{\lambda}-\bar{\boldsymbol{\lambda}}\|_2^2}\\
&= \|[\boldsymbol{a};\boldsymbol{b}_1;\boldsymbol{b}_2;\boldsymbol{\phi}]\|_2\cdot\mathrm{dist}([\boldsymbol{w};\boldsymbol{v};\boldsymbol{\lambda}],\mathbb{O}^*), \qquad (43)
\end{aligned}$$

where the last line follows from (34) and the definition of $(\bar{\boldsymbol{w}},\bar{\boldsymbol{v}},\bar{\boldsymbol{\lambda}})$. Plugging (43) into (38) yields

$$\begin{aligned}
\mathrm{dist}([\boldsymbol{w};\boldsymbol{v};\boldsymbol{\lambda}],\mathbb{O}^*) \le \eta\Bigg(&2\,\|[\boldsymbol{a};\boldsymbol{b}_1;\boldsymbol{b}_2;\boldsymbol{\phi}]\|_2\\
&+\frac{1+\hbar}{\sqrt{\nu}}\sqrt{\|[\boldsymbol{a};\boldsymbol{b}_1;\boldsymbol{b}_2;\boldsymbol{\phi}]\|_2\cdot\mathrm{dist}([\boldsymbol{w};\boldsymbol{v};\boldsymbol{\lambda}],\mathbb{O}^*)}\Bigg).
\end{aligned}$$

Upon solving the above inequality, we obtain

$$\mathrm{dist}([\boldsymbol{w};\boldsymbol{v};\boldsymbol{\lambda}],\mathbb{O}^*) \le \frac{\kappa_1+\sqrt{\kappa_1^2+4\kappa_2}}{2}\,\|[\boldsymbol{a};\boldsymbol{b};\boldsymbol{\phi}]\|_2\,,$$

where $\kappa_1 := \frac{1+\hbar}{\sqrt{\nu}}\eta$ and $\kappa_2 := 2\eta$. Since the above inequality holds for all $[\boldsymbol{a};\boldsymbol{b};\boldsymbol{\phi}]\in\Pi_{\mathrm{KKT}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})$, it follows from (37) that

$$\mathrm{dist}([\boldsymbol{w};\boldsymbol{v};\boldsymbol{\lambda}],\mathbb{O}^*) \le \varsigma\cdot\mathrm{dist}\left(\mathbf{0},\Pi_{\mathrm{KKT}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})\right), \qquad (44)$$

where $\varsigma := \frac{\kappa_1+\sqrt{\kappa_1^2+4\kappa_2}}{2}$. This shows that Π_{KKT} is metrically subregular at any KKT point of Problem (13), which completes Step 1.

2) Step 2: Given any proper extended real-valued function f, we have the equivalence $\boldsymbol{b} = \mathrm{prox}_f(\boldsymbol{a}) \Leftrightarrow \boldsymbol{a}-\boldsymbol{b}\in\partial f(\boldsymbol{b})$ for any vectors $\boldsymbol{a},\boldsymbol{b}$. It follows that

$$\begin{aligned}
\boldsymbol{w}-\widetilde{\boldsymbol{w}} &\in \partial f_T\left(\widetilde{\boldsymbol{w}}\right) - \boldsymbol{C}^\top\boldsymbol{\lambda}, &(45a)\\
\boldsymbol{v}-\widetilde{\boldsymbol{v}} &\in \partial g_T\left(\widetilde{\boldsymbol{v}}\right) - \boldsymbol{\lambda}, &(45b)
\end{aligned}$$

where $\widetilde{\boldsymbol{w}} := \mathrm{prox}_{f_T}(\boldsymbol{w}+\boldsymbol{C}^\top\boldsymbol{\lambda})$ and $\widetilde{\boldsymbol{v}} := \mathrm{prox}_{g_T}(\boldsymbol{v}+\boldsymbol{\lambda})$. Let $[\boldsymbol{w};\boldsymbol{v};\boldsymbol{\lambda}]$ be such that $[\widetilde{\boldsymbol{w}};\widetilde{\boldsymbol{v}};\boldsymbol{\lambda}]\in\mathcal{U}$. We compute

$$\begin{aligned}
&\mathrm{dist}^2(\mathbf{0},\Pi_{\mathrm{KKT}}(\widetilde{\boldsymbol{w}},\widetilde{\boldsymbol{v}},\boldsymbol{\lambda}))\\
&=\mathrm{dist}^2\left(\mathbf{0},\begin{bmatrix}\partial f_T\left(\widetilde{\boldsymbol{w}}\right)-\boldsymbol{C}^\top\boldsymbol{\lambda}\\ \partial g_T\left(\widetilde{\boldsymbol{v}}\right)-\boldsymbol{\lambda}\\ \boldsymbol{C}^\top\widetilde{\boldsymbol{w}}-\widetilde{\boldsymbol{v}}\end{bmatrix}\right) \le \mathrm{dist}^2\left(\mathbf{0},\begin{bmatrix}\boldsymbol{w}-\widetilde{\boldsymbol{w}}\\ \boldsymbol{v}-\widetilde{\boldsymbol{v}}\\ \boldsymbol{C}^\top\widetilde{\boldsymbol{w}}-\widetilde{\boldsymbol{v}}\end{bmatrix}\right)\\
&=\|\boldsymbol{w}-\widetilde{\boldsymbol{w}}\|_2^2 + \|\boldsymbol{v}-\widetilde{\boldsymbol{v}}\|_2^2 + \|\boldsymbol{C}^\top\widetilde{\boldsymbol{w}}-\widetilde{\boldsymbol{v}}\|_2^2
\end{aligned}$$

$$
\begin{aligned}
\leq & \|\boldsymbol{w}-\widetilde{\boldsymbol{w}}\|_2^2+\|\boldsymbol{v}-\widetilde{\boldsymbol{v}}\|_2^2 \\
& +3\|\boldsymbol{C}^\top(\boldsymbol{w}-\widetilde{\boldsymbol{w}})\|_2^2+3\|\boldsymbol{v}-\widetilde{\boldsymbol{v}}\|_2^2+3\|\boldsymbol{C}^\top\boldsymbol{w}-\boldsymbol{v}\|_2^2 \\
\leq & (1+3\|\boldsymbol{C}\|_2^2)\|\boldsymbol{w}-\widetilde{\boldsymbol{w}}\|_2^2+4\|\boldsymbol{v}-\widetilde{\boldsymbol{v}}\|_2^2+3\|\boldsymbol{C}^\top\boldsymbol{w}-\boldsymbol{v}\|_2^2 \\
\leq & \max\left\{1+3\|\boldsymbol{C}\|_2^2,4\right\}\left(\|\boldsymbol{w}-\widetilde{\boldsymbol{w}}\|_2^2+\|\boldsymbol{v}-\widetilde{\boldsymbol{v}}\|_2^2+\|\boldsymbol{C}^\top\boldsymbol{w}-\boldsymbol{v}\|_2^2\right) \\
= & \max\left\{1+3\|\boldsymbol{C}\|_2^2,4\right\}\cdot\|\Pi_{\text{KKT}}^{\text{p}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})\|_2^2, \qquad (46)
\end{aligned}
$$

where the first inequality is due to (45a) and (45b), and the last equality follows from the definition of $\Pi_{\text{KKT}}^{\text{p}}$ in (24). Consequently, we have

$$
\begin{aligned}
& \operatorname{dist}([\boldsymbol{w};\boldsymbol{v};\boldsymbol{\lambda}],\mathbb{O}^*)=\operatorname{dist}([\boldsymbol{w}-\widetilde{\boldsymbol{w}}+\widetilde{\boldsymbol{w}};\boldsymbol{v}-\widetilde{\boldsymbol{v}}+\widetilde{\boldsymbol{v}};\boldsymbol{\lambda}],\mathbb{O}^*) \\
& \leq \|[\boldsymbol{w}-\widetilde{\boldsymbol{w}};\boldsymbol{v}-\widetilde{\boldsymbol{v}};\boldsymbol{0}]\|_2+\operatorname{dist}([\widetilde{\boldsymbol{w}};\widetilde{\boldsymbol{v}};\boldsymbol{\lambda}],\mathbb{O}^*) \\
& \leq \|\Pi_{\text{KKT}}^{\text{p}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})\|_2+\operatorname{dist}([\widetilde{\boldsymbol{w}};\widetilde{\boldsymbol{v}};\boldsymbol{\lambda}],\mathbb{O}^*) \\
& \leq \|\Pi_{\text{KKT}}^{\text{p}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})\|_2+\varsigma\cdot\operatorname{dist}(\boldsymbol{0},\Pi_{\text{KKT}}(\widetilde{\boldsymbol{w}};\widetilde{\boldsymbol{v}};\boldsymbol{\lambda})) \\
& \leq \left(1+\varsigma\sqrt{\max\left\{1+3\|\boldsymbol{C}\|_2^2,4\right\}}\right)\|\Pi_{\text{KKT}}^{\text{p}}(\boldsymbol{w},\boldsymbol{v},\boldsymbol{\lambda})\|_2,
\end{aligned}
$$

where the first inequality is due to the triangle inequality, the second inequality follows from the definition of $\Pi_{\text{KKT}}^{\text{p}}$ in (24), the third inequality follows from (44) and the assumption that $[\widetilde{\boldsymbol{w}};\widetilde{\boldsymbol{v}};\boldsymbol{\lambda}]\in\mathcal{U}$, and the last inequality follows from (46). This shows that the error bound (26) holds in the neighborhood $\mathcal{U}$ with constant $\zeta=1+\varsigma\sqrt{\max\{1+3\|\boldsymbol{C}\|_2^2,4\}}$. In particular, the mapping $\Pi_{\text{KKT}}^{\text{p}}$ is metrically subregular at any KKT point of Problem (13). This completes Step 2.

References

[1] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, 2019.

[2] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, 2019.

[3] 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[4] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, 2015.

[5] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[6] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

[7] C. Hu, L. Cheng, J. Sepulcre, G. El Fakhri, Y. M. Lu, and Q. Li, "A graph theoretical regression model for brain connectivity learning of Alzheimer's disease," in *Proc. 10th IEEE ISBI*, 2013, pp. 616–619.

[8] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.

[9] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. 19th AISTATS*, 2016, pp. 920–929.

[10] X. Wang, Y.-M. Pun, and A. M.-C. So, "Distributionally robust graph learning from smooth signals under moment uncertainty," *IEEE Trans. Signal Process.*, vol. 70, pp. 6216–6231, 2022.

[11] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning based on sparseness of temporal variation," in *Proc. 2019 IEEE ICASSP*, 2019, pp. 5411–5415.

[12] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *Proc. 2017 IEEE ICASSP*, 2017, pp. 2826–2830.

[13] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, "Network inference via the time-varying graphical lasso," in *Proc. the 23rd ACM SIGKDD*, 2017, pp. 205–213.

[14] X. Zhang and Q. Wang, "Time-varying graph learning under structured temporal priors," *arXiv preprint arXiv:2110.05018*, 2021.

[15] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," in *Proc. 2019 ICLR*, 2019.

[16] K. Yamada and Y. Tanaka, "Time-varying graph learning with constraints on graph temporal variation," *arXiv:2001.03346*, 2020.

[17] N. Komodakis and J.-C. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 31–54, 2015.

[18] X. Wang, C. Yao, H. Lei, and A. M.-C. So, "An efficient alternating direction method for graph learning from smooth signals," in *Proc. 2021 IEEE ICASSP*, 2021, pp. 5380–5384.

[19] S. S. Saboksayr and G. Mateos, "Accelerated graph learning from smooth signals," *IEEE Signal Process. Lett.*, vol. 28, pp. 2192–2196, 2021.

[20] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[21] G. Fatima, A. Arora, P. Babu, and P. Stoica, "Learning sparse graphs via majorization-minimization for smooth node signals," *IEEE Signal Process. Lett.*, vol. 29, pp. 1022–1026, 2022.

[22] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, "Hankel matrix rank minimization with applications to system identification and realization," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 3, pp. 946–977, 2013.

[23] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *J. Sci. Comput.*, vol. 66, no. 3, pp. 889–916, 2016.

[24] L. Zhao, Y. Wang, S. Kumar, and D. P. Palomar, "Optimization algorithms for graph Laplacian estimation via ADMM and MM," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4231–4244, 2019.

[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[26] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. SIAM, 1995.

[27] D. Han, D. Sun, and L. Zhang, "Linear rate convergence of the alternating direction method of multipliers for convex composite programming," *Math. Oper. Res.*, vol. 43, no. 2, pp. 622–637, 2018.

[28] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[29] J.-S. Pang, "Error bounds in mathematical programming," *Math. Program.*, vol. 79, no. 1–3, pp. 299–332, 1997.

[30] H. Liu, M.-C. Yue, and A. M.-C. So, "On the estimation performance and convergence rate of the generalized power method for phase synchronization," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2426–2446, 2017.

[31] Z. Zhou and A. M.-C. So, "A unified approach to error bounds for structured convex optimization problems," *Math. Program.*, vol. 165, no. 2, pp. 689–728, 2017.

[32] H. Liu, A. M.-C. So, and W. Wu, "Quadratic optimization with orthogonality constraint: Explicit Łojasiewicz exponent and linear convergence of retraction-based line-search and stochastic variance-reduced gradient methods," *Math. Program.*, vol. 178, no. 1–2, pp. 215–262, 2019.

[33] M.-C. Yue, Z. Zhou, and A. M.-C. So, "A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo–Tseng error bound property," *Math. Program.*, vol. 174, no. 1, pp. 327–358, 2019.

[34] P. Wang, H. Liu, and A. M.-C. So, "Linear convergence of a proximal alternating minimization method with extrapolation for ℓ_1-norm principal component analysis," *SIAM J. Optim.*, 2022.

[35] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Springer Science & Business Media, 2009, vol. 317.

[36] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *arXiv preprint arXiv:1408.5781*, 2014.

[37] C. D. Manning, H. Schütze, and P. Raghavan, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[38] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collection," *ACM Trans. Math. Softw.*, vol. 38, no. 1, pp. 1–25, 2011.

[39] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2009.

[40] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *2009 IEEE CVPR*, 2009, pp. 1746–1753.

[41] A. Natali, E. Isufi, M. Coutino, and G. Leus, "Learning time-varying graphs from online data," *IEEE Open Journal of Signal Processing*, vol. 3, pp. 212–228, 2022.

[42] S. S. Saboksayr, G. Mateos, and M. Cetin, “Online discriminative graph learning from multi-class smooth signals,” *Signal Process.*, vol. 186, p. 108101, 2021.

[43] G. H. Golub and C. F. Van Loan, *Matrix Computations*. JHU Press, 2013.

[44] A. L. Dontchev and R. T. Rockafellar, *Implicit Functions and Solution Mappings*. Springer, 2009, vol. 543.