# Exploiting Structure In Data: Sampling and Signal Processing on Graphs

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Rohan Anilkumar Varma

B.S Electrical Engineering and Computer Sciences, University of California, Berkeley
B.A Economics, University of California, Berkeley
B.A Statistics, University of California, Berkeley
M.S Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

May 2020

# Acknowledgments

Firstly, I must acknowledge the National Science Foundation (grants CIF:1563918 and CIF:1421919), the Department of Electrical and Computer Engineering (ECE) and Carnegie Institute of Technology (CIT) Dean's Tuition Fellowship for their financial support.

I procrastinated until the day before I submitted my thesis to write the acknowledgements section; primarily because I will not be able to do justice to the many people who deserve my immense thanks in words.

I am indebted to my advisor, Jelena Kovačević. Since my first day in graduate school, Jelena believed in me and gave me endless support giving me the freedom to discover and explore the field. Your ability to effectively communicate ideas, ability to step back and ask the broader questions, and endless cheer made me not just a better researcher but a better person. My committee was an immense source of help and guidance from my proposal to my defense. Aarti Singh was a fantastic source of guidance, patience and support throughout. I aspire to have your ability to distill a problem and pick the right questions to ask. Yuejie Chi was a fantastic source of guidance and help. Thank you for introducing me to the wonderful world of non-convexity! Jose Moura was also a source of guidance, the questions he posed made this thesis stronger.

The last five years have been truly memorable where I have grown both academically and as a person. Carnegie Mellon and Pittsburgh has been a wonderful place to study and learn. Anupama Kuruvilla, Mike McCann, Jackie Chen, Filipe Condessa, Siheng Chen, Anuva Kulkarni, Chaojing Duan and Harlin Lee were all great lab-mates; I will miss our lab dinners! A not insignificant chunk of the work in this thesis is either a continuation of work by or a collaboration with Siheng Chen. His endless font of ideas made working with him a pleasure. It also has been a pleasure working with Harlin in my last year. Filipe has also been of immense help combined with good humor. I also must thank Christina Cowan for the endless supply of espresso and always being so cheerful.

Thank you, my Pittsburgh friends, Vivek, Arpita, Ardra, Krishna, Aditya, Ramon, Tom, Claire, Serena, Jason, Rishi and Clint. Petar, it's been a blast, thanks for all the support, from day one of 715! Unisha, thanks for all the support, help, and encouragement, I will forever be indebted to you.

To all my Berkeley friends, thanks for the memories. Berkeley in many ways was the formative years of my life. I couldn't have asked for a better group of friends to go through it. And I must be grateful to Berkeley for its grueling classes so much so that I never complained about classes at Carnegie Mellon! Harman, thanks for being a rock to lean on across the country, and always showing me a great time in Los Angeles (with Bilu!) and San Francisco. I look forward to us meeting more often regardless of where in the world we may be. And I would be remiss not to mention the Rumor Mill, which had me burst into laughter leading to lab-mates looking over more than once. Mihir, thanks for always having my back. As is turning into a somewhat common theme, I promise we will meet more often. To Aniruddha and Naveen, thanks for everything especially the Iceland trip.

To my friends from back in Abu Dhabi, especially Ibra, Senal, Ramana, Raza, Athar, Chaman

and Chunky, thanks for the laughs, and always being ready for a game of Halo. We all live in different corners of the world, yet the activity on mostly virtual platforms is testament to the strength of our friendship.

My grandfather, Vikram Varma, who passed away while I was pursuing my studies, was an inspiration and I'd like to think he would be proud of me. Both my grandmothers have been wonderful sources of inspiration, love and comfort. My aunt, Rashmi, has been a pillar of support. I look forward to visiting London more and spending more time with my little cousins Tanvi and (not so little), Vedant. My cousin, Anu has also been of massive help and support. My extended family, my uncles, aunts, and cousins also have always been supportive in all my endeavors.

To my younger brother Varun, thanks for all your love, support and help. As I finish my PhD, you close another chapter of your life too in your undergraduate career. I can't wait to see what the world has in store for us. The only problem now is that we cannot attend each other's graduation ceremony since they fall on the same day! My parents, Anil and Priya, have always been an unwavering anchor, without whom I wouldn't be here. Thanks for always encouraging me, giving me the freedom to explore what I wanted, and always making me feel better even when I was down. One downside of me moving abroad was that I only got to go home once or twice a year; I look forward to spending more time with you. This dissertation would not have been possible without their warm love, continued patience, and endless support. I dedicate this thesis to them.

# Abstract

With the explosive growth of information and communication, data is being generated at an unprecedented rate from various sources, including multimedia, sensor networks, biological systems, social networks, and physical infrastructure. Research in graph signal processing aims to develop tools for processing such data by providing a framework for the analysis of high-dimensional data defined on irregular graph domains. Graph signal processing extends fundamental signal processing concepts to data supported on graphs that we refer to as graph signals. In this work, we study two fraternal problems: (1) *sampling* and (2) *reconstruction* of signals on graphs. Both of these problems are eminent topics in the field of signal processing over the past decades and have meaningful implications for many real-world problems including semi-supervised learning and active learning on graphs. Sampling is the task of choosing or measuring some representative subset of the signal such that we can interpolate and recover the whole signal. In many settings, acquiring samples is expensive and it is desirable to be able to approximate the signal as efficiently as possible. Signal reconstruction refers the task of recovering the true graph signal given noisy, incomplete, or corrupt measurements. It is evident that these two problems are intimately tied to the underlying graph structure.

In this body of work, we study the tasks of sampling and reconstruction for two complementary classes of graph signals that broadly capture characteristics of most graph-structured data: (1) *smooth signals* that are characterized by slow transitions with respect to graph and (2) *piecewise-smooth signals* that are characterized by localized behavior, abrupt discontinuities or fast transitions over the underlying graph. We examine why a one-size-fits-all approach may not always be appropriate and consequently design algorithms and frameworks specific to each graph signal model. We first present a sampling theory in the spirit of Shannon and Nyquist and study the limits of sampling these two graph signal models under passive and active settings. We present a framework for the reconstruction or estimation of graph signals and investigate the limitations of these methods. Graph trend filtering is a flexible framework for estimation on graphs that is adaptive to inhomogeneity in the level of smoothness and localized characteristics of an observed signal across nodes. We strengthen the graph trend filtering framework by considering a family of possibly non-convex regularizers that exhibit superior reconstruction performance over minimization for the denoising of piecewise smooth graph signals. We study product graphs which are a generative model that allows us to decompose a large real-world large graph into smaller graphs. We develop frameworks for sampling and reconstruction that both lessen the computational complexity and achieve better performance by availing of the inherent structure in product graphs.

The irregularity of the underlying structure of the data in contrast to the regularity in classical signal processing makes studying these problems on graphs challenging but also compelling. A key theme throughout this work is the interplay between the graph structure and the signal that lies on it. We study how structural properties of both the graph and the signal on the graph inform not only how well we can perform these two tasks but also the design of algorithms and strategies to perform them efficiently. Moreover, we illustrate the power of these proposed tools via vignettes on semi-supervised learning and efficient communication in sensor networks.

# Contents

# Listing of figures

# Listing of tables

*To my parents*

# 1

# Introduction

## 1.1 Motivation

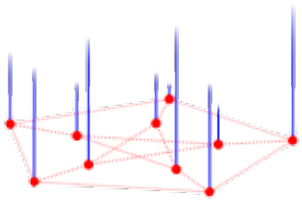


*Figure 1.1: A graph signal supported on a graph*

With the vast growth in information and communication, data is being generated and acquired at an unprecedented rate from various sources, including multimedia, climate, neuro-imaging, social networks, urban infrastructure, biological systems and sensor networks. The dimensionality of the information in this data offers significant challenges both in terms of the computational and sample complexity required to process and glean useful insight from this data. However, such high-dimensional data often have an underlying structure that limits its intrinsic dimensionality. Graphs offer the ability to model this internal structure that is inherently complex and irregular. A rigorous way to formulate the assumption that the data possesses an innate structure with fewer degrees of freedom is to model the data as belonging to a low-dimensional manifold embedded in a high dimensional space. That is, we make a direct association between the *geodesic distance* between two points on a manifold and the *graph distance* between these two points with respect to an underlying graphs. As a result, a graph can be seen as a discrete approximation to a continuous manifold.

Graph signal processing (GSP) was borne of a need to process such graph-structured data in a systematic and mathematically rigorous way[1,2,3]. Algebraic signal processing introduced an axiomatic approach to signal processing and showed how the signal model is generated from a simple filter, the shift, which then determines filtering, convolution, the Fourier transform, and frequency[4,5]. This led to the introduction of using the graph structure as the shift operator and the genesis of graph signal processing. As a result, GSP in many ways builds upon established principles in classical signal processing which deals with signals that are supported on regular structures such as time-series signals or images. The challenges of GSP are principally consequences of the irregularity of the underlying graph structure. For example, it is unclear how to define a translation operator for a graph signal. In addition, as we shall see, simultaneous localization in the vertex and frequency space which is infeasible in classical signal processing on regular grids as a result of the uncertainty principle, is viable on irregular graphs. This has significant implications for many of the problems we study.

Recent work involves graph-based filtering[6], graph-based transforms[6,7,8], sampling and interpolation on graphs[9,10,11], semi-supervised classification on graphs[12,13,14], graph dictionary learning[15,16] and community detection on graphs[17]. For a recent review see[18,2].

Graphs are usually acquired or constructed in one of two different ways. Firstly, the graph may be constructed using the data itself by using a distance, affinity or correlation metric or by learning the graph under some optimization criterion[19,20]. In point clouds and many graph-based semi-supervised classification problems for example, it is natural construct a nearest neighbor graph. Alternatively, the graph may be innate to the data as is the case in urban transportation or road networks and brain networks.

Sampling is the task of choosing or acquiring some subset of the signal with the view of later being able to recover the whole signal. A convenient way to view sampling is through the lens of data compression or dimensionality reduction. Particularly, we aim to efficiently sample the small-

est representative subset of the graph signal that retains the relevant information in the original data such that we can recover the original signal. This problem is important when the labeled data is scarce and expensive whereas unlabeled data is easily available for most semi-supervised classification problems on graphs. We consider two sampling paradigms: (1) *passive sampling*, and (2) *active sampling*. Passive sampling refers to the setting where we are constrained to strategies that are blind to any samples of the signal. That is, we design strategies by considering only the underlying graph structure and any modeling assumptions we have made. In contrast, active or adaptive sampling strategies are able to choose samples in an online fashion: the decision of where to sample next depends on all the observations made previously.

A good sampling strategy is beneficial for active learning and dimensionality reduction with graphs. For example, in semi-supervised learning with graphs, we embed the entire dataset onto a graph and a graph signal, whose nodes represent individual data samples, edges encode the similarities between data samples, and the signal is the class label. We then select a small subset of nodes from this graph to be the observed training data. A smart sampling strategy can help choose the most representative training data, leading to a more effective training process. On the other hand, a good recovery strategy is beneficial for prediction, completion and denoising with graphs. Consider the Netflix problem, for example. We can build a similarity graph of users and movies, and construct a ratings signal on top, but a user typically rates only a few movies. To recommend movies based on a user's preference, we need to predict their preferences for unrated items. This is equivalent to completing a matrix of graph signals from a few random, noisy samples.

Signal reconstruction or estimation refers the task of recovering the underlying signal given noisy or corrupted measurements. This is a well studied problem in data science under the guise of denoising and additionally has applications for inpainting, collaborative filtering, recommender systems and other large-scale data completion problems. Signal reconstruction is then essentially an inverse problem that aims to *reverse* the corruption of the signal. Since noise can have deleterious cascading effects in many downstream tasks, being able to efficiently and accurately reconstruct a signal is of significant importance. When constructing a signal model or a signal prior, the canonical assumption is that the signal is smooth with respect to the graph, that is, the signal coefficients do not vary much over local neighborhoods of the graph. However, this characterization is often insufficient for many real-world signals. It is often the case that there are localized discontinuities in the signal and the signal exhibits a *piecewise* behavior over the graph. As a result, it is necessary to develop representations and algorithms to process and analyze such signals.

## 1.2 Thesis Contributions

We study the tasks of (1) sampling signals and (2) reconstructing signals that lie on

graphs by understanding and exploiting the relationships between the graph signal, its appropriate representation with respect to the underlying graph and the graph itself.

The proposed body of work can be divided into three distinct yet complementary areas: (a) the sampling of smooth graph signals (b) the sampling of piecewise-smooth graph signals (c) the reconstruction of noisy or corrupted graph signals. A common theme throughout this work is the influence of the graph structure on both the limitations of these tasks and the performance of the proposed algorithms. In particular, the irregularity of the graph structure in contrast to the reg-

ular structure in classical time and image signal processing has important implications for both the fundamental limitations of these tasks and the design of algorithms and strategies to perform these tasks efficiently.

Sampling Smooth Graph Signals: We present the analog of the classical sampling theory for bandlimited signals on graphs, and show extensions for efficient sampling and recovery on product graphs. We then study the fundamental statistical limits of sampling smooth graph signals and establish how the structure of the graph drives both the optimal sampling strategy, how well we can expect to do, and the performance of these algorithms. Particularly, we study the performance of passive sampling versus active sampling for smooth signals.

Sampling Piecewise-Smooth Graph Signals: The discontinuities in piecewise-smooth signals make the task of efficiently sampling them substantially more challenging. In this work, we propose studying the differences between passive and active sampling of piecewise smooth signals on graphs and their interplay with the graph structure. Further, we aim to develop efficient sampling procedures and strategies for sampling such signals.

Reconstructing Graph Signals: We present and develop frameworks for the recovery or estimation of noisy or corrupted smooth and piecewise-smooth graph signals. We consider a synthesis-based approach based on signal approximation via a graph-based representation basis while the second analysis-based approach solves an optimization problem that penalizes the signal's variation on the graph. In the case of piecewise smooth signals, we show how using non-convex penalties for graph trend filtering gives superior performance in terms of both denoising and support recovery. Particularly, in the case of piecewise smooth signals, we seek to develop an understanding of the limitations and differences between graph-based total variation denoising and graph-based wavelet thresholding with respect to the underlying graph structure.

Broader Impact: The goal of the proposed tools and frameworks is to provide solutions for real-world applications of graph-structured data. We outline applications in sensor networks and semi-supervised learning.

### 1.2.1 Overview

In this section, we briefly overview the content in the following chapters. In Chapter 2, we aim to concisely present the foundational technical content that forms the basis for the subsequent chapters where we expound upon the specific aims presented in this chapter. We first briefly review the graph signal processing (GSP) framework which is the basis of this work. We then motivate and develop the signal models we study and their graph-based representations. We compare and contrast the two major classes of signals we study, smooth graph signals and piecewise smooth graph signals. Particularly, via an uncertainty principle on graphs, we gain further insight into the differences between how these two signal models interact with the underlying graph. This allows us to motivate the reasoning behind the choices that we make for the best representation for these different classes of signals.

In Chapter 3, we study sampling smooth signals on graphs. We first present a sampling theory for bandlimited graph signals that is analogous to downsampling discrete-time signals such that we can recover the signal perfectly. We extend this framework to show we can sample efficiently on product graphs by using a structured sampling procedure that allows us significant gains in computational complexity. We then study minimax lower bounds for sampling smooth graph signals under both passive and active samplings, and show that active sampling can't fundamentally outperform passive sampling. Further, we present optimal sampling and reconstruction algorithms with respect to the lower bounds for passive sampling and discuss how the underlying graph structure drives their performance. We then discuss recovering smooth graph signals from noisy or corrupted measurements by formulating an optimization problem that minimizes the variation of the signal over the graph. While this has been well studied in previous work, we present a framework for recovering smooth signals on product graphs that exploits the low rank structure of these signals by modeling the signal as a multi-dimensional tensor.

In Chapter 4, we first discuss the reconstruction of piecewise-smooth graph signals from noisy or corrupted measurements and present two approaches. The first approach solves a optimization problem via the graph trend filtering framework which enforces a sparsity constraint on (higher-order) discrete graph differences while the second approach is based on approximating the signal with respect to a graph wavelet basis or dictionary. Further, we strengthen the graph trend filtering framework by considering a large family of possibly non-convex regularizers that exhibit superior reconstruction performance over $\ell_1$ minimization for the denoising of piecewise smooth graph signals. We also seek to compare wavelet thresholding on graphs and graph-based total variation denoising for the estimation of piecewise-constant signals on graphs.

In Chapter 5, we study the sampling of piecewise-smooth signals on graphs. Similarly to Chapter 3, we propose studying passive and active sampling of piecewise smooth signals on graphs. Unlike sampling smooth signals that have no discontinuities, the localized nature of the discontinuities in piecewise smooth signals make the detection of these discontinuities inherently decoupled from the global features of the graph signal. It then follows that the passive sampling of piecewise-smooth graph signals is a significantly harder or even futile task. Consequently, we propose studying the active sampling of piecewise-smooth signals by designing algorithms and strategies. We emphasize here that the results are borne of largely experimental results.

# 2

# Graph Signal Processing and Related Areas

## 2.1 Graph Signal Processing

Two basic approaches to signal processing on graphs have been considered: The first is rooted in spectral graph theory and builds upon the graph Laplacian matrix. Since the graph Laplacian matrix is restricted to be symmetric and positive semi-definite, the spectral graph theory-based approach is applicable only to undirected graphs with real and nonnegative edge weights. The second approach, *discrete signal processing on graphs* ($\text{DSP}_\text{G}$)[6,21], is rooted in the *algebraic signal processing theory*[22,23] and builds on the graph shift operator, which works as the elementary operator that generates all linear shift-invariant filters for signals with a given structure. In general, the tools we propose and derived from and can be generalized for any standard graph representation. While some of the tools can be extended for directed graphs, for clarity and brevity, we only consider undirected graphs with positive weights in this discussion.

Algebraic signal processing introduced an axiomatic approach to signal processing and showed how the signal model is generated from a simple filter, the shift, which then determines filtering, convolution, the Fourier transform, and frequency. This led to the use of weighted, graph adjacency matrices as shifts that generate the graph signal model for signals indexed by nodes of an arbitrary directed or undirected graph.[3,24,25] We note that additionally, this choice is a direct generalization of the classical time signal model. That is, when the signal model is the classical time signal model, the shift and the graph signal model reverts to the classical time shift (delay) and signal model.

We consider a graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{v_0, \ldots, v_{N-1}\}$ is the set of nodes, $\mathcal{E} = \{e_0, \ldots, e_{M-1}\}$ is the set of edges, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the *graph shift operator*, or the weighted adjacency matrix. The edge set $\mathcal{E}$ represents the connections of the graph $G$, which can be either directed or undirected and the edge weight $\mathbf{A}_{j,k} = w_{j,k}$ between nodes $v_j$ and $v_k$ measures the underlying relation between the $j$th and the $k$th node, such as a similarity, a dependency, or a communication pattern. Let a *graph signal* be defined as

$$\mathbf{x} \;=\; \Big[x_0, x_1, \ldots, x_{N-1}\Big]^T \in \mathbb{R}^N,$$

where $x_i$ denotes the signal coefficient at the $i$the node. We note that ordering the signal coefficients corresponds to labeling the nodes of the graph and establishing the adjacency matrix.

The degree of a node is the sum of the weights of the edges from that node. We can write the diagonal degree matrix as $\mathcal{D}$ where the $i$-th entry in its diagonal is the degree of the $i$-th node. It is easy to see that the graph shift operator $\mathbf{A} \in \mathbb{R}^{N \times N}$ is an elementary filtering operation that replaces a signal coefficient at a node with a weighted linear combination of coefficients at its neighboring nodes. Some variations of this graph shift include the normalized adjacency matrix $\mathbf{A}_\text{norm} = \mathcal{D}^{-\frac{1}{2}}\,\mathbf{A}\,\mathcal{D}^{-\frac{1}{2}}$ and transition matrix $\text{P} = \mathcal{D}^{-1}\,\mathbf{A}$. We can also use the graph Laplacian matrix $\boldsymbol{L} \in \mathbb{R}^{N \times N}$ as a graph representation, which is a second-order difference operator on graphs. Some common choices of a graph Laplacian matrix are the unnormalized Laplacian $\boldsymbol{L} = \mathcal{D} - \mathbf{A}$, the normalized Laplacian $\text{I} - \mathcal{D}^{-\frac{1}{2}}\,\mathbf{A}\,\mathcal{D}^{-\frac{1}{2}}$, or the transition Laplacian $\text{I} - \mathcal{D}^{-1}\,\mathbf{A}$. We note that the graph Laplacian $\boldsymbol{L}$ is a symmetric positive semidefinite matrix.

The graph Fourier basis generalizes the traditional Fourier basis and is used to represent the graph signal in the graph spectral domain. The *graph Fourier basis* $\mathbf{V} \in \mathbb{R}^{N \times N}$ is defined to be the eigenvector matrix of the chosen graph representation. When we use the graph shift operator

$\mathbf{A}$,

$$\mathbf{A} = \mathbf{V}\,\Lambda\,\mathbf{V}^{-1}, \tag{2.1}$$

where the $i$-th column vector of $\mathbf{V}$ is the graph Fourier basis vector $\mathbf{v}_i$ corresponding to the eigenvalue $\lambda_i$ as the $i$th diagonal element in $\Lambda$. The *graph Fourier transform* of $\mathbf{x} \in \mathbb{R}^N$ is $\widehat{\mathbf{x}} = \mathbf{U}\,\mathbf{x}$, where $\mathbf{U} = \mathbf{V}^{-1}$ is called the graph Fourier transform matrix. When $\mathbf{A}$ is symmetric, then $\mathbf{U} = \mathbf{V}^T$ is orthonormal; the graph Fourier basis vector $\mathbf{v}_i$ is the $i$th row vector of $\mathbf{U}$. The *inverse graph Fourier transform* is $\mathbf{x} = \mathbf{V}\,\widehat{\mathbf{x}}$. The vector $\widehat{\mathbf{x}}$ represents the frequency coefficients corresponding to the graph signal $\mathbf{x}$, and the graph Fourier basis vectors can be regarded as graph frequency components. In this thesis, we use $\mathbf{V}$ and $\mathbf{U}$ to denote the inverse graph Fourier transform matrix and graph Fourier transform matrix for the chosen graph representation basis, which can be the adjacency matrix, the graph Laplacian, or the transition matrix.

The ordering of the graph Fourier basis vectors depends on the their variation with respect to the underlying graph. Further, the variation of a graph signal $\mathbf{x}$ is defined with respect to the chosen graph representation. When the graph representation is the graph shift $\mathbf{A}$, the variation of $\mathbf{x}$ is defined as

$$\mathrm{S}_{\mathbf{A}}(\mathbf{x}) = \left\| \mathbf{x} - \frac{1}{|\lambda_{\max}(\mathbf{A})|}\,\mathbf{A}\,\mathbf{x} \right\|_2^2,$$

where $\lambda_{\max}(\mathbf{A})$ is the eigenvalue of $\mathbf{A}$ with the largest magnitude. Unless otherwise specified, we assume that the graph shift has been normalized such that $\lambda_{\max}(\mathbf{A}) = 1$. We can show that when the eigenvalues of the graph shift $\mathbf{A}$ are sorted in a nonincreasing order $\lambda_1^{(\mathbf{A})} \geq \lambda_2^{(\mathbf{A})} \geq \ldots \geq \lambda_N^{(\mathbf{A})}$, the variations of the corresponding eigenvectors follow a nondecreasing order $\mathrm{S}_{\mathbf{A}}(\mathbf{v}_1^{(\mathbf{A})}) \leq \mathrm{S}_{(\mathbf{A})}(\mathbf{v}_2^{(\mathbf{A})}) \leq \ldots \leq \mathrm{S}_{(\mathbf{A})}(\mathbf{v}_N^{(\mathbf{A})})$.

When the graph representation is the graph Laplacian matrix, the variation of $\mathbf{x}$ is defined as

$$\mathrm{S}_{\boldsymbol{L}}(\mathbf{x}) = \sum_{i,j=1}^{N} \mathbf{A}_{i,j}(x_i - x_j)^2 = \mathbf{x}^T \boldsymbol{L}\mathbf{x}.$$

We can show that when the eigenvalues of the graph Laplacian $\boldsymbol{L}$ are sorted in a nondecreasing order $\lambda_1^{(\boldsymbol{L})} \leq \lambda_2^{(\boldsymbol{L})} \leq \ldots \leq \lambda_N^{(\boldsymbol{L})}$, the variation of the corresponding eigenvectors follows a nondecreasing order $\mathrm{S}_{\boldsymbol{L}}(\mathbf{v}_1^{(\boldsymbol{L})}) \leq \mathrm{S}_{\boldsymbol{L}}(\mathbf{v}_2^{(\boldsymbol{L})}) \leq \ldots \leq \mathrm{S}_{\boldsymbol{L}}(\mathbf{v}_N^{(\boldsymbol{L})})$.

The variations of a graph Fourier basis vectors allow us to order the graph Fourier basis vectors: the Fourier basis vectors with small variations are considered as *low-frequency* components while the vectors with large variations are considered as *high-frequency* components.[24] The eigenvectors associated with large eigenvalues of the graph shift (small eigenvalues of the graph Laplacian) represent low-frequency components and the eigenvectors associated with small eigenvalues of the graph shift (large eigenvalues of graph Laplacian) represent high-frequency components. In the following discussion, unless stated otherwise, we assume all graph Fourier bases are ordered from low frequencies to high frequencies.

Here, we restate Theorem 1 from[24] that establishes the frequency ordering induced by the total variation functional $\mathrm{TV}_{\mathbf{A}}$ for graphs that have diagonalizable adjacency matrices.

**Theorem 1.** Consider two distinct eigenvalues $\lambda_m, \lambda_n \in \mathbb{R}$ of the adjacency matrix $\mathbf{A}$ with corre-

sponding eigenvectors $\mathbf{v}_m$ and $\mathbf{v}_n$. If the eigenvalues are ordered as

$$\lambda_m \leq \lambda_n$$

,then the total variation of their eigenvectors satisfy

$$\mathrm{TV}_{\mathbf{A}}(\mathbf{v}_n) \leq \mathrm{TV}_{\mathbf{A}}(\mathbf{v}_m)$$

Further, we can show that an ordering of the graph Fourier basis from lowest to highest frequencies based on the graph shift quadratic form $\boldsymbol{S}_2(x)$ coincides with the ordering induced by the total variation $\mathrm{TV}_{\mathbf{A}}$.

## 2.2 Smooth Graph Signals

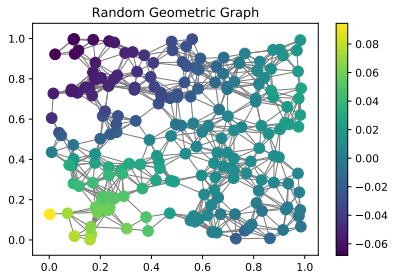



*Figure 2.1: An example of a smooth graph signal on a random geometric graph*

Here, we look at globally smooth graph signals where the signal coefficient at each node is close to the signal coefficients of its neighbors.[26,27,28,29]

**Definition 1.** A graph signal $\mathbf{x} \in \mathbb{R}^N$ is *globally smooth* on a graph $\mathbf{A} \in \mathbb{R}^{N \times N}$ with parameter $\eta \geq 0$, when

$$\mathrm{S}_{\mathbf{A}}(\mathbf{x}) \;\leq\; \eta \left\|\mathbf{x}\right\|_2^2 . \tag{2.2}$$

Denote this class of graph signals by $\mathrm{GS}_{\mathbf{A}}(\eta)$.

**Definition 2.** A graph signal $\mathbf{x} \in \mathbb{R}^N$ is *bandlimited* on a graph $\mathbf{A}$ with parameter $K \in \{0, 1, \cdots, N-1\}$, when the graph frequency components $\widehat{\mathbf{x}}$ satisfies

$$\widehat{x}_k \;=\; 0 \quad \text{for all} \quad k \geq K.$$

We denote this class of graph signals by $\mathrm{BL}_{\mathbf{A}}(K)$.

We can then show conditions on $\eta$ and $K$ such the class of bandlimited graph signals $\mathrm{BL}_{\mathbf{A}}(K)$ is a subset of the class of globally smooth signals $\mathrm{GS}_{\mathbf{A}}(\eta)$. Since smooth signals can contain arbitrary high frequency components, towards characterizing the spectral energy of smooth signals, we can define a generalization of the bandlimited class of signals that allows for a tail after the first $K$ frequency components.

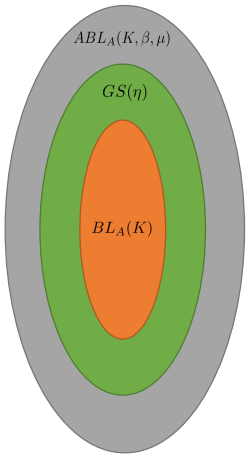



*Figure 2.2: Characterization of smooth and bandlimited graph signals*

**Definition 3.** A graph signal $\mathbf{x} \in \mathbb{R}^N$ is *approximately bandlimited* on a graph $\mathbf{A}$ with parameters $\beta \geq 1$ and $\mu \geq 0$ , when there exists a $K \in \{0, 1, \cdots, N-1\}$ such that its graph Fourier transform $\widehat{\mathbf{x}}$ satisfies

$$\sum_{k=K}^{N-1} (1 + k^{2\beta}) \widehat{x}_k^2 \leq \mu \left\|\mathbf{x}\right\|_2^2 . \tag{2.3}$$

Denote the class of graph signals by $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$.

We can then show conditions on $\eta$ such that the class of globally smooth graph signals $\mathrm{GS}_{\mathbf{A}}(\eta)$ is a subset of the class of the approximately bandlimited graph signals $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$ as seen in Figure 2.2. as seen in Figure 2.2. We note that we can show a similar characterization under the graph Laplacian.

## 2.3 Piecewise Smooth Signals

Piecewise smooth signals exhibit inhomegeneous smoothness over the graph, and are characterized by localized behavior and abrupt discontinuities. In practice, the graph signal may not be necessarily smooth over the entire graph, but only locally within different pieces of the graph. To model inhomogeneous levels of smoothness over a graph, we say that a graph signal $\boldsymbol{\beta}$ is piecewise constant over a graph $G$ if many of the differences $\beta_k - \beta_j$ are zero for $(j,k) \in \mathcal{E}$.

Let $\Delta \in \mathbb{R}^{M \times N}$ be the oriented incidence matrix of $G$, where each row corresponds to an edge. That is, if the edge $e_i \in \mathcal{E}$ connects the $j$th node to the $k$th node $(j < k)$, the $i$th row of $\Delta$ is then

$$\Delta_{i,\ell} = \begin{cases} -(\mathbf{A}_{j,k})\sqrt{|\,\mathbf{A}_{j,k}\,|}, & \ell = j; \\ (\mathbf{A}_{j,k})\sqrt{|\,\mathbf{A}_{j,k}\,|}, & \ell = k; \\ 0, & \text{otherwise}, \end{cases}$$

Let us assume for simplicity that the edge weights are all 1. We then note that

$$\|\Delta \mathbf{x}\|_1 = \sum_{(i,j)\in\mathcal{E}} |x_i - x_j|$$

As a result, $\Delta$ can be interpreted as a *graph difference operator*. Consequently, the difference signal $\boldsymbol{\Delta\beta}$ is sparse and $\|\boldsymbol{\Delta\beta}\|_0$ is small. We can generalize this graph difference operator to characterize piecewise smooth signals on a graph.[30,31,32]

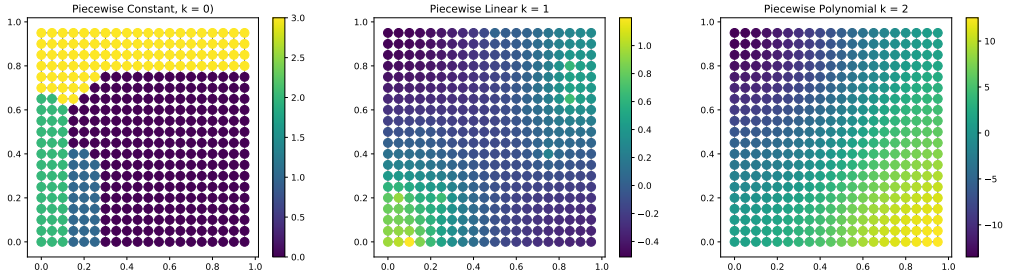



*Figure 2.3: From left to right, piecewise constant, linear and quadratic signals on a 2-dimensional grid graph*

We can generalize this characterization to piecewise linear $(k = 1)$ signals by defining a piecewise linear signal as a signal whose value at a node can be linearly interpolated from the values at neighboring nodes. It is easy to see that this is the same as requiring the second-order differences $\Delta^T \Delta \mathbf{x}$ to be sparse. Similarly, we say that a signal has a piecewise quadratic structure if the differences between the second-order differences defined for piecewise linear signals are mostly zero, that is, if $\Delta\Delta^T\Delta\mathbf{x}$ is sparse. Generalizing this, we can define the following recursive definition of the $k$-th order graph difference operator $\Delta^{(k+1)}$ such that a $k$-piecewise polynomial graph signal is

sparse in $\Delta^{(k+1)}\mathbf{x}$. Let $\Delta^{(1)} = \Delta$.

$$\Delta^{(k+1)} = \begin{cases} \Delta^{(1)T}\Delta^{(k)} & \text{odd } k \\ \Delta^{(1)}\Delta^{(k)} & \text{even } k \end{cases} \tag{2.4}$$

The signal $\mathbf{\Delta\beta} = [(\beta_k - \beta_j)]_{(j,k)\in\mathcal{E}}$ specifies the unweighted pairwise differences of the graph signal over each edge. As a result, $\mathbf{\Delta}$ can be interpreted as a *graph difference operator*. On the other hand , a signal is called smooth over a graph $G$ if $\|\mathbf{\Delta\beta}\|_2^2 = \sum_{(j,k)\in\mathcal{E}}(\beta_k - \beta_j)^2$ is small.

### 2.4 Localization and Uncertainty Principles

In classical signal processing, it is well known that signals cannot be simultaneously localized in both the time and frequency domains.[33,34] Some previous works extend this uncertainty principle to graphs by studying how well a graph signal exactly localize in both the graph vertex and graph spectrum domain.[35,36,37] For any unitary transform $\boldsymbol{Z} \in \mathbb{R}^{N \times N}$, we can show the following $\ell_0$-norm and $\ell_1$-norm based uncertainty principles that are demonstrably tight. They shed some insight into how properties of the representation basis drive how well we can localize in the signal (vertex) domain and the transform domain.

**Theorem 2.** (**Uncertainty principle I**)

$$\|\mathbf{x}\|_0 + \|\boldsymbol{Z}\mathbf{x}\|_0 \geq \frac{2}{\|\boldsymbol{Z}\|_\infty} \text{ and } \|\mathbf{x}\|_1 + \|\boldsymbol{Z}\mathbf{x}\|_1 \geq \frac{2}{\|\boldsymbol{Z}\|_\infty^{\frac{1}{2}}}$$

Particularly, this shows that how well a chosen representation basis can represent a signal is directly linked to the localization of the energy in the column vectors of $\boldsymbol{Z}$. As we shall study later, for the representations we study, this energy localization is inextricably linked to structural properties of the graph. ■

We now look at a concentration-based uncertainty principle with respect to the graph Fourier transform $\mathbf{U}$. For a vertex set $\Gamma \subseteq [N]$, we can define a vertex-projection operator $\boldsymbol{P}_\Gamma$ that project signals onto the set of signals that are non-zero over the vertex set $\Gamma$. We can then define *$\epsilon$-vertex concentrated* signals over $\Gamma$ such that $\|\mathbf{x} - \boldsymbol{P}_\Gamma \mathbf{x}\|_2^2 \leq \epsilon$. Similarly, for a frequency set $\Omega \subseteq [N]$, we can define a spectrum-projection operator $\boldsymbol{Q}_\Omega$ that projects signals onto the set of bandlimited signals over the frequency set $\Omega$. We can then define *$\epsilon$-spectrum concentrated* signals over $\Omega$ such that $\|\mathbf{x} - \boldsymbol{Q}_\Omega \mathbf{x}\|_2^2 \leq \epsilon$.

**Theorem 3.** (**Uncertainty principle II**) Let a unit norm signal $\mathbf{x}$ supported on an undirected graph be $\epsilon_\Gamma$-vertex concentrated and $\epsilon_\Omega$-spectrum concentrated at the same time. Then,

$$|\Gamma| \cdot |\Omega| \geq \frac{(1 - (\epsilon_\Omega + \epsilon_\Gamma))^2}{\|\mathbf{U}_\Omega\|_\infty^2}.$$

where $\mathbf{U}_\Omega$ is the submatrix of the graph Fourier transform matrix indexed by the columns of $\Omega$. ■

In classical signal processing, $\|\mathbf{U}_\Omega\|_\infty = 1/\sqrt{N}$; the lower bound is $O(N)$ and simultaneous localization in the vertex and frequency space is not viable. However, for complex and irregular graphs, the energy of a graph Fourier basis vector may be concentrated on a few elements, that is, $\|\mathbf{U}_\Omega\|_\infty = O(1)$ and simultaneous localization is possible. Empirically, we can observe that these bounds are tight.

Wavelets and Smoothness: Given a graph signal, let us consider its graph Fourier decomposition $\mathbf{x} = \mathbf{U}^T \boldsymbol{\alpha} = \mathbf{V}\boldsymbol{\alpha}$ and its wavelet decomposition $\mathbf{x} = \boldsymbol{W}^T \boldsymbol{\beta}$. Using the uncertainty principle for pairs of bases[33], we have that

$$\|\boldsymbol{\alpha}\|_0 \cdot \|\boldsymbol{\beta}\|_0 \geq \frac{1}{\mu^2} \text{ and } \|\boldsymbol{\alpha}\|_0 + \|\boldsymbol{\beta}\|_0 \geq \frac{2}{\mu}$$

where $\mu = \mu(\mathbf{U}^T, \boldsymbol{W}^T)$ is the mutual coherence between the graph Fourier transform basis $\mathbf{U}$ and the wavelet basis $\boldsymbol{W}$. In general, they are quite incoherent and a signal cannot have sparse representations in both these bases since for a smooth signal, we can show that the wavelet coefficients decay at a certain rate. As a result, this helps motivate and necessitate the development of distinct representations and algorithms for the processing of smooth and the processing of piecewise smooth signals.

While in time and images, Fourier and Haar wavelet bases are generally exhibit bad incoherence (large mutual coherence) because low-order wavelets and low-order frequencies are correlated. However, we can demonstrate that on irregular graphs this is not the case and the mutual coherence is often smaller. This is because vertex-localized wavelets at different scalings can be smooth which is a direct implication of Theorem 3.

# 3

# Sampling and Recovering Smooth Graph Signals

In this chapter, we study sampling smooth signals on graphs on graphs. These sampling algorithms can be considered as extensions of the Nyquist sampling for regular domains to irregular domains. In traditional machine learning approaches to classification, one uses only a labeled set to train the classifier. Labeled instances however are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile, unlabeled data is typically relatively cheap to collect, but there has been few ways to use them. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers. Graphs are a natural way to represent such datasets. That is, each vertex represents one data point to which a label is associated and a graph can be formed by connecting vertices with weights corresponding to the affinity or distance between the data points in some feature space. It is then natural to assume that the *label signal* is smooth on the graph. Since samples are often sparse or expensive, designing efficient sampling and reconstruction tools for semi-supervised classification and active learning is notably valuable.

We first present a sampling theory for bandlimited graph signals that is analogous to downsampling discrete-time signals such that we can recover the signal perfectly. We extend this framework to show we can sample efficiently on product graphs by using a structured sampling procedure that allows us significant gains in computational complexity. We then study minimax lower bounds for sampling smooth graph signals under both passive and active samplings, and show that active sampling does not fundamentally outperform passive sampling. Further, we present optimal sampling and reconstruction algorithms with respect to the lower bounds for passive sampling and discuss how the underlying graph structure drives their performance. We then discuss recovering smooth graph signals from noisy or corrupted measurements by formulating an optimization problem that minimizes the variation of the signal over the graph. While this has been well studied in previous work, we present a framework for recovering smooth signals on product graphs that exploits the low rank structure of these signals by modeling the signal as a multi-dimensional tensor. Finally, we present applications that showcase our algorithms and strategies.

## 3.1 Sampling Theory for Bandlimited Graph Signals

In this section, we consider the classical signal processing task of sampling theory within the framework of $\text{DSP}_\text{G}$. Sampling theory is a key topic in signal processing[38,39]. As the bridge connecting sequences and functions, classical sampling theory shows that a bandlimited function can be perfectly recovered from its sampled sequence if the sampling rate is high enough[40]. More generally, we can treat any decrease in dimension via a linear operator as sampling, and, conversely, any increase in dimension via a linear operator as interpolation[38,41]. Formulating a sampling theory in this context is equivalent to moving between higher- and lower-dimensional spaces while ensuring perfect recovery.

A sampling theory for graphs has interesting implications and applications. For example, given a graph representing friendship connectivity in Facebook, we can just sample a small fraction of users and query their hobbies. We then can recover all users' hobbies. The task of sampling on graphs is, however, not well understood[42,10]. It is challenging because graph signals lie on complex, irregular structure, where many classical concepts are ill-posed, such as downsampling[43]. It is even more challenging to find a graph structure that is associated with the sampled signal coefficients. For example, in the Facebook example, we sample a small fraction of users. An associated

graph structure would allow us to infer new connectivity between those sampled users, even when they are not directly connected in the original graph.

Some previous work on sampling theory[9,42] considers graph signals that are uniquely sampled onto some given subset of nodes. This approach is not consistent with classical sampling theory and applies to undirected graphs only. It also does not explain how a graph structure supports these sampled coefficients.

The assumption that graph signals vary slowly or are smooth over the graph is a natural one to make. Many real world graph signals like sensor network data and biological network data are smooth, or exhibit bandlimited behavior, or have known limited support with respect to the graph Fourier transform. For example, in the context of semi-supervised classification on graphs, each vertex represents one data point to which a label is associated and a graph can be formed by connecting vertices with weights corresponding to the affinity or distance between the data points in some feature space. It is then natural to assume that the *label signal* has slow variation or is *smooth* on the graph and consequently approximately bandlimited. Since labeled instances are rare or expensive to collect, devising efficient yet frugal sampling algorithms on large complex graphs is of significant interest. Here we propose a sampling theory for signals that are supported on either directed or undirected graphs. Perfect recovery is possible for graph signals bandlimited under the graph Fourier transform. We also show that the sampled signal coefficients form a new graph signal whose corresponding graph structure is constructed from the original graph structure. The proposed sampling theory follows Chapter 5 from[38] and is consistent with classical sampling theory. We further establish the connection to the theories of frames with maximal robustness to erasures and compressed sensing, show a principle to choose the optimal sampling operator, and show how random sampling works on circulant graphs and Erdős-Rényi graphs. To handle full-band graphs signals, we propose graph filter banks to force graphs signals to be bandlimited. Finally, we validate the proposed sampling theory on three simulated datasets of Erdős-Rényi graphs, small-world graphs, scale-free graphs, and a real-world dataset of online blogs. We show that for each case, the proposed sampling theory achieves perfect recovery with high probabilities.

**Contributions.** The contributions in this section are

- a novel sampling theory for graph signals, which follows the same paradigm as classical sampling theory;
- a novel approach to construct a graph structure that supports the sampled signal coefficients;
- a novel principle to choose the optimal sampling operator;
- a novel approach to construct graph filter banks to analyze full-band graph signal.

**Outline.** Section 3.1.1 describes the proposed sampling theory for graph signals, and the proposed construction of graph structures for the sampled signal coefficients. The proposed sampling theory is evaluated in in Section 3.5. We then conclude this section and provide pointers to future directions.

In general, $\mathbf{V}$ may not be orthonormal; to restrict its behavior, we assume that

$$\alpha_1 \|\mathbf{x}\|_2^2 \leq \|\mathbf{V}\,\mathbf{x}\|^2 \leq \alpha_2 \|\mathbf{x}\|_2^2, \quad \text{for all } \mathbf{x} \in \mathbb{R}^N, \tag{3.1}$$

where $\alpha_1, \alpha_2 > 0$, that is, $\mathbf{V}$ is a Riesz basis with stability constants $\alpha_1, \alpha_2$[38]. The eigenvalues $\lambda_0, \ldots \lambda_{N-1}$ of $\mathbf{A}$, represent frequencies on the graph[18].

| Symbol | Description | Dimension |
|---|---|---|
| $\mathbf{A}$ | graph shift | $N \times N$ |
| $\mathbf{V}$ | inverse graph Fourier transform matrix | $N \times N$ |
| $\Psi$ | sampling operator | $M \times N$ |
| $\Phi$ | interpolation operator | $N \times M$ |
| $\mathbf{x}$ | graph signal | $N$ |
| $\widehat{\mathbf{x}}$ | graph signal in the frequency domain | $N$ |
| $\mathcal{M}$ | sampled indices | |
| $\mathbf{x}_{\mathcal{M}}$ | sampled signal coeffcients of $\mathbf{x}$ | $M$ |
| $\widehat{\mathbf{x}}_{(K)}$ | first $K$ coeffcients of $\widehat{\mathbf{x}}$ | $K$ |
| $\widehat{\mathbf{x}}_{(-K)}$ | except first $K$ coeffcients of $\widehat{\mathbf{x}}$ | $K$ |
| $\mathbf{V}_{(K)}$ | first $K$ columns of $\mathbf{V}$ | $N \times K$ |
| $\mathbf{V}_{(-K)}$ | except first $K$ columns of $\mathbf{V}$ | $N \times (N-K)$ |
| $\mathbf{V}^{-1}_{(K)}$ | first $K$ rows of $\mathbf{V}^{-1}$ | $K \times N$ |
| $\mathbf{V}^{-1}_{(-K)}$ | except first $K$ rows of $\mathbf{V}^{-1}$ | $(N-K) \times N$ |

*Table 3.1: Key notation used in this chapter*

### 3.1.1 Sampling on Graphs

In this section, we propose a sampling theory for graph signals. We show that perfect recovery is possible for graph signals bandlimited under the graph Fourier transform, and a new graph shift for the sampled signal coefficients is constructed from the original graph shift. A toy example is shown to illustrate the proposed sampling theory. We further analyze the proposed sampling theory by showing the relations to previous theories, a principle to choose the optimal sampling operator, how random sampling works, and a graph filter bank to handle full-band graph signals.

Suppose that we want to sample $M$ coefficients in a graph signal $\mathbf{x} \in \mathbb{R}^N$ to produce a sampled part $\mathbf{x}_{\mathcal{M}} \in \mathbb{R}^M$ $(M < N)$, where $\mathcal{M}$ denotes the sequence of *sampled* indices, $\mathcal{M} \subset \{0, 1, \cdots, N-1\}$ and $|\mathcal{M}| = M$. We then interpolate $\mathbf{x}_{\mathcal{M}}$ to get $\mathbf{x}' \in \mathbb{R}^N$, which recovers $\mathbf{x}$ either exactly or approximately. The sampling operator $\Psi$ is a linear mapping from $\mathbb{R}^N$ to $\mathbb{R}^M$, defined as

$$\Psi_{i,j} = \begin{cases} 1, & j = \mathcal{M}_i; \\ 0, & \text{otherwise}, \end{cases} \tag{3.2}$$

and the interpolation operator $\Phi$ is a linear mapping from $\mathbb{R}^M$ to $\mathbb{R}^N$ (see Figure 3.1),

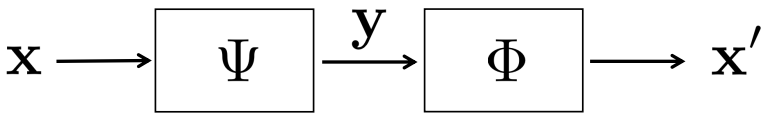



*Figure 3.1: Sampling followed by interpolation*

$$\begin{aligned} \text{sampling}: &\quad \mathbf{x}_{\mathcal{M}} = \Psi\mathbf{x} \in \mathbb{R}^M, \\ \text{interpolation}: &\quad \mathbf{x}' = \Phi\mathbf{x}_{\mathcal{M}} = \Phi\Psi\mathbf{x} \in \mathbb{R}^N. \end{aligned}$$

Perfect recovery happens for all $\mathbf{x}$ when $\Phi\Psi$ is the identity matrix. This is not possible in general because $\text{rank}(\Phi\Psi) \leq M < N$.

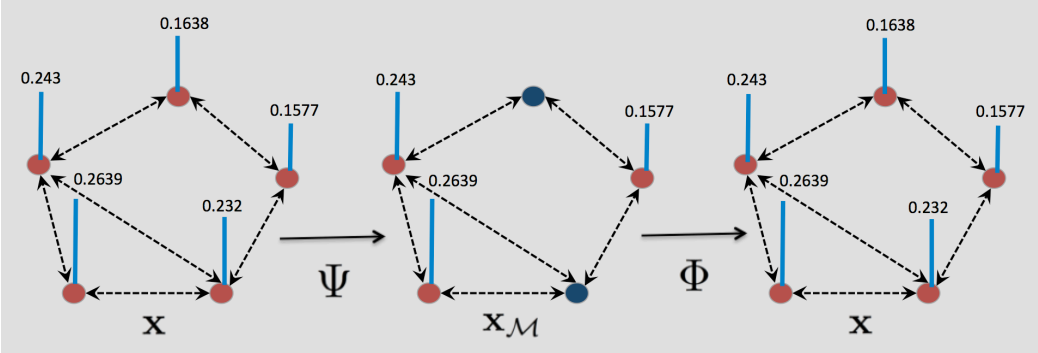



*Figure 3.2: Sampling followed by interpolation. The arrows indicate different edge weights for two nodes.*

We consider bandlimited graph signals here, where perfect recovery is possible.

**Definition 4.** A graph signal is called *bandlimited* when there exists $K \in \{0, 1, \cdots, N-1\}$ such that its graph Fourier transform $\widehat{\mathbf{x}}$ satisfies

$$\widehat{x}_i = 0 \quad \text{for all} \quad i \geq K.$$

The smallest such $K$ is called the *bandwidth* of $\mathbf{x}$. A graph signal that is not bandlimited is called a *full-band graph signal*.

**Definition 5.** The set of graph signals in $\mathbb{R}^N$ with bandwidth of at most $K$ is a closed subspace denoted $\text{BL}_K(\mathbf{V}^{-1})$.

Following Theorem 5.2 in[38], we obtain the following result.

**Theorem 4.** Let $\mathbf{V}_{(K)}$ be the first $K$ columns of $\mathbf{V}$ and let the sampling operator $\Psi$ satisfy

$$\text{rank}(\Psi\,\mathbf{V}_{(K)}) = K.$$

The interpolation operator $\Phi = \mathbf{V}_{(K)}\,\mathbf{U}$, with $\mathbf{U}\,\Psi\,\mathbf{V}_{(K)}$ a $K \times K$ identity matrix, where $\mathbf{U} \in \mathbb{R}^{K\times M}$, achieves perfect recovery:

$$\mathbf{x} = \Phi\Psi\mathbf{x}, \quad \text{for any } \mathbf{x} \in \text{BL}_K(\mathbf{V}^{-1}).$$

Since we do not specify the ordering of frequencies, we can reorder the eigenvalues and permute the corresponding eigenvectors in the graph Fourier transform matrix to choose any band in the graph Fourier domain. The bandlimited restriction is equivalent to requiring limited support in

the graph Fourier domain. Theorem 4 is thus applicable for all graph signals that have limited support in the graph Fourier domain.

The sample size $M$ should be no smaller than the bandwidth $K$. When $M < K$, $\text{rank}(\mathbf{U}\,\Psi\,\mathbf{V}_{(K)}) \leq \text{rank}(\mathbf{U}) \leq M < K$, $\mathbf{U}\,\Psi\,\mathbf{V}_{(K)}$ can never be an identity matrix. Since $\mathbf{U}\,\Psi\,\mathbf{V}_{(K)}$ an identity matrix, $\mathbf{U}$ is the inverse of $\Psi\,\mathbf{V}_{(K)}$ when $M = K$; it is a pseudo-inverse of $\Psi\,\mathbf{V}_{(K)}$ when $M > K$, where the redundancy exists. We discuss the redundancy in Section 3.2.4. For simplicity, we only consider the case where the sample size and the bandwidth are the same, i,e., $M = K$, and $\mathbf{U}$ is invertible. When $M > K$, we simply select $K$ from $M$ sampled signal coefficient to ensure that the sample size and the bandwidth are the same.

From Theorem 4, we see that an arbitrary sampling operator may not lead to perfect recovery even for bandlimited graph signals. The sampling operator should select at least one set of $K$ linearly-independent rows in $\mathbf{V}_{(K)}$. Since $\mathbf{V}$ is invertible, the column vectors in $\mathbf{V}$ are linearly independent and $\text{rank}(\mathbf{V}_{(K)}) = K$ always holds. In other words, at least one set of $K$ linearly-independent rows in $\mathbf{V}_{(K)}$ always exists. When a sampling operator $\Psi$ satisfies $\text{rank}(\Psi\,\mathbf{V}_{(K)}) = K$, we call it a *qualified sampling operator*. Since the graph shift $\mathbf{A}$ is given, one can find such a set independently of the graph signal. Given such a set, Theorem 4 guarantees perfect recovery of bandlimited graph signals without any approximation as in [42] and any probability constraints as in compressed sensing [44]. To find linearly-independent rows in a matrix, fast algorithms exist, such as QR decomposition; see [45,38].

In the previous part, we show that perfect recovery is possible when the graph signals are bandlimited. In the following content, we show that the sampled signal coefficients form a new graph signal, whose corresponding new graph shift is constructed from the original graph shift.

Suppose a graph signal with bandwidth $K$, we express it as

$$\mathbf{x} \;=\; \mathbf{V}\,\widehat{\mathbf{x}} = \mathbf{V}_{(K)}\,\widehat{\mathbf{x}}_{(K)}, \tag{3.3}$$

where $\widehat{\mathbf{x}}_{(K)} \in \mathbb{R}^K$ contains the first $K$ signal coefficients in $\widehat{\mathbf{x}}$. Let $\Psi$ be a sampling operator that samples $M$ coefficients in $\mathbf{x}$ to produce $\mathbf{x}_{\mathcal{M}}$, $\Phi = \mathbf{V}_{(K)}\,\mathbf{U}$ be an interpolating operator, and $\Psi$ be a sampling operator, which satisfies (3.3) in Theorem 4 to perfectly recover $\mathbf{x}$ from $\mathbf{x}_{\mathcal{M}}$. We express the graph signal as

$$\mathbf{x} \;=\; \Phi\Psi\mathbf{x} = \Phi\mathbf{x}_{\mathcal{M}} = \mathbf{V}_{(K)}\,\mathbf{U}\,\mathbf{x}_{\mathcal{M}}. \tag{3.4}$$

Since (3.3) and (3.4) hold for all $\mathbf{x} \in \text{BL}_K(\mathbf{V}^{-1})$, we thus get

$$\widehat{\mathbf{x}}_{(K)} \;=\; \mathbf{U}\,\mathbf{x}_{\mathcal{M}}.$$

Reminding ourselves from Theorem 4 that $\mathbf{U}$ is the invertible when $M = K$, we then get

$$\mathbf{x}_{\mathcal{M}} \;=\; \mathbf{U}^{-1}\,\mathbf{U}\,\mathbf{x}_{\mathcal{M}} = \mathbf{U}^{-1}\,\widehat{\mathbf{x}}_{(K)}.$$

The sampled signal coefficients $\mathbf{x}_{\mathcal{M}}$ can be constructed from the frequency content $\widehat{\mathbf{x}}_{(K)}$ through $\mathbf{U}^{-1}$. In addition, the frequency content $\widehat{\mathbf{x}}_{(K)}$ can be constructed from the sampled signal coefficients $\mathbf{x}_{\mathcal{M}}$ through $\mathbf{U}$, which implies that $\mathbf{x}_{\mathcal{M}}$ is a graph signal associated with the graph Fourier transform matrix $\mathbf{U}$. Since we only use the first $K$ frequencies, the graph shift that is associated

with $\mathbf{x}_{\mathcal{M}}$ is then

$$\mathbf{A}_{\mathcal{M}} = \mathbf{U}^{-1}\,\Lambda_{(K)}\,\mathbf{U} \in \mathbb{R}^{K\times K},$$

where $\Lambda_{(K)} \in \mathbb{R}^{K\times K}$ is a diagonal matrix that samples the first $K$ eigenvalues of $\Lambda$. The previous discussion can be summarized as follows:

**Theorem 5.** Let $\Psi$ be the sampling operator to sample $K$ coefficients in $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V}^{-1})$ to produce $\mathbf{x}_{\mathcal{M}} \in \mathbb{R}^K$ and satisfy

$$\mathrm{rank}(\Psi\,\mathbf{V}_{(K)}) = K.$$

Let $\mathbf{U}$ be $(\Psi\,\mathbf{V}_{(K)})^{-1}$. Then, $\mathbf{x}_{\mathcal{M}}$ is a graph signal associated with the graph shift

$$\mathbf{A}_{\mathcal{M}} = \mathbf{U}^{+}\,\Lambda_{(K)}\,\mathbf{U} \in \mathbb{R}^{K\times K}. \tag{3.5}$$

The graph Fourier transform of $\mathbf{x}_{\mathcal{M}}$ is

$$\widehat{\mathbf{x}}_{\mathcal{M}} \;=\; \mathbf{U}\,\mathbf{x}_{\mathcal{M}} \in \mathbb{R}^K.$$

The inverse graph Fourier transform is

$$\mathbf{x}_{\mathcal{M}} = \mathbf{U}^{-1}\,\widehat{\mathbf{x}}_{\mathcal{M}} \in \mathbb{R}^K.$$

From Theorem 5, we see that the graph shift $\mathbf{A}_{\mathcal{M}}$ is constructed by sampling the rows of the eigenvector matrix and sampling the first $K$ eigenvalues of the original graph shift $\mathbf{A}$. We simply say that $\mathbf{A}_{\mathcal{M}}$ is "sampled" from $\mathbf{A}$, preserving certain information in the graph Fourier domain.

Since the bandwidth of $\mathbf{x}$ is $K$, the first $K$ coefficients in the frequency domain are $\widehat{\mathbf{x}}_{(K)} = \widehat{\mathbf{x}}_{\mathcal{M}}$, and the other $N-K$ coefficients are $\widehat{\mathbf{x}}_{(-K)} = 0$; in other words, the frequency contents are equivalent for the original graph signal $\mathbf{x}$ and the sampled graph signal $\mathbf{x}_{\mathcal{M}}$ after performing their corresponding graph Fourier transforms.

Similarly to Theorem 4, by reordering the eigenvalues and permuting the corresponding eigenvectors in the graph Fourier transform matrix, Theorem 5 is applicable for all graph signals that have limited supports in the graph Fourier domain, and the sampled graph shift $\mathbf{A}_{\mathcal{M}}$ supports the sampled signal coefficients, preserving the corresponding frequency content.

### 3.1.2 Finite Discrete-time Cases

We call the graph that supports a finite discrete-time signal as the *finite discrete-time graph*, which is represented by the cyclic permutation matrix[38,21],

$$\begin{aligned}\mathbf{A} \;&=\; \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix} \\ &=\; \mathbf{V}\,\Lambda\,\mathbf{V}^{-1},\end{aligned} \tag{3.6}$$

where the eigenvector matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{N-1} \end{bmatrix} = \left[ \tfrac{1}{\sqrt{N}} (w^{jk})^* \right]_{j,k=0,\cdots N-1}, \tag{3.7}$$

is the Hermitian transpose of the $N$-point discrete Fourier transform matrix, i.e., $\mathbf{V} = \mathrm{F}^*$, where $*$ is the Hermitian transpose, and $\mathbf{V}^{-1}$ is the $N$-point discrete Fourier transform matrix (F), i.e., $\mathbf{V}^{-1} = \mathrm{F}$, and the eigenvalue matrix is

$$\Lambda = \mathrm{diag} \begin{bmatrix} \lambda_0 & \lambda_1 & \cdots & \lambda_{N-1} \end{bmatrix}, \tag{3.8}$$

where $\lambda_i = w^i$, $w = e^{-2\pi j/N}$. We see that Definitions 6, 7 and Theorem 4 are immediately applicable to finite discrete-time signals.

**Definition 6.** A discrete-time signal is called *bandlimited* when there exists $K \in \{0, 1, \cdots, N-1\}$ such that its discrete Fourier transform $\widehat{\mathbf{x}}$ satisfies

$$\widehat{x}_i = 0 \quad \text{for all} \quad i \geq K.$$

The smallest such $K$ is called the *bandwidth* of $\mathbf{x}$. A discrete-time signal that is not bandlimited is called a *full-band discrete-time signal*.

**Definition 7.** The set of discrete-time signals in $\mathbb{R}^N$ with bandwidth of at most $K$ is a closed subspace denoted $\mathrm{BL}_K(\mathrm{F})$, with F as the discrete Fourier transform matrix.

With this definition of the discrete Fourier transform matrix, the highest frequency is in the middle of the spectrum (although this is just a matter of ordering). From Definitions 6 and 7, we can permute the rows in the discrete Fourier transform matrix to choose any frequency band. Since the discrete Fourier transform matrix is a Vandermonde matrix, any $K$ rows of $\mathrm{F}^*_{(K)}$ are independent[45,38]; in other words, $\mathrm{rank}(\Psi\, \mathrm{F}^*_{(K)}) = K$ always hold when $M \geq K$. We apply now Theorem 4 to obtain the following result.

**Theorem 6.** Let $\mathrm{F}^*_{(K)}$ be the first $K$ columns of $\mathrm{F}^*$ and let the sampling operator $\Psi$ satisfy the sampling number $M$ is no less than the bandwidth $K$. The interpolation operator $\Phi = \mathrm{F}^*_{(K)}\, \mathbf{U}$, with $\mathbf{U}\, \Psi\, \mathrm{F}^*_{(K)}$ a $K \times K$ identity matrix, achieves perfect recovery,

$$\mathbf{x} = \Phi\Psi\mathbf{x}, \quad \text{for any } \mathbf{x} \in \mathrm{BL}_K(\mathrm{F}).$$

From Theorem 6, we can perfectly recover a discrete-time signal when it is bandlimited.

Similarly to Theorem 5, we can show that a new graph shift can be constructed from the finite discrete-time graph. Multiple sampling mechanisms can be done to sample a new graph shift, to obtain an intuitive one, we do as follows. Suppose $\mathbf{x} \in \mathbb{R}^N$ is a finite discrete-time signal, where $N$ is even, and the corresponding finite discrete-time graph is represented by the cyclic permutation matrix, $\mathbf{A}$, as in (3.6). We reorder the frequencies in (3.8), by putting the frequencies with even indices first as

$$\widetilde{\Lambda} = \mathrm{diag} \begin{bmatrix} \lambda_0 & \lambda_2 & \cdots & \lambda_{N-2} & \lambda_1 & \lambda_3 & \cdots & \lambda_{N-1} \end{bmatrix},$$

Correspondingly, we reorder the columns of $\mathbf{V}$ in (3.7) by putting the columns with even indices

first as

$$\widetilde{\mathbf{V}} = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_2 & \cdots & \mathbf{v}_{N-2} & \mathbf{v}_1 & \mathbf{v}_3 & \cdots & \mathbf{v}_{N-1} \end{bmatrix}.$$

One can check that $\widetilde{\mathbf{V}}\widetilde{\Lambda}\widetilde{\mathbf{V}}^{-1}$ is still the same cyclic permutation matrix, where $\widetilde{\mathbf{V}}^{-1}$ is the inverse of $\widetilde{\mathbf{V}}$. Suppose we want to preserve the first $N/2$ frequency contents in $\widetilde{\Lambda}$, the sampled frequencies are then

$$\widetilde{\Lambda}_{(N/2)} = \text{diag}\begin{bmatrix} \lambda_0 & \lambda_2 & \cdots & \lambda_{N-2} \end{bmatrix}.$$

Let a sampling operator $\Psi$ choose the first $N/2$ rows in $\widetilde{\mathbf{V}}_{(N/2)}$,

$$\Psi\widetilde{\mathbf{V}}_{(N/2)} = \left[\tfrac{1}{\sqrt{N}}(w^{2jk})^*\right]_{j,k=0,\cdots N/2-1},$$

which is the Hermitian transpose of the $N/2$ discrete Fourier transform and satisfies $\text{rank}(\Psi\widetilde{\mathbf{V}}_{(N/2)}) = N/2$ in Theorem 5. The sampled graph Fourier transform matrix $\mathbf{U} = (\Psi\widetilde{\mathbf{V}}_{(N/2)}))^{-1}$ is the $N/2$ discrete Fourier transform. The sampled graph shift is then constructed as

$$\mathbf{A}_{\mathcal{M}} = \mathbf{U}^{-1}\,\widetilde{\Lambda}_{(N/2)}\,\mathbf{U},$$

which is exactly the $N/2 \times N/2$ cyclic permutation matrix. Hence, we have shown that by choosing an appropriate sampling mechanism, a smaller finite discrete-time graph is obtained from a larger finite discrete-time graph by using Theorem 5. We note that using a different ordering or sampling operator, would result in a graph shift that can be different and non-intuitive. This is however a matter of choosing different frequency contents.

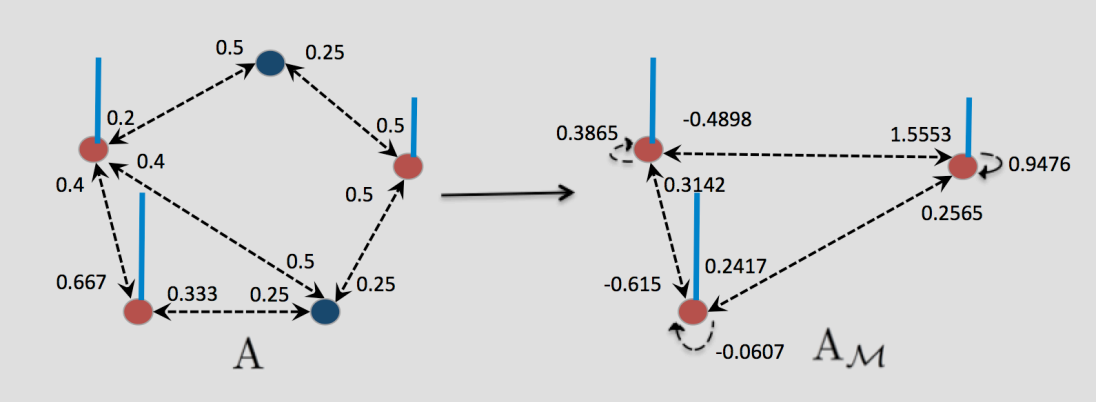



*Figure 3.3: Sampling a graph.*

### 3.1.3 Toy examples

We consider a five-node directed graph with graph shift

$$\mathbf{A} = \begin{bmatrix} 0 & 0.4 & 0.4 & 0 & 0.2 \\ 0.667 & 0 & 0.333 & 0 & 0 \\ 0.5 & 0.25 & 0 & 0.25 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0.5 & 0 \end{bmatrix}.$$

The corresponding inverse graph Fourier transform matrix is

$$\mathbf{V} = \begin{bmatrix} 0.4472 & 0.1936 & 0.253 & 0.3532 & -0.4026 \\ 0.4472 & 0.4034 & 0.1604 & -0.7446 & 0.1782 \\ 0.4472 & 0.0842 & -0.5618 & 0.2862 & 0.362 \\ 0.4472 & -0.6596 & -0.4053 & -0.4706 & -0.5733 \\ 0.4472 & -0.598 & 0.656 & 0.132 & 0.5886 \end{bmatrix},$$

and the frequencies are

$$\Lambda = \text{diag}\begin{bmatrix} 1 & 0.3895 & -0.1161 & -0.444 & -0.829 \end{bmatrix}.$$

We generate a bandlimited graph signal $\mathbf{x} \in \text{BL}_3(\mathbf{V}^{-1})$ as

$$\mathbf{x} = \begin{bmatrix} 0.242 & 0.2639 & 0.232 & 0.1577 & 0.1638 \end{bmatrix}^T.$$

We can check the first three columns of $\mathbf{V}$ to see that all sets of three rows are independent. According to the sampling theorem, we can recover $\mathbf{x}$ perfectly by sampling any three of its coefficients; for example, sample the first,second and the fourth coefficients. Then, $\mathcal{M} = \{1, 2, 4\}$, $\mathbf{x}_{\mathcal{M}} = \begin{bmatrix} 0.242 & 0.2639 & 0.1577 \end{bmatrix}^T$, and the sampling operator

$$\Psi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

We recover $\mathbf{x}$ by using the following interpolation operator (see Figure 3.2)

$$\Phi = \mathbf{V}_{(3)}(\Psi\,\mathbf{V}_{(3)})^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2.7043 & 2.8703 & 0.834 \\ 0 & 0 & 1 \\ 5.0363 & -3.9845 & -0.0518 \end{bmatrix}.$$

The inverse graph Fourier transform matrix is

$$\mathbf{U}^{-1} = \Psi\, \mathbf{V}_{(3)} = \begin{bmatrix} 0.4472 & 0.1936 & 0.253 \\ 0.4472 & 0.4034 & 0.1604 \\ 0.4472 & -0.6596 & -0.4053 \end{bmatrix},$$

and the sampled frequencies are

$$\Lambda_{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.3895 & 0 \\ 0 & 0 & -0.1161 \end{bmatrix},$$

The sampled graph shift is then constructed as

$$\mathbf{A}_{\mathcal{M}} = \mathbf{U}^{-1}\, \Lambda_{(3)}\, \mathbf{U} = \begin{bmatrix} 0.3865 & 0.3142 & 0.2417 \\ -0.615 & -0.0607 & -0.4898 \\ 1.5553 & 0.2565 & 0.9476 \end{bmatrix}.$$

We see that while the sampled graph shift contains self-loops and negative weights, which seems to be dissimilar to $\mathbf{A}$, $\mathbf{A}_{\mathcal{M}}$ perfectly preserves the frequency content of $\mathbf{A}$.

### 3.1.4 Discussions

We extend the proposed sampling theory, and discuss three topics: the relation to the previous work, how to choose a sampling operator, and how to handle full-band graph signals.

#### Relation to frames with maximal robustness to erasures

A frame is a generating system $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N\}$ of $\mathbb{R}^K$, where $N \geq K$, when there exist two constants $A > 0, B < \infty$, such that for all $x \in \mathbb{R}^N$,

$$A \left\|\mathbf{x}\right\|^2 \leq \sum_k |\mathbf{f}_k^T \mathbf{x}|^2 \leq B \left\|\mathbf{x}\right\|^2.$$

We represent the frame as an $N \times K$ matrix with rows $\mathbf{f}_k^T$:

$$\mathcal{F} = \begin{bmatrix} \mathbf{f}_1^T \\ \mathbf{f}_2^T \\ \vdots \\ \mathbf{f}_M^T \end{bmatrix}$$

The frame $\mathcal{F}$ is *maximally robust to erasures* when every $K \times K$ submatrix (obtained by deleting $N - K$ rows of $\mathcal{F}$) is invertible[46]. In[46], the authors show that a polynomial transform matrix is a frame with maximally robust to erasures; in[47], the authors show that many lapped orthogonal transforms and lapped tight frame transforms, are also maximally robust to erasures. It is clear that if the inverse graph Fourier transform matrix $\mathbf{V}$ is maximally robust to erasures, any sampling operator that samples at least $K$ signal coefficients guarantees perfect recovery; in

other words, when a graph Fourier transform matrix happens to be a polynomial transform matrix, sampling any $K$ signal coefficients leads to perfect recovery.

### Relation to compressed sensing

Compressed sensing is a sampling framework to recover sparse signals in a few measurements[48]. The theory asserts that a few samples guarantee to recover the original signals when signals and the sampling approaches are well-defined in some theoretical aspects. To be more specific, given the sampling operator $\Psi \in \mathbb{R}^{M \times N}, M << N$ and the sampled signal $\mathbf{x}_{\mathcal{M}} = \Psi \mathbf{x}$, a sparse signal $\mathbf{x} \in \mathbb{R}^N$, is recovered by solving

$$\min_{\mathbf{x}} \ ||\mathbf{x}||_0, \quad \text{subject to } \mathbf{x}_{\mathcal{M}} = \Psi \mathbf{x}. \tag{3.9}$$

Since the $l_0$ norm is not convex, the optimization is a non-deterministic polynomial-time hard problem. To obtain a computational efficient algorithm, the $l_1$ norm based algorithm, known as the basis pursuit or basis pursuit with denoising, recovers the sparse signal with small approximation error[49,50,51].

In the standard compressed sensing theory, the signals have to be sparse or approximately sparse to gurantee accurate recovery properties. In[52], the authors proposed a general way to perform compressed sensing with non-sparse signals using dictionaries. To be more specific, a general signal $\mathbf{x} \in \mathbb{R}^N$, is recovered by

$$\min_{\mathbf{x}} \ ||\mathcal{D}\mathbf{x}||_0, \quad \text{subject to } \mathbf{x}_{\mathcal{M}} = \Psi \mathbf{x}, \tag{3.10}$$

where $\mathcal{D}$ is a dictionary designed to make $\mathcal{D}\mathbf{x}$ sparse. When specifying $\mathbf{x}$ to be a graph signal, and $\mathcal{D}$ to be the graph Fourier transform of the graph on which the signal resides, $\mathcal{D}\mathbf{x}$ represents the frequency content of $\mathbf{x}$, which is sparse when $\mathbf{x}$ has a small bandwidth. (3.10) recovers a bandlimited graph signal from a few sampled signal coefficients via an optimization approach. The proposed sampling theory deals with the cases where the nonzero frequencies are known, and can be reordered to form a bandlimited graph signal. Compressed sensing deals with the cases where the nonzero frequencies are unknown, which is a more general and harder problem. By taking advantage of knowing the frequencies, the proposed sampling theory only needs $K$ sampled signal coefficients to achieve perfect recovery. On the other hand, compressed sensing needs more sampled signal coefficients to achieve an approximated recovery.

### Full-band graph signal

As shown in Theorem 4, the perfect recovery is achieved when graph signals are bandlimited. To handle full-band graph signals, we propose an approach based on graph filter banks.

Suppose $\mathbf{x}$ is a full-band graph signal, we express it as the addition of two bandlimited signals supported on the same graph, i. e., $\mathbf{x} = \mathbf{x}^l + \mathbf{x}^h$, where

$$\mathbf{x}^l = \mathbf{V} \begin{bmatrix} \mathrm{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}^{-1} \mathbf{x},$$

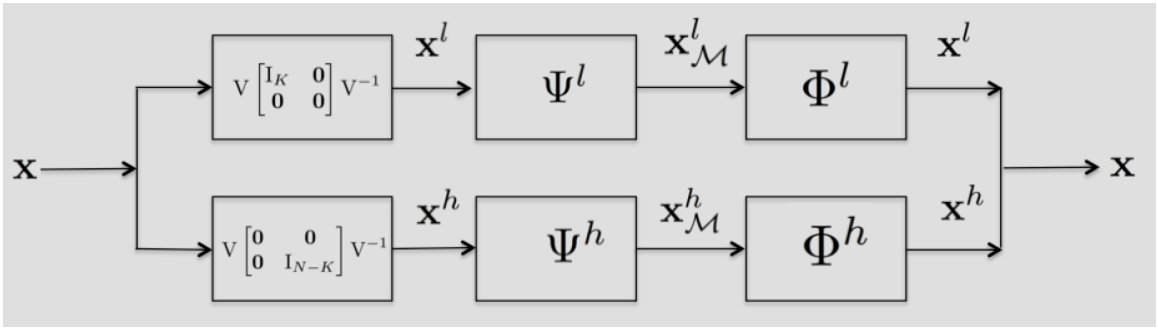



*Figure 3.4: Graph filter bank. We can split the graph signal to two bandlimited graph signals. In each channel, we perform sampling and interpolation, following the paradigm in Theorem 4. Finally, we add the results from both channels to obtain the original full-band graph signal.*

and

$$\mathbf{x}^h = \mathbf{V} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathrm{I}_{N-K} \end{bmatrix} \mathbf{V}^{-1} \mathbf{x}.$$

We see that $\mathbf{x}^l$ contains the first $K$ frequencies, $\mathbf{x}^h$ contains the other $N-K$ frequencies, and both of them are bandlimited. We do sampling and interpolation for $\mathbf{x}^l$ and $\mathbf{x}^h$ in two channels, respectively. We take the first channel as an example. Following the paradigm in Theorems 4 and 5, we use a feasible sampling operator $\Psi^l$ to sample $\mathbf{x}^l$, and obtain the sampled signal coefficients as $\mathbf{x}^l_{\mathcal{M}^l} = \Psi^l \mathbf{x}^l$, with the corresponding graph as $\mathbf{A}_{\mathcal{M}^l}$. We can recover $\mathbf{x}^l$ by using a interpolation operator $\Phi^l$ as $\mathbf{x}^l = \Phi^l \mathbf{x}^l_{\mathcal{M}^l}$. Finally, we add the results from both channels to obtain the original full-band graph signal (also illustrated in Figure 3.4). We see that the above idea can easily be generalized to multiple channels by splitting the original graphs signal into multiple bandlimited graph signals; instead of dealing with a huge graph, we work with multiple small graphs, which is easy for the storage and computation.

## 3.2 Sampling Theory on Product Graphs

Many examples of real-world graph-structured data are multi-modal in nature and importantly have an inherent structure. Product graphs are a graph model that composes graphs from smaller building blocks we call *graph atoms* and represent a concise way to model such data[53,54]. For example, product graph *composition* using a product operator is a natural way to model time-varying signals on a sensor network as shown in Figure 1(b). The graph signal formed by the measurements of all the sensors at all the time steps is supported by the graph that is the product of the sensor network graph and the time series graph. The $k^{th}$ measurement of the $n^{th}$ sensor is indexed by the $n^{th}$ node of the $k^{th}$ copy of the sensor network graph. In[53], a generative model that can effectively model the structure of many large real-world networks was presented by recursively applying the Kronecker product on a base graph that can be estimated efficiently. Consequently, constructing a framework for the efficient sampling and recovery on such product graphs is an important step for tasks such as graph signal recovery, compression, and semi-supervised learning on large-scale and multi-modal graphs.

Multiple types of graph products exist, that is, we can enforce connections across modes in dif-

ferent ways[55]. In the case of the Cartesian product as in Figure 1(b), the measurement of the $n^{th}$ sensor at the $k^{th}$ time step is related to not only to its neighboring sensors at the $k^{th}$ time step but also to its measurements at the $(k-1)^{th}$ and $(k+1)^{th}$ time steps respectively. Hence, constructing a framework for efficient sampling and recovery on such product graphs is an important step for tasks such as graph signal recovery, compression, and semi-supervised learning on large-scale and multi-modal graphs.

In[28,36], a sampling theory for bandlimited signals was presented that can be considered as an extension of Nyquist sampling for regular domains to irregular domains. In this work, extended this sampling theory to product graphs in[56] by showing how to efficiently sample and perfectly recover bandlimited signals on product graphs. Particularly, we show that we do not need to process the whole product graph $\mathbf{A}$ or compute its spectral decomposition which is of complexity $O(N^3)$ and is often computationally prohibitive for large graphs. While the sampling theory characterizes sampling sets that enable perfect recovery for bandlimited signals, it does not prescribe easily implementable, robust sampling strategies. Randomized sampling strategies[57,29], characterized by a probability distribution over the nodes, present a more flexible framework to sample nodes on a graph in the presence of noise that is also easily implementable. Hence, in our work, we further extend these randomized sampling strategies to product graphs by exploiting the structure of product graph. Particularly, as in the case of the sampling theory for product graphs[56], we only need to process the graph atoms the product graph is composed of.

### 3.2.1 Product Graphs

As before, we consider a graph $G = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V} = \{v_0, \ldots, v_{N-1}\}$ is the set of nodes and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the graph shift, or a weighted adjacency matrix. $\mathbf{A}$ Represents the connections of the graph $G$, which can be either directed or undirected.

Product graphs are graphs whose adjacency matrices are composed using the *product* (represented by the square symbol $\square$) of the adjacency matrices of smaller *graph atoms*. Consider two graphs $G_1 = (\mathcal{V}_1, \mathbf{A}_1)$ and $G_2 = (\mathcal{V}_2, \mathbf{A}_2)$ . The graph product of $G_1$ and $G_2$ is the graph $G = G_1 \square G_2 = (\mathcal{V}, \mathbf{A}_1 \square \mathbf{A}_2)$ where $|\mathcal{V}| = |\mathcal{V}_1| \cdot |\mathcal{V}_2|$. The set of nodes $\mathcal{V}$ is the Cartesian product of the sets $\mathcal{V}_1$ and $\mathcal{V}_2$. That is, a node $(u_1, u_2)$ is created for every $u_1 \in \mathcal{V}_1$ and $u_2 \in \mathcal{V}_2$.

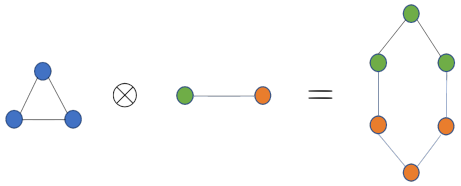


*Figure 3.5: Under the Kronecker product, $(u_1, u_2) \sim (v_1, v_2)$ in the product graph if $u_1 \sim v_1$ and $u_2 \sim v_2$*

Typically, we use one of the Kronecker graph product ($\otimes$, Figure 1(a)), the Cartesian graph product ($\oplus$, Figure 1(b) or the strong graph product ($\boxtimes$) which is a combination of both the Kronecker and Cartesian product to compose a product graphs. Since the product is associative, one can extend the above formulation to define product graphs constructed from multiple graph-atoms.

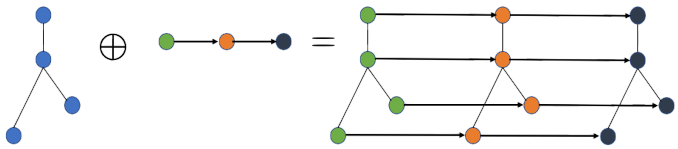


*Figure 3.6: Under the Cartesian product, $(u_1, u_2) \sim (v_1, v_2)$ in the product graph if $u_1 = v_1$ and $u_2 \sim v_2$ or $u_1 \sim v_1$ and $u_2 = v_2$*

Digital images reside on rectangular lattices that are Cartesian products of line graphs for rows and columns. We have already seen how the Cartesian product is a natural way to analyze time-varying signals on graphs by enforcing further connections both across the graph in question and the time graph. A social network with multiple communities can also be represented by the Kronecker graph product of the graph that represents a community structure and the graph that captures the interaction between neighbors. In the context of recommender engines where we have user ratings for different entities at different times, we can view this as a signal lying on the Kronecker product of three graphs, the graph relating the different users, the graph relating the different entities, and the time graph. In the context of multivariate signals on a given graph $\mathbf{A}$ where each node has a multidimensional vector associated with it, we can view this as a signal lying on the product graph constructed by the composition of $\mathbf{A}$ and the covariance matrix of the multivariate data $\mathbf{\Sigma}$.

In the following exposition, for clarity and brevity, we only consider the Kronecker product. However, the results and theorems either hold or can easily be extended to both Cartesian and strong products. We also only consider the graph Fourier transform defined for the graph shift matrix $\mathbf{A}$ but these results can also be extended for when the graph Fourier transform is defined for the graph Laplacian.

We consider a product graph $G = (\mathcal{V}, \mathbf{A}), |\mathcal{V}| = N$, that is constructed from $J$ graph atoms $G_1, G_2, \cdots G_J$, where $G_j = (\mathcal{V}_j, \mathbf{A}^j), |\mathcal{V}_j| = N_j$, using the Kronecker product where $\prod_{j=1}^{J} N_j = N$. We can write the resulting graph shift matrix of the product graph as

$$\mathbf{A} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \otimes \cdots \otimes \mathbf{A}^{(J)} = \otimes_{j=1}^{J} \mathbf{A}^{(j)} \tag{3.11}$$

We can then write the spectral decomposition of the product graph shift $\mathbf{A}$ as

$$\begin{aligned} & \mathbf{A} = \mathbf{V}\,\Lambda\,\mathbf{U} \qquad (3.12) \\ \text{where } & \mathbf{V} = \mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} \otimes \cdots \otimes \mathbf{V}^{(J)} = \otimes_{j=1}^{J} \mathbf{V}^{(j)} \\ & \Lambda = \Lambda^{(1)} \otimes \Lambda^{(2)} \otimes \cdots \otimes \Lambda^{(J)} = \otimes_{j=1}^{J} \Lambda^{(j)} \\ & \mathbf{U} = \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(J)} = \otimes_{j=1}^{J} \mathbf{U}^{(j)} = \mathbf{V}^{-1} \end{aligned}$$

For a given graph atom, $G_j$, the columns of $\mathbf{V}^{(j)}$ and their corresponding frequencies are pairs of the form $(\mathbf{v}_{i_{(j)}}^{(j)}, \lambda_{i_{(j)}}^{(j)})$. Here, $i_{(j)}$ is an index for the nodes in $G_j$ that varies from $(1, 2, \cdots N_j)$ where $N_j = |\mathcal{V}_j|$, the number of nodes in $G_j$.

As a result, under the Kronecker Product, each of the $N$ basis vectors in $\mathbf{V}$ have the form

$$\mathbf{v}^{(1)}_{i_{(1)}} \otimes \cdots \otimes \mathbf{v}^{(j)}_{i_{(j)}} \otimes \cdots \otimes \mathbf{v}^{(J)}_{i_{(J)}}, \lambda^{(1)}_{i_{(1)}} \times \cdots \times \lambda^{(j)}_{i_{(j)}} \times \cdots \times \lambda^{(J)}_{i_{(J)}} \tag{3.13}$$

across all combinations of the indices $(i_{(1)}, \cdots, i_{(j)}, \cdots, i_{(J)})$. For example, if $\mathbf{V}^{(1)} = [\mathbf{v}^{(1)}_1 \,|\, \mathbf{v}^{(1)}_2]$ and $\mathbf{V}^{(2)} = [\mathbf{v}^{(2)}_1 \,|\, \mathbf{v}^{(2)}_2 \,|\, \mathbf{v}^{(2)}_3]$,

$$\begin{aligned}\mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} = [&\mathbf{v}^{(1)}_1 \otimes \mathbf{v}^{(2)}_1 \,|\, \mathbf{v}^{(1)}_1 \otimes \mathbf{v}^{(2)}_2 \,|\, \mathbf{v}^{(1)}_1 \otimes \mathbf{v}^{(2)}_3 \,|\, \cdots \\ &\mathbf{v}^{(1)}_2 \otimes \mathbf{v}^{(2)}_1 \,|\, \mathbf{v}^{(1)}_2 \otimes \mathbf{v}^{(2)}_2 \,|\, \mathbf{v}^{(1)}_2 \otimes \mathbf{v}^{(2)}_3]\end{aligned}$$

Frequency Analysis for Product Graphs Under the Kronecker product, we have seen that the eigenvalues of the graph product matrix $\mathbf{A}$ are the product of combinations of eigenvalues from the sub-graphs that compose the product graph. We now study the ordering of eigenvectors by their total variations induced under the Kronecker Product composition of graphs.

For clarity, we only consider real-valued eigenvalues here. The reasoning below can easily be extended to complex eigenvalues (directed graphs).

**Theorem 7.** For brevity and clarity, let us consider the setting where the product graph is formed by composing $M = 2$ graphs $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ of respective size $N_1$ and $N_2$ whose eigenvector-eigenvalue pairs are $(\mathbf{v}^{(1)}_i, \lambda^{(1)}_i$ and $(\mathbf{v}^{(2)}_j, \lambda^{(2)}_j)$ respectively. Without loss of generality, we also assume the respective sets of eigenvalues of the two graphs are ordered as $\lambda^{(1)}_1 \leq \lambda^{(1)}_2 \leq \cdots \leq \lambda^{(1)}_{N_1}$ and $\lambda^{(2)}_1 \leq \lambda^{(2)}_2 \leq \cdots \leq \lambda^{(2)}_{N_2}$.

As derived in (3.12), $\mathbf{A} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)}$ has eigenvector-eigenvalue pairs of the form $(\mathbf{v}_{i,j}, \lambda_{i,j})$ where

$$\mathbf{v}_{i,j} = \mathbf{v}^{(1)}_i \otimes \mathbf{v}^{(2)}_j \tag{3.14}$$

$$\lambda_{i,j} = \lambda^{(1)}_i \lambda^{(2)}_j \tag{3.15}$$

Clearly $\lambda_{i,j} \leq \lambda_{i,j+1}$ and $\lambda_{i,j} \leq \lambda_{i+1,j}$. Directly applying Theorem 1, we have the following partial ordering of the eigenvectors of $\mathbf{A}$ based on the total variation functional.

$$\mathrm{TV}_{\mathbf{A}}(\mathbf{v}_{i,j}) > \mathrm{TV}_{\mathbf{A}}(\mathbf{v}_{i,j+1}) \tag{3.16}$$

$$\mathrm{TV}_{\mathbf{A}}(\mathbf{v}_{i,j}) > \mathrm{TV}_{\mathbf{A}}(\mathbf{v}_{i+1,j}) \tag{3.17}$$

We can perform analogous characterizations of the partial ordering of eigenvectors induced by the Cartesian graph composition method. In addition, we can also show a congruous partial ordering of eigenvectors of the graph Laplacian induced by both the Kronecker and Cartesian products. We note that in this case, we would use the graph total variation functional based on the quadratic form of the graph Laplacian here $\mathrm{TV}_{\mathbf{A}} = \mathbf{x}^T \boldsymbol{L} \mathbf{x}$.

### 3.2.2 Sampling Theory: Product Graph

In this section, we show how to efficiently sample and recover bandlimited signals on product graphs.

We now consider the product graph $G$ that is composed using the Kronecker product over $J$ graphs $\{G^{(1)}, \cdots, G^{(J)}\}$.

As before, we have a bandlimited graph signal $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ that is associated with the product graph $\mathbf{A}$ and a sampling operator $\Psi$ such that the sampled signal $\mathbf{x}_{\mathcal{M}} = \Psi\mathbf{x}$ is acquired by applying the sampling operator $\Psi$. We showed in Theorem 5 that a sufficient condition to perfectly recover the sampled bandlimited signal $\mathbf{x}_{\mathcal{M}} = \Psi\mathbf{x}$ where $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ is that

$$\mathrm{rank}(\Psi\,\mathbf{V}_{(K)}) = K \tag{3.18}$$

It is straightforward to sample the product graph using the framework constructed in the previous section for a single graph by using the composed graph-shift $\mathbf{A}$ as a whole. Instead, in this section, we look to exploit the structure of the product graph under the Kronecker product composition when we sample the graph. We note here that we are free to order the eigenvectors of $\mathbf{V}$ arbitrarily.

We have seen that we can write any given column vector $\mathbf{v}$ of $\mathbf{V}$ as a particular combination of $J$ column vectors from each of the $\mathbf{V}^{(j)}$:

$$\mathbf{v} = \mathbf{v}^{(1)}_{i_{(1)}} \otimes \cdots \otimes \mathbf{v}^{(j)}_{i_{(j)}} \otimes \cdots \otimes \mathbf{v}^{(J)}_{i_{(J)}} = \bigotimes_{j=1}^{J} \mathbf{v}^{(j)}_{i_{(j)}} \tag{3.19}$$

where $\mathbf{v}^{(j)}_{i_{(j)}}$ is a column of $\mathbf{V}^{(j)}$ indexed by $i_{(j)}$.

Given some subset of $K$ columns of $\mathbf{V}$ over which the signal is bandlimited, we can accordingly re-order the columns in each of $\mathbf{V}^{(j)}$ such that

$$\mathbf{V}_{(K)} \subset \mathbf{V}^{(1)}_{R_1} \otimes \cdots \mathbf{V}^{(j)}_{R_j} \otimes \cdots \mathbf{V}^{(J)}_{R_J} = \bigotimes_{j=1}^{J} \mathbf{V}^{(j)}_{R_j} = \mathbf{V}_S\,. \tag{3.20}$$

$\mathbf{V}^{(j)}_{R_j}$ corresponds to the top $R_j$ columns of $\mathbf{V}^{(j)}$ and $S = \prod_{j=1}^{J} R_j$. We note that $K \leq S \leq K^J$. In addition, any signal that is in $\mathrm{BL}_K(\mathbf{V})$ is also in $\mathrm{BL}_S(\bigotimes_{j=1}^{J} \mathbf{V}^{(j)}_{R_j})$.

**Theorem 8.** Let us consider the sampling scheme where we sample $R_j$ nodes from each of the sub-graphs $G^{(j)}$ using the sampling operator $\Psi^{(j)}$.

Using Theorem 5, for each of the $J$ graph atoms, we can construct appropriate sampling ($\Psi^{(j)}$) and interpolation ($\Phi^{(j)}$) operators corresponding to the subset of columns $R_j$ in $\mathbf{V}^{(j)}$ such that for any $\mathbf{x}^{(j)} \in \mathrm{BL}_{R_j}(\mathbf{V}^{(j)})$, we can sample and perfectly recover such that $\mathbf{x}^{(j)} = \Phi^{(j)}(\Psi^{(j)}\mathbf{x}^{(j)}) = \Phi^{(j)}\mathbf{x}^{(j)}_{\mathcal{M}}$. In addition, $\mathbf{x}^{(j)}_{\mathcal{M}}$ is associated with a sampled graph whose graph shift is $\mathbf{A}^{(j)}_{\mathcal{M}}$.

We now sample $S$ nodes in the product graph corresponding to all combinations of the sampled nodes in the graph atoms. That is, we construct the sampling operator $\Psi$ to sample $S = \prod_{j=1}^{J} R_j$ nodes in the product graph such that $\mathbf{x}_{\mathcal{M}} = \Psi\mathbf{x}$

$$\Psi = \bigotimes_{j=1}^{J} \Psi^{(j)} \tag{3.21}$$

Further, the corresponding interpolation operator $\Phi$ over the product graph is

$$\Phi = \bigotimes_{j=1}^{J} \Phi^{(j)} \tag{3.22}$$

As a result, $\Psi$ and $\Phi$ enable perfect recovery such that for any bandlimited graph signal $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ on the product graph, $\mathbf{x} = \Phi \mathbf{x}_{\mathcal{M}} = \Phi\Psi\mathbf{x}$.

In addition, the sampled graph signal $\mathbf{x}_{\mathcal{M}}$ lies on a sampled product graph. Particularly the sampled product graph can be decomposed as the Kronecker product of the sampled graph for the individual sub-graphs. That is,

$$\mathbf{A}_{\mathcal{M}} = \bigotimes_{j=1}^{J} \mathbf{A}_{\mathcal{M}}^{(j)} \tag{3.23}$$

*Proof.* Let us consider the sampling scheme where we sample $R_j$ nodes from each of the sub-graphs $G^{(j)}$ using the sampling operator $\Psi^{(j)}$. We then compose the full sampling operator

$$\Psi = \Psi^{(1)} \otimes \cdots \otimes \Psi^{(j)} \otimes \cdots \otimes \Psi^{(M)} = \bigotimes_{j}^{J} \Psi^{(j)}$$

such that we sample the nodes on the product graphs corresponding to the combinations of the $R_j$ nodes.

We can then write

$$\Psi\, \mathbf{V}_{(K)} = (\bigotimes_{j=1}^{J} \Psi^{(j)})(\bigotimes_{j=1}^{J} \mathbf{V}_{R_{(j)}}^{(j)}) = \bigotimes_{j=1}^{J} \Psi^{(j)}\, \mathbf{V}_{R_j}^{(j)} \tag{3.24}$$

Hence, to satisfy the condition $\mathrm{rank}(\Psi\, \mathbf{V}_{(K)}) \geq K$ it is sufficient to ensure that for each of the sub-graphs

$$\mathrm{rank}(\Psi^{(j)}\, \mathbf{V}_{R_j}^{(j)}) = R_j \tag{3.25}$$

such that

$$\mathrm{rank}(\Psi\, \mathbf{V}_{(K)}) = \prod_{j}^{J} \mathrm{rank}(\Psi^{(j)}\, \mathbf{V}_{R_j}^{(j)}) = \prod_{j=1}^{J} R_j \geq K \tag{3.26}$$

We have already shown, for a single graph, how to choose a sampling operator $\Psi^{(j)}$ that ensures the above condition holds. We choose $\Psi^{(j)}$ such that $\Psi^{(j)}\, \mathbf{V}_{R_j}^{(j)}$ is full rank, that is, $\mathrm{rank}(\Psi^{(j)}\, \mathbf{V}_{R_j}^{(j)}) = R_j$. The corresponding interpolation operator $\Phi^{(j)}$ is $\Phi^{(j)} = \mathbf{V}_{R_j}^{(j)}) \boldsymbol{W}^{(j)}$ where $\boldsymbol{W}^{(j)} = (\Psi^{(j)}\, \mathbf{V}_{R_j}^{(j)})^{\dagger}$.

As shown before, given an admissible sampling operator $\Psi$ such that $\mathrm{rank}(\Psi\, \mathbf{V}_{(K)}) = K$, the interpolation operator $\Phi$ that ensures perfect recovery can be composed as the product of the interpolation operators of the $M$ graphs, $\Phi^{(j)}$ such that:

$$\Phi = \mathbf{V}_{(K)}\,\boldsymbol{W} = \mathbf{V}_{(K)}(\Psi\,\mathbf{V}_{(K)})^{\dagger} \tag{3.27}$$

$$= (\bigotimes_{j=1}^{M} \mathbf{V}_{R_j}^{(j)})(\bigotimes_{j=1}^{M} \Psi^{(j)} \bigotimes_{j=1}^{M} \mathbf{V}_{R_j}^{(j)})^{\dagger} = (\bigotimes_{j=1}^{M} \mathbf{V}_{R_j}^{(j)})(\bigotimes_{j=1}^{M} (\Psi^{(j)}\,\mathbf{V}_{R_j}^{(j)}))^{\dagger} \tag{3.28}$$

$$= (\bigotimes_{j=1}^{M} \mathbf{V}_{R_j}^{(j)})(\bigotimes_{j=1}^{M} (\Psi^{(j)}\,\mathbf{V}_{R_j}^{(j)})^{\dagger}) = \bigotimes_{j=1}^{M} (\mathbf{V}_{R_j}^{(j)}(\Psi^{(j)}\,\mathbf{V}_{R_j}^{(j)})^{\dagger}) \tag{3.29}$$

$$= \bigotimes_{j=1}^{M} (\mathbf{V}_{R_j}^{(j)}\,\boldsymbol{W}^{(j)}) = \bigotimes_{j=1}^{M} \Phi^{(j)} \tag{3.30}$$

Hence, we have shown how to construct sampling and interpolation operators for bandlimited signals on product graphs that enables perfect recovery. Particularly, we can construct the sampling operator by composing admissible sampling operators on the graph atoms that the product graph is composed of.

This tells us that we don't need to compute the whole product graph $\mathbf{A}$ or its spectral decomposition (GFT basis). Instead we can sample bandlimited graph signals and perfectly recover using only the spectral decompositions of the sub-graphs $\mathbf{A}^{(j)}$. As shown before, a sampled graph signal is supported by a "sampled" graph that preserves it's Graph Fourier transform. The sampled graph is of the form

$$\mathbf{A}_{\mathcal{M}} = \boldsymbol{W}^{\dagger}\Lambda_{(M)}\boldsymbol{W} \in \mathbb{R}^{M\times M}. \tag{3.31}$$

As a result, we can write:

$$\mathbf{A}_{\mathcal{M}} = (\Psi\,\mathbf{V}_{(M)})^{\dagger}\Lambda_{(M)}((\Psi\,\mathbf{V}_{(M)})) \tag{3.32}$$

$$= (\bigotimes_{l=1}^{M} \Psi^{(j)} \bigotimes_{l=1}^{M} \mathbf{V}_{R_j}^{(j)})^{\dagger}(\bigotimes_{l=1}^{M} \Lambda_{(R_{(l)})})(\bigotimes_{l=1}^{M} \Psi^{(j)} \bigotimes_{l=1}^{M} \mathbf{V}_{R_j}^{(j)}) \tag{3.33}$$

$$= (\bigotimes_{l=1}^{M} (\Psi^{(j)}\,\mathbf{V}_{R_j}^{(j)})^{\dagger})(\bigotimes_{l=1}^{M} \Lambda_{(R_j)})(\bigotimes_{l=1}^{M} (\Psi^{(j)}\,\mathbf{V}_{R_j}^{(j)})) \tag{3.34}$$

$$= \bigotimes_{l=1}^{M} (\Psi^{(j)}\,\mathbf{V}_{R_j}^{(j)})^{\dagger}\Lambda_{(R_j)}(\Psi^{(j)}\,\mathbf{V}_{R_j}^{(j)}) = \bigotimes_{l=1}^{M} (\boldsymbol{W}^{(j)})^{\dagger}\Lambda_{(R_j)}\boldsymbol{W}^{(j)} \tag{3.35}$$

$$= \bigotimes_{l=1}^{M} \mathbf{A}_{\mathcal{M}}^{(j)} \tag{3.36}$$

We see that the sampled product graph can be decomposed as the Kronecker product of the sampled graph for the individual graph atoms. □

The sampling and recovery framework for product graphs based the decomposition of the sampling and interpolating operators presented in Theorem 8 is illustrated in Figure 3.7.

### 3.2.3 Toy Example

In this section, we study a toy example that further illustrates Theorem 8. As shown in Figure 3.8, consider a graph $\mathbf{A} = \mathbf{A}_1 \otimes \mathbf{A}_2$ and a bandlimited signal $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ with K=3. The top $K = 3$

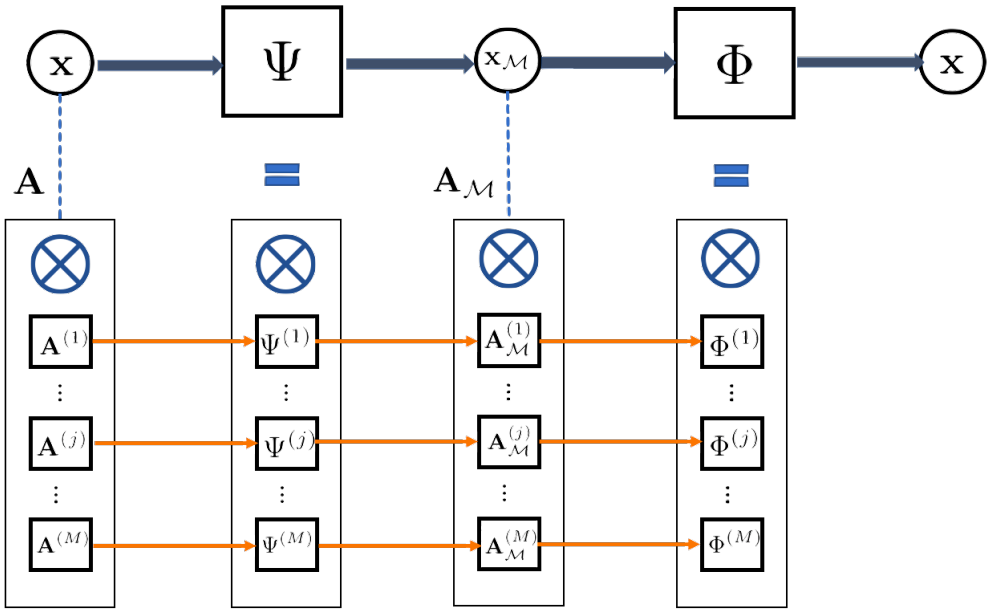



*Figure 3.7: As shown in Theorem 8, we can construct an admissible sampling operator and corresponding interpolation operator by composing sampling and interpolation operators defined respectively on each of the graph atoms. Further, the sampled graph signal lies on a sampled product graph*

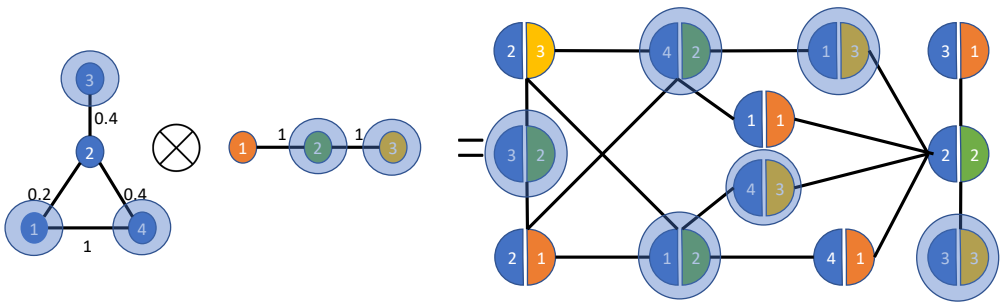



*Figure 3.8: In Section 3.2.3, we consider sampling and recovering a signal* $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ *with* $K = 3$. *The top* $K = 3$ *columns of the ordered GFT basis* $\mathbf{V}$ *corresponds to the pairs* $(1,1),(4,3)$, *and* $(3,3)$ *of the graph atoms respectively. We choose a sample set consisting of nodes (1,3,4) on the* $\mathbf{A}_1$ *and a sampling set (2,3) on* $\mathbf{A}_2$. *The sample sets are marked by the transparent blue circle in the above figure. We then sample nodes corresponding to all combinations of the sampling sets on the product graph. That is, we sample 6 nodes corresponding to the following pairs* $\{(1,2),(1,3),(3,2),(3,3),(4,2),(4,3)\}$ *We can appropriately construct an interpolation operator from the interpolation operators corresponding to the chosen sampling sets on the graph atoms* $\mathbf{A}_1$ *and* $\mathbf{A}_2$ *such that we can ensure perfect recovery for any bandlimited signal* $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$

columns of the ordered GFT basis $\mathbf{V}$ corresponds to the pairs $(1,1),(4,3)$, and $(3,3)$ from the graph atoms respectively. As a result, we can set $R_1 = \{1,3,4\}$ and $R_2 = R_1 = \{1,3\}$ such that $\mathbf{V}_{(K)} \subset \mathbf{V}^{(1)}_{R_1} \otimes \mathbf{V}^{(2)}_{R_2}$. We can then compose sampling and interpolation operators using Theorem 5 for each of the two graphs:

$$\Psi^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \Phi^{(1)} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.69 & 2.21 & -1 \end{bmatrix}$$

$$\Psi^{(2)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \Phi^{(2)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We then compose the sampling and interpolation operators as in Theorem 8 as $\Psi = \Psi^{(1)} \otimes \Psi^{(2)}$ and $\Phi = \Phi^{(1)} \otimes \Phi^{(2)}$. We then sample $|R_1||R_2| = 6$ nodes in the product graph as $\mathbf{x}_{\mathcal{M}} = \Psi\mathbf{x}$ corresponding to combinations of the chosen sampling sets for each of the graph atoms as illustrated in Figure 3.8. We then see that we can sample and perfectly reconstruct any bandlimited signal $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ by verifying that $\Phi\mathbf{x}_{\mathcal{M}} = \mathbf{x}$.

### 3.2.4 Discussions and Extensions

#### Graph Generation under Recursive Kronecker Multiplication

In[53], a generative model that can effectively model the structure of many large real-world networks was presented by recursively applying the Kronecker product on a base graph that can be estimated efficiently. We can consequently leverage our framework to sample graph signals that are supported on a large real-world networks with a substantial reduction in the sample and computational complexity.

#### Smooth Signals

In Theorem 13, it was shown that smooth signals can be well approximated by a graph signal that is bandlimited with respect to the top $K$ columns of the ordered graph Fourier basis $\mathbf{V}$. That is, a smooth graph signal $\mathbf{x}$ can be well approximated by $\mathbf{x}' \in \mathrm{BL}_{\mathbf{A}}(K)$.

Particularly, given the partial ordering induced under the Kronecker product in Theorem 7, we can write

$$\mathbf{V}_{(K)} = \mathbf{V}^{(1)}_{R_{(1)}} \otimes \cdots \mathbf{V}^{(\ell)}_{R_{(\ell)}} \otimes \cdots \mathbf{V}^{(M)}_{R_{(M)}} = \bigotimes_{l=1}^{M} \mathbf{V}^{(\ell)}_{R_{(\ell)}} \tag{3.37}$$

where $\mathbf{V}_{R_{(\ell)}}$ corresponds to the top $R_{(\ell)}$ columns of $\mathbf{V}$. We note that

$$K \leq \prod_{\ell=1}^{M} R_{(\ell)} \leq K + M$$

Hence, we need at most $O(K)$ samples, and we can sample smooth signals on product graphs

nearly optimally.

### Optimal sampling operator

As mentioned in Section 3.1.1, at least one set of $K$ linearly-independent rows in $\mathbf{V}_{(K)}$ always exists. When we have multiple choices of $K$ linearly-independent rows, we aim to find the optimal one to minimize the effect of noise. When we have multiple choices of $K$ linearly-independent rows, which one is optimal? We answer this question from two aspects: the first one is to minimize the turbulence from noise; the second one is to provide robust representation for sampled signal coefficients.

We consider the noise $\mathbf{e}$ is introduced during sampling as follows,

$$\mathbf{x}_{\mathcal{M}} \;=\; \Psi\mathbf{x} + \mathbf{e}$$

The recovered graph signal, $\mathbf{x}'$, is then

$$\mathbf{x}' \;=\; \Phi\mathbf{x}_{\mathcal{M}} \;=\; \Phi\Psi\mathbf{x} + \Phi\mathbf{e} \;=\; \mathbf{x} + \Phi\mathbf{e}.$$

An optimal sampling operator should minimize the turbulence from noise. To bound the turbulence, we have

$$\begin{aligned} & ||\mathbf{x}' - \mathbf{x}||_2 \\ = \;& ||\Phi\mathbf{e}||_2 \\ = \;& ||\,\mathbf{V}_{(K)}\,\mathbf{U}\,\mathbf{e}||_2 \\ \leq \;& ||\,\mathbf{V}_{(K)}\,||_2||\,\mathbf{U}\,||_2||\mathbf{e}||_2. \end{aligned}$$

Since $||\,\mathbf{V}_{(K)}\,||_2$ and $||\mathbf{e}||_2$ are fixed, we want $\mathbf{U}$ to have a small spectral norm. From this aspect, for each feasible $\Psi$, we compute the inverse, or pseudo-inverse of $\Psi\,\mathbf{V}_{(K)}$ to obtain $\mathbf{U}$; the best choice comes from the smallest spectral norm of $\mathbf{U}$.

Since $\mathbf{U}$ is the graph Fourier transform matrix for the sampled signal coefficients, we want $\mathbf{U}$ to span the space well to provide robust and stable representation[58,59]. When $M = K$, $\mathbf{U}$ is a basis that spans $\mathbb{R}^K$; we thus check the condition for the Riesz basis. For each feasible $\Psi$, we compute the inverse of $\Psi\,\mathbf{V}_{(K)}$ to obtain $\mathbf{U}$; the best choice comes from the tightest stability constants of $\mathbf{U}$[38]. When $M > K$, $\mathbf{U}$ is a frame that spans $\mathbb{R}^K$; we thus check the condition for the frame. For each feasible $\Psi$, we compute the pseudo-inverse of $\Psi\,\mathbf{V}_{(K)}$ to obtain $\mathbf{U}$; the best choice comes from the tightest frame bounds of $\mathbf{U}$[38]. Note that for both Riesz basis and frame, the lower bound is the smallest singular value of $\mathbf{U}$ and the upper bound is the spectral norm of $\mathbf{U}$.

Combining these two aspects, we find that the principle to find the optimal sampling operator is as follows: given a sampling operator $\Psi$, we compute the inverse of $\Psi\,\mathbf{V}_{(K)}$ to obtain $\mathbf{U}$. We choose the sampling operator with the minimum spectral norm and the maximum smallest singular value of $\mathbf{U}$.

$$\Psi^{opt} \;=\; \arg\max_{\Psi} \sigma_{\min}(\Psi\,\mathbf{V}_{(K)}), \tag{3.38}$$

where $\sigma_{\min}$ denotes the smallest singular value. Since we restrict the form of $\Psi$ in (5.2), (3.38) is

non-deterministic polynomial-time hard. To solve (3.38), we can use a greedy algorithm as shown in Algorithm 1. Some previous work solved the similar optimization problem for matrix approximation and showed that the greedy algorithm gives a good approximation to the global optimum[60]. Note that $\mathcal{M}$ is the sampling sequence, indicating which row to select, and $(\mathbf{V}_{(K)})_{\mathcal{M}}$ denotes the sampled rows from $\mathbf{V}_{(K)}$. When increasing the number of samples, the smallest singular value of $\Psi\,\mathbf{V}_{(K)}$ grows, and thus, redundant samples make the algorithm robust to noise.

**Algorithm 1** Optimal Sampling Operator via Greedy Algorithm

**Input** $\mathbf{V}_{(K)}$ the first $K$ columns of $\mathbf{V}$
$M$ the number of samples
**Output** $\mathcal{M}$ sampling set

**Function**

while $|\mathcal{M}| < M$
$m = \arg\max_i \sigma_{\min}\left((\mathbf{V}_{(K)})_{\mathcal{M}+\{i\}}\right)$
$\mathcal{M} \leftarrow \mathcal{M} + \{m\}$
end
**return** $\mathcal{M}$

### Product Graph

Since

$$\Psi = \bigotimes_{l=1}^{M} \Psi^{(l)}, \Psi\,\mathbf{V}_{(K)} = \bigotimes_{l=1}^{M} \Psi^{(\ell)}\,\mathbf{V}^{(\ell)}_{R_{(\ell)}},$$

we can write,

$$\Psi^{opt} = \arg\max_{\Psi} \sigma_{\min}(\Psi\,\mathbf{V}_{(K)}) = \arg\max_{\Psi} \sigma_{\min}(\bigotimes_{l=1}^{M} \Psi^{(\ell)}\,\mathbf{V}^{(\ell)}_{R_{(\ell)}}) \tag{3.39}$$

$$= \prod_{l=1}^{M} \arg\max_{\Psi^{(\ell)}} \sigma_{\min}(\Psi^{(\ell)}\,\mathbf{V}^{(\ell)}_{R_{(\ell)}}) \tag{3.40}$$

$$\tag{3.41}$$

Hence this is equivalent to finding the optimal sampling operator for each of the sub-graphs by solving equation 3.38. That is, for each sub-graph indexed by $\ell$, we solve

$$\Psi^{(\ell),opt} = \arg\max_{\Psi^{(l)}} \sigma_{\min}(\Psi^{(l)}\,\mathbf{V}^{(\ell)}_{(K)}) \tag{3.42}$$

such that

$$\Psi^{opt} = \prod_{\ell=1}^{M} \Psi^{(\ell),opt} \tag{3.43}$$

We can use Algorithm 1 for each of the $\ell$ subgraphs to approximate the problem in (3.42).

## 3.3 Randomized Sampling for Graph Signals

### 3.3.1 Uniform Random sampling

In Section 3.6.2, we see that when sampling sufficient signal coefficients, any sampling operator leads to perfect recovery for discrete-time signals. Here we show that similar results are applied for circulant graphs and Erdős-Rényi random graphs.

**Circulant graphs.** A circulant graph is a graph whose adjacency matrix is circulant[43]. The circulant-graph shift, $\mathbf{C}$, can be represented a polynomial of the cyclic permutation matrix, $\mathbf{A}$, whose the corresponding graph Fourier transform is discrete Fourier transform, i.e.,

$$\begin{aligned}\mathbf{C} &= \sum_{i=0}^{L-1} h_i \, \mathbf{A}^i = \sum_{i=0}^{L-1} h_i (\mathrm{F}^* \, \Lambda \, \mathrm{F})^i \\ &= \mathrm{F}^* \left( \sum_{i=0}^{L-1} h_i \Lambda^i \right) \mathrm{F} \, .\end{aligned}$$

where $L$ is the order of the polynomial, and $h_i$ is the coefficient corresponding to the $i$th order. Since the graph Fourier transform matrix of circulant graphs is discrete Fourier transform matrix, we can perfectly recover a circulant-graph signals with bandwidth $K$ by sampling any $M \geq K$ signal coefficients as shown in Theorems 6. In other words, perfect recovery is guaranteed when we randomly sample sufficient signal coefficients.

**Erdős-Rényi graphs.** An Erdős-Rényi graph is constructed by connecting nodes randomly, where each edge is included in the graph with probability $p$ independent from every other edge[61,62]. We aim to show that by sampling $K$ signal cofficients randomly, the singular values of the corresponding $\Psi \, \mathbf{V}_{(K)}$ are well bounded.

**Lemma 1.** Let a graph shift $\mathbf{A}$ represent an Erdős-Rényi random graph on a vertex set of size $N$, obtained by drawing an edge between each pair of vertices, randomly and independently, with probability $p = g(N)\log(N)/N$. Let $\mathbf{V}$ be the eigenvector matrix of $\mathbf{A}$, obeying $\mathbf{V}\,\mathbf{V}^T = N \cdot \mathrm{I}$. Let the sampling number satisfies

$$M \geq K \cdot \frac{\log^{2.2} g(N) \log(N)}{p} \cdot \max(C_1 \log K, C_2 \log \frac{3}{\delta}),$$

for some positive constants $C_1, C_2$. Then,

$$P\left( \left\| \frac{1}{M} (\Psi \, \mathbf{V}_{(K)})^T (\Psi \, \mathbf{V}_{(K)}) - \mathrm{I} \right\|_2 \leq \frac{1}{2} \right) \leq 1 - \delta \tag{3.44}$$

for all the sampling operators $\Psi$ that samples $M$ signal coefficients.

*Proof.* Since the graph shift $\mathbf{A}$ is a real and symmetric matrix, the eigenvector matrix $\mathbf{V}$ is orthogonal and satisfies $\max_{i,j} |\, \mathbf{V}_{i,j} \,| = O\left( \sqrt{\log^{2.2} g(N) \log N / (N^2 p)} \right)$ for $p = g(N)\log(N)/N$[63]. We then plug $\mathbf{V}$ into Theorem 1.2 in[64] and obtain (3.44). □

**Theorem 9.** Let a graph shift $\mathbf{A}$ represent an Erdős-Rényi random graph on a vertex set of size $N$, obtained by drawing an edge between each pair of vertices, randomly and independently, with

probability $p = g(N)\log(N)/N$. Let $\mathbf{V}$ be the eigenvector matrix of $\mathbf{A}$, obeying $\mathbf{V}\,\mathbf{V}^T = N \cdot \mathrm{I}$. Let $\Psi$ be a sampling operator with the sampling number obeying:

$$M \geq K \cdot \frac{\log^{2.2} g(N)\log(N)}{p} \cdot \max(C_1 \log K, C_2 \log \frac{3}{\delta}),$$

for some positive constants $C_1, C_2$. With probability $(1-\delta)$, $\Psi\,\mathbf{V}_{(K)}$ is a frame in $\mathbb{R}^K$ with lower bound $M/2$ and upper bound $3M/2$.

*Proof.* Using Lemma 6, with probability $(1-\delta)$, we have

$$\left\| \tfrac{1}{M}(\Psi\,\mathbf{V}_{(K)})^T(\Psi\,\mathbf{V}_{(K)}) - \mathrm{I} \right\|_2 \quad \leq \frac{1}{2}$$

It is equivalent to that, for all $\mathbf{x} \in \mathbb{R}^K$,

$$\begin{aligned} -\frac{1}{2}\mathbf{x}^T\mathbf{x} \leq \quad & \mathbf{x}^T \left( \tfrac{1}{M}(\Psi\,\mathbf{V}_{(K)})^T(\Psi\,\mathbf{V}_{(K)}) - \mathrm{I} \right) \mathbf{x} \quad \leq \frac{1}{2}\mathbf{x}^T\mathbf{x} \\ \frac{M}{2}\mathbf{x}^T\mathbf{x} \leq \quad & \mathbf{x}^T(\Psi\,\mathbf{V}_{(K)})^T(\Psi\,\mathbf{V}_{(K)})\mathbf{x} \quad \leq \frac{3M}{2}\mathbf{x}^T\mathbf{x} \end{aligned}$$

□

From Theorem 11, we see the singular values of $\Psi\,\mathbf{V}_{(K)}$ are well bounded with high probability. It shows that $\Psi\,\mathbf{V}_{(K)}$ has full rank with high probability; in other words, with high probability, perfect recovery is achieved for Erdős-Rényi graph signals when we randomly sample sufficient signal coefficients.

### 3.3.2 Product Graphs and Decomposability

While the sampling theory discussed above gives conditions on sampling sets that enable perfect recovery for bandlimited signals, it does not prescribe easily implementable robust algorithms to choose these sampling sets. Randomized sampling in this case is particularly favorable especially for large graphs where standard column subset selection or search algorithms may be prohibitive. In this section, we study randomized sampling procedures whereby we sample $M$ nodes proportional to a sampling distribution $\{\pi_i\}$ over the nodes. That is, we sample $M$ nodes without replacement such that in each of the $M$ rounds, the probability of the $i$-th node being selected is proportional to $\pi_i$.

Inspired by the sampling framework discussed in the last section, we want to compose a sampling operator on the product graph from sampling operators we construct on the graph atoms. Consider the following sampling framework: For each of the $J$ graphs, $G_j$, where $j = \{1, \cdots, J\}$, we define a probability distribution $\{\pi^{(j)}\}$ over its nodes and a corresponding sampling operator $\Psi^{(j)}$ that samples the $i$-th node in $G_j$ with probability $\pi_i^{(j)}$. As before, we then compose the sampling operators over the graph atoms using the Kronecker product as $\Psi = \bigotimes_{j=1}^{J} \Psi^{(j)}$ such that the probability of the the $i$-th node in the product graph $G$ is the product of the probabilities of choosing the corresponding nodes on the graph atoms and $\pi_i = \Pi_{j=1}^{J} \pi_{i_{(j)}}^{(j)}$. Similarly to the previous section, this allows us to compose a sampling operator $\Psi$ and probability distribution $\{\pi_i\}$ by only processing the graph atoms and constructing sampling operators $\Psi^{(j)}$ and probability distributions $\{\pi^{(j)}\}$ over each graph atom $G_j$, which is substantially more computationally efficient.

### Uniform Random Sampling

Further, for product graphs composed of graph atoms that belong to the family of graphs, it is sufficient to uniformly randomly sample on each of the graph atoms and compose the sampling operator using the product operator.

### Experimentally-Designed Sampling

Uniform random sampling performs sub-optimally for many real-world irregular graphs and more complex graph models. We now consider experimentally designed sampling, where the sampling distribution is non-uniform and adapted to the graph structure. Particularly, we aim to sample the most informative nodes with respect to the bandlimited class of signals. In[57], a random sampling framework is presented such that only $M = O(K\log(K))$ measurements are sufficient to ensure stable and robust recovery of bandlimited graph signals $\mathrm{BL}_K(\mathbf{V})$ from their samples. It is shown that the graph weighted coherence $\rho_K = \max_i\{\pi_i^{-1/2}\| \mathbf{V}_{(K)}^T \delta_i\|_2\}$ governs the sample complexity for stable and robust recovery. It is then easy to show that the optimal sampling distribution $\{\pi_i^*\}$ that minimizes the graph weighted coherence $\rho_K$ is $\pi_i^* = \| \mathbf{V}_{(K)}^T \delta_i\|_2^2/K$ which also corresponds to the statistical leverage scores of $\mathbf{V}_{(K)}$ and can be computed efficiently. In this section, we generalize this random sampling framework to product graphs by exploiting the structure of product graphs. Towards this, we first show how we can compute this optimal sampling score for a given node of the product graph from the optimal sampling scores of that node's corresponding nodes over the graph atoms for signals in $\mathrm{BL}_S(\mathbf{V})$.

**Lemma 2.** Let $\{\pi^{*(j)}\}$ be the optimal sampling distribution corresponding to the $j$-th graph atom and $\mathbf{V}_{(R_j)}$ such that $\pi_{i_{(j)}}^{*(j)} = \| \mathbf{V}_{(R_j)}^{(j)T} \delta_{i_{(j)}} \|_2^2/R_j$. It then follows that the optimal sampling score for a node on the product graph is simply the product of the sampling scores of the corresponding nodes in the graph atom such that

$$\pi_i^* = \frac{\| \mathbf{V}_{(S)}^T \delta_i \|_2^2}{S} = \bigotimes_{j=1}^{J} \frac{\| \mathbf{V}_{(R_j)}^{(j)T} \delta_{i_{(j)}} \|_2^2}{R_j} = \bigotimes_{j=1}^{J} \pi_{i_{(j)}}^{*(j)}. \tag{3.45}$$

$$\begin{aligned}
\pi_i^2 &= \| \mathbf{V}_{(K)}^T \delta_i \|_2^2 = \delta_i^T \mathbf{V}_{(K)} \mathbf{V}_{(K)}^T \delta_i \\
&= \delta_i^T (\bigotimes_{j=1}^{J} \mathbf{V}_{(R_j)}^{(j)}) (\bigotimes_{j=1}^{J} \mathbf{V}_{(R_j)}^{(j)})^T \delta_i = \delta_i^T (\bigotimes_{j=1}^{J} \mathbf{V}_{(R_j)}^{(j)} \mathbf{V}_{(R_j)}^{(j)T}) \delta_i \\
&= (\bigotimes_{j=1}^{J} \delta_{i_{(j)}})^T (\bigotimes_{j=1}^{J} \mathbf{V}_{(R_j)}^{(j)} \mathbf{V}_{(R_j)}^{(j)T}) (\bigotimes_{j=1}^{J} \mathbf{V}_{(R_j)}^{(j)} \mathbf{V}_{(R_j)}^{(j)T}) (\bigotimes_{j=1}^{J} \delta_{i_{(j)}}) \\
&= \bigotimes_{j=1}^{J} \delta_{i_{(j)}}^T \mathbf{V}_{(R_j)}^{(j)} \mathbf{V}_{(R_j)}^{(j)T} \delta_{i_{(j)}} = \bigotimes_{j=1}^{J} \| \mathbf{V}_{(R_j)}^{(j)T} \delta_{i_{(j)}} \|_2^2 \\
&= \bigotimes_{j=1}^{J} (\pi_{i_{(j)}}^{(j)})^2
\end{aligned}$$

Under the randomized sampling framework over the graph atoms described in Lemma 2, we now provide (optimal) sufficient conditions on the minimum number of samples that ensure a stable embedding of graph signals in $\mathrm{BL}_K(\mathbf{V})$ on the product graph.

**Theorem 10.** Let $\Psi^{(j)}$ sample $M_j$ nodes according to the sampling distribution $\pi^{(j)*}$ such that $\Psi = \bigotimes_{j=1}^{J} \Psi^{(j)}$ samples $M = \Pi_{j=1}^{J} M_j$ nodes. Let $\mathbf{D}^{(j)}$ be a diagonal rescaling matrix such that $\mathbf{D}^{(j)}_{i_{(j)},i_{(j)}} = 1/\sqrt{M_j \pi^{(j)}_{i_{(j)}}}$ and $\mathbf{D} = \bigotimes_{j=1}^{J} \mathbf{D}^{(j)}$. For any $\delta, \epsilon \in (0,1)$ if,

$$M \geq \frac{3}{\delta^2} S \log(\frac{2K}{\epsilon}),$$

where $S = \Pi_{j=1}^{J} R_j$, we have that with probability atleast $1-\epsilon$, $\Psi\mathbf{D}$ represents a stable embedding for any $\mathbf{x} \in \mathrm{BL}_K(\mathbf{V})$ such that

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|\Psi\mathbf{D}\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \tag{3.46}$$

*Proof.* Full proof omitted due to lack of space. The proof is a consequence of Lemma 2 and is in parts constructed similarly to Theorem 3 in[57]. □

**Algorithm 1.** We recover the original graph signal by solving the following optimization problem:

$$\begin{aligned} \mathbf{x}^*_{\mathrm{SP}} &= \mathbf{V}_{(K)} \arg\min_{\widehat{\mathbf{x}}_{(K)}} \left\| \Psi^T \Psi \mathbf{D}^2 \Psi^T \Psi \mathbf{y} - \mathbf{V}_{(K)}\, \widehat{\mathbf{x}}_{(K)} \right\|_2^2 \\ &= (\bigotimes_{j=1}^{J} \Phi^{(j)})\mathbf{y} \end{aligned}$$

where

$$\Phi^{(j)} = \mathbf{V}^{(j)}_{R_j}\, \mathbf{U}^{(j)}_{R_j}\, \Psi^{(j)^T} \Psi^{(j)} \mathbf{D}^{(j)2} \Psi^{(j)T}$$

Hence, we see that we can compose the interpolation operators by only processing the graph atoms. We can now provide lower and upper bounds on the squared error.

**Corollary 1.** Assume we compose a sampling operator with sufficient samples as proposed in Lemma 2 with respect to the optimal sampling distribution $\{\pi_i^*\}$ and use Algorithm 1 to recover the original signal. We then have, with probability atleast $1-\epsilon$,

$$\frac{1}{M\sqrt{1+\delta}} \|\Psi\mathbf{D}\epsilon\|_2 \leq \|\mathbf{x}^*_{\mathrm{SP}} - \mathbf{x}\|_2 \leq \frac{2}{M\sqrt{1-\delta}} \|\Psi\mathbf{D}\epsilon\|_2$$

*Proof.* This is a direct consequence of Theorem 6 in[57] because of the restricted isometry property satisfied in Theorem 10. □

**Remark 1.** Smooth graph signals are bandlimited under a fixed frequency ordering[29]. We can show that with our framework on product graphs, under the Cartesian product, we only need $O(KlogK)$ samples to sample and recover a smooth signal in $BL_K(\mathbf{V})$ which is optimal.

**Remark 2.** We have seen that we do not need to process the whole product graph $\mathbf{A}$ or compute its spectral decomposition (GFT basis) to construct random sampling and interpolation operators on the product graph. Instead, we only need to compute the spectral decompositions of its graph atoms $\mathbf{A}^{(j)}$ that are of size $O(poly(N^{\frac{1}{J}}))$.

In this section, we study randomized sampling procedures whereby we sample $M$ nodes proportional to a sampling distribution $\{\pi_i\}$. That is, we sample $M$ nodes without replacement such that in each of the $M$ rounds, the probability of the $i$-th node being selected is proportional to $\pi_i$.

**Theorem 11.** Let $\mathbf{A}, \mathbf{V}, \Psi$ be defined as in Lemma 6. With probability $(1-\delta)$, $\Psi\,\mathbf{V}_{(K)}$ is a frame with lower bound $M/2$ and upper bound $3M/2$.

For a Kronecker product of random graphs, we can simply randomly sample each sub-graph and ensure perfect recovery with high probability. This follows because

$$\sigma_{max}(A \otimes B) = \sigma_{max}(A)\sigma_{max}(B)$$

and

$$\sigma_{min}(A \otimes B) = \sigma_{min}(A)\sigma_{min}(B)$$

### Graph Weighted Coherence

In[57], a random sampling framework is presented such that only $M = K\log(K)$ measurements are sufficient to ensure stable and robust recovery of bandlimited graph signals from their samples. We can define the graph weighted coherence that governs how many samples we would need as $\rho_K$,

$$\rho_K = \max_i \{\pi_i^{-1/2} \| \mathbf{V}_{(K)}^T\, \delta_i \|_2\}$$

Particularly, it is shown that the optimal sampling distribution $\{\pi_i\}$ to the sampling distribution that minimizes the graph weighted coherence $\rho_K$ is

$$\pi_i = \| \mathbf{V}_{(K)}^T\, \delta_i \|_2$$

We note that this optimal sampling distribution corresponds to the statistical leverage scores of $\mathbf{V}_{(K)}$. In[65], approximately bandlimited graph signals were defined to be a more general class of graph signals that relaxes the requirement of bandlimitedness, but still promotes smoothness by allowing for a tail after the first K frequency components.

A minimax optimal recovery sampling strategy is then presented for such approximately bandlimited signals. Particularly it is shown that the approximate optimal sampling score when the SNR is small and when the SNR is high are respectively $\{\pi\}$ and $\{\sqrt{\pi_i}\}$.

We now show how we can compute the sampling score for a given node of the product graph from the sampling scores of the node's corresponding nodes in the atoms of the product graph.

**Theorem 12.** We have seen how any signal that is in $\mathrm{BL}_K(\mathbf{V})$ is also in $\mathrm{BL}_S(\bigotimes_{j=1}^J \mathbf{V}_{R_j}^{(j)})$. In addition every node $i$ in the product graph corresponds to a tuple of nodes belonging to the graph atoms $(i_{(1)}, \cdots, i_{(j)} \cdots, i_{(J)})$, such that the Kronecker delta vector $\delta_i = \bigotimes_{j=1}^J \delta_{i_{(j)}}$. Let $\{\pi^{(j)}\}$ be the optimal sampling distribution corresponding to the $j$-th graph atom and $\mathbf{V}_{R_j}$ such that $\pi_{i_{(j)}}^{(j)} = \| \mathbf{V}_{R_j}^{(j)T}\, \delta_{i_{(j)}} \|_2$. We can then show that

$$\pi_i = \bigotimes_{j=1}^{J} \pi_{i_{(j)}}^{(j)} \tag{3.47}$$

As a result, the optimal sampling score on a node on the product graph is simply the product of the sampling scores of the corresponding nodes in the graph atom.

*Proof.* We can then write:

$$
\begin{aligned}
\pi_i^2 &= \| \mathbf{V}_{(K)}^T \delta_i \|_2^2 = \delta_i^T \mathbf{V}_{(K)} \mathbf{V}_{(K)}^T \delta_i \\
&= \delta_i^T (\bigotimes_{j=1}^{J} \mathbf{V}_{R_j}^{(j)})(\bigotimes_{j=1}^{J} \mathbf{V}_{R_j}^{(j)})^T \delta_i = \delta_i^T (\bigotimes_{j=1}^{J} \mathbf{V}_{R_j}^{(j)} \mathbf{V}_{R_j}^{(j)T}) \delta_i \\
&= (\bigotimes_{j=1}^{J} \delta_{i_{(j)}})^T (\bigotimes_{j=1}^{J} \mathbf{V}_{R_j}^{(j)} \mathbf{V}_{R_j}^{(j)T})(\bigotimes_{j=1}^{J} \mathbf{V}_{R_j}^{(j)} \mathbf{V}_{R_j}^{(j)T})(\bigotimes_{j=1}^{J} \delta_{i_{(j)}}) \\
&= \bigotimes_{j=1}^{J} \delta_{i_{(j)}}^T \mathbf{V}_{R_j}^{(j)} \mathbf{V}_{R_j}^{(j)T} \delta_{i_{(j)}} = \bigotimes_{j=1}^{J} \| \mathbf{V}_{R_j}^{(j)T} \delta_{i_{(j)}} \|_2^2 \\
&= \bigotimes_{j=1}^{J} (\pi_{i_{(j)}}^{(j)})^2
\end{aligned}
$$

□

### 3.3.3 Graph Filter Banks

We have shown that we can sample and perfectly recover bandlimited signals on product graphs by composing the appropriate sampling and interpolation operators from the product graph atoms. In addition, we can process multi-band signals by sampling optimally on a product graph by constructing filter banks analogously to[66] where to handle full-band graph signals, we proposed an approach based on graph filter banks.

We see that the above idea can easily be generalized to multiple channels by splitting the original graphs signal into multiple bandlimited graph signals; instead of dealing with a huge graph, we work with multiple small graphs, which is easy for the storage and computation.

### 3.3.4 Complexity and Savings

*Sample Complexity:* We have seen that we need at least $K$ samples in order to perfectly recover a bandlimited graph signal $\mathbf{x} \in BL_K(\mathbf{V})$ in the single graph setting. In the product graph sampling framework prescribed above, we need atleast $S$ samples of the graph signal on the product graph where $K \leq S \leq K^J$. Hence, in the worst case, we need $K^J$ samples to ensure perfect recovery.

Smooth signals on graphs are approximately bandlimited under a fixed frequency ordering[26]. We can show that under the Cartesian product, we only need $S \leq K + J$ samples to perfectly recover and sample a smooth signal that is in $BL_K(\mathbf{V})$ which is nearly optimal.

*Computational Complexity*: We note that we do not need to process the whole product graph $\mathbf{A}$ or compute its spectral decomposition (GFT basis) which is of complexity $O(N^3)$ and is often computationally prohibitive for large graphs. Instead we can construct sampling and interpolation operators on the product graph using only the spectral decompositions of its graph atoms $\mathbf{A}^{(j)}$ that are of size $O(poly(N^{\frac{1}{J}}))$. We choose $R_j$ nodes from each of the graphs $G^{(j)}$ and sample


(a) 150 cities in the U.S.

(b) Three sampled cities.

*Figure 3.9: Sampling cities from the U.S. (a) shows an 8-nearest-neighbor graph. We aim to sample three cities, including Los Angeles, New York City and Miami, preserving the first three frequencies contents; (b) shows the sampled graph.*

$S = \prod_{j=1}^{J} R_j$ nodes in the product graph such that each sampled node in the product graph correspond to some combination of the sampled nodes in the graph atoms. Hence, we effectively only need to do *choose* $\sum_{j=1}^{J} R_j$ nodes over the graph atoms. In contrast, in the single graph setting, we need to choose atleast $K$ nodes, where in general $K = O(S)$.

### 3.3.5 Numerical Experiments

In this section, we validate the proposed sampling theory on three classical types of graphs, including Erdős-Rényi graphs, small-world graphs, and scale-free graphs. We show that the perfect recovery is achieved in each type of graphs with high probabilities.

To validate the proposed sampling theory in a real-world graph, we sample a geodesic graph of the cities in the U.S. Due to limited space, we just show one feasible sampling result in Figure 3.9.

We aim to validate the proposed sampling theory for Erdős-Rényi graphs, small-world graphs, and scale-free graphs, investigating success rates of perfect recovery using random sampling.

#### Experimental setup

Suppose that for each graph, we deal with the corresponding graph signals with fixed bandwidth $K = 10$. Given a graph shift, we randomly sample 10 rows from the first 10 columns of graph Fourier transform matrix, and check if the $10 \times 10$ matrix has full rank. Based on Theorem 4, if the $10 \times 10$ matrix has full rank, the perfect recovery is guaranteed. For each given graph shift, we run the random sampling for 100 times, and count the number of success to obtain the success rate.

**Erdős-Rényi graphs.** An Erdős-Rényi random graph is constructed by assigning edges randomly. Each edge exists independently in the graph with a given connection probability[61,62]. As shown in Section IV.C, with high probability, perfect recovery is achieved for Erdős-Rényi graph signals when we randomly sample sufficient signal coefficients. We verify this result experimentally, by randomly sampling Erdős-Rényi graphs with various sizes and connecting probabilities. We vary the size to be 50, 500, and 1000; and the connection probabilities with an interval of 0.01 from 0 to 0.5. For each given size and connection probability, we generate 100 times randomly.

**Small-world graphs.** A small-world graph is a graph where most nodes are not neighbors of one another but most nodes can be reached from the other with a small number of steps[61,62].

In the context of a social network, this results in the small world phenomenon of strangers being linked by a small number of mutual acquaintances or connections. Many empirical graphs that we encounter in the real world show such small-world phenomenon and are well modeled by such models. Online social networks, such as Facebook, the Internet, and gene networks, are examples of such graphs. We use the Watts-Strogatz model to generate such graphs, which includes three variables, size, connection probability, and the rewiring probability[67]. We vary the size to be 50, 500, and 1000; the connection probabilities with an interval of 0.01 from 0 to 0.5, and fix the rewiring probability to be 0.1. For each given size and connection probability, we generate 100 times randomly.

**Scale-free graphs.** A scale-free graph is a graph whose degree distribution follows a power law[61,62], i.e., the fraction $P(d)$ of nodes in the graph having $d$ connections goes asymptotically with

$$P(d) \sim d^{-\gamma},$$

where $\gamma$ is typically between 2 and 3. Many real-world graphs, such as the topology of web pages, the collaborative network of Hollywood actors , the power grid of the United States and the peer-reviewed scientific literature exhibit scale-free phenomenon. Scale-free graphs are dominated by a relatively small proportion of nodes that are hubs of connectivity.

We use the the Barabási-Albert model with the preferential attachment mechanism to generate such graphs, which includes two variables, size, and the number of edges to attach at every step[68]. We vary the size to be 50, 500, and 1000; the number of edges to attach at every step with an interval of 1 from 0 to 10. For each given size and connection probability, we generate 100 times randomly.

#### Results

Figure 3.10 shows success rates for size averaged over 100 random tests for each of three types of graphs. We see that the success rate is close to 100% in each of three types of graphs with various sizes.

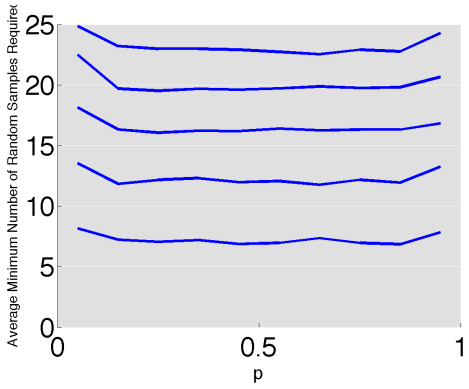
### 3.3.6 Sampling Online Blogs

We aim to validate the proposed sampling theory for online blogs, investigating the success rate of perfect recovery using random sampling, and further classifying the labels of the online blogs.

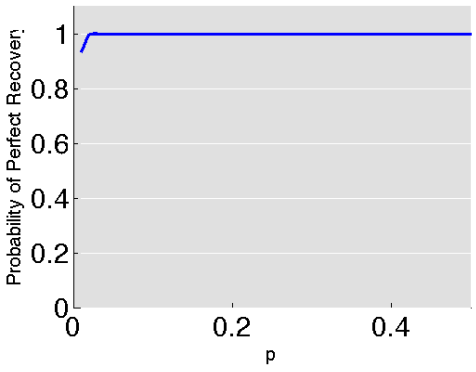
#### Dataset

We consider a dataset of $N = 1224$ online political blogs as either conservative or liberal[69]. We represent conservative labels as $+1$ and liberal ones as $-1$. The blogs are represented by a graph in which nodes represent blogs, and directed graph edges correspond to hyperlink references between blogs. The graph signal here is the label assigned to the blogs, called the labeling signal.

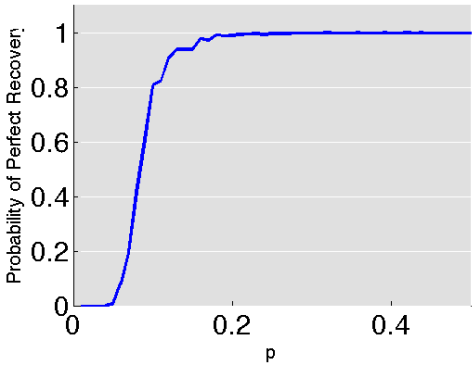
We use the spectral decomposition in (2.1) for this online-blog graph to get the graph frequencies in a descending order and the corresponding graph Fourier transform matrix. We show the graph frequencies in Figure 3.11, and the frequency content of the labeling signal in Figure 3.12. We see that labeling signal is a full-band signal, but approximately bandlimited. The main infor-



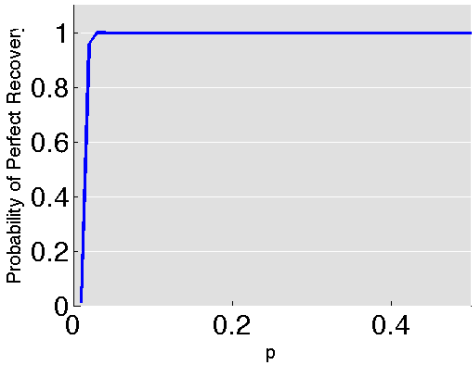

(a) n=50 Erdos-Renyi



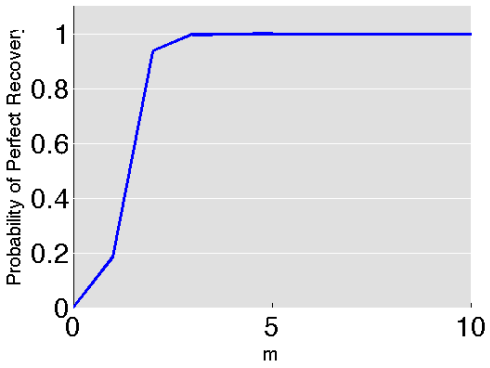

(b) n=500 Erdos-Renyi



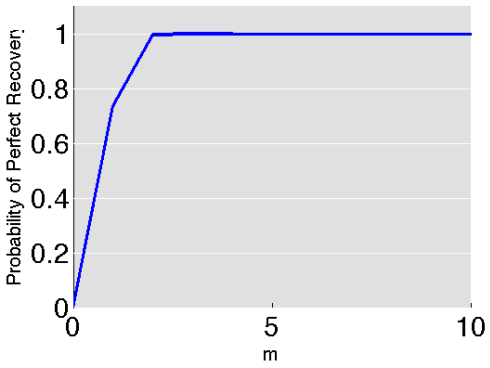

(a) n=50 Small-World




(b) n=500 Small-World




(a) n=50 Scale-free




(b) n=500 Scale-free

*Figure 3.10: Success rates for different graph families*

mation is preserved in the low frequencies. The high frequency contents are introduced to force the elements the labeling signals to be binary integers.

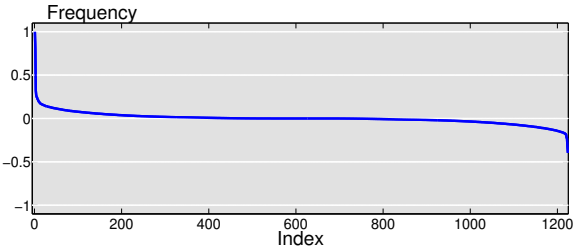



*Figure 3.11: Graph frequencies.*

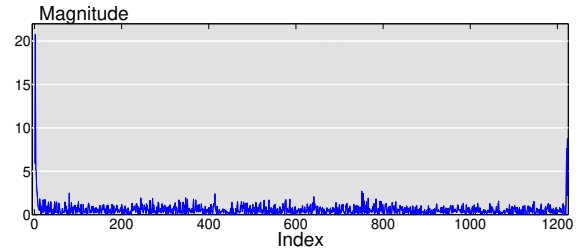



*Figure 3.12: Frequency content of the labeling signal.*

### Experimental setup & results

To investigate the success rate of perfect recovery using random sampling, we vary the bandwidth $K$ of the labeling signal with an interval of 1 from 1 to 10, randomly sample $K$ rows from the first $K$ columns of the graph Fourier transform matrix, and check if the $K \times K$ matrix has full rank. For each bandwidth, we random sample10,000 times, and count the number of success to obtain the success rate. Figure 3.13 shows the resulting success rate. We see that the success rates decrease as increasing the bandwidth, but the success rates are all above 90% when the bandwidth is no greater than 20.

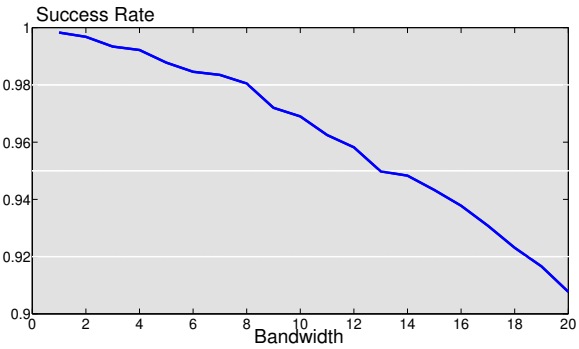



*Figure 3.13: Success rate of online blogs as a function of the bandwidth.*

Since a qualified sampling operator is independent of graph signals, we precompute the qualified sampling operator for the online-blog graph, as discussed in Section 3.1.1. We then sample $M$ labels from the labeling signal by using a qualified sampling operator, and recover the labeling signal by using the corresponding interpolation operator. Since the labeling signal is not bandlimited, it is infeasible to achieve perfect recovery; however, we only care about the sign of labels. We thus set the threshold at zero, so that positive values are set to +1 and negative to 1. Figure 3.14 shows the recovery accuracy by varying the sample size $M$ with an interval of 1 from 1 to 10. We see that the recovery accuracy is as high as 94.44% by sampling only two blogs, and the recovery accuracy gets slightly better as increasing the bandwidth. Comparing to previous results[70], harmonic functions achieve 94.68% by sampling 120 blogs, the graph Laplacian regularization achieves 94.62% by sampling 120 blogs, graph total variation minimization achieves 94.76% by sampling 10 blogs, and graph total variation regularization achieves 94.68% by sampling 10 blogs. We recall that in previous results, we use random sampling; here, we use the qualified sampling operator to choose samples based on the graph structure actively. If we fix the bandwidth to be 2, the success rate, or the probability to get a qualified sampling operator by random sampling, is 99.68%, and the recovery accuracy is 94.44%; in other words, we achieve 94.44% recovery accuracy in a high probability by random sampling.

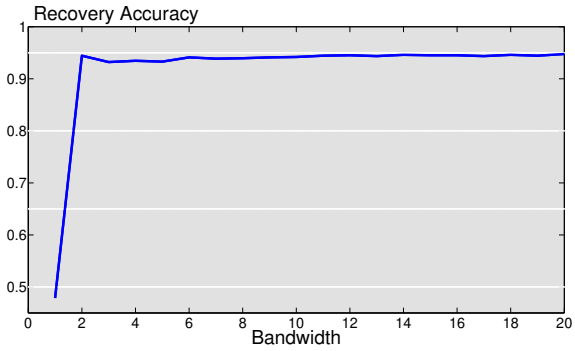



*Figure 3.14: Recovery accuracy of online blogs as a function of the bandwidth.*

In this section, we test our randomized sampling framework on the product graph $\mathbf{A} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)}$ composed over two graph atoms where $\mathbf{A}^{(1)}$ is the Minnesota road graph[71] with $N_1 = 2642$ nodes which we randomly sample according to the optimal sampling distribution illustrated in the heatmap in Figure 3.15 and the path graph ($N_2 = 8$) which we can uniformly randomly sample. We set $K = 100$ such that $R_1 = 40$ and $R_2 = 3$ and $S = R_1 \times R_2 = 120$ and generate a synthetic bandlimited signal on $\mathbf{A}$ with respect to $\mathbf{V}_{(K)}$. We perform the experiment over varying noise settings by injecting the true signal with white gaussian noise such that the noisy signal we sample from has SNR of 5dB, 10dB or 15dB. We sample $M_1$ nodes on $\mathbf{A}^{(1)}$ and $M_2$ nodes on $\mathbf{A}^{(2)}$ with $\Psi^{(1)}$ and $\Psi^{(2)}$ respectively and compose the sampling operator $\Psi = \Psi^{(1)} \otimes \Psi^{(2)}$ to sample $M = M_1 \cdot M_2$ nodes on the product graph $\mathbf{A}$. We recover using the interpolation operators $\Phi^{(1)}$ and $\Phi^{(2)}$ corresponding to each graph atom as described in Algorithm 1. We illustrate the performance of our framework which is consistent with our theoretical analysis in Figure 3.15 where we plot the reconstruction SNR versus the size of the sample set $M$ averaged over 20 iterations.

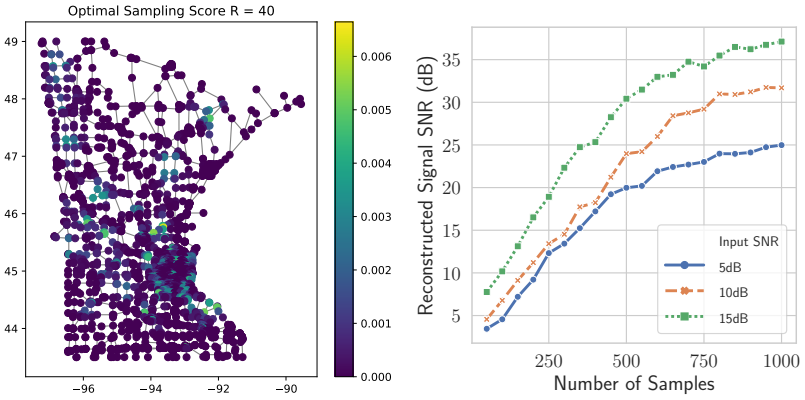



*Figure 3.15: (a) Optimal sampling scores 3.45 for* $\mathbf{A}^{(1)}$ *with* $R_1 = 40$ *(b) Reconstructed signal SNR vs. number of samples*

## 3.4 Fundamental Statistical Limits of Sampling Strategies

We now propose a new class of graph signals, called *approximately bandlimited*, and build a theoretical foundation for understanding the recovery of this class under random sampling, experimentally designed sampling, and active sampling. We propose two recovery strategies under random sampling and experimentally designed sampling, which are unbiased estimators for low frequency components. Both recovery strategies achieve the optimal rate of convergence for each sampling scenario. Our work follows the previous works that studied the theoretical capabilities of passive sampling and active sampling for recovering functions from samples[72,73]. The main difference is that we consider a discrete case and deal with irregular structures. For smooth function, active sampling, experimentally designed sampling, and random sampling have the same performance[72]; however, for the approximately bandlimited class, active sampling has the same rate of convergence with experimentally designed sampling; and experimentally designed sampling outperforms random sampling when supported graphs are irregular.

To validate the recovery strategies, we test on six specific graphs, including a ring graph with $k$ nearest neighbors, an Erdős-Rényi graph, a generalized random key graph, a preferential-attachment graph, a real-world graph from Wikipedia, and a random geometric graph. Surprisingly, the proposed theorems and the experimental results agree that experimentally designed sampling can outperform random sampling on a graph where nodes have similar degrees. This work also shows that graph signal processing is a good framework to study graph structures, and shows a comprehensive explanation that when and why anchor points for clustering and semi-supervised learning on graphs work.

**Contributions.** Our contributions are as follows: we propose

- a new class of smooth graph signals and reveal the relations to existing classes of smooth graph signals;
- minimax lower bounds of the recovery error under three sampling strategies;
- recovery strategies based on random sampling and experimentally designed sampling that achieves optimal rates of convergence;
- a generalized random key graph that has strong cluster patterns;
- an analysis of graph structures from the perspective of signal processing; and
- a comprehensive study that when and why anchor points for clustering and semi-supervised learning on graphs work.

**Outline:.** Section 3.4.1 review the smooth graph signal models and formulate the sampling and recovery strategies ; Section 3.4.2 proposes the minimax lower bound of recovery error; Section 3.4.3 proposes two recovery strategies based on random sampling and experimentally designed sampling; Section 3.4.6 shows the optimal convergence rates of recovery on two types of graphs. The proposed recovery strategies are evaluated in Section 3.5 on six graphs. We then discuss and provides pointers to future directions.

### 3.4.1 Problem Formulation

We now review three classes of smooth graph signals, and show there connections. We next describe the sampling and recovery strategies of interest. In this way, we show the connection between our work and the previous work: graph signal inpainting and sampling theory on graphs.

#### Graph Signal Model

We focus on smooth graph signals, that is, the signal coefficient at each node is close to the signal coefficients of its neighbors. In literature[74,66], two classes of graph signals have been introduced to measure the smoothness on graphs.

**Definition 8.** A graph signal $\mathbf{x} \in \mathbb{R}^N$ is *globally smooth* on a graph $\mathbf{A} \in \mathbb{R}^{N\times N}$ with parameter $\eta \geq 0$, when

$$\|\mathbf{x} - \mathbf{A}\,\mathbf{x}\|_2^2 \;\leq\; \eta\,\|\mathbf{x}\|_2^2\,. \tag{3.48}$$

Denote this class of graph signals by $\mathrm{GS}_{\mathbf{A}}(\eta)$.

Since we normalized the graph shift such that $|\lambda_{\max}(\mathbf{A})| = 1$; when $\eta \geq 4$, all graph signals satisfy (3.48).

**Definition 9.** A graph signal $\mathbf{x} \in \mathbb{R}^N$ is *bandlimited* on a graph $\mathbf{A}$ with parameter $K \in \{0, 1, \cdots, N-1\}$, when the graph frequency components $\widehat{\mathbf{x}}$ satisfies

$$\widehat{x}_k \;=\; 0 \quad \text{for all} \quad k \geq K.$$

Denote this class of graph signals by $\mathrm{BL}_{\mathbf{A}}(K)$.

Note that the original definition just requires $\widehat{\mathbf{x}}$ be $K$-sparse, which is unnecessarily smooth[66]. To show the relations to these two classes, we present the following theorem.

**Theorem 13.** For any $K \in \{0, 1, \cdots, N-1\}$, $\mathrm{BL}_{\mathbf{A}}(K)$ is a subset of $\mathrm{GS}_{\mathbf{A}}(\eta)$, when $\eta \geq (1-\lambda_K)^2$.

*Proof.* Let $\mathbf{x}$ be a graph signal with bandwidth $K$, that is,

$$\mathbf{x} = \sum_{k=0}^{K-1} \widehat{x}_k \mathbf{v}_k,$$

Then, we have

$$\begin{aligned}
|x_i - \sum_{j\in\mathcal{N}_i} \mathbf{A}_{i,j}\, x_j| \;&=\; |\left(\sum_{k=0}^{K-1} \widehat{x}_k \mathbf{v}_k\right)_i - \sum_{j\in\mathcal{N}_i} \mathbf{A}_{i,j} \left(\sum_{k=0}^{K-1} \widehat{x}_k \mathbf{v}_k\right)_j |\\
&=\; |\sum_{k=0}^{K-1} \widehat{x}_k \left((\mathbf{v}_k)_i - \sum_{j\in\mathcal{N}_i} \mathbf{A}_{i,j} (\mathbf{v}_k)_j\right)|\\
&=\; |\sum_{k=0}^{K-1} \widehat{x}_k (1-\lambda_k)(\mathbf{v}_k)_i|\\
&\leq\; (1-\lambda_{K-1})|\sum_{k=0}^{K-1} \widehat{x}_k \cdot (\mathbf{v}_k)_i| \;=\; (1-\lambda_{K-1})|x_i|.
\end{aligned}$$

It is clear that for bandlimited graph signals, the signal coefficient at each node is close to the weighted average of all its neighbors; in other words, bandlimited graph signals are smooth locally, which implies global smoothness,

$$\begin{aligned} \|\mathbf{x} - \mathbf{A}\,\mathbf{x}\|_2^2 &= \sum_{i=0}^{N-1} |x_i - \sum_{j \in \mathcal{N}_i} \mathbf{A}_{i,j}\, x_j|^2 \\ &\leq \sum_{i=0}^{N-1} (1-\lambda_{K-1})^2 |x_i|^2 \;=\; (1-\lambda_{K-1})^2\, \|\mathbf{x}\|_2^2 \,. \end{aligned}$$

It is obvious that a globally smooth graph signal can have arbitrary high-frequency components, thus, the bandlimited class is a subset of the globally smooth class. □

While the recovery of globally smooth graph signals has been studied in [74] (leading to graph signal inpainting), global smoothness is a general requirement, making it hard to provide further theoretical insight [75]. While the recovery of bandlimited graph signals has been studied in [66] (leading to sampling theory on graphs), the bandlimited requirement is a restricted requirement, making it hard to use in the real world applications. A third class of graph signals is thus proposed to relaxes the the bandlimited requirement, but still promotes smoothness [76].

**Definition 10.** A graph signal $\mathbf{x} \in \mathbb{R}^N$ is *approximately bandlimited* on a graph $\mathbf{A}$ with parameters $\beta \geq 1$ and $\mu \geq 0$ , when there exists a $K \in \{0, 1, \cdots, N-1\}$ such that its graph Fourier transform $\widehat{\mathbf{x}}$ satisfies

$$\sum_{k=K}^{N-1} (1 + k^{2\beta}) \widehat{x}_k^2 \leq \mu\, \|\mathbf{x}\|_2^2 \,. \tag{3.49}$$

Denote the class of graph signals by $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$.

The approximately bandlimited class allows a tail after the first $K$ frequency components. The parameter $\mu$ controls the shape of the tail. When $\mu$ is smaller, we allow fewer energy from the high frequency components. The parameter $\beta$ controls the speed of energy decaying. When $\beta$ is larger, we punish the energy from high frequency components more. The class of $\mathrm{BL}_{\mathbf{A}}(K)$ is similar to the ellipsoid constraints in previous literature, where all the frequency components are considered in the constraints; in other words, $\mathrm{ABL}_{\mathbf{A}}(K)$ provides more flexibility for the low frequency components.

The following theorem shows the relationship between $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$ and $\mathrm{GS}_{\mathbf{A}}(\eta)$.

**Theorem 14.** $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$ is a subset of $\mathrm{GS}_{\mathbf{A}}(\eta)$, when

$$\eta \geq \left(1 - \lambda_{K-1} + \sqrt{\frac{4\alpha_2 \mu}{(1+K^{2\beta})}}\right)^2 ;$$

$\mathrm{GS}_{\mathbf{A}}(\eta)$ is a subset of $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$, when

$$\mu \geq \frac{1 + (N-1)^{2\beta}}{(1-\lambda_K)\alpha_1} \eta.$$

From Theorem 14, we see that when choosing proper parameters, $\mathrm{GS}_{\mathbf{A}}(\eta)$ is a subset of $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$.

*Proof.* To show the first statement, let $\mathbf{x} \in \mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$, we have

$$\begin{aligned}
\|\mathbf{x} - \mathbf{A}\,\mathbf{x}\|_2 &= \left\| \left( \sum_{k=0}^{K-1} \widehat{x}_k \mathbf{v}_k + \sum_{k=K}^{N-1} \widehat{x}_k \mathbf{v}_k \right) - \mathbf{A} \left( \sum_{k=0}^{K-1} \widehat{x}_k \mathbf{v}_k + \sum_{k=K}^{N-1} \widehat{x}_k \mathbf{v}_k \right) \right\|_2 \\
&\overset{(a)}{\leq} \left\| \sum_{k=0}^{K-1} \widehat{x}_k \mathbf{v}_k - \mathbf{A} \sum_{k=0}^{K-1} \widehat{x}_k \mathbf{v}_k \right\|_2 + \left\| \sum_{k=K}^{N-1} \widehat{x}_k \mathbf{v}_k - \mathbf{A} \sum_{k=K}^{N-1} \widehat{x}_k \mathbf{v}_k \right\|_2 \\
&\overset{(b)}{\leq} (1 - \lambda_{K-1}) \|\mathbf{x}\|_2 + \sqrt{\alpha_2^2 \sum_{k=K}^{N-1} (1 - \lambda_k)^2 \widehat{x}_k^2} \\
&= (1 - \lambda_{K-1}) \|\mathbf{x}\|_2 + \sqrt{\alpha_2 \sum_{k=K}^{N-1} \frac{(1 - \lambda_k)^2}{(1 + k^{2\beta})} (1 + k^{2\beta}) \widehat{x}_k^2} \\
&\leq (1 - \lambda_{K-1}) \|\mathbf{x}\|_2 + \sqrt{\alpha_2 \max_{k \in \{K, \cdots, N-1\}} \frac{(1 - \lambda_k)^2}{(1 + k^{2\beta})} \sum_{k=K}^{N-1} (1 + k^{2\beta}) \widehat{x}_k^2} \\
&\overset{(c)}{\leq} (1 - \lambda_{K-1}) \|\mathbf{x}\|_2 + \sqrt{\alpha_2 \max_{k \in \{K, \cdots, N-1\}} \frac{(1 - \lambda_k)^2}{(1 + k^{2\beta})} \mu \|\mathbf{x}\|_2^2} \\
&\leq \left( 1 - \lambda_{K-1} + \sqrt{\frac{4\alpha_2 \mu}{(1 + K^{2\beta})}} \right) \|\mathbf{x}\|_2
\end{aligned}$$

where $(a)$ follows from the triangle inequality, $(b)$ from Theorem 13 and (3.1), and $(c)$ from the property of $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$. To show the second statement, let $\mathbf{x} \in \mathrm{GS}_{\mathbf{A}}(\eta)$, we have

$$\begin{aligned}
\sum_{k=K}^{N-1} (1 + k^{2\beta}) \widehat{x}_k^2 &= \sum_{k=K}^{N-1} \frac{1 + k^{2\beta}}{(1 - \lambda_k)^2} (1 - \lambda_k)^2 \widehat{x}_k^2 \\
&\leq \max_{k \in \{K, \cdots, N-1\}} \frac{1 + k^{2\beta}}{(1 - \lambda_k)^2} \sum_{k=K}^{N-1} (1 - \lambda_k)^2 \widehat{x}_k^2 \\
&\leq \frac{1 + (N-1)^{2\beta}}{(1 - \lambda_K)^2 \alpha_1} \eta \|\mathbf{x}\|_2^2
\end{aligned}$$

where the last inequality follows from (3.1). □

From Theorem 14, we see that $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$ is not only more general than $\mathrm{BL}_{\mathbf{A}}(K)$, but describes $\mathrm{GS}_{\mathbf{A}}(\eta)$ in a more controlled way. In this work, we focus on $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$, and study the recovery performance of this class under various sampling scenarios.

### Sampling & Recovery Strategy

We consider the procedure of sampling and recovery as follows: we sample $M$ coefficients in a graph signal $\mathbf{x} \in \mathbb{R}^N$ with noise to produce a noisy sampled signal $\mathbf{y} \in \mathbb{R}^M (M < N)$, that is,

$$\mathbf{y} = \Psi \mathbf{x} + \epsilon \equiv \mathbf{x}_{\mathcal{M}} + \epsilon, \tag{3.50}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_{M \times M})$, and $\mathcal{M} = (\mathcal{M}_0, \cdot, \mathcal{M}_{M-1})$ denotes the sequence of sampled indices, or called *sampling set*, and $\mathcal{M}_i \in \{0, 1, \cdots, N-1\}$, and the sampling operator $\Psi$ is a linear mapping

from $\mathbb{R}^N$ to $\mathbb{R}^M$, defined as

$$\Psi_{i,j} = \begin{cases} 1, & j = \mathcal{M}_i; \\ 0, & \text{otherwise.} \end{cases} \tag{3.51}$$

We then interpolate $\mathbf{y}$ to get $\mathbf{x}' \in \mathbb{R}^N$, which recovers $\mathbf{x}$ either exactly or approximately.

We consider three different sampling strategies: *random sampling* means that sample indices are chosen from from $\{0, 1, \cdots, N-1\}$ independently and randomly; and *experimentally design sampling* means that sample indices can be chosen beforehand; and *active sampling* means that sample indices can be chosen as a function of the sample points and the samples collected up to that instance, that is, $\mathcal{M}_i$ depends only on $\{\mathcal{M}_j, y_j\}_{j<i}$. It is clear that random sampling is a subset of experimentally design sampling, which is again a subset of active sampling.

### 3.4.2 Minimax Lower Bounds

In this section, we study the fundamental limitations of three sampling strategies for recovering $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$ by showing the minimax lower bounds.

We start by introducing some notations[72].

**Definition 11.** For any recovery strategy $(\mathbf{x}^*, \mathcal{M})$, and any vector $\mathbf{x} \in \mathbb{R}^N$, we define the risk of the recovery strategy as

$$R(\mathbf{x}^*, \mathcal{M}, \mathbf{x}) \;=\; \mathbb{E}_{\mathbf{x},\mathcal{M}}[d^2(\mathbf{x}^*, \mathbf{x})],$$

where $\mathbb{E}_{\mathbf{x},\mathcal{M}}$ is the expectation with respect to the probability measure of $\{x_i, y_i\}_{i\in\mathcal{M}}$. We define the maximal risk of a recovery strategy as $\sup_{\mathbf{x}\in\mathbb{R}^N} R(\mathbf{x}^*, \mathcal{M}, \mathbf{x})$.

The goal of this section is to find tight lower bounds for the maximal risk, over all possible recovery strategies, that is, we present bounds of the form,

$$\inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta} \sup_{\mathbf{x}\in\mathbb{R}^N} \mathbb{E}_{\mathbf{x},\mathcal{M}}[d^2(\mathbf{x}^*, \mathbf{x})] \geq c\phi_n^2, \quad \forall\, n \geq n_0, \tag{3.52}$$

where $n_0 \in \mathbb{N}, c > 0$ is a constant, $\phi_n$ is a positive sequence converging to zero, and $\Theta$ is the set of all recovery strategies. The sequence $\phi_n^2$ is denoted as a lower rate of convergence. It is also possible to devise upper bounds on the maximal risk. These are usually obtained through explicit recovery strategies, as will be presented in Section 3.4.3. If (3.52) and

$$\inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta} \sup_{\mathbf{x}\in\mathbb{R}^N} \mathbb{E}_{\mathbf{x},\mathcal{M}}[d^2(\mathbf{x}^*, \mathbf{x})] \leq C\phi_n^2, \quad \forall\, n \geq n_0,$$

hold, where $C > 0$, then $\phi_n$ is said to be the optimal rate of convergence. When talking about optimal rates of convergence, we are interested in the polynomial behavior, a rate of convergence $\phi_n^2$ is equivalent to $n^{-\gamma}$, if and only if given $\gamma_1 < \gamma < \gamma_2$, we have $n^{-\gamma_2} < \phi_n^2 < n^{-\gamma_1}$ for $n$ large enough.

Since we propose general bounds for arbitrary graphs, these bounds involve some parameters that depend on the graph structure, thus, we cannot show lower rates of convergence in this general case. Given a graph structure, we can specify the parameters, and then show the lower rate of convergence based on these general bounds, as will be presented in Section 3.5.

Denote $|\mathcal{M}|$ be the size of the sampling set and $\mathbf{V}_{(2,K)}$ be the sub-matrix of $\mathbf{V}$, containing the $(K+1)$th to the $2K$th columns.

**Theorem 15.** For the class $\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)$, we have the following results.

(1) under the requirements of the random sampling model, we have

$$\begin{aligned}
&\inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\mathrm{rand}}}\sup_{\mathbf{x}\in\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)}\mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\left\|\mathbf{x}^*-\mathbf{x}\right\|_2^2\right)\\
\geq\ &\max_{K\leq\kappa_0\leq N}\frac{c_1\mu\left\|\mathbf{x}\right\|_2^2}{\kappa_0^{2\beta}}\left(1-\frac{c\mu\left\|\mathbf{x}\right\|_2^2}{\sigma^2\kappa_0^{2\beta+2}N}\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2|\mathcal{M}|\right),
\end{aligned}$$

where $c_1>0$ , $0<c<1$, and $\Theta_{\mathrm{rand}}$ denotes the set of all recovery strategies based on random sampling;

(2) under the requirements of the experimentally designed sampling model, we have

$$\begin{aligned}
&\inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\mathrm{exp}}}\sup_{\mathbf{x}\in\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)}\mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\left\|\mathbf{x}^*-\mathbf{x}\right\|_2^2\right)\\
\geq\ &\max_{K\leq\kappa_0\leq N}\frac{c_1\mu\left\|\mathbf{x}\right\|_2^2}{\kappa_0^{2\beta}}\left(1-\frac{c\mu\left\|\mathbf{x}\right\|_2^2}{\sigma^2\kappa_0^{2\beta+2}}\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{\infty,2}^2|\mathcal{M}|\right),
\end{aligned}$$

where $c_1>0$ , $0<c<1$, and $\Theta_{\mathrm{exp}}$ denotes the set of all recovery strategies based on experimentally designed sampling;

(3) under the requirements of the active sampling model, we have

$$\begin{aligned}
&\inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\mathrm{active}}}\sup_{\mathbf{x}\in\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)}\mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\left\|\mathbf{x}^*-\mathbf{x}\right\|_2^2\right)\\
\geq\ &\max_{K\leq\kappa_0\leq N}\frac{c_1\mu\left\|\mathbf{x}\right\|_2^2}{\kappa_0^{2\beta}}\left(1-\frac{c\mu\left\|\mathbf{x}\right\|_2^2}{\sigma^2\kappa_0^{2\beta+2}}\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{\infty,2}^2|\mathcal{M}|\right),
\end{aligned}$$

where $c_1>0$ , $0<c<1$, and $\Theta_{\mathrm{active}}$ denotes the set of all recovery strategies based on active sampling.

For the proof, see Appendix A. From Theorem 15, we see that experimentally designed sampling has the same minimax lower bound as active sampling, which means that collecting the feedback before taking samples does not improve the fundamental limitation; we also see that the three minimax lower bounds depend on the properties of $\mathbf{V}_{(2,\kappa_0)}$, which represents the graph structure. When each row of $\mathbf{V}_{(2,\kappa_0)}$ has roughly similar energies, $\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2$ and $N\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{\infty,2}^2$ are similar; when the energy is concentrated in a few rows, $N\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{\infty,2}^2$ is much larger than $\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2$, in other words, the minimax lower bound of experimental designed sampling is tighter than that of random sampling. This happens in many real-world graphs that have complex, irregular structure. The minimax lower bounds thus show the potential advantage of experimentally designed sampling and active sampling over random sampling. We will elaborate in Section 3.4.6 and 3.5

### 3.4.3 Recovery Strategy

We now propose two recovery strategies based on random sampling and experimentally designed sampling. Since we cannot hope to perform better than the experimentally designed sampling, we do not need a recovery strategy for active sampling. In Section 3.4.1, we showed that a graph

signal is smooth when its energy is mainly concentrated in the low-frequency components. For example, for the class $\mathrm{BL}_{\mathbf{A}}(K)$, all the energy is concentrated in the first $K$ frequency components and the graph signal can be perfectly recovered by using those first $K$ frequency components. The recovery strategies we propose here follow this intuition, by providing unbiased estimators for the low-frequency components.

### Recovery Strategy based on Random Sampling

We consider the following recovery strategy.

**Algorithm 2.** We sample a graph signal $|\mathcal{M}|$ times. Each time, we choose a node $i$ independently and randomly, and take a measurement $y_i$. We then recover the original graph signal by using the following two steps:

$$\begin{aligned} \widehat{x}_k^* &= \frac{N}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \mathbf{U}_{ki}\, y_i, \\ x_i^* &= \sum_{k<\kappa} \mathbf{V}_{ik}\, \widehat{x}_k^*, \end{aligned}$$

where $x_i^*$ is the $i$th component of the recovered graph signal $\mathbf{x}^*$.

Algorithm 2 aims to estimate the first $\kappa$ frequency components, and reconstruct the original graph signal based on these graph frequency components. The only tuning parameter in Algorithm 2 is the bandwidth $\kappa$.

To show the performance of Algorithm 2 for recovering the low-frequency components, we have the following results. Denote $\mathbf{V}_{(\kappa)}$ be the first $\kappa$ columns of the inverse graph Fourier transform matrix $\mathbf{V}$, and $\mathbf{U}_{(\kappa)}$ be the first $\kappa$ rows of the graph Fourier transform matrix $\mathbf{U}$.

**Lemma 3.** Algorithm 2 with bandwidth $\kappa$ provides an unbiased estimator of the first $\kappa$ frequency components, that is,

$$\mathbb{E}\mathbf{x}^* \;=\; \mathbf{V}_{(\kappa)}\, \mathbf{U}_{(\kappa)}\, \mathbf{x}, \quad \text{for all } \mathbf{x},$$

where $\mathbf{x}^*$ is the result of Algorithm 2.

We can further show an upper bound on the recovery error.

**Theorem 16.** For $\mathbf{x} \in \mathrm{ABL}(K, \beta, \mu)$, let $\mathbf{x}^*$ be the result of Algorithm 2 with bandwidth $\kappa \geq K$, we have,

$$\mathbb{E}\left\|\mathbf{x}^* - \mathbf{x}\right\|^2 \;\leq\; \frac{\alpha_2 \mu \left\|\mathbf{x}\right\|_2^2}{\kappa^{2\beta}} + \frac{\alpha_2(\max_j x_j^2 + \sigma^2)}{|\mathcal{M}|} N \left\|\mathbf{U}_{(\kappa)}\right\|_F^2,$$

where $\alpha_2$ is the stability constant of $\mathbf{V}$ in (3.1), $\sigma^2$ is the noise level in (5.1), and $\|\cdot\|_F$ is the Frobenius norm.

The proofs of Lemma 3 and Theorem 16 are merged in Appendix B. The main idea follows from the bias-variance tradeoff. The first term is the bias term, and the second terms is the variance term. Since Algorithm 2 can recover the first $\kappa$ frequency components on expectation, the

bias comes from the other $(N-\kappa)$ frequency components, which can be bounded from the definition of $\mathrm{ABL}(K,\mu,\beta)$ when $\kappa \geq K$. The variance term depends on $\left\|\mathbf{U}_{(\kappa)}\right\|_F^2$, which represents the graph structure.

The advantage of Algorithm 2 is its efficiency, that is, we only need the first $\kappa$ eigenvectors involved with the computation, which is appealing for large-scale graphs; the disadvantage is that when the main energy of an original graph signal is not concentrated in the first $\kappa$ frequency components, the recovered graph signal has a large bias.

#### 3.4.4 Recovery Strategy based on Experimentally Designed Sampling

For experimentally designed sampling, we consider the following recovery strategy.

**Algorithm 3.** We sample a graph signal $|\mathcal{M}|$ times. Each time, we choose a node with probability $w_i = \|i\|_2 / \sum_{j=0}^{N-1} \|j\|_2$, where $i$ is the $i$th column of $\mathbf{U}_{(\kappa)}$, and take a measurement $y_i$. We then recover the original graph signal by using the following two steps:

$$\begin{aligned} \widehat{x}_k^* &= \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \frac{1}{w_i} \mathbf{U}_{ki}\, y_i, \\ x_i^* &= \sum_{k<\kappa} \mathbf{V}_{ik}\, \widehat{x}_k^*. \end{aligned}$$

where $x_i^*$ is the $i$th component of the recovered graph signal $\mathbf{x}^*$.

Similarly to Algorithm 2, Algorithm 5 aims to estimate the first $\kappa$ frequency components, and reconstructs the original graph signal based on these frequency components. The difference comes from the normalization factor. In Algorithm 2, the contribution from each measurement is normalized by a constant, the size of the graph; and in Algorithm 5, the contribution from each measurement is normalized based on the norm of the corresponding column in $\mathbf{U}_{(\kappa)}$. It is similar to leverage scores used in matrix approximation[77], where the goal is to evaluate the contribution from each column to approximating matrix. Note that leverage scores use the norm square, $\|i\|_2^2$, and we use the norm, $\|i\|_2$. When we use the squared norm as probability, the performance is the same as random sampling. We call the probability $w_i$ as the *sampling score* for the $i$th node.

We can show that Algorithm 5 is also an unbiased estimator for recovering the low-frequency components, and potentially has a tighter upper bound.

**Lemma 4.** Algorithm 5 with bandwidth $\kappa$ provides an unbiased estimator of the first $K$ frequency components, that is,

$$\mathbb{E}\mathbf{x}^* = \mathbf{V}_{(\kappa)}\, \mathbf{U}_{(\kappa)}\, \mathbf{x}, \quad \forall\, \mathbf{x},$$

where $\mathbf{x}^*$ is the result of Algorithm 5.

We can further show an upper bound on the recovery error.

**Theorem 17.** For $\mathbf{x} \in \mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)$, let $\mathbf{x}^*$ be the result of Algorithm 5 with bandwidth $\kappa \geq K$, we have,

$$\mathbb{E}\left\|\mathbf{x}^* - \mathbf{x}\right\|^2 \leq \frac{\alpha_2 \mu \|\mathbf{x}\|_2^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2 + \sigma^2)\alpha_2}{|\mathcal{M}|} \left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2.$$

The proofs of Lemma 3 and Theorem 16 are merged in Appendix C. The main idea also follows from the bias-variance tradeoff.

We see that Algorithm 2 and Algorithm 5 have the same bias by recovering the first $\kappa$ frequency components on expectation. When each column of $\mathbf{U}_{(\kappa)}$ has roughly similar energy, $N\left\|\mathbf{U}_{(\kappa)}\right\|_F^2$ and $\left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2$ are similar. However, when the energy is concentrated on a few columns, $N\left\|\mathbf{U}_{(\kappa)}\right\|_F^2$ is much larger than $\left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2$, in other words, Algorithm 5 has a significant advantage over Algorithm 2 when the associated graph structure is irregular.

#### Relation to Graph Signal Inpainting

Graph signal inpainting via variation minimization also aims at recovering smooth graph signals by limited samples[70]. We consider a globally smooth graph signal with the random sampling model, with the following optimization problem as the recovery strategy,

$$\begin{aligned} \mathbf{x}^* &= \arg\min_{\mathbf{x}} \|\mathbf{x}-\mathbf{A}\,\mathbf{x}\|_2^2, && \text{(3.53a)} \\ &\quad \text{subject to} \quad \|\Psi\mathbf{x}-\mathbf{y}\|_2^2 \leq \sigma^2, && \text{(3.53b)} \end{aligned}$$

where $\sigma^2$ is noise level, $\mathbf{y}$ is a vector representation of the noisy measurements (5.1), and $\Psi$ is the sampling operator (5.2). The main difference is that graph signal inpainting via variation minimization focuses on recovery in the vertex domain, and the proposed algorithms focus on recovery in graph spectral domain. The optimum of (3.53) cannot guarantee recovery of the low-frequency components, but it guarantees that the recovered graph signal is close to the measurements. When the noise level is large, we cannot trust these measurements, and then we tend to use Algorithm 2 and 5 to do linear approximation and capture the main shape of the original graph signal. An advantage of (3.53) is that it can be implemented in a distributed manner easily[78,79].

### 3.4.5 Relation to Sampling Theory on Graphs

Sampling theory on graphs considers a bandlimited graph signal with both the random and the experimentally designed sampling models[66]. It shows that for a noiseless bandlimited graph signal, the experimentally designed sampling guarantees perfect recovery, while random sampling cannot, which also implies that active sampling cannot perform better than experimentally designed sampling. The recovery strategy is to solve the following optimization problem,

$$\begin{aligned} \mathbf{x}^* &= \arg\min_{\mathbf{x}\in \mathrm{BL}_{\mathbf{A}}(K)} \|\Psi\mathbf{x}-\mathbf{y}\|_2^2 \\ &= \mathbf{V}_{(K)}(\Psi\,\mathbf{V}_{(K)})^{+}\mathbf{y}, \end{aligned} \tag{3.54}$$

where $\Psi$ is the sampling operator (5.2), $\mathbf{y}$ is a vector representation of the measurements (5.1), and $(\cdot)^+$ is the pseudo-inverse. When the original graph signal is bandlimited, $\mathbf{x} \in \mathrm{BL}_{\mathbf{A}}(K)$, it is clear that the result of (3.54) is an unbiased estimator of $\mathbf{x}$, that is,

$$\mathbb{E}\mathbf{x}^* = \mathbf{V}_{(k)}(\Psi\,\mathbf{V}_{(k)})^{+}\mathbb{E}(\Psi\mathbf{x}+\epsilon) = \mathbf{x}.$$

When the original graph signal is not bandlimited, the result of (3.54) is a biased estimator of the first $K$ frequency components, that is,

$$\begin{aligned}\mathbb{E}\mathbf{x}^* &= \mathbf{V}_{(K)}(\Psi\,\mathbf{V}_{(K)})^{+}\mathbb{E}(\Psi\mathbf{x}+\epsilon)\\ &= \mathbf{V}_{(K)}(\Psi\,\mathbf{V}_{(K)})^{+}\Psi\left(\mathbf{V}_{(K)}\,\widehat{\mathbf{x}}_{(K)}+\mathbf{V}_{(-K)}\,\widehat{\mathbf{x}}_{(-K)}\right)\\ &= \mathbf{V}_{(K)}\,\widehat{\mathbf{x}}_{(K)}+\mathbf{V}_{(K)}(\Psi\,\mathbf{V}_{(K)})^{+}\,\mathbf{V}_{(-K)}\,\widehat{\mathbf{x}}_{(-K)},\end{aligned}$$

where $\mathbf{V}_{(-K)}$ is $\mathbf{V}$ expect for the first $K$ columns, $\widehat{\mathbf{x}}_K$ is the first $K$ components of $\mathbf{x}$, and $\widehat{\mathbf{x}}_{-K}$ is $\mathbf{x}$ expect for the first $K$ components. We see that the signal belonging to the other frequency band also projects onto the first $K$ components. In a sense of recovering the low-frequency components, (3.54) needs fewer samples, but Algorithms 2 and 5 are more reliable.

### 3.4.6 Optimal Rates of Convergence

To discriminate the proposed recovery strategies, we propose two types of graphs, and show that the proposed recovery strategies achieve the optimal rates of convergence on these two.

### 3.4.7 Type-1 Graph

**Definition 12.** A graph $\mathbf{A}\in\mathbb{R}^{N\times N}$ is *type-1*, when

$$|\,\mathbf{V}_{i,j}\,|=O(N^{-1/2}),\quad |\,\mathbf{U}_{i,j}\,|=O(N^{-1/2}),\quad \text{for all } i,j=0,1,\cdots,N-1,$$

where $\mathbf{V},\mathbf{U}$ are the the inverse graph Fourier transform matrix, the graph Fourier transform matrix of $\mathbf{A}$.

For a type-1 graph, each element in $\mathbf{V}$ and $\mathbf{U}$ has roughly similar magnitudes, that is, the energy evenly spread to each element in $\mathbf{V}$, and $\mathbf{U}$. Some examples are discrete-time graphs, discrete-space graphs, and nearest-neighbor graphs.

Based on Theorem 16, we can specify the parameters for a type-1 graph and show the following result.

**Corollary 2.** Let $\mathbf{A}\in\mathbb{R}^{N\times N}$ be a type-1 graph, for the class $\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)$.

- Let $\mathbf{x}^*$ be the results given by Algorithm 2 with the bandwidth $\kappa\geq K$, we have

  $$\mathbb{E}\left(\|\mathbf{x}^*-\mathbf{x}\|_2^2\right)\;\leq\; C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},$$

  where $C>0$, and the rate is achieved when $\kappa$ is in the order of $|\mathcal{M}|^{1/(2\beta+1)}$ and upper bounded by $N$;

- Let $\mathbf{x}^*$ be the results given by Algorithm 5 with the bandwidth $\kappa\geq K$, we have

  $$\mathbb{E}\left(\|\mathbf{x}^*-\mathbf{x}\|_2^2\right)\;\leq\; C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},$$

  where $C>0$, and the rate is achieved when $\kappa$ is in the order of $|\mathcal{M}|^{1/(2\beta+1)}$ and upper bounded by $N$.

When $|\mathcal{M}| \gg N$, we set $\kappa = N$, and then the bias term is zero, and both upper bounds are actually $CN|\mathcal{M}|^{-1}$. We see that Algorithms 2 and 5 have the same convergence rate, that is, experimentally designed sampling does not perform better than random sampling for the type-1 graphs.

Based on Theorem 15 and Corollary 2, we conclude as follows.

**Corollary 3.** Let $\mathbf{A} \in \mathbb{R}^{N\times N}$ be a type-1 graph, for the class $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$.

- Under requirements of the random sampling model, we have

$$c|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}} \leq \inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\mathrm{rand}}} \sup_{\mathbf{x}\in\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}} \left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \leq C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},$$

  where $C > c > 0$, and the rate is achieved when $\kappa$ is in the order of $|\mathcal{M}|^{1/(2\beta+1)}$ and upper bounded by $N$;

- Under requirements of the experimentally designed sampling model, we have

$$c|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}} \leq \inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\exp}} \sup_{\mathbf{x}\in\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}} \left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \leq C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},$$

  where $C > c > 0$, and the rate is achieved when $\kappa$ is in the order of $|\mathcal{M}|^{1/(2\beta+1)}$ and upper bounded by $N$.

The proof of Corollaries 2 and 3 are merged in Appendix D.

We see that in both the random and experimentally designed sampling settings, the lower and upper bounds have the same rate of convergence, which achieves the optimum. In addition, random and experimentally designed sampling have the same optimal rate of convergence and we can conclude that experimentally designed sampling thus does not perform better than random sampling for the type-1 graphs. Both Algorithms 2 and 5 reach the optimal rate of convergence.

#### 3.4.8 Type-2 Graph

**Definition 13.** A graph $\mathbf{A} \in \mathbb{R}^{N\times N}$ is *type-2* with parameter $K_0 > 0$, when

- $\left\|\mathbf{V}_{(2,K)}\right\|_{\infty,2}$ is $O(1)$, for all $K > K_0$, where $\mathbf{V}_{(2,K)}$ is the sub-matrix of $\mathcal{V}$, containing the $(K+1)$th to the $2K$th columns;
- $\left\|h^{(K)}_{T^c}\right\|_1 \leq \alpha \left\|h^{(K)}_{T}\right\|_1$, for all $K \geq K_0$, where $h^{(K)}_i = \sqrt{\sum_{k=0}^{K-1} \mathbf{U}^2_{k,i}}$, $T$ indexes the largest $K$ elements in $h$, $T^c$ indexes the other $(N-K)$ elements, and $\alpha > 0$ is a constant.

A type-2 graph requires that $\mathbf{V}$ and $\mathbf{U}$ to be approximately sparse. When we take the first $K$ rows to form a submatrix, the energy in the submatrix concentrates in a few columns. Note that the second requirement is equivalent to that the sampling scores are approximately sparse. The simulations show that star graphs fall into this type approximately.

Based on Theorems 16, and 17, we conclude the following.

**Corollary 4.** Let $\mathbf{A} \in \mathbb{R}^{N\times N}$ be a type-2 graph with parameter $K_0$, for the class $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$.

- Let $\mathbf{x}^*$ be the results given by Algorithm 2 with the bandwidth $\kappa \geq K$, we have

$$\mathbb{E}\left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \;\leq\; C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},$$

  where $C > 0$, and the rate is achieved when $\kappa$ is in the order of $|\mathcal{M}|^{1/(2\beta+1)}$ and upper bounded by $N$;

- Let $\mathbf{x}^*$ be the results given by Algorithm 5 with the bandwidth $\kappa \geq \max\{K, K_0\}$, we have

$$\mathbb{E}\left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \;\leq\; C|\mathcal{M}|^{-\frac{2\beta}{2\beta+2-\gamma}} \;\leq\; C'|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},$$

  where $C > 0$, the rate is achieved when $\kappa$ is in the order of $|\mathcal{M}|^{1/(2\beta+2-\gamma)}$ and upper bounded by $N$, and

$$\gamma \in [\max\{1, 2\beta + 2 - \frac{\log|\mathcal{M}|}{\log\max\{K, K_0\}}\}, \max\{1, \frac{(2\beta+2)\log N}{(\log N + \log|\mathcal{M}|)}\}].$$

Based on Theorem 15, 16, and 17, we conclude as follows.

**Corollary 5.** Let $\mathbf{A} \in \mathbb{R}^{N\times N}$ be a type-2 graph with parameter $K_0$, for the class $\mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$.

- Under requirements of the random sampling model, we have

$$c|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}} \leq \inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\mathrm{rand}}} \sup_{\mathbf{x}\in\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \leq C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},$$

  where $C > c > 0$, and the rate is achieved when $\kappa$ is in the order of $|\mathcal{M}|^{1/(2\beta+1)}$ and upper bounded by $N$;

- Under requirements of the experimentally designed sampling model, there exists a $\gamma > 1$, we have,

$$c|\mathcal{M}|^{-\frac{2\beta}{2\beta+2-\gamma}} \leq \inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\exp}} \sup_{\mathbf{x}\in\mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \leq C|\mathcal{M}|^{-\frac{2\beta}{2\beta+2-\gamma}},$$

  where $C > 0$, the rate is achieved when $\kappa$ is in the order of $|\mathcal{M}|^{1/(2\beta+2-\gamma)}$ and upper bounded by $N$.

The proof of Corollaries 4 and 5 are merged in Appendix E.

We see that under both the random and experimentally designed sampling settings, the lower and upper bounds have the same rate of convergence, which achieves the optimum. However, experimentally designed sampling has a much larger optimal rate of convergence, and consequently we can conclude that experimentally designed sampling exhibits much better performance than random sampling for this type-2 graph. Only Algorithm 5 reaches the optimal rate of convergence.

### 3.5 Numerical Experiments

In this section, we validate the proposed theorems and recovery strategies on six specific graphs: a ring graph, an Erdős-Rényi graph, a generalized random key graph, a preferential-attachment

graph, a Wikipedia graph, and a random geometric graph. Based on the graph structure, we can label each of them as a type-1 or type-2 graph, and then, for each of these graphs, we compare the empirical performance of recovery strategies based on random and experimentally designed sampling. Surprisingly, the degrees of nodes are not useful to determine whether experimentally designed sampling outperforms random sampling, instead, the weights used in Algorithm 5 is a good indicator!

### 3.5.1 Simulated Graph Signals

For each graph $\mathbf{A}$, we generate 100 graph signals by the following two steps. We first generate the graph frequency components as

$$\widehat{x}_k \begin{cases} \sim \mathcal{N}(1, 0.5^2) & \text{if } k < K, \\ = K^{2\beta}/k^{2\beta} & \text{if } k \geq K. \end{cases} \tag{3.55}$$

We then normalize $\widehat{\mathbf{x}}$ to have norm one, and obtain $\mathbf{x} = \mathbf{V}\widehat{\mathbf{x}}$. It is clear that $\mathbf{x} \in \mathrm{ABL}_{\mathbf{A}}(K, \beta, \mu)$, where $K = 10$ and $\beta$ varies as $0.1, 1$ and 2. During the sampling, we simulate the noise $\epsilon \sim \mathcal{N}(0, 0.01^2)$. In the recovery, we set the bandwidth $\kappa$ to 10 for all the recovery strategies.

### 3.5.2 Type-1 Graph

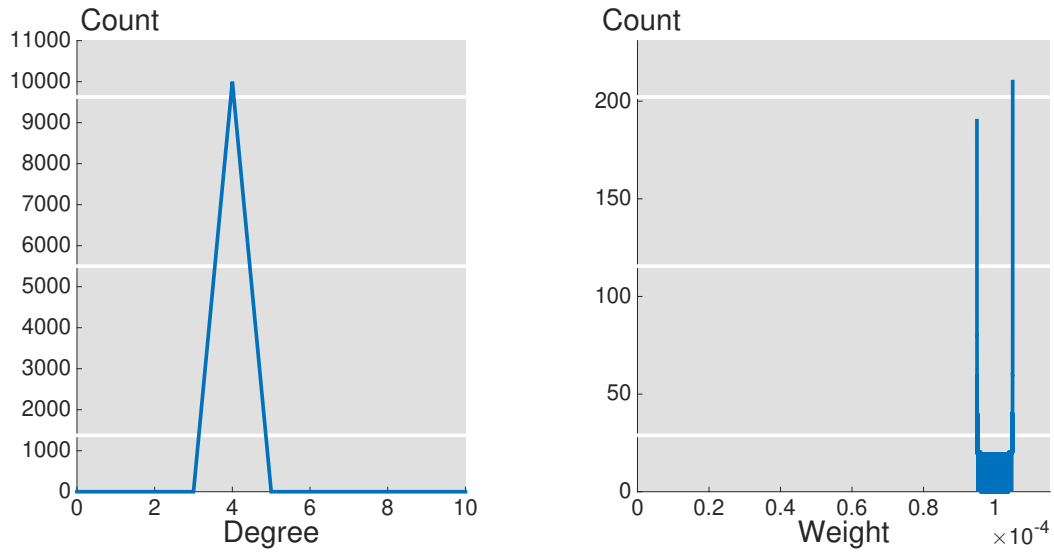
It is generally hard to proof which type a graph belongs to by rigorously following Definition 12, but we can check its graph Fourier transform matrix and label the graph approximately. We find that ring graphs with $k$-nearest neighbors, Erdős-Rényi graphs, and generalized random key graphs belong to type-1. We expect that the recovery strategies based on experimentally designed sampling performs similarly to the recovery strategies based on random sampling.

#### Ring Graph with $k$-nearest Neighbors

We consider a graph with each node connecting to its $k$-nearest neighbors. The eigenvectors are similar to the discrete cosine transform. Since the graph is undirected, $\mathbf{V}$ is orthonormal, $\mathbf{U} = \mathbf{V}^T$. In this case, the energy evenly spreads to each element in $\mathbf{V}, \mathbf{U}$,[80], which approximately follows Definition 12. We simulate a ring graph with 4-nearest neighbors, and generate smooth graph signals as mentioned in Section 3.5.1 on this graph. In the simulation, the ring graph with 4-nearest neighbors has 10,000 nodes, and each node has 4 neighbors.

Figure 3.16 shows the properties of the simulated ring graph. Since every node has 4 neighbors, the degrees are concentrated in 4, and the sampling scores in Algorithm 5 are around $10^{-4}$, which confirms that the graph belongs to type-1.

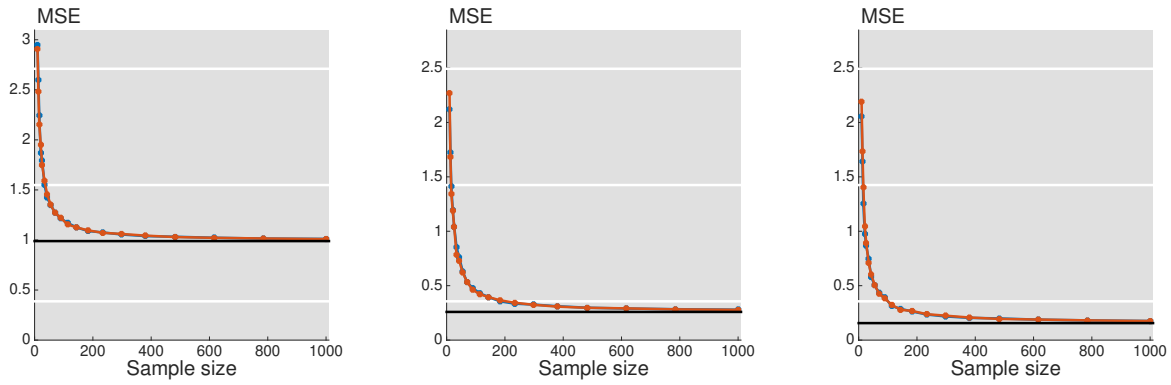
Figure 3.17 compares the performances of Algorithm 2 and 5 with various values of $\beta$ averaged over 100 tests. The blue curve represents Algorithm 2, the red curve represents Algorithm 5, and the black dotted line represented the approximation by the true first $K$ frequency components. In this case, we barely see the blue curve because it overlaps with the red curves. When $\beta$ increases, the fraction of energy from the first $K$ components decreases, and thus, the bias decreases. For each $\beta$, Algorithm 2 and 5 converges to the linear approximation with similar rates. We conclude that the ring graph, as a type-1 graph, results in that the recovery strategies based on experimentally designed sampling performs similarly to the recovery strategies based on random sampling.




(a) Histogram of degrees

(b) Histogram of sampling scores

*Figure 3.16: Properties of the ring graph with 4-nearest neighbors.*




(c) $\beta = 0.1$

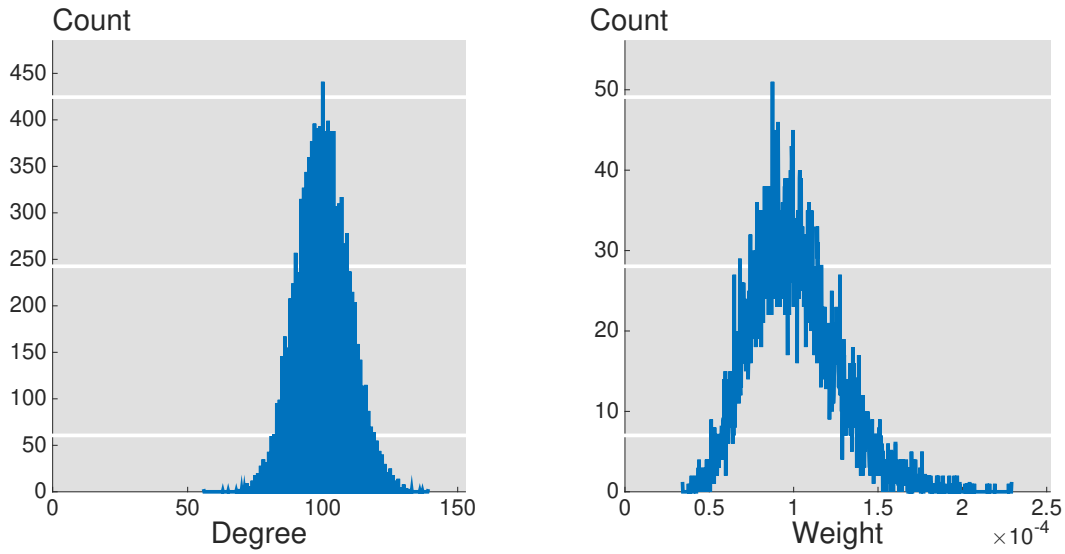
(d) $\beta = 1$

(c) $\beta = 2$

*Figure 3.17: Recovery error comparison of the ring graph with 4-nearest neighbors. The blue curve represents Algorithm 2, the red curve represents Algorithm 5, and the black line represented the approximation by the true first $K$ frequency components.*

### Erdős-Rényi Graph

We consider a random graph where each pair of nodes is connected with some probability, also known as an Erdős-Rényi graph[62]. Since the maximum value of eigenvectors of an Erdős-Rényi graph is bounded by $O(N^{-1/2})$[63], the energy also spreads to each element in $\mathbf{V}$, which follows Definition 12. In the simulation, the Erdős-Rényi graph has 10,000 nodes, and each pair of nodes is connected with probability of 0.01, that is, each node has 100 neighbors on expectation.

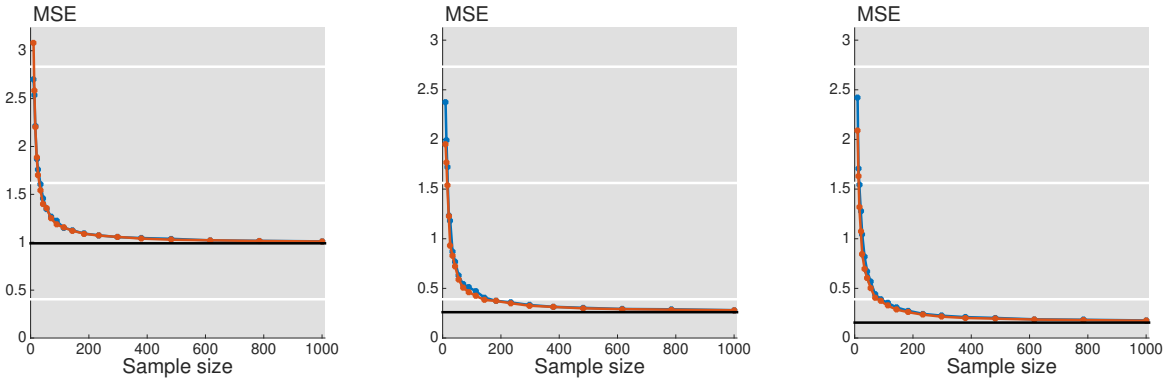



(a) Histogram of degrees

(b) Histogram of sampling scores

*Figure 3.18: Properties of the Erdős-Rényi graph.*

Figure 3.18 shows the properties of the simulated Erdős-Rényi graph. Since the connectivity probability is 0.01, the degrees are around 100, and the sampling scores in Algorithm 5 are around $10^{-4}$, which confirms that the graph belongs to type-1.




(c) $\beta = 0.1$

(d) $\beta = 1$

(c) $\beta = 2$

*Figure 3.19: Recovery error comparison of the Erdős-Rényi graph. The blue curve represents Algorithm 2, the red curve represents Algorithm 5, and the black line represented the approximation by the true first $K$ frequency components.*

Figure 3.19 compares the performances of Algorithm 2 and 5 with various values of $\beta$ averaged over 100 tests. In this case, we still barely see the blue curve because of the overlapping. When $\beta$ increases, the fraction of energy from the first $K$ components decreases, and thus, the bias decreases. For each $\beta$, Algorithm 2 and 5 converges to the linear approximation with similar rates. We conclude that the the Erdős-Rényi graph, as a type-1 graph, results in that the recovery strategies based on experimentally designed sampling performs similarly to the recovery strategies based on random sampling.

### Generalized Random Key Graph

We propose a new graph model here, called a *generalized random key graph*. It is inspired from the random key graph, which is widely used in wireless sensor networks. independently and randomly assigned $R$ distinct keys from the pool of $P$ keys. Two sensor nodes can then establish a secure link between them when they share at least one key in common. Another example is that each people in a social network independently and randomly chooses $R$ hobbies from the pool of the $P$ hobbies. Two people can then establish a friendship between them when they share at least one hobby in common. Some extended work of the random key graph is used in connectivity analysis, clustering analysis, and recommender systems.

To model more realistic scenarios in network science, such as community detection, we propose the generalized random key graph as follows: each node is independently and randomly assigned $R$ distinct keys from the pool of $P$ keys. Two nodes are connected with probability $p_r$ when they share $r$ keys in common, where $0 \leq p_0 \leq p_1 \leq p_2 \leq \cdots \leq p_R \leq 1$. The random key graph is a special case when $p_0 = 0$ and $p_1 = p_2 = \cdots = p_R = 1$, which can be regarded a noiseless case.

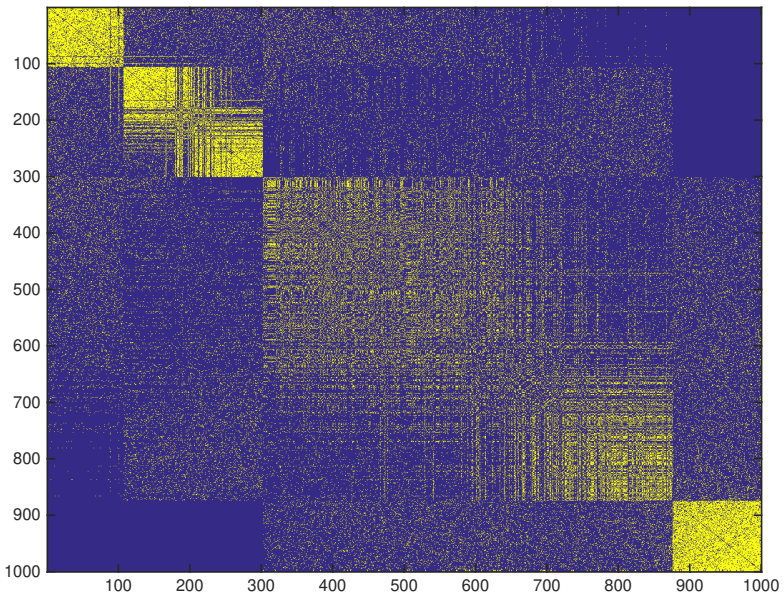
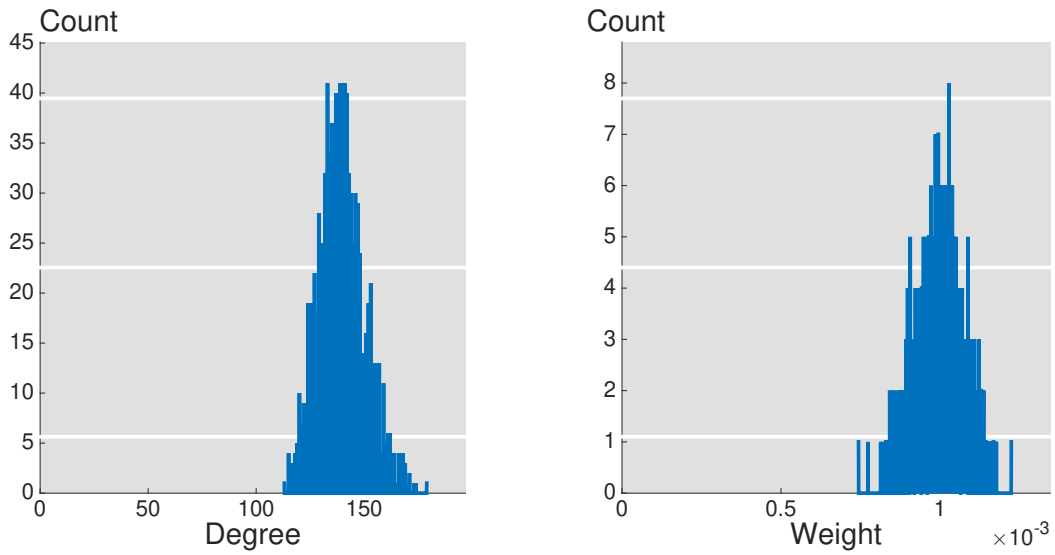



*Figure 3.20: A generalized random key graph.*

In the simulation, the generalized random key graph has $1,000$ nodes, where each node selects 2 keys from the pool of 5 keys, in total of 10 clusters. The connectivity probability is $p_0 = 0, p_1 = 0.1, p_2 = 0.8$. Figure 3.20 shows the graph shift.

Figure 3.21 shows the properties of the simulated generalized random key graph. The degrees are around 140, and the sampling scores in Algorithm 5 are around $10^{-3}$, which confirms that the graph belongs to type-1.

Figure 3.22 compares the performances of Algorithm 2, 5 and sampling theory on graphs with optimal sampling operator 3.54 with various values of $\beta$ averaged over 100 tests. In this case, we

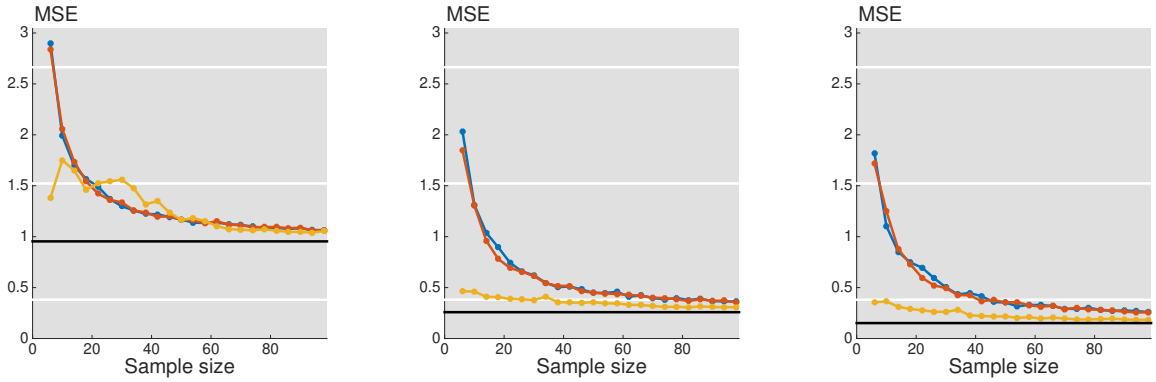



(a) Histogram of degrees

(b) Histogram of sampling scores

*Figure 3.21: Properties of the generalized random key graph.*

still barely see the blue curve because of the overlapping. When $\beta$ increases, the fraction of energy from the first $K$ components decreases, and thus, the bias decreases. For each $\beta$, Algorithm 2 and 5 converges to the linear approximation with similar rates. When $\beta = 0.1$, that is, lots of high-frequency components exist, sampling theory on graphs with optimal sampling operator does not performs stably. When $\beta = 2$, that is, few high-frequency components exist, sampling theory on graphs with optimal sampling operator provides the best performance. Since we proof that Algorithm 5 is optimal in terms of convergence rates, sampling theory on graphs with optimal sampling operator is better by some constant. Note that in many real-world applications, the improvement on constants is also valuable. For example, in semi-supervised learning, we want to label fewer data samples without losing classification accuracy[81]. We conclude that the the generalized random key graph, as a type-1 graph, results in that the recovery strategies based on experimentally designed sampling performs similarly to the recovery strategies based on random sampling.




(c) $\beta = 0.1$ (d) $\beta = 1$ (c) $\beta = 2$

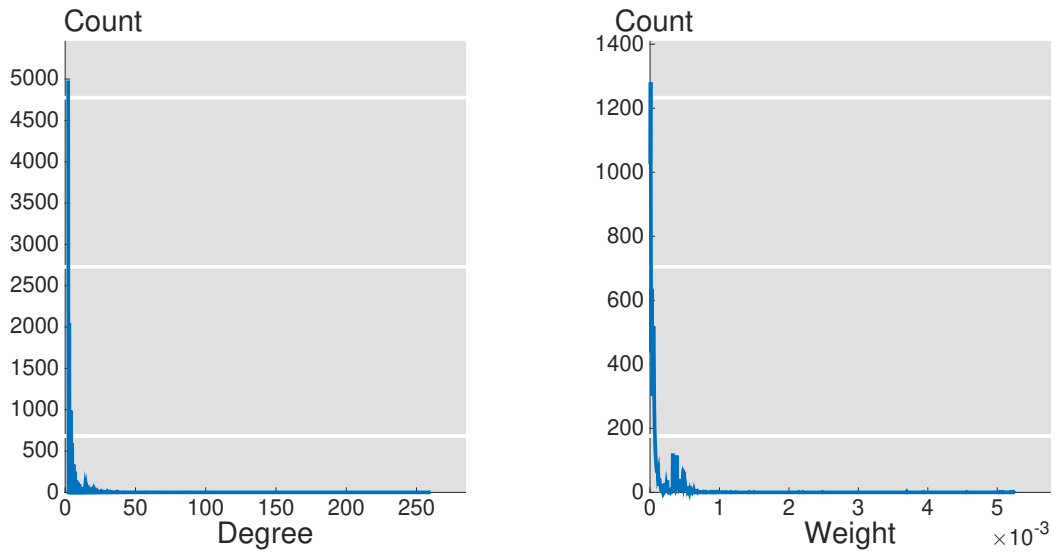
*Figure 3.22: Recovery error comparison of the generalized random key graph. The blue curve represents Algorithm 2, the red curve represents Algorithm 5, the orange curve represents sampling thoery on graphs with optimal sampling operator* (3.54)*, and the black line represented the approximation by the true first 4 frequency components.*

### 3.5.3 Type-2 Graph

To obtain the asymptotic rate, we need the requirement in Definition 13 holds for all $K > K_0$ , which is strict. In many real-world problems, we often consider a small sample size and a small bandwidth, that is, some bias is tolerable. In this case, we just need the requirement in Definition 13 holds for only $K = K_0 \ll N$. We call those graphs as the general type-2 graphs. Based on Theorems 16 and 17, for the general type-2 graphs, Algorithms 2 and 5 have the same bias, but Algorithms 5 has a much lower variance. We find that preferential attachment graphs, some real-world graphs, and generalized random key graphs belong to general type-2. We expect that the recovery strategies based on experimentally designed sampling outperforms the recovery strategies based on random sampling.

#### Preferential Attachment Graph

We consider a graph where the more connected a node is, the more likely it is to receive new links, known as a preferential attachment graph[61,62]. It is well known that the degree distribution of a preferential attachment graph follows the power law. We conjecture that it belongs to type-2. We simulate a preferential attachment graph with 10,000 nodes by using the Barabási-Albert model, where new nodes are added to the network one at a time. Each new node is connected to 2 existing nodes with a probability that is proportional to the number of links that the existing nodes already have.



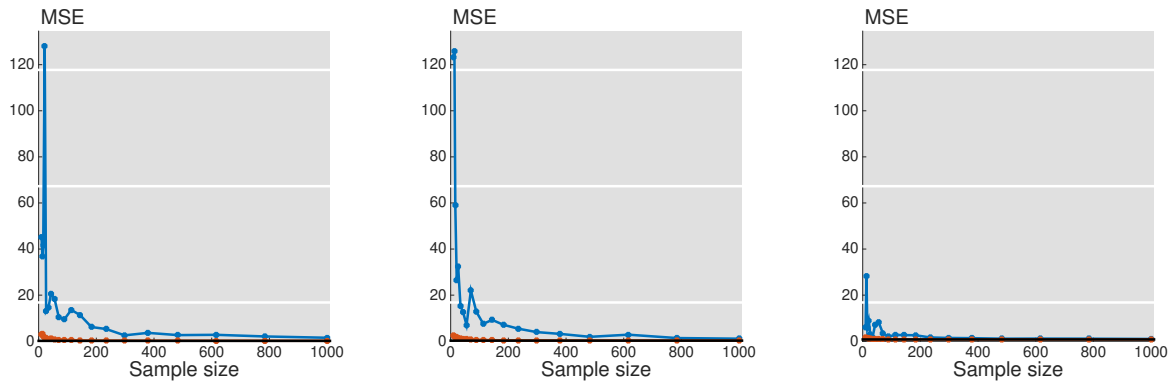

(a) Histogram of degrees

(b) Histogram of sampling scores

*Figure 3.23: Properties of the preferential attachment graph.*

Figure 3.23 shows the properties of the simulated preferential attachment graph. Both distributions of the degrees and the sampling scores in Algorithm 5 are skewed, and follow the power law, which confirms that the graph belongs to general type-2.

Figure 3.24 compares the performances of Algorithm 2 and 5 with various values of $\beta$ averaged over 100 tests. When $\beta$ increases, the fraction of energy from the first $K$ components decreases,



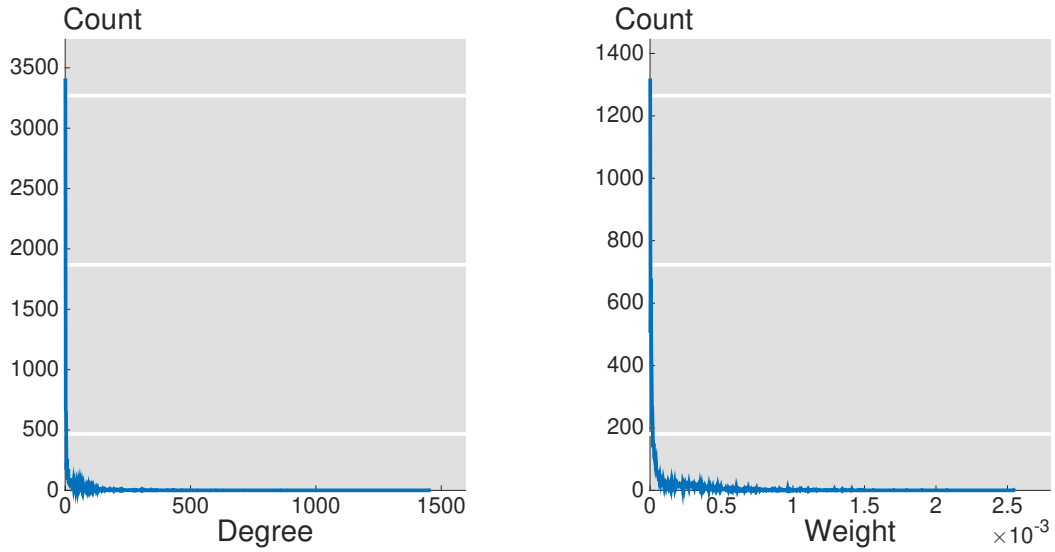

(c) $\beta = 0.1$ (d) $\beta = 1$ (c) $\beta = 2$

*Figure 3.24: Recovery error comparison of the preferential attachment graph. The blue curve represents Algorithm 2, the red curve represents Algorithm 5, and the black line represented the approximation by the true first $K$ frequency components.*

and thus, the bias decreases. For each $\beta$, Algorithm 5 converges to the linear approximation much faster than Algorithm 2. We conclude that the the preferential attachment graph, as a general type-2 graph, results in that the recovery strategies based on experimentally designed sampling outperforms the recovery strategies based on random sampling.

### Wikipedia Graph

We consider a real-world graph that represents Wikipedia adminship election. A small part of Wikipedia contributors are administrators, who are users with access to additional technical features that aid in maintenance. In order for a user to become an administrator a request for adminship is issued and the Wikipedia community via a public discussion or a vote decides who to promote to adminship. Using the latest complete dump of Wikipedia page edit history, all administrator elections and vote history data were extracted[82]. The graph has 8297 nodes and 228080 edges. Many online social networks follow the power law. We then conjecture that the Wikipedia graph has a similar property with a preferential attachment graph, and belongs to general type-2.



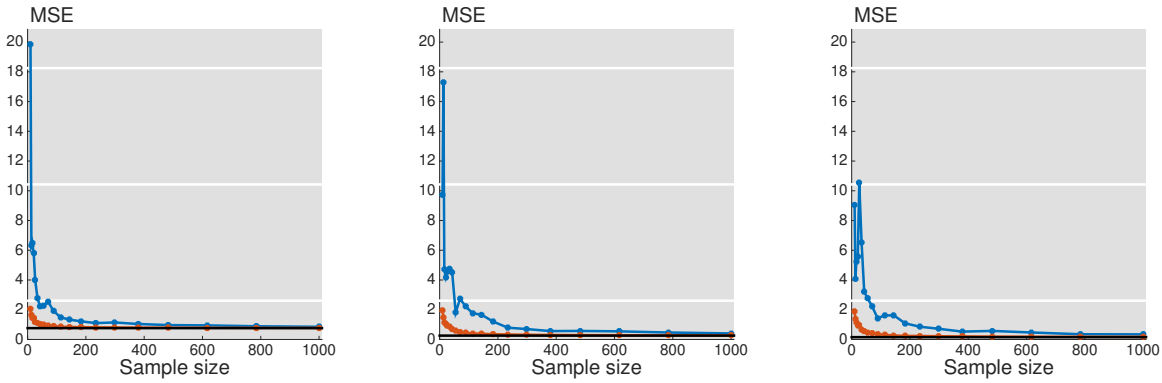

(a) Histogram of degrees

(b) Histogram of sampling scores

*Figure 3.25: Properties of the Wikipedia graph.*

Figure 3.25 shows the properties of the simulated preferential attachment graph. Similarly to the preferential attachment graph, both distributions of the degrees and the sampling scores in Algorithm 5 are skewed, and follow the power law, which confirms that the graph belongs to general type-2.

Figure 3.26 compares the performances of Algorithm 2 and 5 with various values of $\beta$ averaged over 100 tests. When $\beta$ increases, the fraction of energy from the first $K$ components decreases, and thus, the bias decreases. For each $\beta$, Algorithm 5 converges to the linear approximation much faster than Algorithm 2. We conclude that the the Wikipedia graph, as a general type-2 graph, results in that the recovery strategies based on experimentally designed sampling outperforms the recovery strategies based on random sampling.

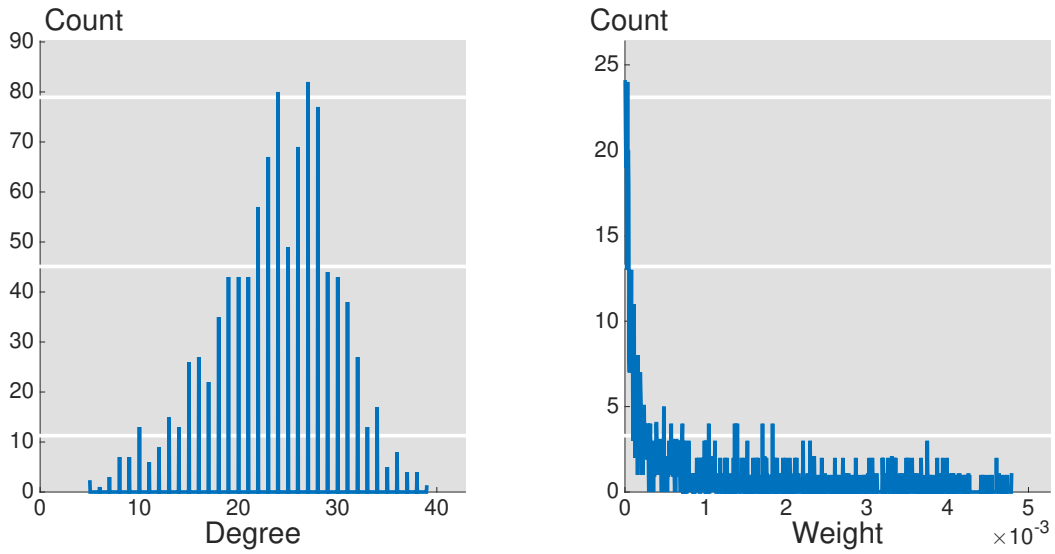



(c) $\beta = 0.1$ (d) $\beta = 1$ (c) $\beta = 2$

*Figure 3.26: Recovery error comparison of the Wikipedia graph. The blue curve represents Algorithm 2, the red curve represents Algorithm 5, and the black line represented the approximation by the true first $K$ frequency components.*

### Random Geometric Graph

We consider a spatial graph where each of nodes is assigned random coordinates in the box $[0,1]^d$ and each edge appears when the distance between two nodes is in a given range[83]. It is known that given proper parameters, the degree distribution of a random geometric graph is the same with an Erdős-Rényi graph. We conjecture that similarly to an Erdős-Rényi graph, a random geometric graph belongs to type-1. In the simulation, the random geometric graph has $1,000$ nodes, lying in the box $[0,1]^2$, and two nodes are connected when the Euclidean distance is less than 0.09.




(a) Histogram of degrees

(b) Histogram of sampling scores

*Figure 3.27: Properties of the random geometric graph.*

Figure 3.27 shows the properties of the simulated random geometric graph. Before this, we have shown five graphs. For each of them, the distributions of degrees and sampling scores are similar, which seems that the degree distribution is a good indicator for identifying the type of a graph. Surprisingly, for the random geometric graph, the degree distribution is bell-shaped and the distribution of sampling scores is skewed! It means that our previous conjecture is wrong and a graph can belongs to general type-2, even when each node has a similar degree. In other words, recovery strategies based on experimentally designed sampling can outperform those based on random sampling on a graph where each node has a similar degree. It is shown[83] that the cluster properties are different between a random geometric graph and an Erdős-Rényi graph. The sampling scores can capture these cluster properties through the graph Fourier transform matrix, and works as a proper indicator of when experimentally designed sampling can outperform random sampling.

We simulate a graph signal with bandwidth $K = 4$ and $\beta = 1$, in a way as shown in (3.55). Figure 3.28 shows the first 20 graph frequency componentsof such a graph signal, the black dotted line indicates the cut-off frequency. We see that there is a decay after the cut-off frequency.

Figure 3.29 (a)-(d) show the first 4 graph Fourier basis vectors. Each basis vector captures cer-

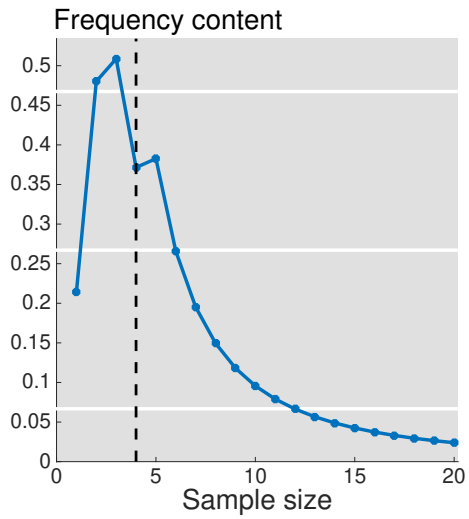
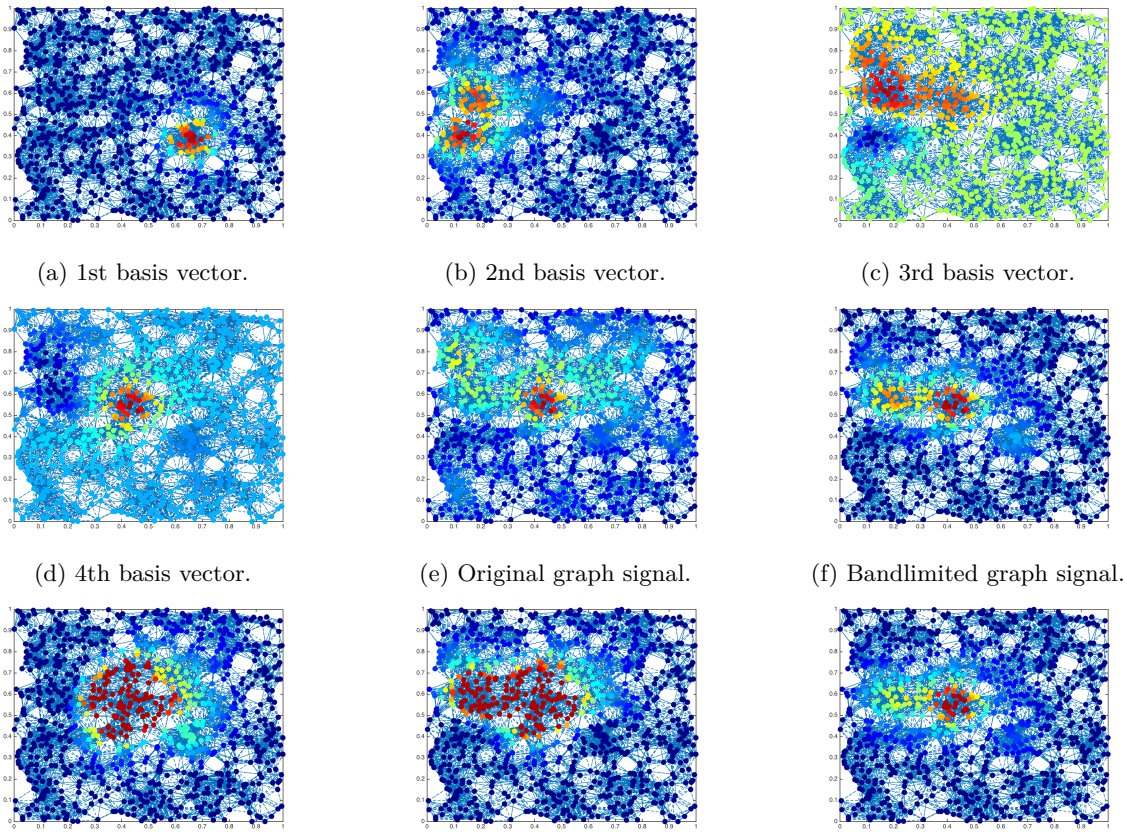



*Figure 3.28: First 20 graph frequency components.*

tain local cluster patterns. Figure 3.29 (e) shows the original graph signal whose frequency content is in Figure 3.28, and Figure 3.29 (f) shows the corresponding graph signal approximated by the first 4 basis vectors. Figure 3.29 (g)-(i) show the recovered graph signals of Algorithm 2, 5, and sampling theory on graphs with optimal sampling operator (3.54), when only 4 samples can be taken for each algorithm. We see that sampling theory on graphs with optimal sampling operator almost perfectly recover the bandlimited graph signals with only 4 samples, and Algorithm 5 works much better than Algorithm 2.

Figure 3.30 compares the performances of Algorithm 2, 5 and sampling theory on graphs with optimal sampling operator (3.54) averaged over 100 tests.

We see that as experimentally designed sampling operators, both Algorithm 5 and sampling theory on graphs with optimal sampling operator converge much faster than Algorithm 2. We conclude that the the random geometric graph, as a general type-2 graph, results in that the recovery strategies based on experimentally designed sampling outperforms the recovery strategies based on random sampling.



(a) 1st basis vector. (b) 2nd basis vector. (c) 3rd basis vector.

(d) 4th basis vector. (e) Original graph signal. (f) Bandlimited graph signal.

(g) Random sampling. (h) Experimentally designed sampling. (i) Optimal sampling.

*Figure 3.29: Graph signals on the random geometric graph.*

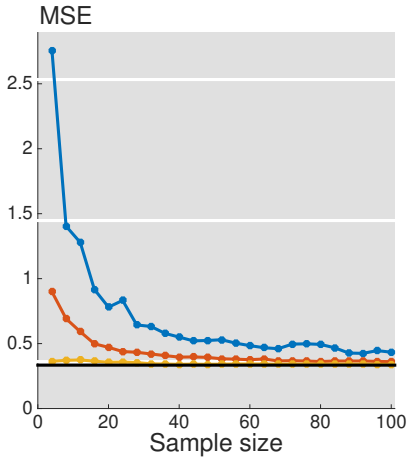



*Figure 3.30: Recovery error comparison of the random geometric graph. The blue curve represents Algorithm 2, the red curve represents Algorithm 5, the orange curve represents sampling thoery on graphs with optimal sampling operator* (3.54)*, and the black line represented the approximation by the true first 4 frequency components.*

### 3.5.4 Discussion

This work inspire the following ideas.

- The degree distribution does not tell everything on a graph. Degree only consider the information of neighbors, which is limited. Many other patterns may be more meaningful, such as clusters;
- Graph Fourier transform matrix is useful for understanding a graph structure. Here, we compute sampling scores from the graph Fourier transform matrix, and find that the random geometric graph has local cluster patterns. We believe that the graph Fourier transform matrix can be useful to study a graph structure in many other case;
- Experimentally designed sampling is useful for semi-supervised learning with graphs. Many algorithms of semi-supervised learning work based on graphs that are constructed from a given dataset[81]. The graph is often constructed by modeling each node as a data sample and connecting two nodes by a edge if the distance between their features is in a given range, which is similar to the construction of random geometric graphs. Many adaptive algorithms are proposed to find anchor points on graphs, which is essentially experimentally designed sampling on graphs[84]. Those work does not study when and why experimentally designed sampling can work, however, we give a comprehensive explanation;
- Even we prove Algorithm 5 is optimal in terms of convergence rate, we still see that sampling theory on graphs with optimal sampling operator outperforms Algorithm 5 in the generalized random key graphs and the random geometric graph. Since we want to sample very little in many applications, and asymptotic rates cannot tell the performance when we take only a few samples, better sampling and recovery strategies based on experimentally designed sampling on graphs are still appealing.

## 3.6 Spectrum-Blind Sampling of Graph Signals

The sampling theory studied in Section 3.1.1 recovers bandlimited graph signals where the authors consider settings where we are aware that the main energy of a graph signal is in the low frequencies; we call this setting *spectrum-aware*. In contrast, here we are unaware of the shape of graph signals in the graph spectral domain; we call this setting *spectrum-blind*.[85] More precisely, we now consider a relaxation from the definition of bandlimited graph signals where we know graph signals are sparse in the graph spectrum domain, but do not know the supports. We consider a class of *$K$-spectrally sparse graph signals*:

$$\mathcal{X}_K = \{\mathbf{x} \in \mathbb{R}^N : \|\widehat{\mathbf{x}}\|_0 \leq K, \text{ where } \widehat{x} = \mathbf{U}\,\mathbf{x}\}.$$

Here, bandlimited graph signals here do not necessarily mean low-pass because we do not specify the ordering of frequencies. The bandlimited restriction is equivalent to limiting the number of nonzero elements in the graph Fourier domain with known supports. The elements in $\mathcal{X}_K$ are the same as those in $\mathrm{BL}_{\mathbf{A}}(K)$, but $\mathcal{X}_K$ is blind to the ordering of frequencies. $\mathcal{X}_K$ is thus a useful characterization for graph signals with more complex multiband structures. We consider sampling graph signals $\mathbf{x} \in \mathcal{X}_K$. We approach this problem by formulating an optimization problem that enforces sparsity on the graph Fourier transform of $\mathbf{x}$. In this section, we formulate the problem of signal recovery on graphs and describe the sampling and recovery framework. We study spectrum-blind signal recovery and propose algorithms for both random sampling and experimental designed sampling with which we can ensure reliable recovery.

### 3.6.1 Problem Formulation

We now review a class of bandlimited graph signals and generalize this class to the spectrum-blind setting. We next describe the sampling and recovery strategies. Based on Definition 6, the bandlimited graph signals do not necessarily mean low-pass because we do not specify the ordering of frequencies. The bandlimited restriction is equivalent to limiting the number of nonzero elements in the graph Fourier domain with known supports. Note that this setting is spectrum-aware because we assume the ordering of frequencies is known to ensure the first $K$ frequency components are nonzero.

We now consider a relaxation from the definition of bandlimited graph signals where we know graph signals are sparse in the graph spectrum domain, but do not know the supports. We consider a class of $K$-spectrally sparse graph signals.

$$\mathcal{X}_K = \{\mathbf{x} \in \mathbb{C}^N : \|\widehat{\mathbf{x}}\|_0 \leq K, \text{ where } \widehat{x} = \mathbf{V}^{-1}\,x\}.$$

The elements in $\mathcal{X}_K$ are the same as those in $\mathrm{BL}_{\mathbf{A}}(K)$, but $\mathcal{X}_K$ is blind to the ordering of frequencies. $\mathcal{X}_K$ is thus a useful characterization for graph signals with more complex multiband structures. We consider such graph signals $\mathbf{x} \in \mathcal{X}_K$ in the following discussion.

#### Sampling and Recovery Strategy

Suppose that we want to sample $M$ coefficients of a graph signal $\mathbf{x} \in \mathbb{C}^N$ to produce a sampled signal $\mathbf{x}_{\mathcal{M}} \in \mathbb{C}^M$ ($M < N$), where $\mathcal{M} = (\mathcal{M}_0, \cdots, \mathcal{M}_{M-1})$ denotes the sequence of *sampled*

indices, and $\mathcal{M}_i \in \{0, 1, \cdots, N-1\}$. We then interpolate $\mathbf{x}_\mathcal{M}$ to get $\mathbf{x}' \in \mathbb{C}^N$, which recovers $\mathbf{x}$ either exactly or approximately. The sampling operator $\Psi$ is a linear mapping from $\mathbb{C}^N$ to $\mathbb{C}^M$, defined as

$$\Psi_{i,j} = \begin{cases} 1, & j = \mathcal{M}_i; \\ 0, & \text{otherwise}, \end{cases} \tag{3.56}$$

and the interpolation operator $\Phi$ is a mapping from $\mathbb{C}^M$ to $\mathbb{C}^N$ .

We use the sampling operator to get the sampled graph signal $\mathbf{x}_\mathcal{M} = \Psi\mathbf{x} \in \mathbb{C}^N$ and $\mathbf{x}' = \Phi\mathbf{x}_\mathcal{M} \in \mathbb{C}^N$ recovers $\mathbf{x}$ either exactly or approximately. We consider two sampling strategies: *random sampling* means that sampling indices are chosen from from $\{0, 1, \cdots, N-1\}$ independently and randomly; and *experimentally designed sampling* means that sampling indices can be chosen beforehand. It is clear that random sampling is a subset of experimentally design sampling[27].

### Random Sampling

We consider random sampling where the subsampled graph signal $\mathbf{x}_\mathcal{M}$ is a random subset of the graph signal $\mathbf{x}$. That is, the sampling operator $\Psi$ is formed by sampling $\mathcal{M}$ rows of the identity matrix $I_n$ from a uniform distribution. We note that since $\mathbf{x}_\mathcal{M} = \Psi\mathbf{x}$ and $\mathbf{x} = \mathbf{V}\,\widehat{x}$, consequently, $\mathbf{x}_\mathcal{M} = (\Psi\,\mathbf{V})\widehat{\mathbf{x}}$. Denoting the sampled graph Fourier matrix $(\Psi\,\mathbf{V})$ as $\mathbf{V}_\mathcal{M}$, it is natural to formulate this as an $\ell_0$-norm optimization problem which is often intractable. We instead seek to minimize surrogate measures, such as the $\ell_1$-norm, that lead to more tractable computational methods as in the following formulation:

**Algorithm 4.** Let $\Psi$ be a random sampling operator and the noisy measurements $\mathbf{x}_\mathcal{M} = \mathbf{V}_\mathcal{M}\,\widehat{\mathbf{x}} + e$ with $\|e\|_2 \leq \epsilon$. We recover $x \in \mathcal{X}_K$ by solving the following optimization problem:

$$\begin{aligned} \underset{\widehat{x}}{\text{minimize}} \quad & \|\widehat{\mathbf{x}}\|_1, \\ \text{subject to} \quad & \|\mathbf{x}_\mathcal{M} - \mathbf{V}_\mathcal{M}\,\widehat{\mathbf{x}}\|_2 \leq \epsilon. \end{aligned} \tag{3.57}$$

In the following discussion, we restrict ourselves to undirected graphs and consequently symmetric graph shifts. The following theorem shows the recovery performance of Algorithm 4. .We initially assume that the graph Fourier matrix has uniformly bounded entries. The matrix formed by choosing $M$ rows of such a matrix uniformly at random is sufficiently incoherent (satisfies the restricted isometry property) with high probability when the number of measurements is on the order of $K \log^4 N$[86,87].We shall consequently show families of graphs whose graph Fourier matrices satisfy this property both empirically and theoretically.

**Theorem 18.** Let $M \geq C\mu^2 \log(N)^4 K$, where $C$ is some constant and $\mu = \sqrt{N}\max_{i,j}|(\mathbf{V})_{i,j}|$ such that $\mathbf{V}$ has sufficiently well-bounded entries. The recovery error of the solution $\widehat{\mathbf{x}}^*$ of in Algorithm 4 is bounded with high probability as

$$\|\widehat{\mathbf{x}}^* - \widehat{\mathbf{x}}\|_2 \leq C\Big[\epsilon + \frac{\|\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_K\|_2}{\sqrt{K}}\Big], \tag{3.58}$$

where $\widehat{\mathbf{x}}_K$ denotes the vector of the $K$ largest coefficients (in magnitude) of $\widehat{\mathbf{x}}$.

The proof is a direct consequence of results in classical compressed sensing[88,89,86,87]. This result says that the recovery error is at most proportional to the norm of the noise in the samples

and the tail of the signal. We therefore apply the same arguments used to prove that the RIP holds for partially bounded orthogonal matrices for appropriate random sampling operators, for graph Fourier matrices that are partially bounded orthogonal matrices.

Given the reconstruction of $\widehat{\mathbf{x}}$, we can recover the full graph signal $\mathbf{x}^*$ by simply applying the inverse graph Fourier transform $\mathbf{x}^* = \mathbf{V}\,\widehat{\mathbf{x}}^*$.

We note that (5) can be formulated as a linear program. Consequently, we have shown a spectrum-blind framework under which uniform random sampling of a sufficient number of coefficients of the graph signal allows the robust recovery of the original graph signal with high probability.

### 3.6.2 Extensions to Graph Models

In the previous section, we note that the restricted isometry property and incoherence condition between the sparsity and measurement bases does not necessarily hold for all families of graphs. We now specifically look at families of graphs where we can use random sampling for reliable spectrum-blind recovery.

#### Circulant Graphs and the Finite Discrete-Time Signal

Since the discrete Fourier transform matrix diagonalizes the circulant graph and is hence its graph Fourier transform matrix[66], we see that Theorem 18 is immediately applicable to signals supported on circulant graphs. Specifically, for a partial Fourier matrix, $\mu = \sqrt{N}\max_{i,j}|(\mathbf{V}_K)_{i,j}| = 1$, we have that the minimum number of random samples required to recover a $K$-sparse $\widehat{\mathbf{x}}$ is

$$M \geq C(\log N)^4 K. \tag{3.59}$$

Additionally, the graph that supports a finite discrete-time signal is called the *finite discrete-time graph*, which is represented by the cyclic permutation matrix[38,21], also a circulant graph. Consequently, the bound in (3.59) corresponds to a standard result in classical compressed sensing when using Fourier measurements.

#### Erdős-Rényi Random Graphs

An Erdős-Rényi graph is constructed by connecting nodes randomly, where each edge is included in the graph with probability $p$ independent from every other edge[61,62]. We aim to derive bounds on the minimum number of random samples required such that we can recover the graph signal using the formulation in Algorithm 4.

**Corollary 6.** Let a graph shift $\mathbf{A}$ represent an Erdős-Rényi random graph on a node set of size $N$, obtained by drawing an edge between each pair of vertices, randomly and independently, with probability $p = g(N)\log(N)/N$ where $g(\cdot)$ is some positive function. Let $\mathbf{V}$ be the eigenvector matrix of $\mathbf{A}$, obeying $\mathbf{V}\,\mathbf{V}^T = N \cdot \mathrm{I}$. Let the sampling number satisfy

$$M \geq CK \cdot \frac{\log^{2.2} g(N)\log^5(N)}{Np},$$

for some positive constant $C$. Then, the recovery error of the solution $\widehat{\mathbf{x}}^*$ of in Algorithm 4 follows (3.58).

Due to lack of space we omit the proof which is based on results in[90]. We note the dependency on $p$, and see that as $p$ decreases or as the graph is more sparse, we expect to require more samples for the partial graph Fourier matrix to capture sufficient information. This can be verified empirically.

### 3.6.3 Experimentally Designed Sampling

As mentioned in the previous section, for graphs where we do not have sufficient incoherence between the measurement and sparsity bases, we cannot hope to perform random sampling for reliable recovery using the same order of measurements. This is the case for some irregularly structured graphs. In this section, we study whether we can hope to do better by studying the graph structure. We also restrict ourselves to undirected graphs and consequently symmetric graph shifts. More specifically, we now consider experimentally designed sampling where the samples are designed based on the graph structure. We can use the notion of local coherence introduced in[91], and extend sparse recovery guarantees to a richer class of graphs beyond strictly incoherent systems. The result shows that regions of the sensing basis that are more coherent with the sparsity basis should be sampled with a higher density. In Section 3.6.4, we demonstrate superior performance for irregular graph structures using experimentally designed sampling as opposed to random sampling.

**Algorithm 5.** [91] Let $\Psi$ be a sampling operator designed as follows: we assign a weight $\kappa_j$ to each node where $\kappa_j = \sup_{1\leq k\leq N} |\sqrt{N}\,\mathbf{V}_{j,k}\,|$. We sample a graph signal $|\mathcal{M}|$ times. Each time, we choose a node with probability $\kappa_j^2$ proportional to the square of maximum element in the row corresponding to the node in the graph Fourier matrix $\mathbf{V}$ . Given noisy measurements $\mathbf{x}_{\mathcal{M}} = \Psi x + e$, we recover the original graph signal by solving the following optimization program:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|\,\mathbf{V}^{-1}\,x\|_1, \\ \text{subject to} \quad & \|\mathbf{x}_{\mathcal{M}} - P\Psi x\|_2 \leq \sqrt{M}\epsilon, \end{aligned}$$

where $x_i^*$ is the $i$th component of the recovered graph signal $\mathbf{x}^*$ and $P$ is a diagonal matrix with diagonal entries $p_{k,k} = 1/\sqrt{c}\kappa_k$.

The following theorem shows the recovery performance of Algorithm 5.

**Theorem 19.** Let $M \geq K(\frac{1}{N}\sum_{j=1}^{N}\kappa_j^2)\log^4(N)$. The recovery error of the solution $\widehat{\mathbf{x}}^*$ of in Algorithm 5 is bounded as

$$\|x - x^*\|_2 \leq \frac{1}{\sqrt{K}}\|\,\mathbf{V}^{-1}\,x - \mathbf{V}^{-1}\,x^*\|_1 + \epsilon. \tag{3.60}$$

We omit the proof; it follows from[91].

Since random sampling is a subset of experimentally designed sampling, we naturally expect experimentally designed sampling to always perform at least as well as random sampling. We expect the performance improvement to be more significant for irregularly structured graphs whose Fourier matrices tend to exhibit a larger degree of coherence. We empirically compare the performance of experimentally designed sampling versus random sampling for different graph families in the next section.

### 3.6.4 Simulations

In this section, we validate the proposed framework for spectrum-blind graph signal recovery by showing that reliable recovery is achieved for each of the different families of graphs with high probabilities for both random and experimentally-designed sampling. We demonstrate the (linear) tradeoff in terms of the average number of minimum number of samples required to ensure perfect recovery versus the sparsity of the signal in addition to showing the efficacy of experimentally-designed sampling.

**Erdős-Rényi graphs.** As shown in Section 3.6.2, with high probability, perfect recovery is achieved for Erdős-Rényi graph signals when we randomly sample a sufficient number of signal coefficients. We verify this result experimentally, by randomly sampling Erdős-Rényi graphs with various sizes and connecting probabilities.

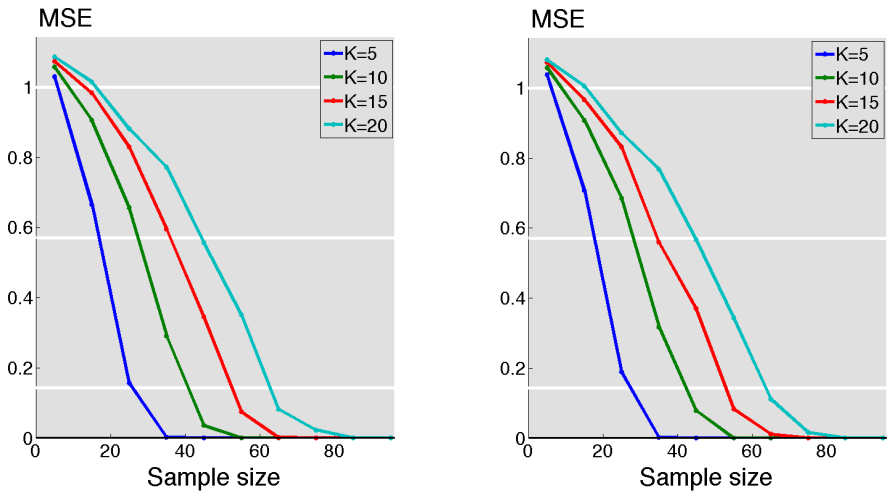
**Scale-free graphs.** A scale-free graph is a graph that is dominated by a relatively small number of nodes that are hubs of connectivity. Its degree distribution follows a power law[61,62]. Many real-world graphs, such as the topology of web pages, the collaborative network of Hollywood actors, the power grid of the United States exhibit the scale-free phenomenon.

### 3.6.5 Experimental setup

Suppose that for each graph, we construct a unit norm graph signal whose spectral support is chosen randomly from a uniform distribution on the set of all supports with a given sparsity $K$. Given a graph shift, we sample $M$ times and perform the $\ell_1$ minimization procedure in Algorithm 4 100 times, and calculate the average mean square recovery error as a function of the samples acquired; these are shown in Figure 3.31. We vary the sparsity K to be 5,10,15,20 and the number of samples acquired $M$ from 0 to 100.

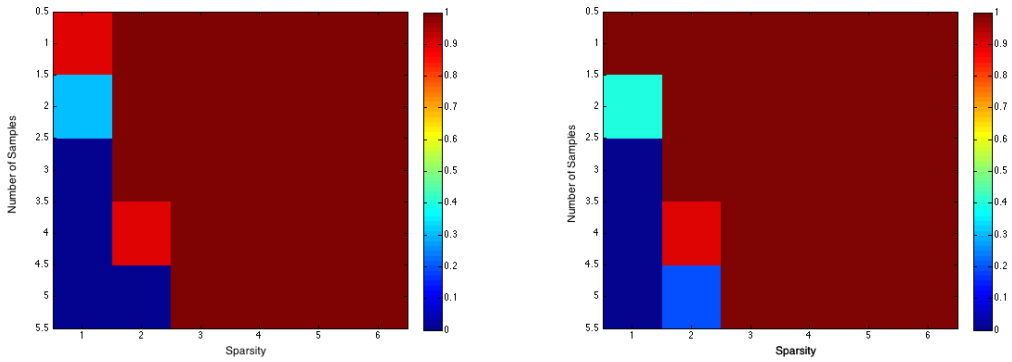
We also compare experimentally designed sampling to random sampling; the results are shown in Figure 3.32 and Figure 3.33 for Erdős-Rényi and scale-free graphs, respectively. We use graphs of size $N = 1000$ and vary sparsity levels and the number of samples acquired to generate a heat map that shows the frequency of perfect recovery for both random sampling and experimentally designed sampling. In the figures, each square corresponds to the number of samples acquired (y-axis) at a particular level of sparsity (x-axis), the color of which signifies the frequency of perfect recovery with respect to to the color bar. Hence, we see that while experimentally designed sampling always works better, the improvement is amplified for scale-free graphs and we are able to perfectly recover with a higher success rate. This conforms with our intuition since the scale-free graph exhibits a more irregular structure and is less incoherent with respect to the measurement basis.

We further empirically study mean square recovery for Erdős-Rényi graphs of size $N = 200$ as a function of the number of samples acquired by varying the connection probability . The results are shown in Figure 3.32. We see in accordance with Theorem 6 that for low density graphs, we expect to require slightly more samples on average to ensure reliable recovery. Nevertheless, we are able to exhibit reliable recovery for the rich family of Erdős-Rényi graphs.



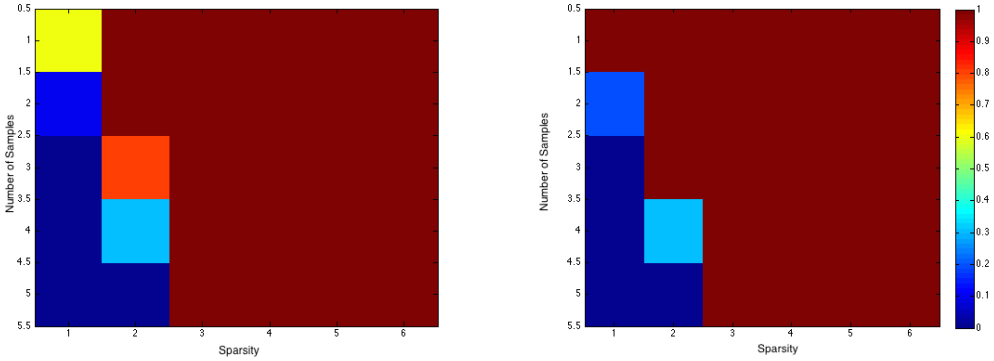

(a) Erdős-Rényi Graph (b) Scale-Free Graph

*Figure 3.31: Mean square recovery error performance using random sampling as a function of the number of samples acquired for Erdős-Rényi and scale-free graphs of size N=200. The blue curve corresponds to a sparsity level in the spectral domain of $K = 5$, the green to $K = 15$ and the cyan to $K = 20$. For Erdős-Rényi random graphs, we use edge-presence probability $p = 0.05$ and add 5 edges at each step for scale-free graphs using the Barabasi-Albert preferential attachment model.*




(a) Random sampling (b) Experimentally designed sampling

*Figure 3.32: Frequency of perfect recovery for varying sparsity (y-axis) and varying number of samples (x-axis) for both random sampling and experimentally designed sampling on Erdős-Rényi graphs*




(a) Random sampling

(b) Experimentally designed sampling

*Figure 3.33: Frequency of perfect recovery for varying sparsity (y-axis) and varying number of samples (x-axis) for both random sampling and experimentally designed sampling on scale-free graphs.*

## 3.7 Reconstruction of Smooth Graph Signals

In this section, we study the estimation or reconstruction of smooth signals on graphs from noisy, incomplete, or corrupted measurements. Reconstructing signals from noisy observations is a well-studied problem in signal processing and has applications for inpainting, collaborative filtering, recommender systems and other large-scale data completion problems. We briefly introduce previous work on signal recovery in the graph signal processing literature and Laplacian regularization before presenting a framework for the estimation of smooth signals on product graphs. By structuring the signal as a tensor, we enforce a low-rank condition on the tensor to help recover the signal.

In the basic setting, we assume we measure a noisy or corrupted signal $\mathbf{y}$ and seek to reconstruct $\mathbf{x}$ from $\mathbf{y}$.

$$\mathbf{y} = \mathbf{x} + \epsilon$$

Broadly, there are two frameworks for signal reconstruction or estimation. The first, which we refer to as the *synthesis* framework, generally consists of constructing an appropriate basis or dictionary over the graph for the class of signals, and then regressing the observed signal $\mathbf{y}$ over this basis. We have already seen in Section 2.2 how seen how the graph Fourier basis promotes smoothness in the sense that smooth signals are approximately bandlimited with respect to the graph Fourier basis. A straightforward way to recover the signal would be to choose the best $K$ such that we project the signal onto the corresponding bandlimited space. In this section, however we study graph signal reconstruction based on the *analysis* framework where we formulate an optimization problem that regularizes the smoothness criterion with a penalty function. While in the below discussion we only consider this setting where we only have i.i.d Gaussian noise $\epsilon$, we can extend a general form of graph signal recovery to a flexible optimization problem formulation that accounts for outliers and multiple signals.

### 3.7.1 Graph Total Variation and Laplacian Regularization

A natural proxy for the smoothness of a signal on a graph is its variation over the graph $S_2(\mathbf{x})$ which can be defined with respect to the graph adjacency matrix where $\mathrm{S}_{\mathbf{A}}(\mathbf{x}) = \left\|\mathbf{x} - \frac{1}{|\lambda_{\max}(\mathbf{A})|}\mathbf{A}\,\mathbf{x}\right\|_2^2$ or with respect to the graph Laplacian $\mathrm{S}_{\boldsymbol{L}}(\mathbf{x}) = \mathbf{x}^T\boldsymbol{L}\mathbf{x}$ as we saw in Section 2.1. Regularization with respect to $\mathrm{S}_{\boldsymbol{L}}(\mathbf{x})$ on graphs has been well-studied in previous work in the context of Laplacian regularization.[92,93]

From Section 2.2, we see that the class of bandlimited signals is a subset of globally smooth signals. Rather than explicitly enforcing bandlimitedness, we formulate a convex optimization problem that minimizes the graph variation of the graph signal. Minimizing $\mathrm{S}(\mathbf{x})$ enforces the estimated signal to be smooth with most of the energy lying in the subspace of low graph frequencies. For the basic reconstruction problem, we solve the following problem[27]:

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathbf{y} - \mathbf{x}\|_2 + \lambda\,\mathrm{S}(\mathbf{x}) \tag{3.61}$$

While this is the basic formulation,[27] presents a more flexible framework that can deal with multiple signals as well as outliers. The approaches described in[27] however, do not exploit the inherent structure of product graphs, and instead treat the graph holistically.

The graph Fourier basis promotes smoothness in the sense that smooth signals are approximately bandlimited with respect to the graph Fourier basis. A straightforward way to recover the signal under the synthesis framework would then be to choose the best $K$ such that we project the signal onto the corresponding bandlimited space. We refer to this method as **GFProj**.

### 3.7.2 Representations for Signals on Product Graphs:

We consider a product graph $G = (\mathcal{V}, \mathbf{A}), |\mathcal{V}| = N$, that is constructed from $N$ graph atoms $G_1, \cdots G_N$ , where $G_j = (\mathcal{V}_j, \mathbf{A}^j), |\mathcal{V}_j| = N_j$, using the Kronecker product where $\prod_{j=1}^{J} N_j = N$. We can write the resulting graph shift matrix of the product graph as $\mathbf{A} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \otimes \cdots \otimes \mathbf{A}^{(J)} = \otimes_{j=1}^{J} \mathbf{A}^{(j)}$. In general, we have seen that we can write both the spectral graph Fourier decomposition as well as the multiresolution decomposition in the form $\mathbf{A}^{(j)} = \boldsymbol{F}^{(j)} \Sigma^{(j)} \boldsymbol{F}^{(j)T}$. We can then write the decomposition of the product graph shift as $\mathbf{A} = \boldsymbol{F} \Sigma \boldsymbol{F}^T$ where:

$$\boldsymbol{F} = \boldsymbol{F}^{(1)} \otimes \boldsymbol{F}^{(2)} \otimes \cdots \otimes \boldsymbol{F}^{(J)} = \otimes_{j=1}^{J} \boldsymbol{F}^{(j)} \tag{3.62}$$

$$\Sigma = \Sigma^{(1)} \otimes \Sigma^{(2)} \otimes \cdots \otimes \Lambda^{(J)} = \otimes_{j=1}^{J} \Sigma^{(j)} \tag{3.63}$$

This construction can easily be extended to dictionaries and frames. When $\boldsymbol{F}$ corresponds to the graph multiresolution wavelet transform, we note that this is analogous to separable wavelet construction by tensorization on images and $d$-dimensional grids.[94]

### 3.7.3 Product Graphs and Low Rank Structure

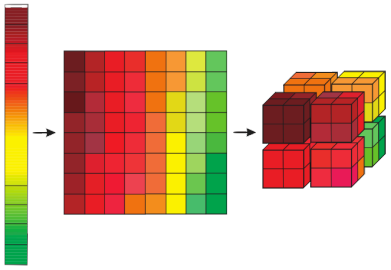


*Figure 3.34: A graph signal on a product graph composed from k-atoms can be structured as a k-th order tensor*

We now study the same problem on product graphs but try to exploit some of the inherent structure in the graph. A natural way to organize signals on product graphs is using tensors.[95,96] In the case of a signal lying on a product of three graphs ,we want to represent the data by a 3rd order tensor $\boldsymbol{\mathcal{X}}$ where the $(i, j, k)$-th entry in the tensor $\boldsymbol{\mathcal{X}}$ indicates the signal value corresponding to the the node in the product graph corresponding to the tuple of the $i$-th node in $\boldsymbol{A}_1$ , the $j$-th node in $\boldsymbol{A}_2$, and the $k$-th node in $\boldsymbol{A}_3$. In the context of recommender engines for example, this would in turn correspond to the i-th user's rating of the j-th entity at the k-th time instant.

Tensor factorization is a decomposition method for high dimensional data that is used to estimate the prominent factors in some signal. Similarly to matrix factorization, PCA, and in graph signal processing, transforms such as spectral graph wavelets and the graph Fourier transform, tensor decomposition allows us to detect latent structure in graph data. Typical tasks built on top of this include denoising, inpainting, and anomaly detection. We study the spectral decomposition of such product graphs and show how an extension of the smoothness assumption to signals on product graphs leads to the corresponding tensor $\boldsymbol{\mathcal{Y}}$ possessing a low-dimensional structure. In this section, we formulate an optimization problem for signal recovery that exploits this low-dimensional structure.[97] For ease of exposition, we use 3-way tensors, the Kronecker tensor product and the graph Laplacian in the following discussion, but the

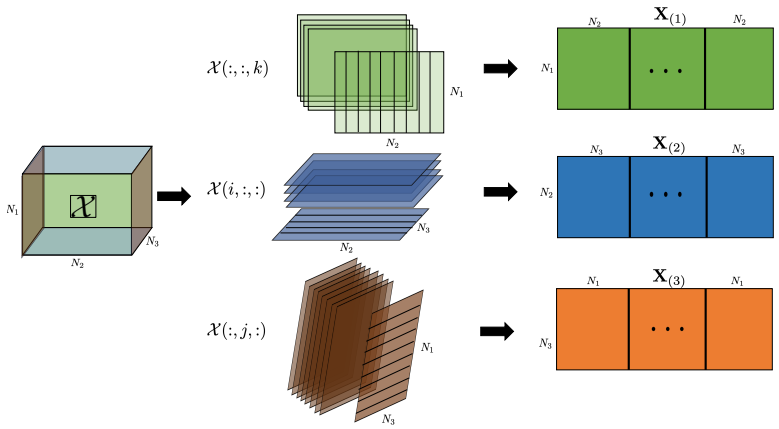



*Figure 3.35: Mode unfoldings of a third-order tensor along each of its three modes.*

framework described can easily be generalized for n-way tensors, other tensor products, and other graph representations.

## 3.8 Low-Dimensional Tensors and Product Graphs

It is straightforward to see that we can leverage the formulations presented above directly on product graphs without incorporating the structure of the product graph. However, in the following discussion, we study the reconstruction of smooth signals on product graphs by exploiting the low-dimensional structure of the signal on the product graph.

A natural way to organize signals on product graphs is by using tensors that can be thought of as generalizations of a matrix to higher dimensions.[95,96]. In the case of a signal lying on a product of three graphs ,we represent the data by a 3rd order tensor $\boldsymbol{\mathcal{X}}$ where the $(i_1, i_2, i_3)$-th entry in the tensor $\boldsymbol{\mathcal{X}}$ indicates the signal value corresponding to the node in the product graph corresponding to the tuple of the $i_1$-th node in $\mathbf{A}_1$ , the $i_2$-th node in $\mathbf{A}_2$, and the $i_3$-th node in $\mathbf{A}_3$. In the context of recommender engines for example, this would in turn correspond to the $i$-th user's rating of the $j$-th entity at the $k$-th time instant. Hence, a signal $\mathbf{x} \in \mathbb{R}^N$ associated with the product graph defined in Section 3.2.1 can be organized as a $J$-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N_1 \times N_2 \cdots \times N_J}$.

We largely adopt the nomenclature and notation in[95] for tensors. Fibers are the higher order analog of rows and columns in matrix and are obtained by fixing every index but one. The mode-$j$ unfolding of the tensor $\boldsymbol{\mathcal{X}}$, $\boldsymbol{X}_{(j)} \in \mathbb{R}^{N_j \times (N/N_j)}$ arranges the mode-$j$ fibers as the columns of the matrix. The $j$-mode (matrix) product of a tensor $\boldsymbol{\mathcal{X}}$ with a matrix $\Phi$ denoted by $\boldsymbol{\mathcal{X}} \times_j \Phi$ corresponds to multiplying each mode-$j$ fibre by $\Phi$. Let $\bigtimes_{j=1}^{J} F_j$ be short-form for multiplying a tensor along each mode by $\mathcal{F}_j$. That is, $\mathcal{G}(\bigtimes_{j=1}^{J} F_j) = \mathcal{G} \times_1 \mathcal{F}_1 \times_2 \mathcal{F}_2 \cdots \times_J \mathcal{F}_J$. The $CP$-rank of a tensor is defined as the minimum number of rank-one tensors that generate $\boldsymbol{\mathcal{X}}$ as their sum. In general, computing the $CP$-rank of a tensor is difficult, in fact, it is an NP-hard problem. An alternative notion of the rank of a tensor, the *n-rank*,is the tuple of the ranks of the mode-$n$ unfoldings which is easy to compute and yields a greater degree of flexibility. As a result, in this work, we only consider the $n$-rank of a tensor to quantify the low dimensional structure of the tensor. Any signal $\mathbf{x}$ on a product graph $\mathbf{A}$ can be decomposed as $\mathbf{x} = \sum_{i=1}^{R} \mathbf{x}_i^{(1)} \otimes \mathbf{x}_i^{(2)} \cdots \otimes \mathbf{x}_i^{(J)}$ such that each

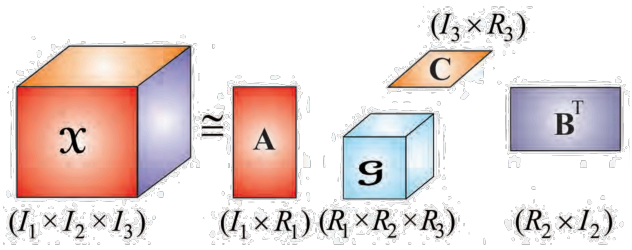



*Figure 3.36: Tucker Decomposition*

$\mathbf{x}^{(j)}$ lies on the respective graph atom $\mathbf{A}_j$. We can study the decomposition of smooth signals on product graphs and show how an extension of the smoothness assumption to signals on product graphs leads to the corresponding tensor $\boldsymbol{\mathcal{X}}$ possessing a low-dimensional structure. We can then also show that the $j$-mode fibers of the tensor $\boldsymbol{\mathcal{X}}$ corresponding to $\mathbf{x}$ are smooth with respect to the $j$-th graph atom $\mathbf{A}_j$. As a result, in the following algorithms and frameworks, we exploit this low-dimensional structure of the graph signal tensor for smooth signal recovery.

### 3.8.1 Reconstruction via Smooth Tucker Decomposition

Similarly to matrix factorization, PCA, and in graph signal processing, transforms such as spectral graph wavelets and the graph Fourier transform, tensor decomposition allows us to detect latent structure in graph data. The Tucker decomposition decomposes a tensor into a *core tensor* and multiple matrices which correspond to different core scalings along each mode.

The Tucker decomposition (Figure 3.36) approximates a tensor with a core-tensor and factor matrices $\boldsymbol{F}_1, \boldsymbol{F}_2$, and $\boldsymbol{F}_3$ as below .

$$\begin{aligned} \underset{\boldsymbol{\mathcal{X}}}{\text{minimize}} \quad & \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}\|_F \\ \text{subject to} \quad & \boldsymbol{\mathcal{X}} = \mathcal{G} \times_1 \boldsymbol{F}_1 \times_2 \boldsymbol{F}_2 \times_3 \boldsymbol{F}_3 \end{aligned} \tag{3.64}$$

We note that the canonical polyadic decomposition decomposition (CPD) is a special case of the Tucker decomposition where the core tensor $\mathcal{G}$ is constrained to be super-diagonal. While we formulate decompositions in the graph setting inspired by both the CP and Tucker decomposition formulations; below, we only consider the Tucker decomposition[95,96]. Such decompositions of the graph signal tensor can be interpreted as signal compression on product graphs. We note that by setting each of the factor matrices equal to (some subset of) the columns of the GFT basis $\mathbf{V}^{(i)}$ of the graph atoms $\boldsymbol{A}_i$, we can enforce bandlimitedness of the product signal on the graph. That is, setting $\boldsymbol{F}_1 = \mathbf{V}^{(1)}_{K_1}$, $\boldsymbol{F}_2 = \mathbf{V}^{(2)}_{K_2}$ and $\boldsymbol{F}_3 = \mathbf{V}^{(3)}_{K_3}$ explicitly enforces smoothness of the graph signal on the product graph. This is the direct analog of approximating the signal with respect to the graph Fourier basis by projecting the signal to a low-frequency subspace of the graph.

Therefore, the Tucker decomposition can be seen as a higher-order PCA[95,96,98]. For a $J$-th order tensor, the Tucker decomposition approximates a tensor $\boldsymbol{\mathcal{X}}$ with a core-tensor $\mathcal{G}$ and $J$ column-wise orthonormal factor matrices $\mathcal{F}_j \in \mathbb{R}^{N_j x R_j}$, $P_j \leq N_j$ $j = \{1, \cdots, J\}$ such that $\boldsymbol{\mathcal{X}} = \mathcal{G}(\bigtimes_{j=1}^{J} F_j)$. Under such a decomposition, the $n$-rank of $\boldsymbol{\mathcal{X}}$ is simply the tuple of the ranks of the mode-$j$ unfoldings, $(R_1, R_2 \cdots R_J)$. We note however that, in general, we need to estimate or fix the $n$-rank

beforehand which is often difficult or unwieldy.

#### Synthesis

We now formulate the direct analog of the synthesis approach on a single graph in **GFProj**, where we project the signal onto the low-frequency bandlimited subspace spanned by the graph Fourier basis vectors, to the Tucker decomposition and product graphs. We note that by setting each of the factor matrices equal to the leading $R_j$ columns of the GFT basis $\mathbf{V}^{(j)}$ of the graph atoms $\mathbf{A}_j$, we can enforce bandlimitedness of the product signal on the graph. That is, by setting $\mathcal{F}_j = \mathbf{V}_{R_j}^{(j)}$ we can explicitly enforce smoothness of the graph signal on the product graph. We call this algorithm **TD-S**.

#### Analysis

Under the analysis framework, we now formulate an optimization problem that enforces in addition to the Tucker decomposition structure described above, a smoothness regularizer. Particularly, given a set of graph signals on a product graph, we aim to find a low-rank decomposition that explicitly enforces smoothness not only across edges within the same mode but also across modes of the tensor or product graph. We can define the optimization problem as follows:

$$\begin{aligned} \underset{\boldsymbol{\mathcal{X}}}{\arg\min} \quad & \|\boldsymbol{\mathcal{Y}} - \mathcal{G}(\bigtimes_{j=1}^{J} F_j)\|_F^2 + \lambda g(\mathcal{F}_i) + \gamma h(\mathcal{F}_i) \\ \text{subject to} \quad & g(\mathcal{F}_i) = \sum_{j=1}^{J} tr(\mathcal{F}_j^T \boldsymbol{L}_j \mathcal{F}_j), \\ & h(\mathcal{F}_i) = tr((\otimes_{j=1}^{J} F_j)^T \boldsymbol{L} (\otimes_{j=1}^{J} F_j)), \\ & \mathcal{F}_i^T \mathcal{F}_i = \boldsymbol{I}, \forall i \end{aligned} \tag{3.65}$$

$g(\cdot)$ enforces smoothness within modes while $h(\cdot)$ enforces smoothness across modes of the tensor or product graph. Since the above formulation is convex over each of the variables we are optimizing over, we can solve it in an alternating fashion by solving for each of $\mathcal{F}_j$ while leaving the other factor matrices $\mathcal{F}_{(-j)}$ fixed. Due to lack of space, we omit detailed derivations and only present the framework in Algorithm 2 which we refer to as **TD-A**. It is closely related to and is a generalization of previous work in[99].

### 3.8.2 Reconstruction via the Nuclear Norm of Unfoldings

In the algorithms presented in Section 3.8.1, we saw that we needed to fix or estimate the $n$-rank beforehand which can often be inconvenient. In this section, we alleviate this inflexibility by presenting a more direct optimization formulation. The sum of each of the ranks of the mode-$j$ unfoldings in the $n$-rank tuple $\sum_{j=1}^{J} R_j$ has been proposed as a proxy for the $n$-rank[100,101,102]. We can then use the nuclear norm as a convex surrogate for the rank as is done in many matrix completion problems[103]. We also enforce smoothness of each of the mode-$j$ fibers with respect to the

**Algorithm 2 (TD-A):** Tucker Decomposition via Alternating Least Squares

**Inputs:** $\boldsymbol{\mathcal{Y}}$, $R_j, \forall j \in \{1 \cdots J\}$ and parameters $\lambda$, $\gamma$
**Initialize:**
  $\mathcal{F}_j^{(0)} = I, \forall j$
**repeat**
  **for** `j ← 1 to J` **do**
    $\mathcal{M}_j^{(k+1)} \leftarrow Y_{(j)} \bigotimes_{j=1, j\neq i}^{J} \mathcal{F}_j^{(k)}$
    $u_j \leftarrow \prod_{j=1, j\neq i}^{J} \text{tr}(\mathcal{F}_j^{(k)T} \boldsymbol{D}_j \mathcal{F}_j^{(k)})$
    $v_j = \prod_{j=1, j\neq i}^{J} \text{tr}(\mathcal{F}_j^{(k)T} \mathbf{A}_j \mathcal{F}_j^{(k)})$
    $\boldsymbol{H}_j^{(k+1)} \leftarrow \mathcal{M}_j^{(k+1)} \mathcal{M}_j^{(k+1)T} - (\lambda + \gamma u_j)\boldsymbol{D}_j + (\lambda + \gamma v_j)\, \mathbf{A}_j$
    $\mathcal{F}_j^{(k+1)} \leftarrow$ top $R_j$ eigenvectors of $\boldsymbol{H}_j^{(k+1)}$
  **end for**
  $k \leftarrow k + 1$
**until** convergence
$\mathcal{G} = \boldsymbol{\mathcal{Y}}(\bigtimes_{j=1}^{J} \mathcal{F}_j^{(k)T})$

$j$-th graph atom and define the following convex optimization problem:

$$\underset{\boldsymbol{\mathcal{X}}}{\text{minimize}} \quad \|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\|_F^2 + \sum_{j=1}^{J} [\alpha \,\text{tr}(\boldsymbol{X}_{(j)}^T \boldsymbol{L}_j \boldsymbol{X}_{(j)}) + \beta \|\boldsymbol{X}_{(j)}\|_*] \tag{3.66}$$

We solve this via the alternating direction method of multipliers (ADMM) framework for separable optimization problems[104,100]. Towards this, we introduce $J$ tensor variables $\boldsymbol{Z}_1, \cdots \boldsymbol{Z}_J$ which represent the $J$ different mode-$j$ unfoldings $\boldsymbol{X}_{(1)}, \cdots, \boldsymbol{X}_{(J)}$ of the tensor $\boldsymbol{\mathcal{X}}$ such that the mode-$j$ unfolding of $\boldsymbol{Z}_j$, $\boldsymbol{Z}_{j,(j)} = \boldsymbol{X}_{(j)}, \forall j \in \{1, 2, \cdots J\}$. We can rewrite (3.66) in the form $f(\boldsymbol{\mathcal{X}}) + \sum_{j=1}^{J} g_j(\boldsymbol{Z}_j)$:

$$\begin{aligned} &\underset{\boldsymbol{\mathcal{X}}, \boldsymbol{Z}_j}{\text{minimize}} \quad \|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\|_F^2 + \sum_{j=1}^{J} [\alpha \,\text{tr}(\boldsymbol{Z}_{(j)}^T \boldsymbol{L}_j \boldsymbol{Z}_{(j)}) + \beta \|\boldsymbol{Z}_{(j)}\|_*] \\ &\text{subject to} \quad \boldsymbol{Z}_j = \boldsymbol{\mathcal{X}}, \forall j \end{aligned} \tag{3.67}$$

We can then write the augmented Lagrangian where $\mathbf{W}_j$ are the Lagrange variables and $\mu$ is the penalty parameter as:

$$\begin{aligned} \underset{\boldsymbol{\mathcal{X}}, \boldsymbol{Z}_{\{j\}}, \mathbf{W}_{\{j\}}}{\text{minimize}} \quad & \|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}\|_F^2 + \sum_{j=1}^{J} [\alpha \,\text{tr}(\boldsymbol{Z}_{j,(j)}^T \boldsymbol{L}_j \boldsymbol{Z}_{j,(j)}) + \\ & \beta \|\boldsymbol{Z}_{j,(j)}\|_* - < \mathbf{W}_j, \boldsymbol{\mathcal{X}} - \boldsymbol{Z}_j > + \frac{\mu}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{Z}_j\|_F^2] \end{aligned} \tag{3.68}$$

Due to limitations on space, we do not present detailed derivations when we minimize 3.68 over $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{Z}_j$ respectively. However, we note that the subproblem when solving for $\boldsymbol{Z}_j$ is

$$\begin{aligned} \underset{\boldsymbol{Z}_{j,(j)}}{\text{minimize}} \quad & \alpha \,\text{tr}(\boldsymbol{Z}_{j,(j)}^T \boldsymbol{L}_j \boldsymbol{Z}_{(j)}) + \beta \|\boldsymbol{Z}_{j,(j)}\|_* + \\ & < \boldsymbol{W}_{j,(j)}, \boldsymbol{Z}_{j,(j)} > + \frac{\mu}{2} \|\boldsymbol{X}_{(j)} - \boldsymbol{Z}_{j,(j)}\|_F^2 \end{aligned} \tag{3.69}$$

We can solve this convex problem by generalized gradient descent[105] by solving a proximity function in each step with respect to the nuclear norm. We define $\mathcal{D}_\tau(\boldsymbol{C})$ to be the operator that shrinks the singular values of $\boldsymbol{C}$ by soft-thresholding the singular values of $\boldsymbol{C}$ by $\tau$. We call the resulting sub-algorithm **GD-Z** and give an overview in Algorithm 3. We refer to the overall algorithm as **NNFold**, the pseudocode for which is presented in Algorithm 4.

**Algorithm 3** (**GD-Z**): Gradient Descent Algorithm for 3.69

**Inputs:** $\boldsymbol{W}_{j,(j)}, \boldsymbol{X}_{(j)}$, and parameters $\alpha$, $\beta$, $\mu$
**Initialize:**
    $\boldsymbol{Z}_{j,(j)} = 0$
**repeat** until convergence
    Choose step size $t$ by backtracking line search
    $\boldsymbol{Z}_{j,(j)} \leftarrow \mathcal{D}_{t\beta}(\boldsymbol{Z}_{j,(j)} - t[2\mu(\boldsymbol{Z}_{j,(j)} - \boldsymbol{X}_{(j)}) + \boldsymbol{W}_{j,(j)} + 2\alpha \boldsymbol{L}\boldsymbol{Z}_{j,(j)}])$
**until** termination
**return** $\boldsymbol{Z}_{j,(j)}$

**Algorithm 4** (**NNFold**): ADMM algorithm for 3.66

**Inputs:** $\boldsymbol{\mathcal{Y}}$, and parameters $\alpha$, $\beta$, $\mu$
**Initialize:**
    $\boldsymbol{X}_{(j)}^{(0)}, \boldsymbol{Z}_{(j)}^{(0)}, \boldsymbol{W}_{(j)}^{(0)} = 0, \forall j$
**repeat** until convergence
    $\boldsymbol{\mathcal{X}}^{(k+1)} \leftarrow \frac{1}{J\mu - 2} \sum_{j=1}^{J} (\mathbf{W}_j^{(k)} + \mu \boldsymbol{Z}_j^{(k)}) - 2\boldsymbol{\mathcal{Y}}$
    **for** `j` $\leftarrow$ `1 to J` **do**
        $\boldsymbol{Z}_{j,(j)}^{(k+1)} \leftarrow$ **GD-Z**$(\boldsymbol{W}_{j,(j)}^{(k)}, \boldsymbol{X}_{(j)}^{(k)}, \alpha, \beta, \mu)$
        $\mathbf{W}_j^{(k+1)} \leftarrow \mathbf{W}_j^{(k)} - \mu(\boldsymbol{\mathcal{X}}^{(k+1)} - \boldsymbol{Z}_j^{(k+1)})$
    **end for**
    $k \leftarrow k + 1$
**until** termination
**return** $\boldsymbol{\mathcal{X}}^{(k)}$

**Theorem 20.** Algorithm 4 **NNFold** converges if the optimal solution set is nonempty such that every limit point of the sequence $\{\boldsymbol{\mathcal{X}}^{(k)}\}$ is an optimal solution.

### 3.9 Numerical Experiments

We construct a synthetic ground truth smooth signal on a product graph $\mathbf{A}$ composed using the Kronecker product of a random geometric graph, star graph and a chain graph each of which has 25 nodes. We use a heat diffusion model over the product graph such that the graph signal tensor has varying $CP$-ranks, $r = 2, 4, 6$. We add noise so the signal we denoise over $\boldsymbol{\mathcal{Y}}$ has an SNR of 5dB. We compare the algorithms **GFProj** (Section 3.8), **GTV** (Section 3.8), **TD-S**(Section 3.8.1), **TD-A**(Section 3.8.1), and **NNFold**(Section 3.8.2). The results are shown in the Table 3.2. We see that **TD-A**, **NNfold** that exploit the product structure consistently outperform **GFProj** and **GTV** while **NNfold** tends to outperform the Tucker decomposition based methods especially for more complex signals.

For the same synthetic smooth signal, we compare Algorithms **TD-A** and **NNfold** for denoising for different levels of noise. The results are shown in Figure 3.37. While at high SNR levels,

| | $r = 2$ | $r = 4$ | $r = 6$ |
|---|---|---|---|
| **GFProj** | 1.85e-3 | 4.6e-2 | 4.9e-2 |
| **GTV** | 7.82e-4 | 8.12e-3 | 8.88e-3 |
| **TD-S** | 1.22e-3 | 9.14e-3 | 1.23e-2 |
| **TD-A** | 9.21e-5 | 6.42e-4 | 3.2e-3 |
| **NNFold** | 2.02e-4 | 5.98e-4 | 9.69e-4 |

*Table 3.2: MSE for denoising smooth signal on product graph using each of the 5 discussed algorithms*

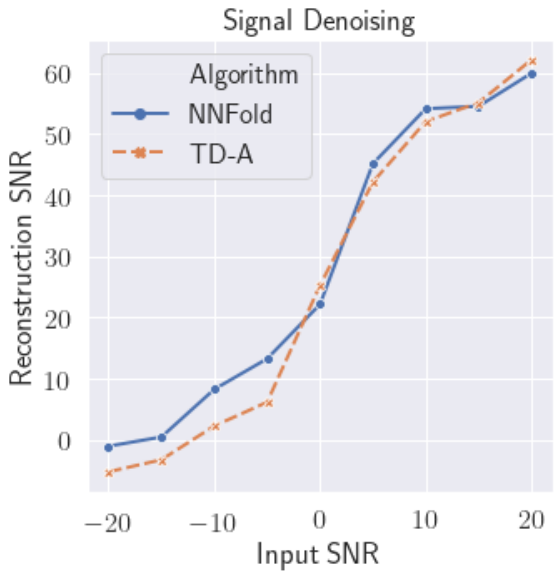



*Figure 3.37: Reconstructed signal SNR for varying levels of noise*

they are both very similar, at low SNR levels or in noisy settings, **NNfold** performs significantly better.

**Computational Complexity:** The graph atoms $\mathbf{A}^{(j)}$ the product graph is composed of are of size $O(poly(N^{\frac{1}{7}}))$. Since we only perform computationally heavy operations like matrix inversion and singular value decomposition ($O(N^3)$) on structures derived from the graph atoms, the algorithms yield significant computational gains over algorithms that do not exploit the structure in product graphs.

Product graphs are a pragmatic and flexible framework for modeling many kinds of multi-modal real-world graph structured data. In this work, we studied the reconstruction of noisy smooth signals on product graphs. We exploited the low-dimensionality of these signals on product graphs by modeling them as low-rank tensors. Our motivations in this work are two-fold in that in addition to better reconstruction performance, we can also gain computational savings. We presented two main algorithms the first of which is based on the Tucker decomposition while the second is based on the the nuclear norm of the mode-$j$ unfoldings of the tensor. Further, we presented numerical experiments that showcase the superior performance of these algorithms with respect to algorithms that do not exploit the structure of product graphs.

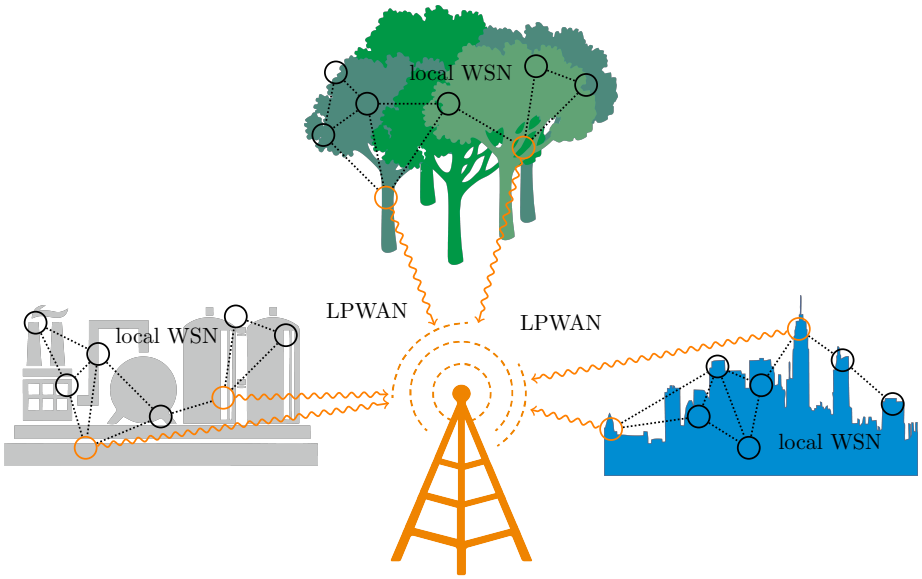



*Figure 3.38: A grand view of systems made of local Wireless Sensor Networks that communicate their readings to a geographically separated hub.*

## 3.10 Applications: Rakeness-Based design of Compressed Sensing for Wireless Sensor Networks

Signals on multiple graphs may model an IoT consisting of local wireless sensor networks (WSNs) performing sets of acquisitions that must be sent to a central hub that may be far from the measurement field. Rakeness-based design of compressed sensing is exploited to allow the administration of the trade off between local communication and the long range transmission needed to reach the hub.

Extensive Montecarlo simulations incorporating real world figures in terms of communication consumption show potential power savings from 25% to almost 50% with respect to a direct approach not exploiting local communication and rakeness.

From an application point of view, signals on graphs fit into a number of scenarios where the relationship between samples is not a simple ordering in time. In unstructured frameworks, the locations at which samples are acquired imply some relationship between them (like the temperature at different spots that are thermically connected in different ways or the consumption of computers in an inhomogeneous company local network) that can be modeled by a generic graph. Moreover, the sensors themselves may belong to a Wireless Sensor Network (WSN) whose nodes have local communication capabilities (that can also be modeled by a graph) and finally deliver their acquisitions to a central hub by means of long range transmissions in some Wide Area Network (WAN).

Figure 3.38 gives an intuitive representation of these structures that suggest exploring the trade-off between local communication/processing and direct transmission to the hub. For example, assuming that the ratio between the typical distance covered by long-range and short-range communications is $10^2$ (tens of meters to kilometers) and that no particular directivity can be provided by sensor nodes antennas, one expects that the ratio between entailed powers is of the order of $10^4$. This is matched by actual consumption of current implementations. For example, Blue-

tooth Low Energy modules come with energy-per-bit efficiencies in the range from 31 nJ/bit to 46 nJ/bit while LoRaWAN implementations exhibit energy efficiencies in the range 19 $\mu$J/bit to 220 $\mu$J/bit so that one may expect a ratio $\epsilon$ between short- and long-range efficiencies between $\epsilon_{\min} = 1.4 \times 10^{-4}$ and $\epsilon_{\max} = 2.4 \times 10^{-3}$. This is more than enough to allow substantial local data exchange before a single long-range transmission is attempted.

Though intuitively straightforward, such a trade-off is not trivial to administer. In our case, we will address it by exploiting a further prior that is commonly valid for real-world signals, i.e., the fact that they have non-white second-order statistics that can be modeled as a further weighted graph connecting the same vertices.

Hence, the signal is ultimately characterized by three graphs: the one representing the structure of its support, the one describing the connectivity of the WSN acquiring it, and the one expressing its second-order statistics. From this point of view, we are dealing with a *signal on multiple graphs*.

### 3.10.1 Acquisition of multiple graph signals

Acquisition largely benefits from priors on the signal, for example, classical frequency domain information allows us to sample signals in a subset of the time instants. What we address here is the efficient acquisition of graph signals exploiting the prior that a they are known to be *sparse* in their Fourier domain, i.e, that $\xi$ has at most $\kappa \ll n$ non-zero components. The graph providing the Fourier basis will be named the *sparsity graph* of the signal.

This is the natural setting in which Compressed Sensing (CS)[89] may be employed. In fact, for a certain $m$, where $m < n$ one may find $m \times n$ matrices $S$ such that the measurements in the vector $y = (y_0, \dots, y_{m-1})^\top = Sx = SU\xi$ can be post-processed to yield the original $x$ despite the fact that $S$ (and thus $SU$) is rectangular. In the graph framework, the easiest case is when $y_j = x_{v_j}$ for certain vertices $v_0, \dots, v_{m-1} \in V$, i.e., when the signal is subsampled and the matrix $S$ is made of $m$ rows of the $n \times n$ identity matrix[106,85].

Instead, we consider measurements of the form $y_j = \sum_{u \in W_j} S_{j,u} x_u$ for certain $W_j \subseteq V$, assuming that one may use local communication to collect the signal values at the vertices $w \in W_j$, compute $y_j$ and send it to the hub. This is precisely the scenario sketched in the introduction, where acquired values can be propagated locally by the WSN with an energy cost per individual communication (a *hop*) that is only $\epsilon$-times the cost of transmitting $y_j$ to the hub.

Usually, one cannot arbitrarily choose the vertices in $W_j$ since, for example, they must correspond to nodes that are geometrically close. We model this with a *sampling graph* that connects two vertices of $V$ if one of them can communicate a value to the other.

The sampling strategy is a generalization of single-vertex sampling scheme that takes into account the sampling graph constraint. To compute the $j$-th measurement $y_j$ we randomly select a vertex $v_j \in V$. Assuming that the sampling graph is connected, a distance $h(v_j, u)$ is defined from every vertex $u \in V$. Given a *hop budget* $H$ we select a subset $W_j \subseteq V$ such that $\sum_{u \in W_j} h(v_j, u) \leq H$. This can be effectively done by modifying the classical Dijkstra algorithm giving the minimum spanning tree, so that it adds a new vertex to the tree only if there are enough hops left to go from that vertex to the root. The vertices in $W_j$ belong to a minimum cost tree that routes signal samples to $v_j$, where $y_j$ can be computed.

This is exemplified in Figure 3.39 where the largest red disk represents the randomly chosen

root $v_j$ and we are given a hop budget $H = 16$. The 3 nearest neighbors of $v_j$ are included in $W_j$ and consume a total of 1 hop each to communicate their values to the root along the red solid edges. Four nodes can connect to the nearest neighbors of the root by means of red dashed edges and thus can communicate their value with 2 hops each. Since the budget is not exhausted by these 11 hops we may add further vertices. However, we cannot accomodate all the vertices that communicate with the root with 3 hops. In this case, a random node is selected among the candidates to satisfy our budget.

Once these signal values are collected, the root may linearly combine them in multiple ways by adopting different weighting coefficients, thus producing more than one measurement. This sample reuse saves communication costs but limits the diversity that can be exploited in computing the measurements. Hence, given a certain $M$ and $\Delta m = \lceil m/M \rceil$, measurements are taken from independently drawn roots $v_j$ and neighborhoods $W_j$ for $j = 0, \ldots, \Delta m - 1$. Then, we assume $v_j = v_{j \pmod{\Delta m}}$ and $W_j = W_{j \pmod{\Delta m}}$ for $j \geq \Delta m$.

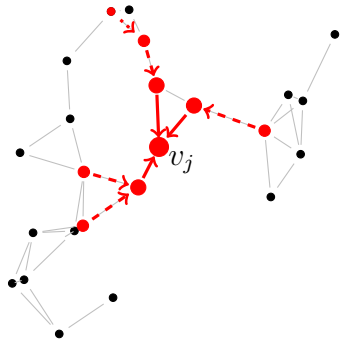



*Figure 3.39: A generalization of single-vertex sampling. In the graph nodes are connected only if they are closer than a certain threshold.*

As far as coefficients are concerned, the most trivial, CS-inspired, option is to take each non-null entry of $S$ to be the realization of an independent normal random variable. We will denote this classical choice as the *random* option.

### 3.10.2 Correlation graph and Rakeness-Based CS

Independently of their sparsity, most signals feature some sort of energy *localization* that can be detected by considering their correlation matrix $\mathcal{X} = \mathbf{E}[xx^\top]$ and verifying that its eigenvalues are not identical and, thus, there are subspaces along which most of the energy of $x$ concentrates. Localization and sparsity are different priors since the subspaces along which energy concentrates does not need to be $\kappa$-dimensional canonical subspaces in the sparsity reference system.

It is a graph prior since the matrix $\mathcal{X}$ is a symmetric matrix that can be interpreted as the incidence matrix of a complete, graph where the edge between vertex $v'$ and vertex $v''$ has a weight $\mathbf{E}[x_{v'} x_{v''}]$.

The exploitation of such a prior to optimize CS for time-domain signals has been investigated based on the *rakeness* concept [107]. The basic observation is that it is convenient to design the statistics of the coefficients $S_{j,u}$ such that $y_j$ is, on the average, able to *rake* as much energy as possible from the signal. Due to the random nature of the signal, observing only its maximum-energy component (the so-called principal component) is not enough to reconstruct it and energy maximization should be tempered by the need to span the whole signal space. This results in a design flow that solves a constrained maximization problem that depends on the correlation of the signal to be acquired. This optimization problem outputs the correlation matrix of the process that should be used to generate the coefficients $S_{j,u}$ to improve the acquisition capabilities and

performance of CS[108].

Hence, as a second option, instead of drawing the coefficients as random independent normals, for each vertex subset $W_j$ the correlation subgraph with incidence matrix $\mathcal{X}_{|W_j}$ linking the vertices in $W_j$ is used as an input for textcolorredthe rakeness-based design of the corresponding set of coefficients. We use the simplest version of the aforementioned design flow such that the correlation matrix of the set of coefficients turns out to be

$$\Sigma_{|W_j} = \frac{1}{2}\left(\frac{\mathcal{X}_{|W_j}}{\mathrm{tr}\left(\mathcal{X}_{|W_j}\right)} + \frac{I_{n_j}}{n_j}\right) \tag{3.70}$$

where $n_j = |W_j|$ is the cardinality of $W_j$ and $I_{n_j}$ is the $n_j \times n_j$ matrix. Given $\Sigma_{|W_j}$ symmetric and positive semidefinite the non-null entries of the $j$-th row of $S$ are realizations of jointly-Gaussian random variables whose correlation matrix is $\Sigma_{W_j}$.

### 3.10.3 Empirical evidence

To assess the effectiveness of the proposed approach we perform a Montecarlo analysis of a few configurations. In all trials $n = 128$ while the sparsity level is taken as $\kappa \in \{6, 12, 24\}$ to explore priors with different strengths.

In each trial the *sampling graph* is a realization of a Geometric random graph with $n$ nodes uniformly distributed in $[0, 1]^2$ with connections if their distance is less than 0.15 (label Geo-0.15). Hop budgets $H \in \{64, 128, 256\}$ are considered. The *sparsity graph* can either be the same as the *sampling graph* or the realization of one of the following random graphs:

- Erdös-Rényi graph with probability of connection equal to 0.1 (label ER-0.1)
- Barabasi-Albert graph whose construction starts from a 10-vertices ER-0.1 and connects every new vertex to 5 previous vertices (label BA-10-5)
- Watts-Strogatz graph with 6 neighbors in the initial ring and with a rewiring probability equal to 0.3 (label WS-6-0.3)

In all cases possibly non-connected realizations are discarded.

To simulate localization, the $\kappa$ non-zero components in $\xi$ are selected with a non-uniform probability. This probability distribution is communicated to neither the sampling mechanism nor the reconstruction algorithm. What is known by the sampling stage is only the correlation matrix $\mathcal{X} = \mathbf{E}[xx^\top]$ from which the various correlation submatrices $\mathcal{X}_{|W_j}$ are taken to compute (3.70).

White Gaussian noise is added to the samples giving them an Intrinsic Signal-to-Noise-Ratio ISNR $= 60\,\mathrm{dB}$. Reconstruction is obtained by Basis Pursuit with De-noising (BPDN) as implemented by SPGL1[109].

Performance is evaluated as the Probability of Correct Reconstruction (PCR) defined as the probability that the relative error in the reconstruction corresponds to a loss of not more than 6 dB with respect to the ISNR, i.e., $\texttt{PCR} = \Pr\left\{{}^{\|x\|}/_{\|x-\hat{x}\|} \geq 54\,\mathrm{dB}\right\}$.

The qualitative features of all the observed trends coincide. Figure 3.40 reports how the PCR depends on the number of measurements in three cases that correspond to $\kappa = 6, 12, 24$, i.e., to progressively weakening sparsity priors. The vertex-only option (black dotted track) is taken as a reference.

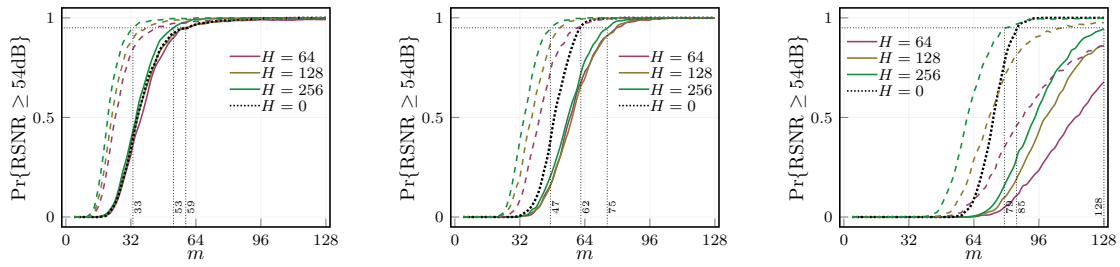



*Figure 3.40: PCR plotted against $m$ for different configurations. Track color indicates the available hop budget ($H = 0$ signifying vertex-only sampling). Solid lines correspond to random CS, dashed lines correspond to rakeness-based CS. The number of measurements needed to guarantee a PCR of 95% is highlighted for vertex-only sampling ($H = 0$) and for the best random and rakeness-based options. In a) $\kappa = 6$, the sparsity graph is the same Geo-0.15 used for sampling, and each vertex contributes not more than $M = 4$ measurements. In b) $\kappa = 14$, the sparsity graph is WS-6-0.3, and each vertex contributes not more than $M = 8$ measurements. In c) $\kappa = 24$, the sparsity graph is ER-0.1, and each vertex contributes not more than $M = 16$ measurements.*

In all those plots as well as in all tested cases, the position of the continuous tracks shows that if the samples collected by local communication are combined with purely random coefficients no gain is obtained. Local communication can be traded for long-range one only if we exploit the correlation graph by means of rakeness-based CS. An optimized choice of the coefficients leverages the availability of multiple samples to compute more informative measurements. Hence, the same reconstruction quality can be obtained at the hub even if less measurements are sent to it through long-range transmission.

This points toward a possible power saving. To quantify this, we normalize to 1 the energy needed by a long-range transmission so that the cost of a short-range transmission gets normalized to the ratio $\epsilon$ discussed in the Introduction. With this, the power needed by the collection of samples and transmission of the measurements is $P_{\mathrm{CS}} = m^{\mathrm{CS}} + \epsilon H \left\lceil m^{\mathrm{CS}}/M \right\rceil$, where $m^*$ is the number of measurement needed to achieve the prescribed performance, $M$ is the maximum number of measurement that each node can compute with the samples it collected, and $H$ is the hop budget constraining sample collection. This compares favorably with $P_{\mathrm{VS}} = m^{\mathrm{VS}}$, i.e., with the power (equal to the number of measurements) needed to achieve the same performance level by simple vertex-sampling.

Figure 3.41 reports the ratio $P_{\mathrm{CS}}/P_{\mathrm{VS}}$ when the desired PCR is set to 95% and in all the cases we tested in an extensive Monte-Carlo simulation.

Though it is evident that as the sparsity prior $\kappa$ increases, our framework looses its ability of allowing substantial subsampling and thus power saving, rakeness-based CS is almost always able to yield substantial power saving. Actual saving depends on the relationship between the sparsity graph and the sampling graph and on the value of $\epsilon$, but in most of the non-extreme cases, at least 25% of the power is unnecessary if rakeness-based CS is adopted.

Hence, rakeness-based CS applied to multiple-graph signals is an effective way to administer the trade-off between short- and long-range communication in a quite common IoT scenario that sees the interplay of local WSN and geographic information hubs. It is estimated that its exploitation may yield not less than 25% of power saving.

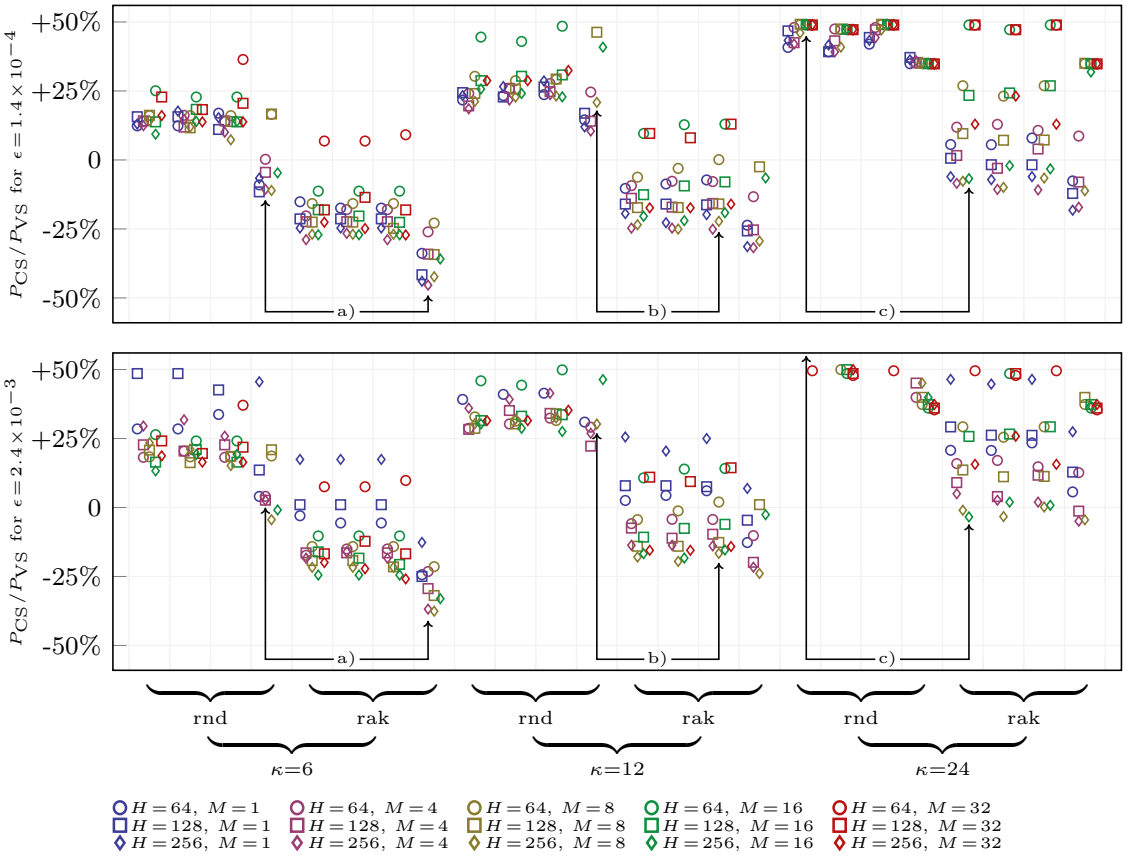



*Figure 3.41: Power saving with respect to vertex-only sampling in all the tested configurations. Each group of 4 points with the same shape and color correspond to the 4 sparsity graph (ER-01, BA-10-2, WE-10-0.6, and Geo-0.51). The color of a point indicates the available hop budget $H$, while its shape indicates the maximum number of measurements $M$ provided by each vertex. Different sparsities $\kappa$ are shown and for each sparsity, random and rakeness-based CS is considered. The upper plot considers a ratio between the energy needed by short-range and long-range communication equal to $\epsilon = \epsilon_{\min} = 1.4 \times 10^{-4}$. The lower plot considers $\epsilon = \epsilon_{\max} = 2.4 \times 10^{-3}$. Highlighted points correspond to the a), b), and c) plots of Figure 3.40.*

## 3.11 Applications: Bayesian Sampling and Recovery of Smooth Signals on Graphs

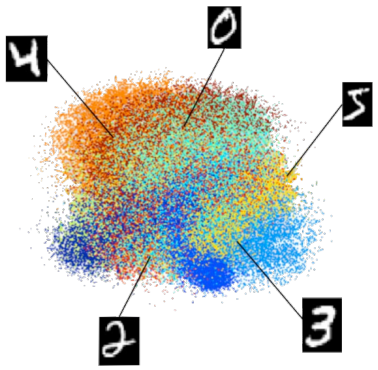


*Figure 3.42: Graph representation of the USPS Digit dataset*

In the semi-supervised learning setting, it is often of interest to provide a confidence score or a level of uncertainty in addition to our decision. Towards this goal, in this work, we introduce a Bayesian treatment of semi-supervised learning on graphs via sampling theory. We build an appropriate prior distribution to model smooth signals or label distributions on a graph. We can then perform semi-supervised learning on a graph by sampling the graph using the optimal random sampling distribution and then recovering the label distribution. When constructing the posterior, a key issue that needs to be overcome is that the label space is discrete while the graph signal is continuous. We study the posterior distribution of the labels under this framework and introduce scalable numerical methods, for MCMC-based sampling to sample from the posterior distribution which is often intractable. We can estimate the mean and variance of the posterior distribution over each node which we argue capture the confidence score and the uncertainty for the estimate at the node. We specifically study relationships between uncertainty quantification and the graph structure. We use real-world data to further demonstrate and validate our framework.

## 3.12 Applications: Energy-Efficient Route Planning for Autonomous Aerial Vehicles

We use graph signal sampling and recovery techniques to plan routes for autonomous aerial vehicles. We proposed a novel algorithm that plans an energy-efficient flight trajectory by considering the influence of wind. We model the weather stations as nodes on a graph and model wind velocity at each station as a smooth graph signal.[110]

# 4

# Reconstruction of Piecewise Smooth Graph Signals

Signal estimation from noisy observations is a classic problem in signal processing and has applications in signal inpainting, collaborative filtering, recommendation systems and other large-scale data completion problems. Since noise can have deleterious, cascading effects in many downstream tasks, being able to efficiently and accurately filter and reconstruct a signal is of significant importance.

In graph signal processing, a common assumption is that the graph signal is smooth with respect to the graph, that is, the signal coefficients do not vary much over local neighborhoods of the graph. However, this characterization is insufficient for many real-world signals that exhibit spatially inhomogeneous levels of smoothness over the graph. In social networks for example, within a given community or social circle, users' profiles tend to be homogeneous, while within a different social circle they will be of different, yet still have homogeneous values. Consequently, the signal is often characterized by large variations between regions and small variations within regions such that there are localized discontinuities and patterns in the signal. As a result, it is necessary to develop representations and algorithms to process and analyze such *piecewise smooth* graph signals.

In this chapter, we study the denoising of the class of piecewise smooth graph signals (including but not limited to piecewise constant graph signals), which is complementary to the class of smooth graph signals that exhibit homogeneous levels of smoothness over the graph. The reconstruction of smooth graph signals has been well studied in previous work both within graph signal processing[1,111] as well as in the context of Laplacian regularization[112,113]. In this chapter, we develop frameworks and algorithms for the reconstruction and sampling of piecewise-smooth graph signals. We follow a similar structure to the presentation in Chapter 3 for smooth signals where we first present a framework for reconstructing or estimating piecewise smooth signals as defined in Section 2.3. Particularly, we want to understand the difference between graph total variation denoising and wavelet smoothing on graphs. As in Section 3.7, we study methods from both the analysis framework for signal estimation and the synthesis framework. We present both the graph trend filtering formulation which falls under the analysis framework, and wavelet thresholding which falls under the synthesis framework. We also propose studying how these two frameworks are fundamentally different both in terms of their empirical performance and theoretical properties

## 4.1 Graph Trend Filtering

### 4.1.1 Introduction

The Graph Trend Filtering (GTF) framework[30], which applies total variation denoising to graph signals[114], is a particularly flexible and attractive approach that regularizes discrete graph differences using the $\ell_1$ norm. Although the $\ell_1$ norm based regularization has many attractive properties[115], the resulting estimates are biased toward zero for large coefficients. To alleviate this bias effect, non-convex penalties such as the Smoothly Clipped Absolute Deviation (SCAD) penalty[116] and the Minimax Concave Penalty (MCP)[117] have been proposed as alternatives. These penalties behave similarly to the $\ell_1$ norm when the signal coefficients are small, but tend to a constant when the signal coefficients are large. Notably, they possess the so-called *oracle property*: in the asymptotics of large dimension, they perform as well as the case where we know in advance the

support of the sparse vectors[118]-[122].

In this work, we strengthen the GTF framework in[30] by considering a large family of possibly non-convex regularizers, including SCAD and MCP that exhibit superior reconstruction performance over $\ell_1$ minimization for the denoising of piecewise smooth graph signals. Furthermore, we extend the GTF framework to allow vector-valued signals, e.g. time series, on each node of the graph, which greatly broadens the applicability of GTF to applications in social networks[123], gene networks, and semi-supervised classification[124,125].

Through theoretical analyses and empirical performance, we demonstrate that the use of non-convex penalties improves the performance of GTF in terms of both reduced reconstruction error and improved support recovery, i.e. how accurately we can localize the discontinuities of the piecewise smooth signals. Our contributions can be summarized as follows:

- Theoretically, we derive the statistical error rates of the signal estimates, defined as first-order stationary points of the proposed GTF estimator. We derive the rates in terms of the noise level and the alignment of the ground truth signal with respect to the underlying graph, without making assumptions on the piecewise smoothness of the ground truth signal. The better the alignment, the more accurate the estimates. Importantly, the estimators do not need to be the global minima of the proposed non-convex problem, which are much milder requirements and important for the success of optimization. For denoising vector-valued signals, the GTF estimate is more accurate when each dimension of the signal shares similar patterns across the graph.
- Algorithmically, we propose an ADMM-based algorithm that is guaranteed to converge to a critical point of the proposed GTF estimator.
- Empirically, we demonstrate the performance improvements of the proposed GTF estimators with non-convex penalties on both synthetic and real data for signal estimation, support recovery, event detection, and semi-supervised classification.

The rest of this section is organized as follows. Section 4.1.2 reviews related works and their relationships to our work and an introduction to graph trend filtering. Section 4.2 presents the proposed GTF framework with non-convex penalties and vector-valued graph signals. Section 4.3 develops its performance guarantees, and Section 4.4 presents an efficient algorithm based on ADMM. Numerical performance of the proposed approach is examined on both synthetic and real-world data for denoising and semi-supervised classification in Section 4.5. Throughout this paper, we use boldface letters $\boldsymbol{a}$ and $\boldsymbol{A}$ to represent vectors and matrices respectively. The transpose of $\boldsymbol{A}$ is denoted as $\boldsymbol{A}^\top$. The $\ell$-th row of a matrix $\boldsymbol{A}$ is denoted as $\boldsymbol{A}_{\ell\cdot}$, and the $j$-th column of a matrix $\boldsymbol{A}$ is denoted as $\boldsymbol{A}_{\cdot j}$. The cardinality of a set $T$ is denoted as $|T|$. For any set $T \subseteq \{1, 2, ..., r\}$ and $\mathbf{x} \in \mathbb{R}^r$, we denote $(\mathbf{x})_T \in \mathbb{R}^{|T|}$ such that $x_\ell \in (\mathbf{x})_T$ if and only if $\ell \in T$ for $\ell \in \{1, 2, ..., r\}$. Similarly, we define a submatrix $\boldsymbol{A}_{T\cdot} \in \mathbb{R}^{|T|\times d}$ of $\boldsymbol{A} \in \mathbb{R}^{r\times d}$ that corresponds to pulling out the rows of $\boldsymbol{A}$ indexed by $T$. The $\ell_2$ norm of a vector $\boldsymbol{a}$ is defined as $\|\boldsymbol{a}\|_2$, and the spectral norm of a matrix $\boldsymbol{A}$ is defined as $\|\boldsymbol{A}\|$. The pseudo-inverse of a matrix $\boldsymbol{A}$ is defined as $\boldsymbol{A}^\dagger$. For a function $h(\mathbf{x}) : \mathbb{R}^p \to \mathbb{R}$, we write $\nabla_{\mathbf{x}} h(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*}$ to denote the gradient or subdifferential of $h(\mathbf{x})$, if they exist, evaluated at $\mathbf{x} = \mathbf{x}^*$. When the intention is clear, this may be written concisely as $\nabla h(\mathbf{x}^*)$. We also follow the standard asymptotic notations. If for some constants $C, N > 0$, $|f(n)| \leq C|g(n)|$ for all $n \geq N$, then $f(n) = O(g(n))$; if $g(n) = O(f(n))$, then $f(n) = \Omega(g(n))$. Finally, Table 4.1 summarizes some key notations used here for convenience.

| Symbol | Description | Dimension |
|---|---|---|
| $\boldsymbol{\Delta}$ | oriented incidence matrix | $m \times n$ |
| $\boldsymbol{\Delta}^{(k+1)}$ | $k$th order graph difference operator | $r \times n$ |
| $\boldsymbol{\beta}$ | scalar-valued graph signal | $n$ |
| $\boldsymbol{B}$ | vector-valued graph signal | $n \times d$ |
| $\mathbf{y}$ | noisy observation of $\boldsymbol{\beta}$ | $n$ |
| $\boldsymbol{Y}$ | noisy observation of $\boldsymbol{B}$ | $n \times d$ |
| $\boldsymbol{\Delta}_{\ell\cdot}$ | $\ell$-th row of $\boldsymbol{\Delta}$ | $n$ |
| $\boldsymbol{B}_{\cdot j}$ | $j$-th column of $\boldsymbol{B}$ | $n$ |
| $\|\boldsymbol{\Delta}^{(k+1)}\|$ | spectral norm of $\boldsymbol{\Delta}^{(k+1)}$ | 1 |

*Table 4.1: Key notation used in this chapter*

### 4.1.2 Related Work and Connections

Estimators that adapt to spatial inhomogeneities have been well studied in the literature via regularized regression, total variation and splines[126]-[128]. Most of these methods involve locating change points or knots that denote a distinct change in the behavior of the function or the signal.

For example, in one of the earliest relevant works[126], least-squares regression penalized with total variation penalties[127,128] are shown to be least squares splines with locally data-adaptive placed *knots*.

Our work is most related to the spatially adaptive GTF estimator introduced in[30] that *smoothens* or *filters* noisy signals to promote piecewise smooth behavior with respect to the underlying graph structure; see also[129]. In the same spirit as[126], the fused LASSO and univariate trend filtering framework developed in[114,130,31] use discrete difference operators to fit a time series signal using piecewise polynomials. The GTF framework generalizes univariate trend filtering by generalizing a path graph to arbitrarily complex graphs. Specifically, by appropriately defining the discrete difference operator, we can enforce piecewise constant, piecewise linear, and more generally piecewise polynomial behaviors over the graph structure. In comparison to previous work[30], in this chapter, we have significantly expanded its scope by allowing vector-valued data over the graph nodes and a broader family of possibly non-convex penalties.

We note that while a significant portion of the relevant literature on GTF or the fused LASSO has focused on the sparsistency or support recovery conditions under which we can ensure the recovery of the location of the discontinuities or knots[131,132], in this work, we study the asymptotic error rates of our estimator with respect to the mean squared error. Our analysis of error rates leverages techniques in[133,134] that result in sharp error rates of total variation denoising via oracle inequalities, which we have carefully adapted to allow *non-convex* regularizers. The obtained error rates can be translated into bounds on support recovery or how well we can localize the boundary by leveraging techniques in[135].

Employing a graph-based regularizer that promotes similarities between the signal values at connected nodes has been investigated by many communities, such as graph signal processing, machine learning, applied mathematics, and network science. The Network LASSO proposed in[123], which is similar to the GTF framework with multi-dimensional or vector-valued data, focused on the development of efficient algorithms without any theoretical guarantees. The recent works by Jung et al.[125,136,124] have analyzed the performance of Network LASSO for semi-supervised learning when the graph signal is assumed to be *clustered* according to the labels using the network

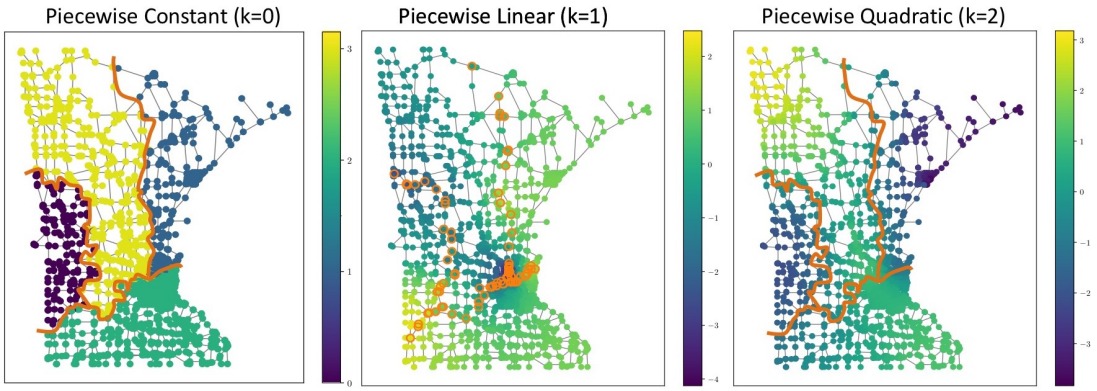



*Figure 4.1: Illustration of piecewise smooth signals on the Minnesota road graph. From left to right: piecewise constant ($k = 0$), piecewise linear ($k = 1$), and piecewise quadratic ($k = 2$) graph signals. Note that the highlighted change points, i.e. the support of $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star$, are edges for even $k$ and nodes for odd $k$.*

null space property and the network compatibility condition inspired by related concepts in compressed sensing[137]. In contrast, our analysis does not make assumptions on the graph signal, and the error rate is adaptive to the alignment of the signal and the graph structure used in denoising.

A well-studied generalization of the sparse linear inverse problem is when there are multiple measurement vectors (MMV), and the solutions are assumed to have a *common sparsity pattern*[138,139,140]. Sharing information across measurements, and thereby exploiting the conformity of the sparsity pattern, has been shown to significantly improve the performance of sparse recovery in compressive sensing and sparse coding[141,145]. Motivated by these works, we consider vector-valued graph signals that are regarded as multiple measurements of scalar-valued graph signals sharing discontinuity patterns.

There are a few variants of non-convex penalties that promote sparsity such as SCAD, MCP, weakly convex penalties, and $\ell_q$ $(0 \leq q < 1)$ minimization[118,146]. In this thesis, we consider and develop theory for a family of non-convex penalties parametrized similarly to that in[118] with SCAD and MCP as our prime examples, although it is valid for other non-convex penalties.

More broadly, a significant amount of work has been devoted towards image denoising. Previous work related to signal recovery include Gaussian smoothing, Wiener local empirical filtering, and wavelet thresholding methods. Signal inpainting reconstructs lost or deteriorated parts of signals, including images and videos. Standard techniques include total variation-based methods which have enjoyed widespread popularity[111,127,128], image model-based methods and sparse representations. It is natural to view graph trend filtering as a generalization of total variation defined on an image (a 2d grid) to arbitrary weighted graphs. Some of the first works generalizing total variation to graph total variation include[129]. Further, in the context of graphs, signal recovery or denoising algorithms have been presented for *globally* smooth signals[26] especially in the context of graph Laplacian regularization[112,113,147,148,149,150,151,152]. However, as we discuss later, these are not locally adaptive which limits their efficacy in many settings.

### 4.1.3 Denoising Piecewise Smooth Graph Signals via GTF

Assume we observe a noisy signal $\mathbf{y}$ over the graph under i.i.d Gaussian noise:

$$\mathbf{y} = \boldsymbol{\beta}^\star + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}), \tag{4.1}$$

and seek to reconstruct $\boldsymbol{\beta}^\star$ from $\mathbf{y}$ by leveraging the graph structure. When $\boldsymbol{\beta}$ is a smooth graph signal, Laplacian smoothing [112,113,147–149] can be used, which solves the following problem:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\Delta} \boldsymbol{\beta}\|_2^2, \tag{4.2}$$

where $\lambda > 0$. However, it cannot localize abrupt changes in the graph signal when the signal is piecewise smooth.

Graph trend filtering (GTF) [30] is a flexible framework for estimation on graphs that is adaptive to inhomogeneity in the level of smoothness of an observed signal across nodes. The $k$th order GTF estimate is defined as:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\Delta}^{(k+1)} \boldsymbol{\beta}\|_1, \tag{4.3}$$

which can be regarded as applying total variation or fused LASSO with the graph difference operator $\boldsymbol{\Delta}^{(k+1)}$ [114,32]. The sparsity-promoting properties of the $\ell_1$ norm have been well-studied [153]. Consequently, applying the $\ell_1$ penalty in GTF sets many of the (higher-order) graph differences to zero while keeping a small fraction of non-zero values. GTF is then *adaptive* over the graph; its estimate at a node adapts to the smoothness in its localized neighborhood.

**Remark 3.** We can use mixed piecewise penalties to encourage different kinds of piecewise polynomial behavior by stacking the graph difference matrices of different orders. For example, we can use a regularizer $\lambda \|\boldsymbol{\Delta}^{(l+1)} \boldsymbol{\beta}\|_1 + \gamma \|\boldsymbol{\Delta}^{(m+1)} \boldsymbol{\beta}\|_1$ and optimize (4.3) with $\boldsymbol{\Delta}^{(k+1)}$ replaced by $\tilde{\boldsymbol{\Delta}} = \left[\boldsymbol{\Delta}^{(l+1)\top}; \gamma/\lambda \boldsymbol{\Delta}^{(m+1)\top}\right]^\top$. In this chapter, however, we only consider the basic graph difference operator defined for a fixed $k$.

## 4.2 Vector-Valued GTF with Non-Convex Penalties

In this section, we first extend GTF to allow a broader family of non-convex penalties and then extend it to handle vector-valued signals over the graph.

### 4.2.1 (Non-)convex Penalties

The $\ell_1$ norm penalty considered in (4.3) is well-known to produce biased estimates , which motivates us to extend the GTF framework to a broader class of sparsity-promoting regularizers that are not necessarily convex. We wish to minimize the following generalized $k$th order GTF loss function:

$$f(\boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{y} - \boldsymbol{\beta}\|_2^2 + g(\boldsymbol{\Delta}^{(k+1)} \boldsymbol{\beta}; \lambda, \gamma), \quad \boldsymbol{\beta} \in \mathbb{R}^n, \tag{4.4}$$

where

$$g(\boldsymbol{\Delta}^{(k+1)} \boldsymbol{\beta}) \triangleq g(\boldsymbol{\Delta}^{(k+1)} \boldsymbol{\beta}; \lambda, \gamma) = \sum_{\ell=1}^{r} \rho((\boldsymbol{\Delta}^{(k+1)} \boldsymbol{\beta})_\ell; \lambda, \gamma)$$

is a regularizer defined as the sum of the penalty function $\rho(\cdot;\lambda,\gamma):\mathbb{R}\to\mathbb{R}$ applied element-wise to $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}$. Here, $r=m$ for even $k$ and $r=n$ for odd $k$ to account for different dimensions of $\boldsymbol{\Delta}^{(k+1)}$; see (2.4). We will refer to the GTF estimator that minimizes $f(\boldsymbol{\beta})$ as *scalar-GTF*.

Similarly to[118,120,154], we consider a family of penalty functions $\rho(\cdot;\lambda,\gamma)$ that satisfies the following assumptions.

**Assumption 1.** Assume $\rho(\cdot;\lambda,\gamma)$ satisfies the following:

(a) $\rho(t;\lambda,\gamma)$ satisfies $\rho(0;\lambda,\gamma)=0$, is symmetric around 0, and is non-decreasing on the real non-negative line.

(b) For $t\geq 0$, the function $t\mapsto\frac{\rho(t;\lambda,\gamma)}{t}$ is non-increasing in $t$. Also, $\rho(t;\lambda,\gamma)$ is differentiable for all $t\neq 0$ and sub-differentiable at $t=0$, with $\lim_{t\to 0^+}\rho'(t;\lambda,\gamma)=\lambda$. This upper bounds $\rho(t;\lambda,\gamma)\leq\lambda|t|$.

(c) There exists $\mu>0$ such that $\rho(t;\lambda,\gamma)+\frac{\mu}{2}t^2$ is convex.

Many penalty functions satisfy these assumptions. Besides the $\ell_1$ penalty, the non-convex SCAD[116] penalty

$$\rho_{\text{SCAD}}(t;\lambda,\gamma)=\lambda\int_0^{|t|}\min\left(1,\frac{(\gamma-u/\lambda)_+}{\gamma-1}\right)du,\gamma\geq 2, \tag{4.5}$$

and the MCP[117]

$$\rho_{\text{MCP}}(t;\lambda,\gamma)=\lambda\int_0^{|t|}\left(1-\frac{u}{\lambda\gamma}\right)_+du,\quad\gamma\geq 1 \tag{4.6}$$

also satisfy them. We note that Assumption 1 (c) is satisfied for SCAD with $\mu\geq\mu_{\text{SCAD}}=\frac{1}{\gamma-1}$ and for MCP with $\mu\geq\mu_{\text{MCP}}=\frac{1}{\gamma}$. Fig. 4.2 illustrates the $\ell_1$, SCAD and MCP penalties for comparison. While the non-convexity means that in general, we may not always find the global optimum of $f(\boldsymbol{\beta})$, it often affords us many other advantages. SCAD and MCP both taper off to a constant value and hence apply less shrinkage for higher values. As a result, they mitigate the bias effect while promoting sparsity. Further, they are smooth and differentiable for $t\geq 0$ and are both upper bounded by the $\ell_1$ penalty for all $t$.

### 4.2.2 Vector-Valued GTF

In many applications, the signals on each node are in fact multi-dimensional or *vector-valued*, e.g. time series in social networks, multi-class labels in semi-supervised learning, feature vectors of different objects in feature selection. Therefore, it is natural to consider an extension to the graph signal denoising problem, where the graph signal on each node is a $d$-dimensional vector instead of a scalar. In this scenario, we define a vector-valued graph signal to be piecewise smooth if it is piecewise smooth in each of its $d$ dimensions, and assume their discontinuities to coincide over the same small set of edges or nodes. Further, we denote the vector-valued signal of interest as $\boldsymbol{B}^\star\in\mathbb{R}^{n\times d}$, such that the $i$th row of the matrix $\boldsymbol{B}$ corresponds to the $i$th node of the graph. The noise model for the observation matrix $\boldsymbol{Y}\in\mathbb{R}^{n\times d}$ is defined as

$$\boldsymbol{Y}=\boldsymbol{B}^\star+\boldsymbol{E}, \tag{4.7}$$

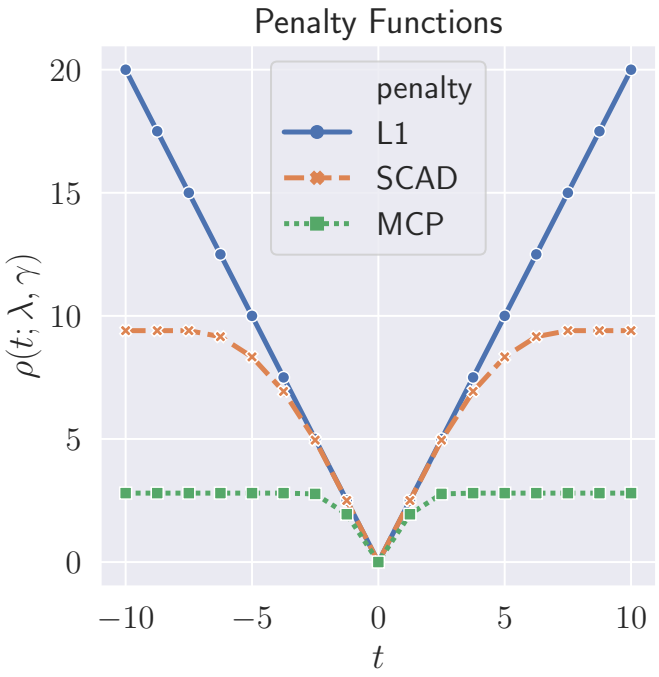



*Figure 4.2: Illustration of $\rho(\cdot;\lambda,\gamma)$ for $\ell_1$, SCAD ($\gamma = 3.7$), and MCP ($\gamma = 1.4$), where $\lambda = 2$. Both SCAD and MCP move towards $\ell_1$ as $\gamma$ increases.*

where each element of $\boldsymbol{E} \in \mathbb{R}^{n\times d}$ is drawn i.i.d from $\mathcal{N}(0,\sigma^2)$. A naïve approach is to estimate each column $\boldsymbol{B}_{\cdot j}$ of $\boldsymbol{B}$ separately via scalar-GTF:

$$\min_{\boldsymbol{B}\in\mathbb{R}^{n\times d}} \sum_{j=1}^{d} f(\boldsymbol{B}_{\cdot j}). \tag{4.8}$$

However, this formulation does not take full advantage of the multi-dimensionality of the graph signal. Instead, when the columns of $\boldsymbol{B}$ are correlated, coupling them can be beneficial such that we encourage the sharing of information across dimensions or features. For example, if one column $\boldsymbol{B}_{\cdot i}$ exhibits strong piecewise smoothness over the graph, and therefore has compelling evidence about the relationship between nodes, sharing that information to a related column $\boldsymbol{B}_{\cdot j}$ can improve the overall denoising and filtering performance. As a result, we formulate a *vector-GTF* problem as follows:

$$\min_{\boldsymbol{B}\in\mathbb{R}^{n\times d}} \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{B}\|_{\mathrm{F}}^2 + h(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B};\lambda,\gamma), \tag{4.9}$$

where the new penalty function $h(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}) \triangleq h(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B};\lambda,\gamma) : \mathbb{R}^{r\times d} \to \mathbb{R}$ is the sum of $\rho(\cdot;\lambda,\gamma)$ applied to the $\ell_2$ norm of each row of $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B} \in \mathbb{R}^{r\times d}$:

$$h(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}\,;\lambda,\gamma) = \sum_{\ell=1}^{r} \rho\left(\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_{\ell\cdot}\|_2;\lambda,\gamma\right). \tag{4.10}$$

By enforcing sparsity on $\left\{\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_{\ell\cdot}\|_2\right\}_{1\le l\le r}$, we are coupling $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}_{\cdot j}$ to be of similar sparsity patterns across $j = 1,\ldots,d$. Note the difference from (4.8), where elements of $(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_{\ell\cdot}$

can be set to zero or non-zero independently.

### 4.3 Theoretical Guarantees and Statistical Limits

In this section, we present the error rates and support recovery guarantees of the generalized GTF estimators, namely scalar-GTF (5.3) and vector-GTF (4.9), under the AWGN noise model. Before continuing, we first define a few useful quantities. Let $C_G$ be the number of connected components in the graph $G$, or equivalently, the dimension of the null space of $\boldsymbol{\Delta}^{(k+1)}$. Further, let $r$ be the number of rows of $\boldsymbol{\Delta}^{(k+1)}$, and $\zeta_k$ be the maximum $\ell_2$ norm of the columns of $\boldsymbol{\Delta}^{(k+1)\dagger}$.

Throughout this section, we assume the penalty function, in addition to Assumption 1, also satisfies the following.

**Assumption 2.**

$$\mu < \frac{1}{\|\boldsymbol{\Delta}^{(k+1)}\|^2}.$$

**Proposition 1** (Bound on $\zeta$)**.** If the graph Laplacian matrix has $\lambda_2(\boldsymbol{L}) > 0$, then $\zeta \leq 1/\lambda_2(\boldsymbol{L})^{\frac{k+1}{2}}$ for odd $k$, and $\zeta \leq \sqrt{2}/\lambda_2(\boldsymbol{L})^{\frac{k}{2}+1}$ for even $k$.

#### 4.3.1 Error Rates of the Global Minimizers

We first bound the error rates of the global minimizer of the generalized GTF estimators (5.3) and (4.9), whose proof is given in Appendix B.1.1.

**Theorem 21** (Error bounds of the GTF minimizer)**.** Assume $\mu < \frac{1}{\|\boldsymbol{\Delta}^{(k+1)}\|^2}$. Fix $\delta \in (0,1)$. For scalar-GTF (5.3), let $\bar{\boldsymbol{\beta}}$ to be its minimizer. Set $\lambda = \sigma\zeta\sqrt{2\log(\frac{er}{\delta})}$, then

$$\frac{\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} \leq \frac{4g(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)}{n(1-\mu\|\boldsymbol{\Delta}^{(k+1)}\|^2)} \tag{4.11}$$

$$+ \frac{\sigma^2}{n}\Big(C_G + 2\sqrt{2C_G\log\left(\frac{1}{\delta}\right)}\Big) \tag{4.12}$$

with probability at least $1-2\delta$. Similarly, for vector-GTF (4.9), let $\overline{\boldsymbol{B}}$ to be its minimizer. Set $\lambda = \sigma\zeta\sqrt{2d\log(\frac{edr}{\delta})}$, then

$$\frac{\|\overline{\boldsymbol{B}} - \boldsymbol{B}^\star\|_{\mathrm{F}}^2}{dn} \leq \frac{4h(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}^\star)}{dn(1-\mu\|\boldsymbol{\Delta}^{(k+1)}\|^2)} \tag{4.13}$$

$$+ \frac{\sigma^2}{n}\Big(C_G + 2\sqrt{2C_G\log\left(\frac{d}{\delta}\right)}\Big) \tag{4.14}$$

with probability at least $1-2\delta$.

Moreover, $\lambda_2(\boldsymbol{L}) \geq \frac{4}{nD}$, where $D$ is the diameter of the graph. Consequently, we get faster rates when the graph is well-connected and has a small diameter. When $\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 = O(1)$ and the graph is fully connected such that $C_G = 1$, the estimate $\bar{\boldsymbol{\beta}}$ converges in probability with respect to its average squared error at the rate $\zeta\sqrt{\log r}/n$. It is shown in[30] that $\zeta \leq \frac{1}{\lambda_2(\boldsymbol{L})^{(k+1)/2}}$, where $\lambda_2(\boldsymbol{L})$ is the smallest non-zero eigenvalue of the graph Laplacian matrix $\boldsymbol{L} = \boldsymbol{\Delta}^{(1)T}\boldsymbol{\Delta}^{(1)}$ and quantifies the algebraic connectivity of the graph[155]. Moreover, one can bound $\lambda_2(\boldsymbol{L}) \geq \frac{4}{nD}$,

where $D$ is the diameter of the graph. Consequently, we get faster rates when the graph is well-connected and has a small diameter.

**Proposition 2** (Bound on $\zeta$)**.** If the graph Laplacian matrix has $\lambda_2(\boldsymbol{L}) > 0$, then $\zeta \leq 1/\lambda_2(\boldsymbol{L})^{\frac{k+1}{2}}$ for odd $k$, and $\zeta \leq \sqrt{2}/\lambda_2(\boldsymbol{L})^{\frac{k}{2}+1}$ for even $k$.

Using Proposition 2, we can further specialize the rates in Theorem 21 for some representative graphs to gain further insights.

- **Chain graph:** For univariate trend filtering,

$$\frac{\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} = O\left(\sqrt{\frac{\log n}{n}} n^k \left\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\right\|_1\right).$$

- **$d$-regular graphs and Erdős-Rényi random graphs:** For $d$-regular graphs as well as Erdős-Rényi random graphs with edge probability $p \in (0,1)$ such that $d = np$,

$$\frac{\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} = O\left(\frac{\sqrt{\log(nd)}}{nd^{\frac{k+1}{2}}} \left\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\right\|_1\right).$$

*Proof.* Proof and further discussion deferred to Appendix. □

**Theorem 22** (Error bound of extended GTF minimizer)**.** Define $\overline{\boldsymbol{B}}$ to be the minimizer of the extended GTF loss function in (4.9), and $\boldsymbol{B}^\star$ as defined in (4.7). Fix $\delta \in (0,1)$. Then setting $\lambda = \sigma\zeta\sqrt{2\log(\frac{er}{\delta})}$, and for a penalty function $\rho(\cdot;\lambda,\gamma)$ such that $\mu < \frac{1}{\|\boldsymbol{\Delta}^{(k+1)}\|_2^2}$,

$$\frac{\|\overline{\boldsymbol{B}} - \boldsymbol{B}^\star\|_{\mathrm{F}}^2}{n} \leq O\left(\frac{C}{n}\right) + \frac{4\lambda^2 r(\sqrt{d}-1)^2}{(1-\mu\|\mathcal{D}\|_2^2)^2 n} + \frac{4h_\lambda(\mathcal{D}\boldsymbol{B}^\star)}{(1-\mu\|\mathcal{D}\|_2^2)n} \tag{4.15}$$

with at least probability $1-2\delta$.

### 4.3.2 Error Rates of First-order Stationary Points

Due to non-convexity, global minima of the proposed GTF estimators may not be attainable. Therefore, it is more desirable to understand the statistical performance of any first-order stationary points of the GTF estimators by considering oracle inequalities. We call $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^n$ a stationary point of $f(\boldsymbol{\beta})$, if it satisfies

$$0 \in \nabla_{\boldsymbol{\beta}} f(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}}.$$

We further introduce the compatibility factor, which generalizes the notion used in [133] to allow vector-valued signals.

**Definition 14** (Compatibility factor)**.** Let $\boldsymbol{\Delta}^{(k+1)}$ be fixed. The compatibility factor $\kappa_{T,d}$ of a set $T \subseteq \{1,2,\ldots,r\}$ is defined as $\kappa_{\varnothing,d} = 1$, and for nonempty set $T$,

$$\kappa_{T,d}(\boldsymbol{\Delta}^{(k+1)}) = \inf_{\boldsymbol{B}\in\mathbb{R}^{n\times d}} \left\{\frac{\sqrt{|T|}\cdot\|\boldsymbol{B}\|_{\mathrm{F}}}{\sum_{\ell\in T}\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_{\ell\cdot}\|_2}\right\}.$$

To further build intuition, consider $\sqrt{|T|}\kappa_{T,1}(\boldsymbol{\Delta})^{-1} = \sup_{\boldsymbol{\beta}\in\mathbb{R}}\{\|(\boldsymbol{\Delta})_T\boldsymbol{\beta}\|_1/\|\boldsymbol{\beta}\|_2\}$. This is precisely the definition of $\|(\boldsymbol{\Delta})_T\|_{1,2}$, an induced norm of the $|T| \times n$ submatrix of $\boldsymbol{\Delta}$. If we consider signals with fixed power $\|\boldsymbol{\beta}\|_2^2 = 1$, $\|(\boldsymbol{\Delta})_T\boldsymbol{\beta}\|_1$ will depend on how much the $T$ edges are connected to each other. Together with $\|(\boldsymbol{\Delta})_T\boldsymbol{\beta}\|_1 \leq \sqrt{|T|}\|(\boldsymbol{\Delta})_T\boldsymbol{\beta}\|_2$, $\kappa_{T,1}(\boldsymbol{\Delta})$ can be related to the restricted eigenvalue condition, which is often used to bound the performance of LASSO[137]. With slight abuse of notation, we write $\kappa_T := \kappa_{T,d}$.

We have the following oracle inequality that is applicable to the stationary points of the GTF estimators, whose proof is given in Appendix B.1.3. The proof follows a construction that is similar to Theorem 2 in[133]. The oracle inequality holds for any $\widehat{\boldsymbol{\beta}}$ that satisfies the first order optimality condition. This mild condition on $\widehat{\boldsymbol{\beta}}$ that they are stationary points allows us to use our ADMM-based algorithm and non-convex penalties. This is a key difference with[30] Theorem 3 and[156] Theorem 1 which holds for global minima which we cannot always guarantee when using non-convex penalties.

We stress that although GTF was motivated by piecewise smooth graph signals, Theorem 23 holds for any graph $G$ and graph signal $\boldsymbol{\beta}^\star$.

**Theorem 23** (Oracle inequality of GTF stationary points)**.** Assume $\mu < 1/\|\boldsymbol{\Delta}^{(k+1)}\|^2$. Fix $\delta \in (0,1)$. For scalar-GTF (5.3), let $\widehat{\boldsymbol{\beta}}$ be a stationary point. Set $\lambda = \sigma\zeta_k\sqrt{2\log\left(\frac{er}{\delta}\right)}$, then

$$\frac{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} \leq \inf_{\boldsymbol{\beta}\in\mathbb{R}^n}\left\{\frac{\|\boldsymbol{\beta} - \boldsymbol{\beta}^\star\|_2^2 + 4g((\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta})_{T^c})}{n}\right\} + \frac{2\sigma^2\left[C_G + 2\sqrt{2C_G\log(\frac{1}{\delta})} + \frac{8\zeta_k^2|T|}{\kappa_T^2}\log(\frac{er}{\delta})\right]}{n(1-\mu\|\boldsymbol{\Delta}^{(k+1)}\|^2)} \tag{4.16}$$

with probability at least $1-2\delta$ for any $T \subseteq \{1,2,...,r\}$. Similarly, for vector-GTF (4.9), let $\widehat{\boldsymbol{B}}$ be a stationary point. Set $\lambda = \sigma\zeta_k\sqrt{2d\log(\frac{edr}{\delta})}$, then

$$\frac{\|\widehat{\boldsymbol{B}} - \boldsymbol{B}^\star\|_{\mathrm{F}}^2}{dn} \leq \inf_{\boldsymbol{B}\in\mathbb{R}^{n\times d}}\left\{\frac{\|\boldsymbol{B} - \boldsymbol{B}^\star\|_{\mathrm{F}}^2 + 4h((\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_{T^c})}{dn}\right\} + \frac{2\sigma^2\left[C_G + 2\sqrt{2C_G\log(\frac{d}{\delta})} + \frac{8\zeta_k^2|T|}{\kappa_T^2}\log(\frac{edr}{\delta})\right]}{n(1-\mu\|\boldsymbol{\Delta}^{(k+1)}\|^2)} \tag{4.17}$$

with probability at least $1-2\delta$ for any $T \subseteq \{1,2,...,r\}$.

**Remark 4.** Recall that $\mu$ is defined in Assumption 1 (c), which characterizes how "non-convex" the regularizer is, and dictates the inflection point in Fig. 4.2. The assumption $\mu < 1/\|\boldsymbol{\Delta}^{(k+1)}\|^2$ in Theorem 23 therefore implicitly constrains the level of non-convexity of the regularizer. Take MCP in (B.2) for example: since $\mu \geq 1/\gamma$, we can guarantee the existence of a valid $\mu$ such that $\mu < 1/\|\boldsymbol{\Delta}^{(k+1)}\|^2$ as long as we set $\gamma > \|\boldsymbol{\Delta}^{(k+1)}\|^2$.

Theorem 23 allows one to select $\boldsymbol{\beta}$ and $T$ to optimize the error bounds on the right hand side of

(4.16) and (4.17). For example, pick $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$ in (4.16) (hence an "oracle") to have

$$\frac{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} \le \frac{4g((\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T^c})}{n} + \frac{2\sigma^2\left[C_G^\delta + 8\zeta_k^2\kappa_T^{-2}|T|\log(\frac{er}{\delta})\right]}{n(1-\mu\|\boldsymbol{\Delta}^{(k+1)}\|^2)}, \tag{4.18}$$

where $C_G^\delta = C_G + 2\sqrt{2C_G\log(\frac{1}{\delta})}$.

- By setting $T$ as an empty set, we have

$$\frac{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} \le \frac{4g(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)}{n} + \frac{2\sigma^2 C_G^\delta}{n(1-\mu\|\boldsymbol{\Delta}^{(k+1)}\|^2)}, \tag{4.19}$$

  which suggest that the reconstruction accuracy improves when the ground truth $\boldsymbol{\beta}^\star$ is better aligned with the graph structure, and consequently the value of $g(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)$ is small.

- On the other hand, by setting $T$ as the support of $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star$, we achieve

$$\frac{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} \le \frac{2\sigma^2\left[C_G^\delta + 8\zeta_k^2\kappa_T^{-2}\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_0\log(\frac{er}{\delta})\right]}{n(1-\mu\|\boldsymbol{\Delta}^{(k+1)}\|^2)},$$

  which grows linearly as we increase the sparsity level $\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_0$.

Similar discussions can be conducted for vector-GTF by choosing $\boldsymbol{B} = \boldsymbol{B}^\star$ in (4.17). More importantly, we can directly compare the performance of vector-GTF with scalar-GTF, which was formulated for vector-valued graph signals in (4.8). The error bound of vector-GTF pays a small price in the order of $\log d$, but is tighter than scalar-GTF if $h((\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}^\star)_{T^c}) \ll \sum_{j=1}^d g((\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}^\star_{\cdot j})_{T^c})$. This suggests that vector-GTF is much more advantageous when the support sets of $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}^\star_{\cdot j}$ for $j = 1, \ldots, d$ overlap, i.e. when the local discontinuities and patterns in $\boldsymbol{B}^\star_{\cdot j}$ are shared.

### 4.3.3 Comparison with Scalar-GTF using $\ell_1$ Regularization

We compare our error bound for scalar-GTF, that is, on $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2/n$, with[30] Theorem 3, which is obtained for GTF with the $\ell_1$ penalty, reproduced below for convenience.

**Theorem 24** (Basic error bound of $\ell_1$ GTF minimizer)**.** If $\lambda = \Theta(\sigma\zeta_k\sqrt{\log r})$, then $\widehat{\boldsymbol{\beta}}$, the minimizer of (4.3), satisfies

$$\frac{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} = O\left(\frac{\lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1}{n} + \frac{\sigma^2 C_G}{n}\right).$$

The above bound is comparable to our bound in the special case of setting $T$ to an empty set, i.e. (4.19). The first term of the bound in (4.19) is upper bounded by that of Theorem 24. The non-convex regularization yields especially tighter bounds when $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star$ contains large coefficients, so that $g(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star) \ll \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1$. On the other hand, the second term of (4.19) contains $1 - \mu\|\boldsymbol{\Delta}^{(k+1)}\|^2$ in the denominator, which makes it an upper bound of the second term in Theorem 24. This gap can be brought down by choosing a larger $\gamma$, which allows one to pick a smaller $\mu$, as mentioned in Remark 4. However, as $\gamma \to \infty$, non-convex SCAD and MCP also

tends to $\ell_1$, which erases the improvement from using non-convex regularizers in the first term of the bound. This indicates a trade-off in the overall error bound based on $\gamma$, or the "non-convexity" of the regularizers chosen for scalar-GTF.

To sum up, despite being non-convex, we can guarantee that any stationary point of the proposed GTF estimator possesses strong statistical guarantees.

### 4.3.4 Error Rates for Erdős-Rényi Graphs

We next specialize Theorem 23 to the Erdős-Rényi random graphs using spectral graph theory[157]. Let $d_{\max}$ and $d_0$ respectively be the maximum and expected degree of the graph. It is known that for any graph it holds[30]

$$\zeta_k \leq \lambda_{\min}(\boldsymbol{\Delta}^{(2)})^{-\frac{k+1}{2}}, \tag{4.20}$$

where $\lambda_{\min}(\boldsymbol{\Delta}^{(2)})$ is the smallest *non-zero* eigenvalue of the graph Laplacian matrix $\boldsymbol{\Delta}^{(2)}$. Moreover, we have $\|\boldsymbol{\Delta}^{(k+1)}\|^2 = (\lambda_{\max}(\boldsymbol{\Delta}^{(2)}))^{k+1}$, and $d_{\max}+1 \leq \lambda_{\max}(\boldsymbol{\Delta}^{(2)}) \leq 2d_{\max}$[155]. Next, we present a simple lower bound on $\kappa_T$, which is proved in Appendix B.1.4.

**Proposition 3** (Bound on $\kappa_T$)**.** *$\kappa_T$ is bounded for any $T$ and $d$ as*

$$\kappa_T(\boldsymbol{\Delta}^{(k+1)}) \geq (2d_{\max})^{-\frac{k+1}{2}}.$$

[1].

**Chain graph (Univariate trend filtering):** for a chain graph, we have $d_{\max} = 2$, and $\|\boldsymbol{\Delta}^{(k+1)}\|^2 = O(1)$, $r = O(n)$, and $\zeta = O(n^{k+1/2})$[30]. Therefore, with probability at least $1-2n^{-10}$,

$$\frac{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} \lesssim \frac{\sigma^2 \log n}{n} + \|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_0 \cdot n^{2k} \log n.$$

For an Erdős-Rényi random graph, if $d_0 = \Omega(\log(n))$, we have $d_{\max} = O(d_0)$ almost surely[158] Corollary 8.2 and $C_G = 1$. Furthermore, $\lambda_{\min}(\boldsymbol{\Delta}^{(2)}) = \Omega(d_0 - \sqrt{d_0})$[30,157,159], and $r = n$ for odd $k$ and $r = O(nd_0)$ for even $k$. Therefore, with probability at least $1-n^{-10}$, we have

$$\frac{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2}{n} \lesssim \frac{\sigma^2\sqrt{\log n}}{n} + \min\left\{\frac{g(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)}{n}, \frac{\sigma^2\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_0 \log n}{n}\right\},$$

where $g(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star) \lesssim \frac{\sigma\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1\sqrt{\log n}}{d_0^{(k+1)/2}}$ by plugging in $g(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star) \leq \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1$.

These results are also applicable to $d_0$-regular Ramanujan graphs[159].

### 4.3.5 Support Recovery

An alternative yet important metric for gauging the success of the proposed GTF estimators is support recovery, which aims to localize the discontinuities in the piecewise smooth graph signals,

[1] For $k > 2$, the piecewise $k$th order polynomial signals are indistinguishable from smooth signals for large $k$, we can safely assume $k \leq 2$ for most use cases.

i.e. the support set of $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star$, that is

$$S_k(\boldsymbol{\beta}^\star) = \left\{ t \in \{1, \cdots, r\} : (\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_t \neq 0 \right\}. \tag{4.21}$$

In particular, for odd $k$, the discontinuities correspond to graph nodes; and for even $k$, they correspond to the edges. Let $\widehat{\boldsymbol{\beta}}$ be the GTF estimate of the graph signal. The quality of the support recovery can be measured using the *graph screening distance*[135]. For any $t_1 \in S_k(\boldsymbol{\beta}^\star)$ and $t_2 \in S_k(\widehat{\boldsymbol{\beta}})$, let $d_G(t_1, t_2)$ denote the length of the shortest path between them. The distance of $S_k(\widehat{\boldsymbol{\beta}})$ from $S_k(\boldsymbol{\beta}^\star)$ is then defined as

$$\begin{aligned} & d_G(S_k(\widehat{\boldsymbol{\beta}}) | S_k(\boldsymbol{\beta}^\star)) \\ &= \begin{cases} \max\limits_{t_1 \in S_k(\boldsymbol{\beta}^\star)} \min\limits_{t_2 \in S_k(\widehat{\boldsymbol{\beta}})} d_G(t_1, t_2), & \text{if } S_k(\boldsymbol{\beta}^\star) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}. \end{aligned} \tag{4.22}$$

Interestingly, Lin et.al.[135] showed recently that under mild assumptions, one can translate the error bound into a support recovery guarantee. We see that the rates on the graph screening distance or how well we can localize the boundary then depend on both the jump height $H_n$ between clusters, and the diameter of the clusters. Specifically, letting $R_n$ be the RHS of (4.16) that bounds the error $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2/n$ in Theorem 23, we have

$$d_G(S_k(\widehat{\boldsymbol{\beta}}) | S_k(\boldsymbol{\beta}^\star)) = \begin{cases} O\left(\frac{R_n}{H_r^2}\right), & k = 0 \\ O\left(\frac{R_n^{1/3}}{H_r^{2/3}}\right), & k = 1 \end{cases}, \tag{4.23}$$

where $H_r$ quantifies the minimum level of discontinuity, defined as the minimum absolute value of the non-zero values of $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star$, i.e.

$$H_r = \min_{t \in S_k(\boldsymbol{\beta}^\star)} |(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_t|. \tag{4.24}$$

Consequently, this leads to support recovery guarantees of the proposed GTF estimators. Numerical experiment in Section 4.5.1 verifies the superior performance of the non-convex regularizers over the $\ell_1$ regularizer for support recovery.

## 4.4 ADMM Algorithm and its Convergence

There are many algorithmic approaches to optimize the vector-GTF formulation in (4.9), since scalar-GTF (5.3) can be regarded as a special case with $d = 1$. In this section, we illustrate the approach adopted in this work, which is the Alternating Direction Method of Multipliers (ADMM) framework for solving separable optimization problems[104].

Via a change of variable as $\boldsymbol{Z} = \boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}$, we can transform (4.9) to

$$\min_{\boldsymbol{B} \in \mathbb{R}^{n \times d}} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{B}\|_{\mathrm{F}}^2 + h(\boldsymbol{Z}; \lambda, \gamma) \quad \text{s.t.} \quad \boldsymbol{Z} = \boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}.$$

**Algorithm 5** ADMM for solving (4.9)

**Inputs:** $\boldsymbol{Y}, \boldsymbol{\Delta}^{(k+1)}$, and parameters $\lambda$, $\gamma$, $\tau$
**Initialize:**
  $\boldsymbol{B} \leftarrow \boldsymbol{Y}$ or $\boldsymbol{B}_{\texttt{init}}$ if given.
  $\mathcal{D} \leftarrow \boldsymbol{\Delta}^{(k+1)}$, $\boldsymbol{Z} \leftarrow \mathcal{D}\boldsymbol{B}$, $\boldsymbol{U} \leftarrow \mathcal{D}\boldsymbol{B} - \boldsymbol{Z}$
  $\boldsymbol{X} \leftarrow (\boldsymbol{I} + \tau\mathcal{D}^\top\mathcal{D})^{-1}$
**repeat**
  **for** $j \leftarrow$ `1 to num_cols(`$\boldsymbol{B}$`)` **do**
    $\boldsymbol{B}_{\cdot j} \leftarrow \boldsymbol{X}(\tau\mathcal{D}^\top(\boldsymbol{Z}_{\cdot j} - \boldsymbol{U}_{\cdot j}) + \boldsymbol{Y}_{\cdot j})$
  **end for**
  **for** $\ell \leftarrow$ `1 to num_rows(`$\mathcal{D}\boldsymbol{B}$`)` **do**
    $\boldsymbol{Z}_{\ell\cdot} \leftarrow \texttt{Prox}_\rho(\|\mathcal{D}_{\ell\cdot}\boldsymbol{B} + \boldsymbol{U}_{\ell\cdot}\|_2; \lambda/\tau)$
  **end for**
  $\boldsymbol{U} \leftarrow \boldsymbol{U} + \mathcal{D}\boldsymbol{B} - \boldsymbol{Z}$
**until** termination

Its corresponding Lagrangian can be written as:

$$\mathcal{L}(\boldsymbol{B}, \boldsymbol{Z}, \boldsymbol{U}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{B}\|_{\mathrm{F}}^2 + h(\boldsymbol{Z}; \lambda, \gamma) + \frac{\tau}{2}\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B} - \boldsymbol{Z} + \boldsymbol{U}\|_{\mathrm{F}}^2 - \frac{\tau}{2}\|\boldsymbol{U}\|_{\mathrm{F}}^2, \tag{4.25}$$

where $\boldsymbol{U} \in \mathbb{R}^{r\times d}$ is the Lagrangian multiplier, and $\tau$ is the parameter. Alg. 5 shows the ADMM updates based on the Lagrangian in (4.25). Recall the proximal operator is defined as $\texttt{Prox}_f(\boldsymbol{v}; \alpha) = \operatorname{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \boldsymbol{v}\|_2^2 + \alpha f(\mathbf{x})$ for a function $f(\cdot)$. $\ell_1$, SCAD and MCP all admit closed-form solutions of `Prox`, which are simple thresholding operations[160]. Furthermore, we have the following convergence guarantee for Alg. 5, whose proof is provided in Appendix B.1.5.

**Theorem 25.** Let $\tau \geq \mu$, then Alg. 5 converges to a stationary point of (4.9).

In addition, we provide a detailed time complexity analysis of Alg. 5 in Table 4.2. Note that since $\boldsymbol{\Delta}$ is a sparse matrix with exactly $2m$ non-zero entries, Alg. 5 can run much faster when $k = 0$. As a preprocessing step for each $\mathcal{D}$, we compute $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{S} \in \mathbb{R}^{n\times n}$, the eigenvectors and eigenvalues of $\mathcal{D}^\top\mathcal{D}$, exactly once. $\boldsymbol{X} = \boldsymbol{V}(1 + \rho\boldsymbol{S})^{-1}\boldsymbol{V}^\top$ can then be initialized very efficiently for all experiments that use $\mathcal{D}$.

| | $k \geq 1$ | $k = 0$ |
|---|---|---|
| $\mathcal{D}^\top\mathcal{D}$ eigen decomposition | $O(rn^2 + n^3)$ | $O(m^2 + n^3)$ |
| $\boldsymbol{Z}$ initialization | $O(rnd)$ | $O(md)$ |
| $\boldsymbol{X}$ initialization | $O(n^2)$ | $O(n^2)$ |
| $\boldsymbol{B}$ update | $O(d(nr + n^2))$ | $O(d(m + n^2))$ |
| $\mathcal{D}\boldsymbol{B}$ calculation | $O(rnd)$ | $O(md)$ |
| $\boldsymbol{Z}$,$\mathcal{U}$ update | $O(rd)$ | $O(rd)$ |
| Total after $t$ iterations | $O(tdrn + tdn^2)$ | $O(tdm + tdn^2)$ |

*Table 4.2: Time complexity analysis of Alg. 5.*

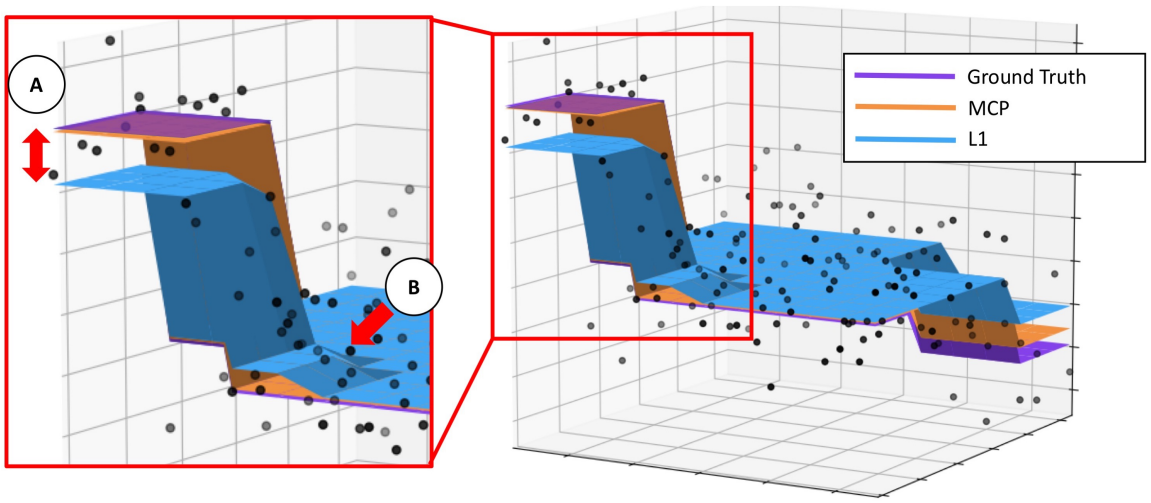



*Figure 4.3: Scalar-GTF with MCP (orange) has much lower bias than scalar-GTF with $\ell_1$ (blue) when estimating a piecewise constant signal over a 12 × 12 grid graph. See highlighted regions pointed by red arrows in A and B. The scatter points correspond to a noisy signal with 5dB SNR.*

## 4.5 Numerical Experiments

For the following experiments, we fixed $\gamma = 3.7$ for SCAD, $\gamma = 1.4$ for MCP. Further, the graphs we use in the following experiments satisfy Assumption 2 for this choice of $\gamma$. Unless explicitly mentioned, we tuned $\lambda$ and $\frac{\tau}{\lambda}$ for each experiment using the Hyperopt toolbox[161]. To meet the convergence criteria in Theorem 25, we enforced $\tau \geq 1/\gamma$. SCAD/MCP were warm-started with the GTF estimate with $\ell_1$ penalty. Python packages PyGSP[162,71] and NetworkX[163] were used to construct and plot graphs. The input signal SNR was calculated as $10\log_{10}(\|\boldsymbol{B}^\star\|_{\mathrm{F}}/\sigma^2 nd)$, while the reconstructed signal SNR was calculated as $10\log_{10}(\|\boldsymbol{B}^\star\|_{\mathrm{F}}/\|\widehat{\boldsymbol{B}} - \boldsymbol{B}^\star\|_{\mathrm{F}})$, where $\widehat{\boldsymbol{B}}$ was the reconstruction. Computation time was measured with MacBook Pro 2017 with an 2.9 GHz Intel Core i7 and 16GB RAM. Our code is available at https://github.com/HarlinLee/nonconvex-GTF-public.

### 4.5.1 Denoising via GTF with Non-convex Regularizers

We first highlight via synthetic examples two important advantages that non-convex regularizers provide over the $\ell_1$ penalty.

- **Bias Reduction:** We demonstrate the reduction in signal bias in Fig. 4.3 for the graph signal defined over a 12 × 12 2D-grid graph, using both the $\ell_1$ penalty and the MCP penalty. Clearly, the MCP estimate (orange) has less bias than the $\ell_1$ estimate (blue), and can recover the ground truth surface (purple) more closely.
- **Support Recovery:** We illustrate the improved support recovery performance of non-convex penalties on localizing the boundaries for a piecewise constant signal on the Minnesota road graph, shown in Fig. 4.5. Particularly, we look at how well our estimator localizes the support of $\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star$, that is, the discontinuity of the piecewise constant graph signal by looking at how well we can classify an edge as connecting two nodes in the same piece or being a cut edge across two pieces. By sweeping the regularization parameter $\lambda$, we obtain the ROC curve in Fig. 4.4, i.e. the true positive rate versus the false positive rate of

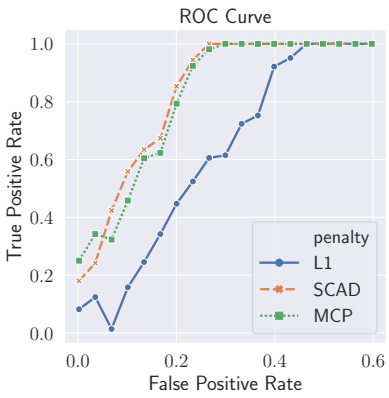



*Figure 4.4: The ROC curve for classifying whether an edge lies on a boundary for the Minnesota road graph signal shown in Fig. 4.5. The input SNR of the noisy piecewise constant signal is* $7.8dB$.

classifying a cut edge correctly, and see that scalar-GTF with MCP and SCAD consistently outperforms the scalar-GTF with $\ell_1$ penalty.

Then, we compare the performance of GTF using non-convex regularizers such as SCAD and MCP with that using the $\ell_1$ norm more rigorously. For the ground truth signal $\boldsymbol{\beta}^\star$, we construct a piecewise constant signal on a $20 \times 20$ 2D-grid graph and the Minnesota road graph [162] as shown in the left panel of Fig. 4.5, and add different levels of noise following (4.1). We recover the signal by scalar-GTF with Alg. 5, and plot the SNR of the reconstructed signal versus the SNR of the input signal *in solid lines* in the middle panel of Fig. 4.5, averaged over 10 and 20 realizations, respectively. SCAD/MCP consistently outperforms $\ell_1$ in denoising graph signals defined over both regular and irregular structures.

### 4.5.2 Denoising Vector-valued Signals via GTF

We compare the performance of vector-GTF in (4.9) with (4.8), which applies scalar-GTF to each column of the vector-valued graph signal. The convex $\ell_1$ norm, and the non-convex SCAD and MCP are employed. We reuse the same ground truth graph signals over the 2D-grid graph and the Minnesota road graph constructed in Section 4.5.1 in Fig. 4.5. $d$ independent noisy realizations of the graph signal are concatenated to construct a noisy vector-valued graph signal with dimension $d = 10$ on the 2D-grid graph and with $d = 20$ on the Minnesota road graph. We recover the vector-valued signal by minimizing vector-GTF (4.9) with Alg. 5.

The middle panel of Fig. 4.5 plots the average SNR of the reconstructed signal versus the average SNR of the input signal *in dotted lines*. We emphasize that the performance of (4.8) is the same as applying scalar-GTF to each realization, which is shown in the middle panel of Fig. 4.5 in solid lines. As before, SCAD/MCP consistently outperforms $\ell_1$ in denoising signals over both regular and irregular graphs. Furthermore, as expected, due to the sharing of information across realizations, vector-GTF consistently outperforms scalar-GTF, especially in the low SNR regime.

The right panel of Fig. 4.5 plots the computation time versus the gain in SNR from denoising via vector-GTF. 10 trials are performed for each regularizer with the input signal SNR fixed at 20dB. Parameter tuning and eigen decomposition of $\boldsymbol{\Delta}^{(2)}$ are preprocessing steps, and hence they

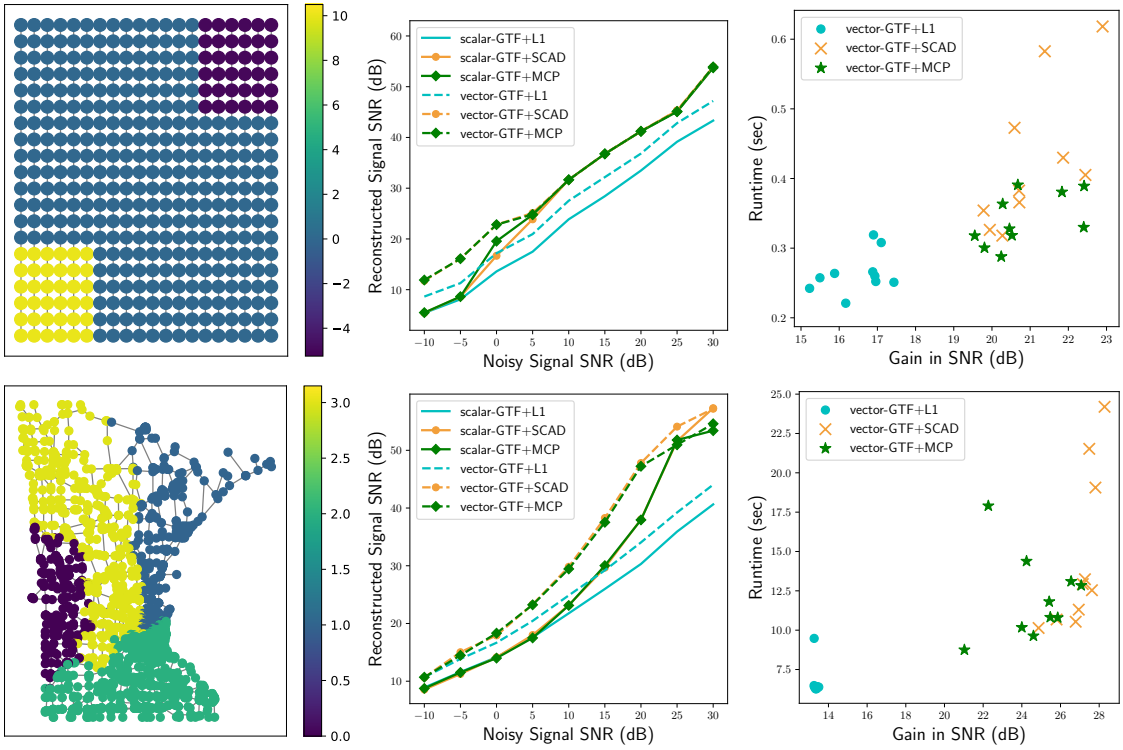



*Figure 4.5: The left panel shows the ground truth piecewise constant signals on* $20 \times 20$ *2D-grid graph (top), and Minnesota road graph (bottom). The middle panel shows their corresponding plots of input signal SNR versus reconstructed signal SNR, averaged over 10 and 20 realizations, respectively. Finally, the right panel plots the computation time against gain in SNR from denoising via vector-GTF. 10 trials were performed for each regularizer, where the input signal SNR was fixed at 20dB.*

are not included in the time measurement; but for reference, the eigen decomposition took 0.025 and 2.5 seconds for 2D-grid and Minnesota graphs, respectively. Since GTF with non-convex regularizers are warm-started by the $\ell_1$ estimate, the runtime for $\ell_1$ GTF is added to the SCAD/MCP runtime. Overall, running vector-GTF with SCAD/MCP after once with $\ell_1$ takes more time, but with large benefits in the denoising performance. Even with the additional computation time, Vector-GTF runs reasonably fast; with the Minnesota road network, where $n = 2642$ and $m = 3304$, computation takes less than 25 seconds.

We further investigate the benefit of sharing information across measurements or realizations in the following experiment, using the same ground truth signal on the 2D-grid graph. We stack eight noisy realizations of this same piecewise constant signal to build a vector-valued signal. We construct these noisy measurements by scaling each one of them differently and randomly such that each will have SNR $\sim \log_{10} \text{Uniform}[-10, 30]$dB under (4.1). This has the effect of rendering some measurements more *informative* than others, and potentially allowing vector-GTF to reap the benefits of sharing information across measurements. We recover the 8-dimensional graph signal via Alg. 5 using $\ell_1$, SCAD, and MCP regularizers, and in Table 4.3, report the input signal and reconstructed signal SNRs for each measurement in addition to the average SNRs. $\lambda$ is fixed at $0.5\sigma^2$.

| | Average | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 |
|---|---|---|---|---|---|---|---|---|---|
| Input SNR (dB) | 8.7 | -14 | 0 | 0 | 3.5 | 5.8 | 12 | 29 | 34 |
| Vector-GTF + $\ell_1$ | 29 | **10** | **20** | **23** | **26** | **36** | **37** | 39 | 38 |
| Scalar-GTF + $\ell_1$ | 21 | 0 | 11 | 13 | 16 | 18 | 26 | 41 | 45 |
| Vector-GTF + SCAD | **32** | **10** | **20** | 22 | 25 | **36** | 35 | **49** | **61** |
| Scalar-GTF + SCAD | 29 | 0 | 15 | 17 | 25 | 35 | 34 | 47 | 60 |
| Vector-GTF + MCP | **32** | **10** | **20** | 22 | 25 | **36** | 35 | **49** | **61** |
| Scalar-GTF + MCP | 29 | 0 | 15 | 22 | 24 | 30 | 33 | **49** | 60 |

*Table 4.3: Noisy input and reconstructed signal SNRs for eight measurements of varying input SNRs, rounded to two significant figures. Highest reconstructed signal SNR for each measurement is in **bold**.*

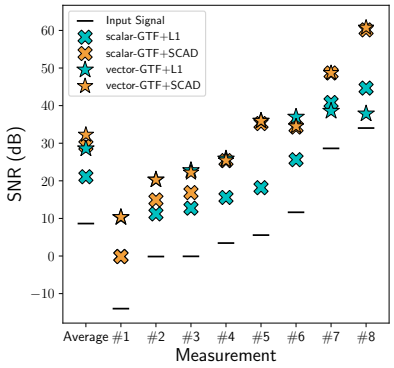



*Figure 4.6: Noisy input and reconstructed signal SNRs for each snapshot of a piecewise constant signal on a 20 × 20 2D-grid graph, as shown in Fig. 4.5. Stars show reconstructed signal SNRs from vector-GTF, while crosses are from scalar-GTF. MCP (not shown) performed similarly to SCAD.*

First of all, notice that as before, using SCAD/MCP generally achieves results with higher SNR than using $\ell_1$, and that on average, minimizing (4.9) outperforms minimizing (4.8). The effect of sharing information across measurements is most apparent in low SNR settings, when information about the boundaries of the graph signal can be borrowed from higher SNR signals to improve the estimation. On the other hand, sharing information with noisier signals does not help denoising signals with high input SNR. However, it is worth noting that, unlike $\ell_1$, SCAD/MCP does not see decrease in its performance in the high SNR settings.

### 4.5.3 Event Detection with NYC Taxi Data

To further illustrate graph trend filtering on a real-world dataset, we consider the road network of Manhattan where the nodes correspond to junctions[164]. We map the pickups and dropoffs of the NYC taxi trip dataset to the nearest road junctions, and define the total count at that junction to be the signal value on the corresponding graph node. The signal of interest, plotted on the top left panel of Fig. 4.7, is the difference between the *event* graph signal on the day of NYC Gay Pride parade, 12-2pm on June 26, 2011, and the *seasonal average* graph signal at the same time during the 8 nearest Sundays. During the event, no pickups and dropoffs could occur in the areas

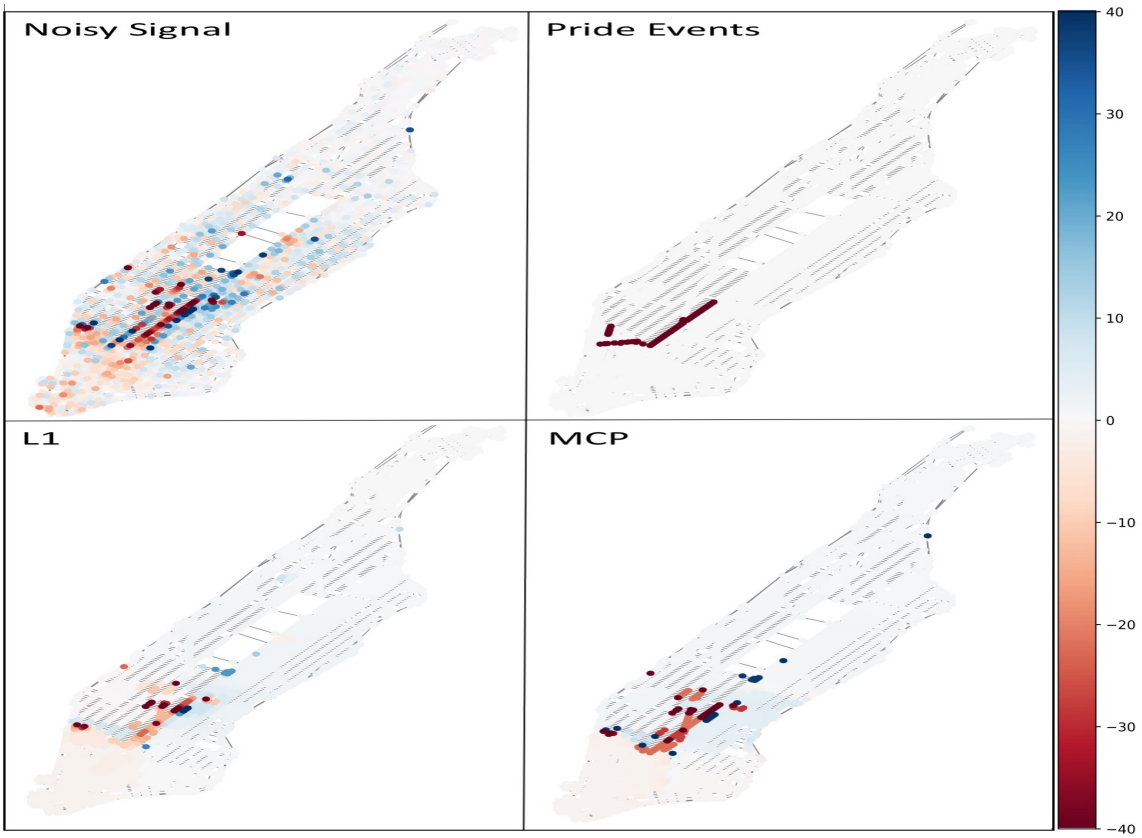



*Figure 4.7: Top left: the noisy signal on the Manhattan road network is the change in the taxi pickup and dropoff count during the 2011 NYC Gay Pride. Top right: areas of Pride events, where the traffic was blocked off. Bottom: the GTF estimates using $\ell_1$ and MCP. The GTF estimate with MCP better detects and localizes the event, compared to the one using $\ell_1$ penalty.*

shown in the top right panel of Fig. 4.7 . We denoise the signal via GTF using both $\ell_1$ and MCP, where we chose $\lambda$ such that $\|\boldsymbol{\Delta}\widehat{\boldsymbol{\beta}}\|_0 \approx 200$. Once again, we observe the GTF estimate with MCP produces sharper traces around the parade route, indicating better capabilities of event detection and localization.

### 4.5.4 Semi-supervised Classification

Graph-based learning provides a flexible and attractive way to model data in semi-supervised classification problems when vast amounts of unlabeled data are available compared to labeled data, and labels are expensive to acquire[112,113,149]. One can construct a nearest-neighbor graph based on the similarities between each pair of samples, and hope to propagate the label information from labeled samples to unlabeled ones. We move beyond our original problem in (5.3) to a $K$-class classification problem in a semi-supervised learning setting, where for a given dataset with $n$ samples, we observe a subset of the one-hot encoded class labels, $\boldsymbol{Y} \in \mathbb{R}^{n\times K}$, such that $Y_{ij} = 1$ if $i$th sample has been observed to be in $j$th class, and $Y_{ij} = 0$ otherwise. A diagonal indicator matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ denotes samples whose class labels have been observed. Then, we can define the modified absorption problem[113,30,149] using a variation of GTF to estimate the unknown class probabil-

| | | Heart | Wine quality | Wine | Iris | Breast | Car |
|---|---|---|---|---|---|---|---|
| # of samples ($n$) | | 303 | 1599 | 178 | 150 | 569 | 1728 |
| # of classes ($K$) | | 2 | 6 | 3 | 3 | 2 | 4 |
| $k=0$ | $\ell_1$ | 0.148 | 0.346 | 0.038 | 0.036 | 0.042 | 0.172 |
| | SCAD | 0.148 | *0.353* | 0.038 | 0.033 | 0.042 | **0.149** |
| | p-value | 1. | *0.06* | 1. | 0.27 | 1. | **0.06** |
| | MCP | 0.144 | 0.351 | 0.037 | 0.035 | 0.040 | **0.148** |
| | p-value | 0.23 | 0.18 | 0.34 | 0.34 | 0.35 | **0.05** |
| $k=1$ | $\ell_1$ | 0.143 | 0.351 | 0.034 | 0.039 | 0.035 | 0.104 |
| | SCAD | 0.144 | 0.350 | 0.034 | 0.039 | 0.035 | 0.104 |
| | p-value | 0.30 | 0.43 | 0.34 | 1. | 0.71 | 0.66 |
| | MCP | *0.146* | 0.350 | 0.034 | 0.039 | **0.034** | 0.103 |
| | p-value | *0.05* | 0.44 | 0.34 | 1. | **0.02** | 0.23 |

*Table 4.4: Misclassification rates averaged over 10 trials, with p-values from running sampled t-tests between SCAD/MCP misclassification rates and the corresponding rates using $\ell_1$. Cases where non-convex penalties perform better than $\ell_1$ with p-value below 0.1 are highlighted in **bold**, and where they perform worse are in italic.*

ities $\boldsymbol{B} \in \mathbb{R}^{n\times K}$:

$$\widetilde{\boldsymbol{B}} = \operatorname{argmin}_{\boldsymbol{B}\in\mathbb{R}^{n\times K}} \frac{1}{2}\|\boldsymbol{M}(\boldsymbol{Y}-\boldsymbol{B})\|_{\mathrm{F}}^2 + \sum_{j=1}^{K} g(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}_{\cdot j};\lambda,\gamma) + \epsilon\|\boldsymbol{R}-\boldsymbol{B}\|_{\mathrm{F}}^2, \tag{4.26}$$

where $\boldsymbol{R} \in \mathbb{R}^{n\times K}$ (set to be uniform in the experiment) is a fixed prior belief, and $\epsilon > 0$ determines how much emphasis to be given to the prior belief. The labels $\widetilde{\boldsymbol{Y}}$ can be estimated using $\widetilde{\boldsymbol{B}}$ such that $\widetilde{Y}_{ij} = 1$ if and only if $j = \arg\max_{1\le\ell\le K} \widetilde{B}_{i\ell}$, and otherwise $\widetilde{Y}_{ij} = 0$. Note that this can be completely separated into $K$ scalar-GTF problems, one corresponding to each class.

We applied the algorithm in (4.26) to 6 popular UCI classification datasets[165] with $\epsilon = 0.01$. For each dataset, we normalized each feature to have zero mean and unit variance, and constructed a 5-nearest-neighbor graph of the samples based on the Euclidean distance between their features, with edge weights from the Gaussian radial basis kernel. We observed the labels of 20% of samples in each class randomly. Table 4.4 shows the misclassification rates averaged over 10 repetitions, which demonstrates that the performance using non-convex penalties such as SCAD/MCP are at least competitive with, and often better than, those with the $\ell_1$ penalty.

## 4.6 Wavelets and Multiresolution Analysis on Graphs

Multiresolution analysis and splines are standard representation tools for piecewise smooth signals. A multiresolution analysis represents and analyzes signals at different resolution scales by recursively decomposing a signal into *coarse* and *detail* subspaces.[166,94,167,168] This is particularly interesting on graphs with respect to piecewise smooth signals that possess localized behavior. Multiresolution analysis is usually formulated axiomatically as a sequence of nested subspaces $V_0 \supset V_1 \supset V_2 \cdots \supset V_L$ of increasing smoothness by repeatedly splitting each $V_j$ into a smoother part $V_{j+1}$ and a rougher part $W_{j+1}$. Typically, as $j$ increases, the the elements in $V_j$ are increasingly less localized.

### 4.6.1 Local-set-based Representations

In this section, we present the local-set-based representations and show the advantages to represent the class of piecewise-constant graph signals.

#### Basics of Local Sets

Local sets are used in the previous works on graph cuts and graph signal reconstruction[169,170]. Here we use the same idea to decompose a graph structure.

**Definition 15.** Let the node set $\mathcal{V}$ divide into a series of node sets $\{S_i\}_{i=1}^{C}$. We call them *local sets* when they satisfy

- the subgraph corresponding to each node set is connected, that is, $G_{S_i}$ is connected for all $i$;
- any two node set is disjoint, that is, $S_i \cap S_j = \emptyset$;
- the union of the node sets is $\mathcal{V}$, that is, $\bigcup_i S_i = \mathcal{V}$.

There may exist various divisions of local sets, which will be discussed later. The local sets decompose a graph structure into multiple smaller pieces. Instead of studying a huge graph, we instead study multiple local sets.

We assign a graph signal to each local set. We represent a local set $S$ by using $\mathbf{1}_S \in \mathbb{R}^N$, where

$$(\mathbf{1}_S)_i = \begin{cases} 1, & i \in S; \\ 0, & \text{otherwise.} \end{cases}$$

For each local set signal, we measure its smoothness using normalized variation

$$\mathrm{S}_p(S) = \frac{1}{\|\mathbf{1}_S\|_p^p} \|\Delta \mathbf{1}_S\|_p^p .$$

For unweighted graphs, $\mathrm{S}_0(S) = \mathrm{S}_1(S) = \mathrm{S}_2(S)$. For a given local set, it measures how hard it is to cut the boundary edges and make $G_S$ an isolated subgraph. We normalize the variation by the size of the local set, which implies that given the same cut cost, a larger local set is more smooth than a smaller local set.

### 4.6.2 Multiresolution Local Sets

We aim to construct a series of local sets in a multiresolution fashion. We first define the multiresolution analysis on graphs.

**Definition 16.** A general multiresolution analysis on graphs consists of a sequence of embedded closed subspaces

$$V_0 \subset V_1 \subset V_2 \cdots \subset V_K$$

such that

- upward completeness
$$\bigcup_{i=0}^{K} V_i = \mathbb{R}^N;$$
- downward completeness
$$\bigcap_{i=0}^{K} V_i = \{c\mathbf{1}_{\mathcal{V}}, c \in \mathbb{R}\};$$
- existence of basis There exists an orthonormal basis $\{\Phi\}_i$ for $V_K$.

Compared with the original multiresolution analysis, the complete space here is $\mathbb{R}^N$ instead of $\mathcal{L}_2(\mathbb{R})$. As a result of the discrete nature of a graph; we remove scale invariance and translation invariance from the above axiomatic definition because the rigorous definitions of scaling and translation onb graphs are still unclear. This is the reason we call it *general multiresolution analysis on graphs.*

#### General Construction

The main idea behind this proposed construction is to build a connection between the subspaces and local sets: a bigger subspace corresponds to a finer resolution on the graph vertex domain, or more localized local sets. We initialize such that $S_{0,1} = \mathcal{V}$ corresponds to $V_0$, that is, $V_0 = \{c_0\mathbf{1}_{S_{0,1}}, c_0 \in \mathbb{R}\}$. We then partition $S_{0,1}$ into two disjoint local sets $S_{1,1}$ and $S_{1,2}$, which corresponds to $V_1$, where $V_1 = \{c_1\mathbf{1}_{S_{1,1}} + c_2\mathbf{1}_{S_{1,2}}, c_1, c_2 \in \mathbb{R}\}$. In this manner, we recursively partition a larger local set into two smaller local sets. For the $i$th level subspace, we have $V_i = \sum_{j=1}^{2^i} c_j\mathbf{1}_{S_{i,j}}$ and then, we partition $S_{i,j}$ into $S_{i+1,2j-1}, S_{i+1,2j}$ for all $j = 1, 2, \cdots, 2^i$. We call $S_{i,j}$ is the parent set of $S_{i+1,2j-1}, S_{i+1,2j}$ and $S_{i+1,2j-1}, S_{i+1,2j}$ are the children set of $S_{i,j}$. When $|S_{i,j}| \leq 1$, $S_{i+1,2j-1} = S_{i,j}$ and $S_{i+1,2j} = \emptyset$. In the finest resolution, each local set corresponds to an individual node or an empty set. In other words, we build a binary decomposition tree that partitions a graph structure into multiple local sets. The $i$th level of the decomposition tree corresponds to the $i$th level subspace. The maximum level of the decomposition $K$ depends on how we partition the local sets. $K$ ranges from $N$ to $\lceil \log N \rceil$, where $N$ corresponds to a partitioning of one node at each step and $\lceil \log N \rceil$ corresponds to an even partition at each step.

It is clear that the proposed construction of local sets satisfies three requirements in Definition 16. The initial subspace $V_0$ has the coarsest resolution. Through the partition, local sets zoom into an increasingly finer resolution on the graph vertex domain. The subspace $V_K$ with finest resolution zooms into each individual node and is a basis that spans the entire $\mathbb{R}^N$. We also revisit

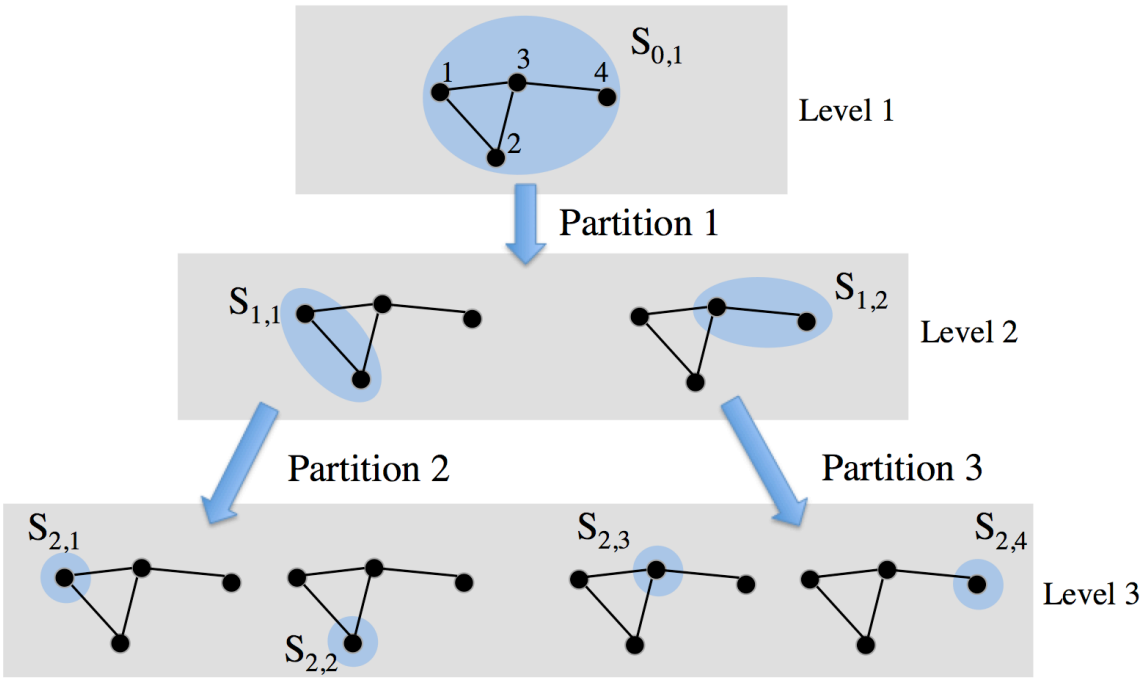



*Figure 4.8: Local set decomposition tree. In each partition, we decompose a node set into two disjoint connected set and generate a basis vector to the wavelet basis. $S_{0,1}$ is in Level 0, $S_{1,1}, S_{1,2}$ are in Level 1, and $S_{2,1}, S_{2,2}, S_{2,3}, S_{2,4}$ are in Level 2.*

scale invariance and translation invariance. The original scale invariance requires that when $f(t) \in V_0$, we have $f(2^m t) \in V_m$, which is ill-posed because graphs are naturally finite and discrete; the original translation invariance requires that when $f(t) \in V_0$, then $f(t-n) \in V_0$, which is ill-posed because graphs are irregular. The essence of scale and translation invariance is to use the same function with its scaled versions and translates to span different subspaces. The proposed construction still promotes similar attributes. The scaling function is $\mathbf{1}_S$; the hierarchy of partition is similar to the scaling and translation, that is, when $\mathbf{1}_{S_{i,j}} \in \mathcal{V}_i$, then $\mathbf{1}_{S_{i+1,2j-1}}, \mathbf{1}_{S_{i+1,2j}} \in \mathcal{V}_{i+1}$, and when $\mathbf{1}_{S_{i+1,2j-1}} \in \mathcal{V}_{i+1}$ then $\mathbf{1}_{S_{i+1,2j}} \in \mathcal{V}_{i+1}$.

To summarize the construction, we build a local set decomposition tree by recursively partitioning a local set into two disjoint local sets until that all the local sets are individual nodes. We now show a toy example in Figure 4.8. In Partition 1, we partition the entire node set $S_{0,1} = \mathcal{V} = \{1,2,3,4\}$ into two disjoint local sets $S_{1,1} = \{1,2\}, S_{1,2} = \{3,4\}$. Thus, $V_1 = \{c_1 \mathbf{1}_{\mathbf{S_{1,1}}} + c_2 \mathbf{1}_{\mathbf{S_{1,2}}}, c_1, c_2 \in \mathbb{R}\}$. Similarly, in Partition 2, we partition $S_{1,1}$ into two disjoint connected sets $S_{2,1} = \{1\}, S_{2,2} = \{2\}$; in Partition 3, we partition $S_{1,2}$ into $S_{2,3} = \{3\}, S_{2,4} = \{4\}$. Thus, $V_2 = \{c_1 \mathbf{1}_{\mathbf{S_{2,1}}} + c_2 \mathbf{1}_{\mathbf{S_{2,2}}} + c_3 \mathbf{1}_{\mathbf{S_{2,3}}} + c_4 \mathbf{1}_{\mathbf{S_{2,4}}}, c_1, c_2, c_3, c_4 \in \mathbb{R}\} = \mathbb{R}^4$.

### Graph Partition Algorithm

The graph partition is the key step to construct the local sets. From the perspective of promoting smoothness of graph signals, we partition a local set $S$ into two disjoint local set $S_1, S_2$ by solving

the following optimization problem

$$\begin{aligned} \min_{S_1,S_2} \quad & \mathrm{S}_0(S_1) + \mathrm{S}_0(S_2) \\ \text{subject to :} \quad & S_1 \cap S_2 = \emptyset, S_1 \cup S_2 = S, \\ & G_{S_1} \text{ and } G_{S_2} \text{ are connected.} \end{aligned} \tag{4.27}$$

Ideally, we aim to solve 4.27 to obtain two children local sets, however, it is nonconvex and hard to solve. Instead, we consider three relaxed methods to partition a graph.

The first method is based on spectral clustering[169]. We first obtain the graph Laplacian matrix of a local set and compute the eigenvector corresponding to the second smallest eigenvalue of the graph Laplacian matrix. We then set the median number of the eigenvector as the threshold; we put the nodes whose corresponding values in the eigenvector are no smaller than the threshold into a children local set and put the nodes whose corresponding values in the eigenvector are smaller than the threshold into the other children local set. This method approximately solves 4.27 by ignoring the second constraint; it guarantees that two children local sets have the same number of nodes, but does not guarantee that they have the same number of nodes are connected.

The second method is based on spanning tree. To partition a local set, we first obtain the maximum spanning tree of the subgraph and then find a balance node in the spanning tree. The balance node partition the spanning tree into two subtrees with the closet number of nodes[171]. We remove the balance node from the spanning tree, the resulting largest connected component form a children local set and the other nodes including the balance node forms the other children local set. This method approximately solves 4.27 by approximating a subgraph by the corresponding maximum spanning tree; it guarantee that two children local sets are connected, but does not guarantees that they have the same number of nodes. When the original sbugraph is highly connected, the spanning tree loses some connection information and the shape of the local set may not capture the community in the subgraph.

The third method is based on the 2-means clustering. We first randomly select 2 nodes as the community center and assign every other node to its nearest community center based on the geodesic distance. We then recompute the community center for each community by minimizing the summation of the geodesic distances to all the other nodes in the community and assign node to its nearest community center again. We keep doing this until the community centres converge after a few iterations. This method is inspired by the classical $k$-means clustering; it also guarantees that two children local sets are connected, but does not guarantee that they have the same number of nodes.

In general, the proposed construction of local sets is not restricted to any particular graph partition algorithm; depending on the applications, the partition step can also be implemented by many other existing graph partition algorithms.

### Dictionary Representations

We collect the local sets based on the levels in a ascending order and represent them in a dictionary, whose atom corresponds to each local set, that is,

$$\mathcal{D} = \{\mathbf{1}_{S_{i,j}}\}_{i=0,j=1}^{i=K,j=2^i}.$$

We call it the *local-set-based dictionary*. When we remove the empty sets, the local-set-based dictionary has $2N-1$ atoms, that is, $\mathcal{D} \in \mathbb{R}^{N\times(2N-1)}$; each atom is a piecewise-constant graph signal with various sizes and localizing various parts of a graph. The local set dictionary provides a redundant representation for any graph signal. We later show that it is particularly good for representing piecewise-constant graph signals.

### 4.6.3 Wavelet Basis

We construct a wavelet basis based on the local set dictionary. We combine two local sets partitioned from the same parent local set to form a basis vector. Let the local sets $S_{i+1,2j-1}, S_{i+1,2j}$ have the same parent local set $S_{i,j}$, the basis vector combing these two local sets is

$$\sqrt{\frac{|S_{i+1,2j-1}||S_{i+1,2j}|}{|S_{i+1,2j-1}|+|S_{i+1,2j}|}}\left(\frac{1}{|S_{i+1,2j-1}|}\mathbf{1}_{S_{i+1,2j-1}} - \frac{1}{|S_{i+1,2j}|}\mathbf{1}_{S_{i+1,2j}}\right).$$

To represent in a matrix form, the wavelet basis is

$$\mathbf{W} = \mathcal{D}\,\mathbf{U},$$

where

$$\begin{aligned}\mathbf{U} &= \begin{bmatrix} \frac{1}{\|1\|_2} & 0 & \cdots & 0 \\ 0 & g(2,3) & \cdots & 0 \\ 0 & -g(3,2) & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g(2N-2,2N-1) \\ 0 & 0 & \cdots & -g(2N-1,2N-2) \end{bmatrix} \\ &\in \mathbb{R}^{(2N-1)\times N},\end{aligned}$$

and $i$ is the $i$th column of $\mathcal{D}$, and

$$g(i,j) = \sqrt{\frac{\|j\|_0}{(\|i\|_0+\|j\|_0)\,\|i\|_0}}.$$

The matrix $\mathbf{U}$ acts like a downsampling matrix that combines two consecutive column vectors in $\mathcal{D}$ to form one column vector in $\mathbf{W}$ and the function $g(\cdot,\cdot)$ reweights the column vectors in $\mathcal{D}$ to ensure that each column vector in $\mathbf{W}$ has norm 1 and sums to 0.

Another explanation is that when we recursively partition a node set into two local sets, each partition generates a wavelet basis vector. We still use Figure 4.8 as an example. In Partition 1, we partition the entire node set $S_{0,1} = \{1,2,3,4\}$ into $S_{1,1} = \{1,2\}, S_{1,2} = \{3,4\}$ and generate a

basis vector

$$\begin{aligned} & \sqrt{\frac{|S_{1,1}||S_{1,2}|}{|S_{1,1}|+|S_{1,2}|}}\left(\frac{1}{|S_{1,1}|}\mathbf{1}_{S_{1,1}}-\frac{1}{|S_{1,2}|}\mathbf{1}_{S_{1,2}}\right) \\ = \quad & \frac{1}{2}\begin{bmatrix}1 & 1 & -1 & -1\end{bmatrix}; \end{aligned}$$

in Partition 2, we partition $S_{1,1}$ into two disjoint connected sets $S_{2,1}=\{1\}, S_{2,2}=\{2\}$ and generate a basis vector

$$\begin{aligned} & \sqrt{\frac{|S_{2,1}||S_{2,2}|}{|S_{2,1}|+|S_{2,2}|}}\left(\frac{1}{|S_{2,1}|}\mathbf{1}_{S_{2,1}}-\frac{1}{|S_{2,2}|}\mathbf{1}_{S_{2,2}}\right) \\ = \quad & \frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1 & 0 & 0\end{bmatrix}; \end{aligned}$$

in Partition 3, we partition $S_{1,2}$ into $S_{2,3}=\{3\}, S_{2,4}=\{4\}$ and generate a basis vector

$$\begin{aligned} & \sqrt{\frac{|S_{2,3}||S_{2,4}|}{|S_{2,3}|+|S_{2,4}|}}\left(\frac{1}{|S_{2,3}|}\mathbf{1}_{S_{2,3}}-\frac{1}{|S_{2,4}|}\mathbf{1}_{S_{2,4}}\right) \\ = \quad & \frac{1}{\sqrt{2}}\begin{bmatrix}0 & 0 & 1 & -1\end{bmatrix}. \end{aligned}$$

We summarize the construction of the *local-set-based wavelet basis* in Algorithm 6.

**Algorithm 6** Local-set-based Wavelet Basis Construction

**Input** $G(\mathcal{V},\mathcal{E},\mathbf{A})$ graph
**Output** $\mathbf{W}$ wavelet basis

**Function**

initialize a stack of node sets $\mathbb{S}$ and a set of vectors $\mathbf{W}$
push $S=\mathcal{V}$ into $\mathbb{S}$
add $\mathbf{w}=\frac{1}{\sqrt{|S|}}\mathbf{1}_S$ into $\mathbf{W}$
while the cardinality of the largest element of $\mathbb{S}$ is bigger than 1
 pop up one element from $\mathbb{S}$ as $S$
 partition $S$ into two disjoint connected sets $S_1, S_2$
 push $S_1, S_2$ into $\mathbb{S}$
 add $\mathbf{w}=\sqrt{\frac{|S_1||S_2|}{|S_1|+|S_2|}}\left(\frac{1}{|S_1|}\mathbf{1}_{S_1}-\frac{1}{|S_2|}\mathbf{1}_{S_2}\right)$ into $\mathbf{W}$
end
**return W**

### 4.6.4 Analysis

We now analyze some properties of the proposed construction of the local sets and wavelet basis. The main results are

- the local-set-based dictionary provides a multiresolution representation;
- there exists a tradeoff between smoothness and fine resolution in partitioning the local sets;
- the local-set-based wavelet basis is an orthonormal basis;
- the local-set-based wavelet basis promotes sparsity for piecewise-constant graph signals.

**Theorem 26.** The proposed construction of local sets satisfies the multiresolution analysis on graphs.

We have shown this in the previous section. We state it again here for completeness. In the original multiresolution analysis, more localization in the time domain leads to more high-frequency components. Here we show a result that is similar in spirit.

**Theorem 27.** A series of local sets with a finer resolution is less smooth, that is, for all $i$,

$$\sum_{j=1}^{2^i} \mathrm{S}_0(S_{i,j}) \leq \sum_{j=1}^{2^{i+1}} \mathrm{S}_0(S_{i+1,j}).$$

*Proof.* We first show that the sum of variations of two children local sets is larger than the variation of the parent local set.

$$\begin{aligned} & \mathrm{S}_0(S_{i+1,2j-1}) + \mathrm{S}_0(S_{i+1,2j}) \\ = \quad & \frac{1}{|S_{i+1,2j-1}|} \left\|\boldsymbol{\Delta}\mathbf{1}_{S_{i+1,2j-1}}\right\|_0 + \frac{1}{|S_{i+1,2j}|} \left\|\boldsymbol{\Delta}\mathbf{1}_{S_{i+1,2j}}\right\|_0 \\ \overset{(a)}{\geq} \quad & \frac{1}{|S_{i,j}|} \left(\left\|\boldsymbol{\Delta}\mathbf{1}_{S_{i+1,2j-1}}\right\|_0 + \left\|\boldsymbol{\Delta}\mathbf{1}_{S_{i+1,2j}}\right\|_0\right) \\ \overset{(b)}{\geq} \quad & \frac{1}{|S_{i,j}|} \left\|\boldsymbol{\Delta}\mathbf{1}_{S_{i,j}}\right\|_0 \;=\; \mathrm{S}_0(S_{i,j}), \end{aligned}$$

where $(a)$ follows from that the cardinality of the the parent local set is larger than either of its children local sets and $(b)$ follows from that we need to cut a boundary to partition two children local sets. Since every local set in the $i$th level has two children local sets in the $i+1$th level and every local set in the $i+1$th level has a parent local set in the $i$th local, we sum them together over $j$ and obtain Theorem 27. □

Theorem 27 shows that by zooming in on the graph vertex domain, the partitioned local sets get less smooth; in other words, we have to tradeoff smoothness to obtain a finer resolution.

We next show that the local-set-based wavelet basis is a valid orthonormal basis.

**Theorem 28.** The local-set-based wavelet basis construction is an orthonormal basis.

*Proof.* First, we show each vector has norm one.

$$\begin{aligned} & \left\| \sqrt{\frac{|S_1||S_2|}{|S_1|+|S_2|}} \left( \frac{1}{|S_1|}\mathbf{1}_{S_1} - \frac{1}{|S_2|}\mathbf{1}_{S_2} \right) \right\|_2^2 \\ \overset{(a)}{=} \quad & |S_1| \left( \sqrt{\frac{|S_1||S_2|}{|S_1|+|S_2|}} \frac{1}{|S_1|} \right)^2 + |S_2| \left( \sqrt{\frac{|S_1||S_1|}{|S_1|+|S_2|}} \frac{1}{|S_2|} \right)^2 \\ = \quad & 1, \end{aligned}$$

where $(a)$ follows from that $S_1 \cap S_2 = \emptyset$. Second, we show each vector is orthogonal to the other vectors. We have

$$\mathbf{1}^T \mathbf{w} = \sqrt{\frac{|S_1||S_2|}{|S_1|+|S_2|}} \left( \sum_{i \in S_1} \frac{1}{|S_1|} - \sum_{i \in S_2} \frac{1}{|S_2|} \right) = 0.$$

Thus, each vector is orthogonal to the first vector, $\mathbf{1}_{\mathcal{V}}/\sqrt{|\mathcal{V}|}$. Each other individual vector is generated from two node sets. Let $S_1, S_2$ generate $\mathbf{w}_i$ and $S_3, S_4$ generate $\mathbf{w}_j$. Due to the construction, there are only two conditions, two node sets of one vector belongs to one node set of the other vector, and all four node sets do not share element with each other. For the first case, without loss of generality, let $(S_3 \cup S_4) \cap S_1 = S_3 \cup S_4$, we have

$$\begin{aligned} \mathbf{w}_i^T \mathbf{w}_j &= \sqrt{\frac{|S_1||S_2|}{|S_1|+|S_2|}\frac{|S_3||S_4|}{|S_3|+|S_4|}}\left(\sum_{i\in S_3}\frac{1}{|S_3|} - \sum_{i\in S_4}\frac{1}{|S_4|}\right) \\ &= 0. \end{aligned}$$

For the second case, the inner product between $\mathbf{w}_i$ and $\mathbf{w}_j$ is zero because their supports do not match. Third, we show that $\mathbf{W}$ spans $\mathbb{R}^N$. Since we recursively partition the node set until the cardinalities of all the node sets are smaller than 2, there are $N$ vectors in $\mathbf{W}$. The statement follows □

We show that the local-set-based wavelet basis is a good representation for piecewise-constant graph signals because it promotes sparsity.

**Theorem 29.** Let $\mathbf{W}$ be the output of Algorithm 6 and $L$ be the maximum level of the decomposition in Algorithm 6 . For all $\mathbf{x} \in \mathbb{R}^N$, we have

$$\left\|\mathbf{W}^T\mathbf{x}\right\|_0 \leq \|\Delta\mathbf{x}\|_0 L.$$

*Proof.* When an edge $e \in \text{Supp}(\Delta\mathbf{w})$, where Supp denotes the indices of nonzero elements, we say that the edge $\mathbf{e}$ is activated by the vector $\mathbf{w}$. Since each edge is activated at most once in each decomposition level, each edge is activated by at most $L$ basis elements. Let activations$(e)$ be the number of basis elements in $\mathbf{W}$ that activates $e$.

$$\left\|\mathbf{W}^T\mathbf{x}\right\|_0 \leq \sum_{e\in\text{Supp}(\Delta\mathbf{w})} \text{activations}(e) \leq \|\Delta\mathbf{x}\|_0 L.$$

□

The maximum level of the decomposition is determined by the choice of graph partition algorithm. Theorem 29 shows that what matters is the cardinality of each local set, instead of the shape of each local set. To achieve the best sparse representation, we should partition each local set as evenly as possible. Note that when the partition is perfectly even, the resulting wavelet basis is the same as the classical Haar wavelet basis.

**Corollary 7.** Let the local-set-based wavelet basis evenly partition the node set each time. For all $\mathbf{x} \in \text{PC}_G(K)$, we have

$$\left\|\mathbf{W}^T\mathbf{x}\right\|_0 \leq K\lceil\log N\rceil.$$

We see that the local-set-based wavelet basis leads to a sparse representation for the piecewise-constant graph signals. Furthermore, we conjecture that the proposed construction is the optimal orthonormal basis to promote sparsity for piecewise-constant graph signals.

In general, the graph difference operator provides more sparse representation than the local-set-based wavelet basis, however, the graph difference operator is not necessarily a one-to-one mapping and is bad at reconstruction; the graph difference operator only focuses on the pairwise relationship. On the other hand, the local-set-based wavelet basis is good at reconstruction and provides a multiresolution view in the graph vertex domain.

The even partition minimizes the worst case; it does not necessarily mean that the even partition is always well suited for all applications. For example, a graph has two communities, a huge one and a tiny one, which hints that a piecewise-constant graph signal sits on a part of either of two communities. In this case, we cut a few edges to partition two communities and assign a local set for each of them, instead of partitioning the huge community to make sure that two local sets have the same cardinality.

### 4.6.5 Review & Discussion

In this section, we review some previous related works and discuss their relations to the above exposition

There are mainly two approaches to design a representation for graph signals: one is based on the graph Fourier domain and the other one is based on the graph vertex domain.

The representations based on the graph Fourier domain are based upon the spectral properties of the graph. The most fundamental representation based on the graph Fourier domain is the graph Fourier transform, which is the eigenvectors of a matrix that represents a graph structure[172,6]. Based on the graph Fourier transform, people propose various versions of multiresolution transforms on graphs, including diffusion wavelets[173], spectral graph wavelets[7], graph quadrature mirror filter banks[174], windowed graph Fourier transform[175], polynomial graph dictionary[176]. The main idea is to construct a series of graph filters on the graph Fourier domain, which are localized on both the vertex and graph Fourier domains. The advantages of the representations on the graph Fourier domain are: first, it avoids the complex and irregular connectivity on the graph vertex domain because each frequency is independent; second, it is efficient, because the construction is simply to determine a series of filter coefficients, where the computation is often accelerated by the polynomial approximation; third, it is similar to the design of the classical wavelets. However, there are two shortcomings: first, it loses the discrete nature of a graph. That is, the construction is not directly based on the graph frequencies; instead, it proposes a continuous kernel from which values are sampled; second, the localization on the graph vertex domain is worse than the representations based on the graph vertex domain. It is true that this construction provides better localization on the graph Fourier domain. However, the concept of localization on the graph Fourier domain is often vague, abstract, and is often less important in most real-world applications.

The representations based on the graph vertex domain are based on the connectivity properties of the graph. The advantages are: first, it provides better locality on the graph vertex domain and is easier to visualize; second, it provides a better understanding of the connectivity of a graph, which is avoided by the graph Fourier transform for the representations on the graph Fourier domain. Some examples of the representations include multiscale wavelets on trees[177], graph wavelets for spatial analysis[178], spanning tree wavelet basis[171]. The proposed representations also adopt this approach.

- Spanning tree wavelet basis proposes a localized basis on a spanning tree. The proposed work is mainly inspired by this work and the proposed representations generalize the results from a spanning tree to a general graph.
- Multiscale wavelets on trees provides a hierarchy tree representation for a dataset. It proposes a wavelet-like orthonormal basis based on a balanced binary tree, which is similar to our proposed representations. This previous work focuses on high dimensional data and the representation properties for a smooth signal; the proposed work focuses on a graph structure and representation properties for a piecewise-constant signal;
- Graph wavelets for spatial traffic analysis proposes a general wavelet representation on graphs. The wavelet basis vectors are not generated from a single function, that is, the wavelet coefficients at different scales and locations are different; the proposed representations resemble the Haar wavelet basis in spirit and are generated from a single indicator function.

## 4.7 Multiresolution Analysis and Wavelets on Graphs

Classical wavelets are constructed by translating and scaling a single *mother wavelet*. The transform coefficients are then given by the inner products of the input function with these translated and scaled waveforms. Directly extending this construction to arbitrary weighted graphs is problematic, as it is unclear how to define scaling and translation on an irregular graph. The spectral graph wavelets[179] define a multiresolution analysis by defining scaling with respect to the graph spectral domain. Vertex-based multiresolution analyses however have greater interpretability and may be more computationally efficient as computing the spectral decomposition of a graph is generally computationally expensive. Previous work in GSP[180,181] has established vertex-based multiresolution analyses wavelets on graphs but there is a lack of a unifying framework that would allow a more rigorous analysis. Towards this, we propose using the Matrix Multiresolution Factorization (MMF) framework introduced by Kondor et al[182] to generalize and analyze vertex-based graph wavelets and multiresolution analyses on graphs. Multiresolution matrix factorization (MMF) uncovers soft hierarchical organization in matrices, characteristic of naturally occurring large networks and the covariance structure of large collections of random variables, without enforcing a hard hierarchical clustering.

### 4.7.1 Matrix Multiresolution Factorization

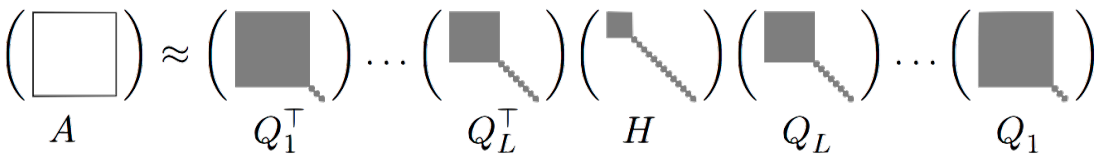



*Figure 4.9: MMF Factorization: The resulting factorizations, provide a natural way to define multiresolution on graphs*

MMF formulates the matrix factorization analog of multiresolution analysis on finite sets. MMF uncovers soft hierarchical organization in matrices, characteristic of naturally occurring large net-

works. A multiresolution analysis with respect to a symmetric matrix $\mathbf{A}$ consists of a sequence of spaces $V_L \subset \cdots V_2 \subset V_1 = \mathbb{R}^n$ where $V_l$ has an orthonormal basis $\Phi_l$. $\Phi_l$ and the complementary space $W_l$ has a basis $\Psi_l$ such that the wavelets in $W_l$ are increasingly localized. Further, each element in $\Phi_l$ and $\Psi_l$ can be sparsely approximated by elements in $\Phi_{l-1}$. The key idea is that each $V_{l-1} \to V_l \oplus W_l$ basis transformation can be represented by a sparse orthogonal transformation $\boldsymbol{Q}_l$. Hence in a multiresolution matrix factorization up to depth $L$,we decompose the matrix $\mathbf{A}$ by a sequence of sparse orthogonal transforms as

$$\mathbf{A} = (\boldsymbol{Q}_1^T \boldsymbol{Q}_2^T \cdots \boldsymbol{Q}_L^T)\boldsymbol{H}(\boldsymbol{Q}_L \cdots \boldsymbol{Q}_2 \boldsymbol{Q}_1) \tag{4.28}$$

where $\boldsymbol{H}$ is diagonal outside of a block of size $d_L \times d_L$ where $d_L = dim(V_L)$ (Figure 4.9). Note that when $\boldsymbol{H}$ is diagonal, this corresponds to the graph Fourier transform. The wavelets correspond to the rows of $\boldsymbol{Q}_L \cdots \boldsymbol{Q}_2 \boldsymbol{Q}_1$. As a result denoting the wavelet transform as $\boldsymbol{W} = \boldsymbol{Q}_L \cdots \boldsymbol{Q}_2 \boldsymbol{Q}_1$, the wavelet coefficients can be analyzed as $\boldsymbol{\alpha} = \boldsymbol{W}\mathbf{x}$ and synthesized as inverse wavelet transform $\mathbf{x} = \boldsymbol{W}^T\boldsymbol{\alpha}$. Here we only give a cursory overview of the framework and omit further details. We note that the MMF framework generalizes the multiresolution construction presented in previous works[180,181]. In fact, the Haar transforms on graphs based on recursive bipartitioning corresponds to a particular series of sparse orthogonal rotations.

#### Separable Multiresolution Wavelets on Product Graphs

Let us consider the product graph $\mathbf{A} = \mathbf{A}_1 \otimes \mathbf{A}_2$, and define multiresolution analyses on each of $\mathbf{A}_1$ and $\mathbf{A}_2$.

$$\begin{aligned} V_{n+1} &= V_{n+1}^{(1)} \otimes V_{n+1}^{(2)} && (4.29)\\ &= (V_n^{(1)} \oplus W_n^{(1)}) \otimes (V_n^{(2)} \oplus W_n^{(2)}) && (4.30)\\ &= (V_n^{(1)} \otimes V_n^{(2)}) \oplus (V_n^{(1)} \otimes W_n^{(2)}) \oplus (W_n^{(1)} \otimes V_n^{(2)}) \oplus (W_n^{(1)} \otimes W_n^{(2)}) && (4.31)\\ &= V_n \otimes W_n && (4.32) \end{aligned}$$

such that $W_n = (V_n^{(1)} \otimes W_n^{(2)}) \oplus (W_n^{(1)} \otimes V_n^{(2)}) \oplus (W_n^{(1)} \otimes W_n^{(2)})$

This is analogous to separable wavelet construction by tensorization on images and $d$-dimensional grids.

### 4.7.2 Wavelet Denoising

We define the class of piecewise-compressible graph signals signals under the graph wavelet transform $\boldsymbol{W}$ defined under the multiresolution analysis formulation in Section 4.7.

**Definition 17.** Let $\boldsymbol{\alpha} = \boldsymbol{W}\mathbf{x}$ be the wavelet coefficients such that $\alpha_{(k)}$ refers to the $k$-th largest entry in the coefficient vector $\boldsymbol{\alpha}$. If there exists a constant $C$ such that

$$|\alpha_{(k)}| \leq C\frac{\|\Delta^{(1)}\mathbf{x}\|_1}{k} \tag{4.33}$$

then the signal is *piecewise-compressible*.

We then study the wavelet smoothing problem that enforces sparsity with respect to the wavelet coefficients

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{y}-\mathbf{x}\|_2 + \lambda \|\boldsymbol{W}\mathbf{x}\|_1 \tag{4.34}$$

We note that for the Haar graph wavelet construction[181], we can show that piecewise constant signals are compressible under the transform. Further, a rather remarkable result shows that natural images with bounded variation are compressible under the 2-dimensional Haar wavelet transform[183]. In addition, this result can be extended to $d$-dimensional grids.

In this section, we then try and rigorously understanding the difference between trend filtering and wavelet denoising on graphs. Empirically, we make the observations that trend filtering is more robust to noise than wavelet smoothing. We study piecewise compressible signals and the performance of the soft-thresholding estimator. We can then show the following theoretical guarantee on the recovery performance.[133]

**Theorem 30.** We consider problem 4.34. It is clear that this problem can be solved by soft-thresholding the wavelet coefficients. Let $\widehat{\mathbf{x}} = \mathbf{W}^T\tau(\mathbf{W}\mathbf{x};\lambda)$ be the reconstructed signal where $\tau$ is the soft-thresholding operator with respect to $\lambda$. Then, for large enough $n$, we can show that up to a constant factor,

$$\frac{1}{n}\mathbb{E}\|\widehat{\mathbf{x}}-\mathbf{x}^*\|_2^2 \leqslant \frac{logN}{N}(\sigma^2 + \sigma\|\Delta\mathbf{x}^*\|_1) \tag{4.35}$$

That is, we can show an adaptive rate for the MSE in terms of the true $\|\Delta\mathbf{x}^*\|_1$.

## 4.8 Applications of Graph Wavelets and Haar Multiresolution Analysis

In this section, we study various applications of the proposed multiresolution local sets and the corresponding representations. The applications include, approximation, denoising, and estimation. For each application, we design a specific algorithm and provide an extensive empirical evaluation with respect to other state of the art algorithms.

### 4.8.1 Approximation

Approximation is a standard task to evaluate a representation and is similar in many ways to compression. The goal is to use a few expansion coefficients to approximate a graph signal. We compare the graph Fourier transform[6], the windowed graph Fourier transform[175], the local-set-based wavelet basis and dictionary. The graph Fourier transform is the eigenvector matrix of the graph shift and the windowed graph Fourier transform provides vertex-frequency analysis on graphs. For the local-set-based wavelet basis and dictionary, we also consider three graph partition algorithms, including spectral clustering, spanning tree and 2-means.

#### Algorithm

Since the graph Fourier transform and the local-set-based wavelet bases are orthonormal bases, we consider nonlinear approximation, that is, after expanding in with a representation, we should choose the $K$ largest-magnitude expansion coefficients so as to minimize the approximation error. Let $\{\phi_i \in \mathbb{R}^N\}_{i=1}^N$ be an orthonormal basis and $\mathbf{x} \in \mathbb{R}^N$ be a signal. The nonlinear approximation

to $\mathbf{x}$ is

$$\mathbf{x}' = \sum_{k \in \mathcal{I}_K} \langle \mathbf{x}, \phi_k \rangle \, \phi_k, \tag{4.36}$$

where $\mathcal{I}_K$ is the index set of the $K$ largest-magnitude expansion coefficients. When a basis promotes sparsity for $\mathbf{x}$, only a few expansion coefficients are needed to obtain a small approximation error.

Since the the windowed graph dictionary and the local-set-based dictionaries are redundant, we solve the following sparse coding problem,

$$\begin{aligned} \mathbf{x}' = \quad & \arg\min_{\mathbf{a}} \quad \|\mathbf{x} - \mathcal{D}\mathbf{a}\|_2^2, \\ & \text{subject to :} \quad \|\mathbf{a}\|_0 \leq K, \end{aligned} \tag{4.37}$$

where $\mathcal{D}$ is a redundant dictionary and $\mathbf{a}$ is a sparse code. The idea is to use a linear combination of a few atoms from $\mathcal{D}$ to approximate the original signal. When $\mathcal{D}$ is an orthonormal basis, the closed-form solution is exactly (4.36). We solve (4.37) by using the orthogonal matching pursuit, which is a greedy algorithm [184].

### Experiments

We test the four representations on two datasets, including the Minnesota road graph [185] and the U.S city graph [74].

For the Minnesota road graph, we simulate a piecewise-constant graph signal by randomly picking 5 nodes as community centers and assigning each other node to its nearest community center based on the geodesic distance. We assign a random integer to each community. The simulated graph signal is shown in Figure 4.10. The signal contains 5 piecewise constants and 84 inconsistent edges. The frequency coefficients and the wavelet coefficients obtained by using three graph partition algorithms are shown in Figure 4.10(b), (c), (d) and (e). The sparsities of the wavelet coefficients for spectral clustering, spanning tree, and 2-means are 364, 254, and 251, respectively; the proposed wavelet bases provide much better sparse representations than the graph Fourier transform. The evaluation metric of the approximation error is the normalized mean square error, that is,

$$\text{Normalized MSE} = \frac{\|\mathbf{x}' - \mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2},$$

where $\mathbf{x}'$ is the approximation signal and $\mathbf{x}$ is the original signal. Figure 4.10(f) shows the approximation errors given by the four representations. The x-axis is the number of coefficients used in approximation, which is $K$ in (4.36) and (4.37) and the y-axis is the approximation error, where lower means better. We see that the local-set-based wavelet with spectral clustering and local-set-based dictionary with spectral clustering provides much better performances and the windowed graph Fourier transform catches up with graph Fourier transform around 15 expansion coefficients. Figure 4.10(g) and (h) compares the local-set-based wavelets and dictionaries with three different partition algorithms, respectively. We see that the spanning tree and 2-means have similar performances, which are better than spectral clustering. This is consistent with the sparsities of the wavelet coefficients, where the wavelet coefficients of spanning tree and 2-means are more

sparse than those of spectral clustering.

The U.S city graph is a network representation of 150 weather stations across the U.S. We assign an edge when two weather stations are within 500 miles. The graph includes 150 nodes and 1033 undirected, unweighted edges. Based on the geographical area, we partition the nodes into four communities, including the north area (N), the middle area (M), the south area (S), and the west area (W). The corresponding piecewise-constant graph signal is

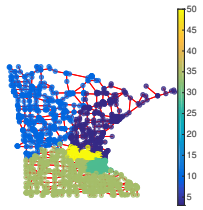
$$\mathbf{x} = \mathbf{1}_N + 2 \cdot \mathbf{1}_M + 3 \cdot \mathbf{1}_S + 4 \cdot \mathbf{1}_W. \tag{4.38}$$

The graph signal is shown in Figure 4.11(a), where dark blue indicates the north area, the light indicates the middle area, the dark yellow indicates the south area and the light yellow indicates the west area. The signal contains 4 piecewise constants and 144 inconsistent edges.

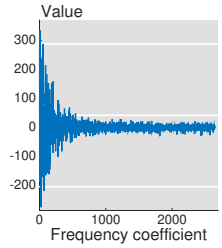
The frequency coefficients and the wavelet coefficients obtained by using three graph partition algorithms are shown in Figure 4.11(b), (c), (d) and (e). The sparsities of the wavelet coefficients for spectral clustering, spanning tree, and 2-means are 45, 56, and 41, respectively; the proposed wavelet bases provide much better sparse representations than the graph Fourier transform.

The evaluation metric of the approximation error is also the normalized mean square error. Figure 4.11(f) shows the approximation errors given by the four representations. Similarly to Figure 4.11(d), the local-set-based wavelet with spectral clustering and local-set-based dictionary with spectral clustering exhibiting much better performance. The windowed graph Fourier transform catches up with graph Fourier transform around 25 expansion coefficients. Figure 4.11(g) and (h) compares the local-set-based wavelets and dictionaries with three different graph partition algorithms, respectively. We see that partitioning using spectral clustering provides the best performance.

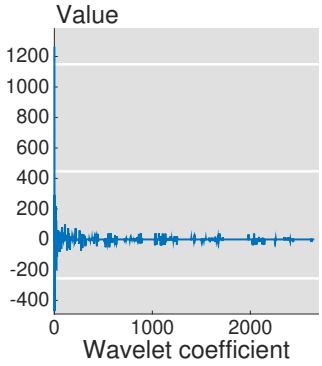
To summarize the task of approximation, the proposed local set based representations provide a reliable approximation for a piecewise-constant graph signal because it promotes sparsity.



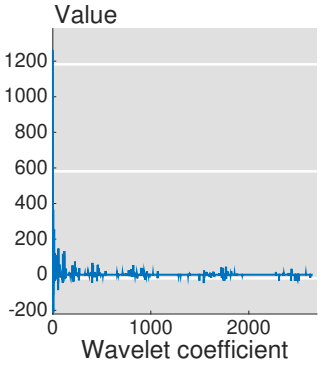
(a) Signal.



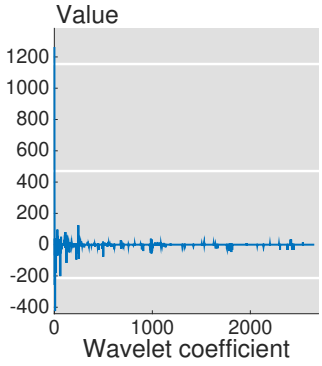

(b) Frequency coefficients.



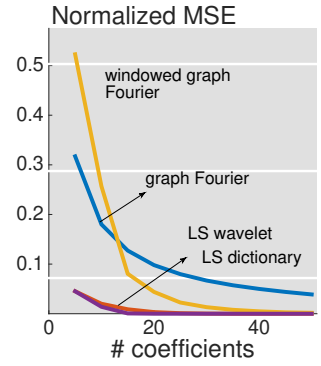

(c) Wavelet coefficients from spectral clustering.



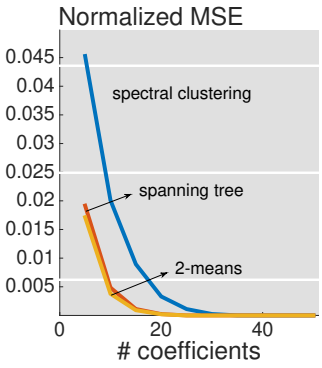

(d) Wavelet coefficients from spanning tree.



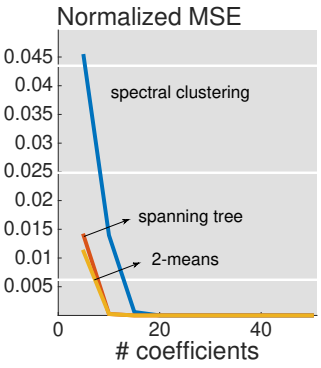

(e) Wavelet coefficients from 2-means.




(f) Error comparison.

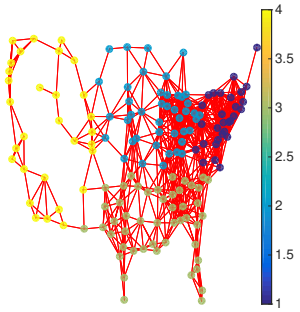



(g) Error (Wavelet)

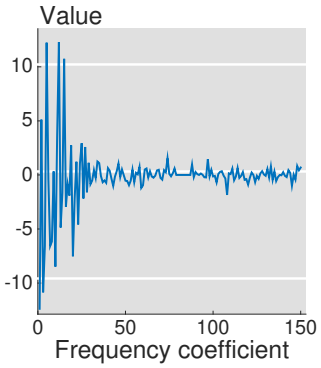



(h) Error (Dictionary).

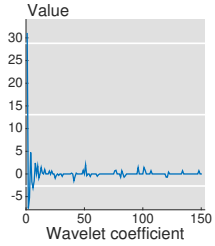
*Figure 4.10: Approximation on the Minnesota road graph.*



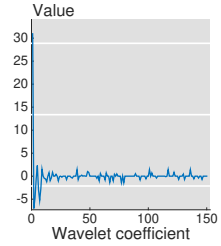
(a) Signal.



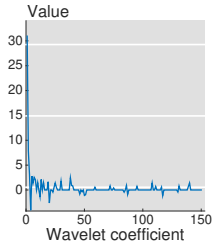

(b) Frequency coefficients.



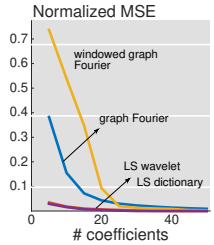

(c) Wavelet coefficients from spectral clustering.



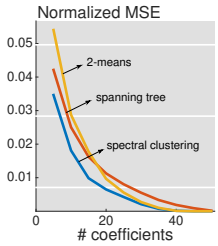

(d) Wavelet coefficients from spanning tree.



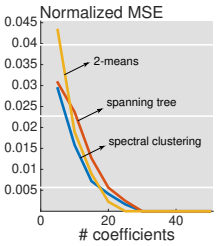

(e) Wavelet coefficients from 2-means.




(f) Error comparison.




(g) Error (wavelet).




(h) Error (dictionary).

*Figure 4.11: Approximation on the U.S city graph.*

### 4.8.2 Denoising

Denoising is one of the most important tasks in image processing. The goal is to remove noise from a signal. Let $\mathbf{x}$ be a piecewise-constant graph signal, $\mathbf{e}$ be noise, and $\mathbf{y} = \mathbf{x} + \mathbf{e}$ is a noisy graph signal. We aim to obtain $\mathbf{x}'$ from $\mathbf{y}$ and minimize the difference between $\mathbf{x}'$ and $\mathbf{x}$.

Similarly to Section 4.8.1, we compare the graph Fourier transform, the vertex-frequency graph dictionary, the local set wavelet basis and dictionary. For the local-set-based wavelet basis and dictionary, we consider three graph partition algorithms, including spectral clustering, spanning tree and 2-means. We also compare with trend filtering on graphs, which is particularly designed for the goal of denoising a signal defined on a graph[186].

#### Algorithm

For the graph Fourier transform and the PC wavelet basis, we use nonlinear approximation to obtain the denoised graph signal. Since $\mathbf{x}$ is unknown, we approximate $\mathbf{y}$ by using a few expansion coefficients. We assume that the main information about the noiseless graph signal is concentrated in those expansion coefficients and the other expansion coefficients are contaminated by noise. We thus obtain a high-quality graph signal by removing those noise-contaminated expansion coefficients. Similarly, for the vertex-frequency graph dictionary and the local set dictionary, we obtain the denoised graph signal by solving (4.37) by replacing $\mathbf{x}$ to $\mathbf{y}$. Trend filtering on graphs considers the following optimization problem.

$$\mathbf{x}' = \quad \arg\min_{\mathbf{x}} \quad \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu \|\Delta \mathbf{x}\|_1 ,$$

where $\Delta$ is the graph difference operator and $\mu$ is a tuning parameter that balances the approximation to $\mathbf{y}$ and the sparsity of $\Delta\mathbf{x}$. Since we know that $\mathbf{x}$ is piecewise constant, the difference $\Delta\mathbf{x}$ should be small.

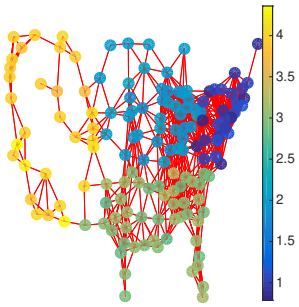
#### Experiments

We test those algorithms on the U.S city graph and use the same piecewise-constant graph signal (4.38) as the noiseless graph signal. We consider two noise levels. Let the noise $\mathbf{e} \sim \mathcal{N}(0, \sigma^2)$, where $\sigma$ varies as 0.2, which corresponds to a low noise level, and 0.4, which corresponds to a high noise level.

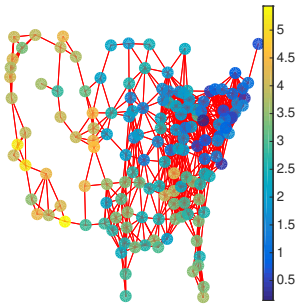
The evaluation metric used to evaluate the denoising error is still the normalized mean square error. Figure 4.12(a) shows the noisy graph signal with low noise level. The noise-to-signal ratio, $\|\mathbf{e}\|_2 / \|\mathbf{x}\|_2 = 7\%$. Figure 4.12(c) shows the denoising errors given by the five algorithms. The x-axis is the number of coefficients used in denoising, which is $K$ in (4.36) and (4.37) and the y-axis is the denoising error, where lower means better. The horizontal line in black indicates the best performance of trend filtering on graphs by tuning the parameter $\mu$. We see that trend filtering on graphs provide much better performances than the four representation-based algorithms in the case of the low noise level, and the local-set-based wavelets and dictionaries outperform the graph Fourier transform and windowed graph Fourier transform. Figure 4.12(e) and (g) compares the local-set-based wavelets and dictionaries with three different partition algorithms, respectively. We see that the spectral clustering and 2 means are slightly better than spanning tree.

Figure 4.12(b) shows the noisy graph signal with low noise level. The noise-to-signal ratio, $\|\mathbf{e}\|_2 / \|\mathbf{x}\|_2 = 14\%$. Figure 4.12(d) shows the denoising errors given by the five algorithms. We see that the local-set-based wavelets and dictionaries are slightly better than trend filtering on graphs in the case of high noise level. Figure 4.12(f) and (h) compares the local-set-based wavelets and dictionaries with three different partition algorithms, respectively. We see that three graph partition algorithms have similar performances.

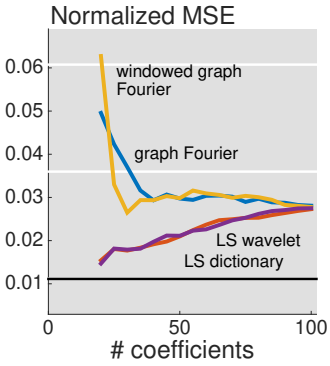
To summarize the task of denoising, the proposed local set based representations works well to remove noise from a piecewise-constant graph signal when the noise level is low. When the noise level is high, trend filtering on graphs works better than the representation-based algorithms because trend filtering on graphs penalizes the sparsity of $\mathbf{\Delta x}$, which emphasizes the pairwise differences.



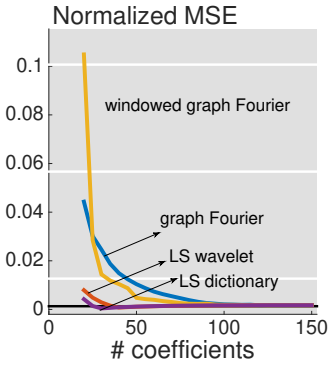

(a) Signal in low noise level.



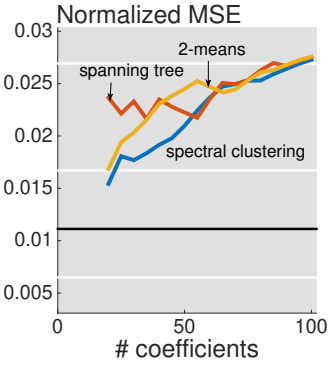

(b) Signal in high noise level.



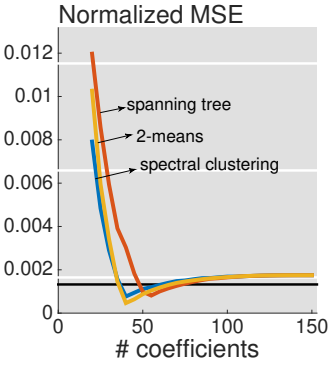

(c) Error in low noise level.



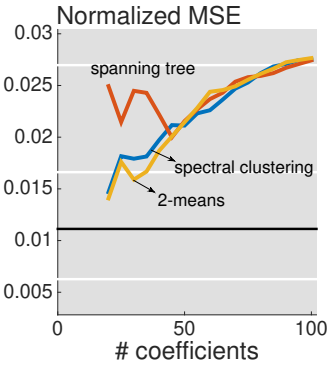

(d) Error in high noise level.



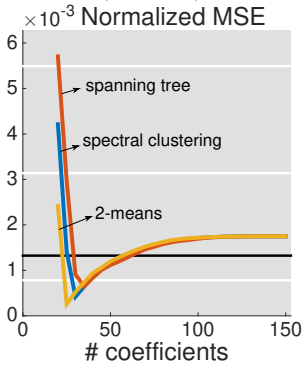

(e) Error in low noise level. (wavelet).




(f) Error in high noise level. (wavelet).




(g) Error in low noise level. (dictionary).




(h) Error in high noise level. (dictionary).

*Figure 4.12: Denoising on the U.S city graph.*

### 4.8.3 Case Study: Epidemics process

Epidemics process has been modeled as the diffusion process of ideas/opinions/beliefs/innovations over a finite-sized, static, connected social network[187]. In the terminology of epidemics, if the state of each node is either susceptible or infected, it is usually model by the susceptible-infected-susceptible (SIS) model. Nodes that are infected have a certain rate ($\gamma$) to recover and return to be susceptible; nodes that are susceptible can be contagious if infected by its neighboring infected nodes with a certain rate ($\beta$).

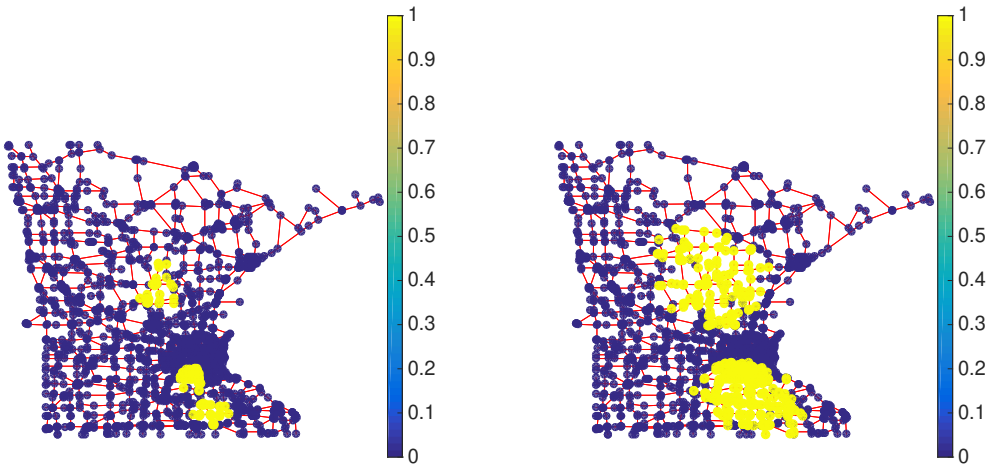
Here we adopt the SIS model on network, which takes the network structure into account and help us estimate the macroscopic behavior of an epidemic outbreak[62]. In SIS model on network, $\beta$ is the infection rate that quantifies the probability per unit time that the infection will be transmitted from an infective individual to a susceptible one, $\gamma$ is the recovery (or healing) probability per unit time that an infective individual recovers and becomes susceptible again. To be more accurate, the infection rate studied here is a part of endogenous infection rate, which has the form of $\beta d$, where $d$ is the number of infected neighbors of the susceptible node[187,188]. Since $\beta d$ dependents on the structure of the network, $\beta$ is referred to as the topology dependent infection rate, and since recovery is a spontaneous action and the recovery probability is identical for all the infective nodes, $\gamma$ is considered to be network topology independent[187].

We consider a task to estimate the disease incidence, or the percentage of the infected nodes at each time step. A simple method is that, in each time, we randomly sample some nodes, query their states, and calculate the percentage of the infected nodes. This method provides an unbiased estimator to estimate the disease incidence. However, this method has two shortcomings: first, it loses information on graphs and cannot tell which nodes are infected; second, since it is a random approach, it needs a huge number of samples to ensure a reliable estimation.

We can model the states of nodes as a graph signal where 1 represents infective and 0 represents susceptible. When the topology dependent infection rate is high and the healing probability is low, the infection spreads locally; that is, nodes in the same community get infected in a same time and the corresponding graph signal is piecewise constant. We can use the sampling and recovery algorithm in Section 5.1 and then calculate the percentage of the infected nodes in the recovered graph signal. In this way, we can visualize the graph and tell which nodes may be infected because we recover the states of all the nodes; we also avoid the randomness effect because the algorithm is based on the experimentally designed sampling.

We simulate an epidemics process over the Minnesota road graph by using the SIS model. We set $\gamma$ be 0.1, and $\beta$ be 0.6. In the first day, we randomly select three nodes to be infected and diffuses it for 49 days. Figure 4.13 shows the states of nodes in the 10th day and the 20th day. We see that three small communities are infected in the 10th day; these communities are growing bigger in the 20th day. Since the healing probability is nonzero, a few susceptible nodes still exist within the communities.

We compare the results of two algorithms: one is based on random sampling followed with calculating the percentage of infection within the sampled nodes; the second is based on the local-set-based recovery algorithm following with calculating the percentage of infection within the recovered graph signal. The evaluation metric is the frequency that the result of the local-set-based




(a) 10th day.

(b) 20th day.

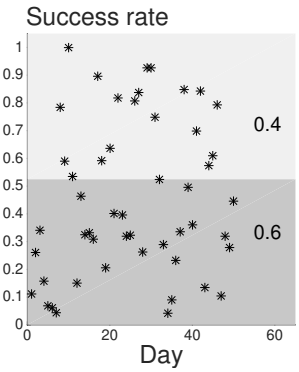
*Figure 4.13: Epidemics process over the Minnesota road graph. Yellow indicates infection and blue indicates susceptible.*

recovery algorithm is closer to the groundtruth, that is,

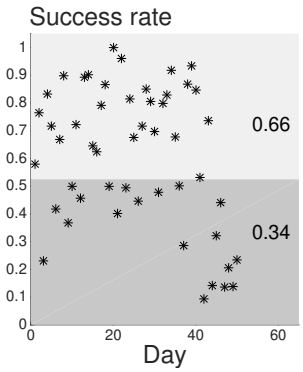
$$\text{Success rate} \;=\; \frac{1}{M}\sum_{i=1}^{M} \mathrm{I}(|\hat{x}^{(2)} - x_0| < |\hat{x}_i^{(1)} - x_0|),$$

where $x_0$ is the ground truth of the percentage of infection, $\hat{x}_i^{(1)}$ is the estimation of the random algorithm in the $i$th trials, $\hat{x}^{(2)}$ is the estimation of the local-set-based recovery algorithm, and $M$ is the total number of random trials; we choose $M = 1000$ here. The success rate measures the frequency with which the local-set-based recovery algorithm has a better performance. When the success rate is bigger than 0.5, the local-set-based recovery algorithm is better; When the success rate is smaller than 0.5, the random algorithm is better. Figure 4.14 shows the success rates given by the local-set-based recovery with three different graph partition algorithms. In each figure, the x-axis is the day (50 days in total); the y-axis is the success rate; the darker region means that local-set-based recovery algorithm fails and the lighter region means that local-set-based recovery algorithm successes; and the number shows the percentage of success or fail within 50 days. We see that given 100 samples, the local-set-based recovery algorithms are slightly worse than the random algorithm; given 1000 samples, the local-set-based recovery algorithms are slightly better than the random algorithm.

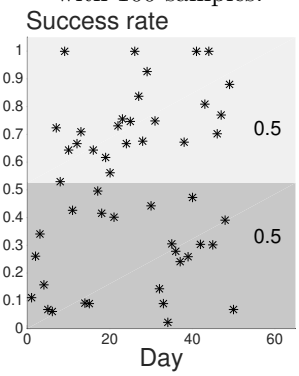
In Figure 4.15, we show the recovered states by the local-set-based recovery algorithm with 2-means partition on the 20th day. When having a few samples, the local-set-based recovery algorithms can recover the states in general, but cannot zoom into details and provide accurate estimations; when taking more samples, the local-set-based recovery algorithms recover the states better and provide better estimations. We see that the local-set-based recovery algorithm with 2-means partition provides both good estimation and good visualization.



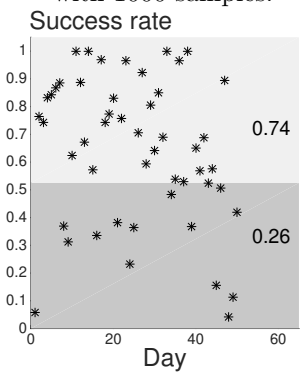

(a) Spectral clustering with 100 samples.



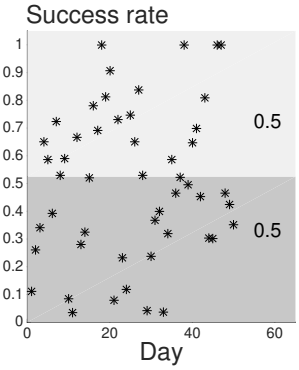

(b) Spectral clustering with 1000 samples.



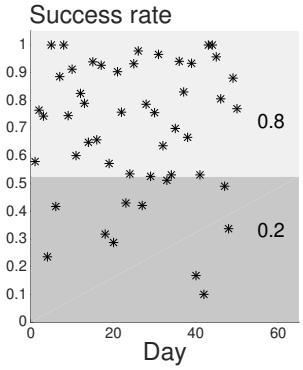

(c) Spanning tree with 100 samples.




(d) Spanning tree with 1000 samples.

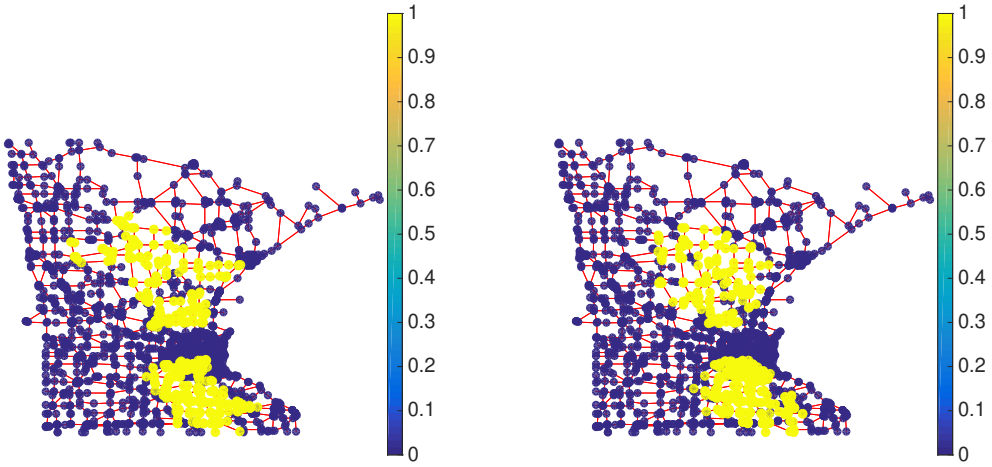



(e) 2-means with 100 samples.




(f) 2-means with 1000 samples.

*Figure 4.14: Success rate of estimating the disease incidence.*




(e) 2-means with 100 samples.

(f) 2-means with 1000 samples.

*Figure 4.15: Recovery of the node state on the* 20*th day.*

# 5

# Sampling Piecewise Smooth Graph Signals

The assumption that graph signals vary slowly or are smooth over the graph is a natural one to make. However, in social networks, within a given community or social circle, users' profiles tend to be homogeneous, while within a different social circle they will be different, yet still homogeneous. Such signals are characterized by large variation between regions or pieces and slow variation within pieces. In this work, we study the sampling and reconstruction of such piecewise-smooth graph signals that exhibit a spatially inhomogeneous level of smoothness over regions of the graph and have abrupt, localized discontinuities. This class of piecewise-smooth signals is complementary to the class of smooth graph signals that exhibit spatially homogeneous levels of smoothness over the graph. The sampling of such smooth signals has been well studied in previous work both within the field of graph signal processing as well as in the context of Laplacian regularization.

In the context of semi-supervised classification on graphs, each vertex represents one data point to which a label is associated and a graph can be formed by connecting vertices with weights corresponding to the affinity or distance between the data points in some feature space. It is then natural to assume that the *label signal* is piecewise-smooth on the graph. Since samples are often sparse or expensive, designing efficient sampling and reconstruction tools for semi-supervised classification and active learning is notably valuable.

In this chapter, we develop frameworks and algorithms for the sampling of piecewise-smooth graph signals. We study sampling piecewise smooth and particularly piecewise-constant signals on a graph. As before, we seek to develop algorithms and strategies for sampling piecewise-smooth signals. We study both passive and active sampling on graphs when sampling piecewise-smooth signals. Further, we seek to understand the influence of the underlying graph structure like in Chapter 3.

## 5.1 Sampling and Recovery Using Haar Wavelets

Here we sample and recover using the Haar graph wavelet introduced in the previous chapter. The goal of sampling and recovery is to collect a few samples from a graph signal, and then to recover the original graph signal from those samples either exactly or approximately.

### Algorithm

We consider the following recovery algorithm based on the multiresolution local sets. Let $m$ be the number of samples. We use the multiresolution decomposition of the local sets as shown in Section 4.6.1. Instead of obtaining a full decomposition tree, we partition the local sets until we obtain $m$ leaf local sets. Those local sets may not be in the same decomposition level , but their union still covers the entire space. For the local sets in the same level of the decomposition tree, we first partition the one that has the largest number of nodes. For each leaf local set, we choose a center that has the minimum summation of the geodesic distances to all the other nodes in the leaf local set. We use those centers for the $m$ leaf local sets as the sampled set. Let $\mathbf{x} \in \mathbb{R}^N$ be a piecewise-constant graph signal, $\mathcal{M} = (\mathcal{M}_1, \cdots, \mathcal{M}_m)$ be the designed sampled set, with each sampled node $\mathcal{M}_j$ be the center of the $j$th leaf local set $S_j$. The recovered graph signal is

$$\mathbf{x}' = \sum_{j=1}^{m} x_{\mathcal{M}_j} \mathbf{1}_{S_j}.$$

We obtain a simple upper bound for the recovery error of this algorithm.

**Theorem 31.** Let the original graph signal $\mathbf{x} \in \mathrm{PC}_G(K)$. The recovery error is bounded as

$$\sum_{i=1}^{N} \mathrm{I}(x_i \neq x_i') \leq K \max_{j=1,\cdots,m} |S_j|,$$

where $\mathrm{I}(\cdot)$ is the indicator function.

*Proof.* The error happens only when there exists at least one inconsistent edge in a community. Since there are $K$ inconsistent edges, we make errors in at most $K$ communities. The worst case is that each error is made in the one of the largest $K$ communities. □

Theorem 31 shows that the size of the largest community influences the recovery error. When we use the even partition, the size of the largest local set is minimized, which minimizes the upper bound. Similar to Theorem 29, Theorem 31 also shows the importance of the even partition again. This algorithm studies the graph structure before taking samples, which belongs to the experimentally designed sampling. In the classical regression for piecewise-constant functions, it is known that experimentally designed sampling has the same performance with random sampling asymptotically[72]. When we restrict the problem setting to sample only a few nodes from a finite graph, however, random sampling can lead to the uneven partition where some communities are much larger than the others. As a deterministic approach, the experimentally designed sampling minimizes the error bound and is better than random sampling when the sample size is small.

We also consider two other recovery algorithms, including trend filtering on graphs and harmonic functions. For trend filtering on graphs, we consider

$$\mathbf{x}' = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^n} \frac{1}{2}\|(\mathbf{y}-\boldsymbol{\beta})_{\mathcal{M}}\|_2^2 + \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}\|_1,$$

where $\mathcal{M}$ is the sampling node set obtained by random sampling. We want to push the recovered graph signal to be close to the original one at the sampled nodes and to be piecewise constant. For harmonic functions, we consider

$$\mathbf{x}' = \quad \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^n} \tfrac{1}{2}\|(\mathbf{y}-\boldsymbol{\beta})_{\mathcal{M}}\|_2^2 + \mu\,\|\Delta\boldsymbol{\beta}\|_2^2\,,$$

where $\|\Delta\boldsymbol{\beta}\|_2^2 = \boldsymbol{\beta}^T\,\mathrm{L}\,\boldsymbol{\beta}$ and L is the graph Laplacian matrix. Harmonic functions are proposed to recover a smooth graph signal which can be treated as an approximation of a piecewise-constant graph signal. When we obtain the solution, we assign each coefficient to its closest constant in $\mathbf{x}$.

### Experiments

We test the four representations on two datasets, including the Minnesota road graph[185] and the U.S city graph[74].

For the Minnesota road graph, we simulate a piecewise-constant graph signal by randomly picking 5 nodes as community centers and assigning each other node a community label based on the geodesic distance. We assign a random integer to each community. We still use the simulated graph signal in Figure 4.10(a).

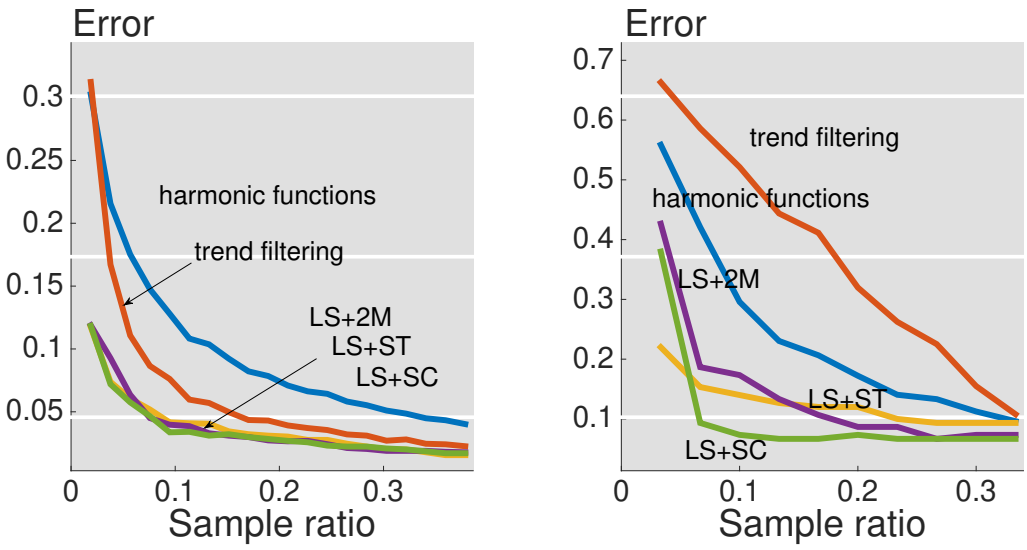
The evaluation metric of the recovery error is the percentage of mislabel, that is,

$$\text{Error} = \frac{\sum_{i=1}^{N}\mathrm{I}(x_i \neq x_i')}{N},$$

where $\mathbf{x}'$ is the recovered signal and $\mathbf{x}$ is the ground-truth signal. Figure 5.1(a) shows the recovery errors given by three algorithms. The x-axis is the ratio between the number of samples and the total number of nodes and the y-axis is the recovery error, where lower means better. Since harmonic functions and trend filtering are based on random sampling, the results are averaged over 50 runs. SC indicates spectral clustering, ST indicates spanning tree and 2M indicates 2-means. We see that the local-set-based recovery algorithms are better than harmonic functions and trend filtering, especially when the sample ratio is small.

For the U.S city graph, we use the same piecewise-constant graph signal (4.38) as the ground truth. The evaluation metric of the recovery error is the percentage of mislabel. Figure 5.1(b) shows the recovery errors given by three recovery strategies with two different sampling strategies. Similarly to the recovery of the Minnesota road graph, we see that the local-set-based recovery algorithms are better than harmonic functions and trend filtering, especially when the sample ratio is small.

To summarize the task of sampling and recovery, the proposed center-assign algorithm is simple and useful in the recovery. The experimentally designed sampling based on local sets tries to minimizes the upper bound in Theorem 31 and make each local set have similar sizes. It provides a deterministic approach to choose sampled nodes; it works better than random sampling when




(a) Minnesota road graph.

(b) U.S. city graph

*Figure 5.1: Comparison of recovery errors. LS+SC represents the local-set-based recovery algorithm with spectral clustering partition; LS+ST represents the local-set-based recovery algorithm with spanning tree partition; LS+2M represents the local-set-based recovery algorithm with 2-means partition.*

the sample ratio is small and has a similar asymptotic performance to random sampling.

## 5.2 Sampling Piecewise Smooth Signals on Graphs via Graph trend Filtering

The graph trend filtering (GTF) framework[30], which applies total variation denoising on graphs[114], is a particularly flexible and attractive approach to process piecewise-smooth graph signals that is based on minimizing the $\ell_1$ norm of discrete graph differences. In this work, we present an extension to the GTF framework under the sampling setting, that is, where we only partially observe the signal.

Most sampling strategies fall under the umbrellas of either (1) passive sampling where there is no feedback and we simply sample the space without any knowledge of key signal characteristics, or (2) active sampling where we can incorporate feedback in a sequential process. Unlike sampling smooth signals that have no discontinuities, the localized nature of the discontinuities in piecewise-smooth signals make the detection of these discontinuities inherently decoupled from the global or neighborhood features of the graph signal. It then follows that the passive sampling of piecewise-smooth graph signals is a significantly harder or even futile task than the same for globally smooth signals. For the latter, it is often sufficient to sample such that we uniformly cover the space. Consequently, we propose studying the active sampling of piecewise-smooth signals by designing algorithms and strategies that incorporate feedback. Particularly, we develop active sampling methods that can capitalize on the localized nature of the boundary by focusing the sampling process in the estimated vicinity of the boundary.

### 5.2.1 Sampling

We consider the procedure of sampling and recovery as follows: we sample $M$ coefficients in a graph signal $\boldsymbol{\beta} \in \mathbb{R}^N$ with Gaussian noise to produce a noisy sampled signal $\mathbf{y} \in \mathbb{R}^M (M < N)$, that is,

$$\mathbf{y} \;=\; \boldsymbol{\Psi}\boldsymbol{\beta} + \epsilon \;\equiv\; \boldsymbol{\beta}_{\mathcal{M}} + \epsilon, \tag{5.1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 \, \mathrm{I}_{M \times M})$, and $\mathcal{M} = (\mathcal{M}_1, \cdots, \mathcal{M}_M)$ denotes the sampling set where $\mathcal{M}_i \in \{1, \cdots, N\}$. The sampling operator $\boldsymbol{\Psi}$ is a linear mapping from $\mathbb{R}^N$ to $\mathbb{R}^M$, defined as

$$\boldsymbol{\Psi}_{i,j} = \begin{cases} 1, & j = \mathcal{M}_i; \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

We then reconstruct $\boldsymbol{\beta}$ from $\mathbf{y}$ to get $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^N$.

Passive sampling refers to the setting where we are constrained to strategies that are blind to any samples of the signal. That is, we design strategies by considering only the underlying graph structure and any modeling assumptions we have made. In contrast, active or adaptive sampling strategies are able to choose samples in an online fashion by allowing feedback: the decision of where to sample next depends on all the observations made previously. While it's obvious that good active sampling strategies should never perform worse than passive sampling strategies, we aim to develop active sampling strategies that are able to achieve substantial gains in performance.

Under passive sampling, we can consider two different sampling settings: *random sampling* where the sample indices are chosen from from $\{1, \cdots, N\}$ independently and uniformly randomly;

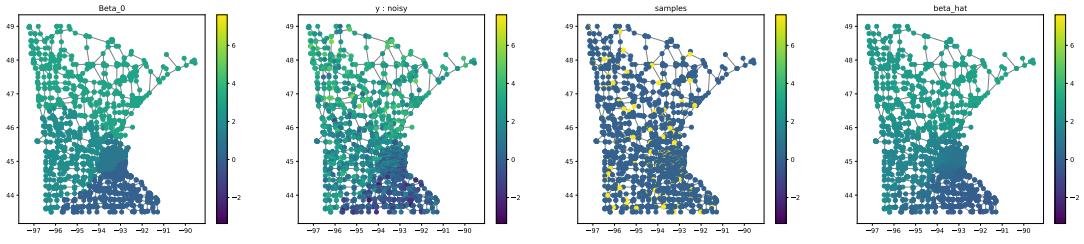



*Figure 5.2: Example of sampling a piecewise-constant (k=0) graph signal with 4 pieces on the Minnesota road graph with a random 5% of samples. From left to right, we have the true signal, the noisy signal, the location of the samples, and the reconstructed signal. Noisy input signal SNR = 5dB, Reconstructed signal SNR = 12.8dB*

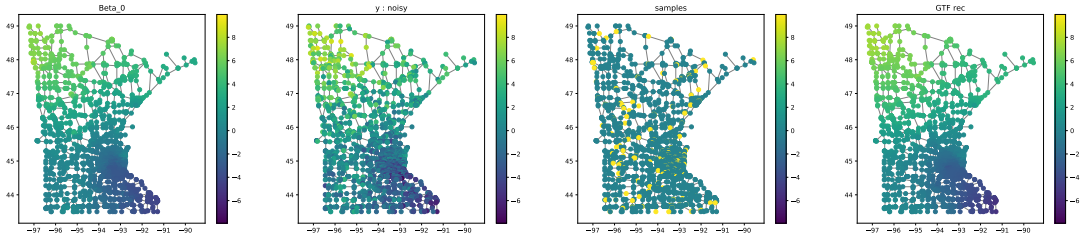



*Figure 5.3: Example of sampling a piecewise-linear (k=1) graph signal on the Minnesota road graph with a random 5% of samples. From left to right, we have the true signal, the noisy signal, the location of the samples, and the reconstructed signal. Noisy input signal SNR = 5dB, Reconstructed signal SNR = 14.5dB*

and *experimentally designed sampling* where the sample indices can be chosen beforehand based on the graph structure. Since we do not a-priori make any assumptions or have any information on the location of the boundary or discontinuities of the piecewise-smooth graph signal, we can show that experimentally designed sampling does not outperform random sampling and in fact, can often be detrimental. In other words, these discontinuities are fundamentally dissociated from samples outside their locations on the graph unlike globally smooth signals where key characteristics of the signal are *spread* out over local neighborhoods and consequently some nodes can be more informative than others. Consequently, we only consider uniform random sampling for passive sampling. We note that previous work that has studied the fundamental limits of passive and active sampling on graphs for globally smooth signals, has shown that active sampling *does not* fundamentally outperform passive sampling. However, experimentally designed sampling outperforms random sampling for irregular graphs where some nodes can be more informative than others.

## 5.3 Sampling via Graph Trend Filtering

Graph trend filtering (GTF)[30] is a flexible framework for estimation on graphs that is adaptive to inhomogeneity in the level of smoothness and localized characteristics of an observed signal across

nodes. The $k$th order GTF estimate is defined as:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}\|_1, \tag{5.3}$$

which can be regarded as applying total variation or fused lasso with the graph difference operator $\boldsymbol{\Delta}^{(k+1)}$ [114,32]. The sparsity-promoting properties of the $\ell_1$ norm have been well-studied [153]. Consequently, applying the $\ell_1$ penalty in GTF sets many of the graph differences to zero while keeping a small fraction of nonzero values. GTF is then *adaptive* over the graph; its estimate at a node adapts to the smoothness in its localized neighborhood.

Under the sampling and recovery framework, we propose solving following modified version of the GTF formulation GTF-S:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}\|_1, \tag{5.4}$$

**Remark 5.** Note that we can use mixed piecewise penalties to encourage different kinds of piecewise polynomial behavior by stacking the graph difference matrices since we can transform $\lambda\|\boldsymbol{\Delta}^{(l+1)}\|_1 + \gamma\|\boldsymbol{\Delta}^{(m+1)}\|_1$ as $\|\boldsymbol{\Delta}\|_1$ where

$$\boldsymbol{\Delta} = \begin{bmatrix} \lambda\boldsymbol{\Delta}^{(l+1)} \\ \hline \gamma\boldsymbol{\Delta}^{(m+1)} \end{bmatrix}$$

In the following exposition however, we only consider the basic graph difference operator for a given $k$.

We solve this GTF-S formulation in (5.4) via the alternating direction method of multipliers (ADMM) framework for solving separable optimization problems [104]. Via a change of variable defining $\boldsymbol{\eta} = \boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}$, we can write the transformed problem:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\eta}\|_1 \quad \text{s.t.} \quad \boldsymbol{\eta} = \boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}$$

and its corresponding Lagrangian as:

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\eta}, ) = \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\eta}\|_1 + \frac{\tau}{2}\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta} - \boldsymbol{\eta} + \|_2^2 \\ - \frac{\tau}{2}\|\|_2^2 \end{aligned} \tag{5.5}$$

where$\breve{}$ is the Lagrangian multiplier, and $\tau$ the parameter. Algorithm 7 shows the ADMM updates based on the Lagrangian in (5.5). For an appropriately chosen $\tau$, the algorithm converges in a fixed number of iterations. In Fig. 5.2 and Fig. 5.3, we illustrate with an example the sampling and recovery of a piecewise-constant and piecewise-linear graph signal on the Minnesota road graph [71] with the GTF-S framework.

### 5.3.1 Theoretical Analysis

We now present bounds on the error rate of the GTF-S fit $\|\boldsymbol{\Psi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_2$ that help elucidate the relationship between the sample complexity (number of samples $M$ needed for accurate reconstruction) with respect to structural properties of the graph and complexity of the boundary $\|\boldsymbol{\Delta}^{(k+1)}\beta\|_1$. For simplicity, let us assume the graph is fully connected, that is there is only 1 con-

**Algorithm 7** ADMM Optimization for GTF-S

**Inputs:** $\mathbf{y}, \mathbf{\Psi}, \mathbf{\Delta}^{(k+1)}$, and parameters $\lambda$, $\tau$
**Initialize:**
    $\mathcal{D} \leftarrow \mathbf{\Delta}^{(k+1)}$, $\boldsymbol{\eta} \leftarrow \mathcal{D}\boldsymbol{\beta}$, $\leftarrow \mathcal{D}\boldsymbol{\beta} - \boldsymbol{\eta}$,
    $\boldsymbol{\beta} \leftarrow \mathbf{y}$ or $\boldsymbol{\beta}_{init}$ if given.
**repeat**
    $\boldsymbol{\beta} \leftarrow (\mathbf{\Psi}^T\mathbf{\Psi} + \tau\mathcal{D}^T\mathcal{D})^{-1}(\tau\mathcal{D}^T(\boldsymbol{\eta}-) + \mathbf{\Psi}^T\mathbf{y})$
    **for** `i ← 1 to length(`$\mathcal{D}\boldsymbol{\beta}$`)` **do**
        $\eta_i \leftarrow \text{prox}_\rho([\mathcal{D}\boldsymbol{\beta}]_i + u_i; \lambda/\tau)$
                                ▷ $\text{prox}_\rho(t;\alpha)$ = soft-thresholding operator on $t$ with $\alpha\rho$
    **end for**
    $\leftarrow + \mathcal{D}\boldsymbol{\beta} - \boldsymbol{\eta}$
**until** termination

nected component, the dimension of the null space of $\mathbf{\Delta}^{(k+1)}$. Note that if there were multiple connected components, the problem becomes fully separable over each connected component.

**Proposition 4.** On a fully connected graph, we have $\mathbf{\Delta}^{(k+1)\dagger}\mathbf{\Delta}^{(k+1)} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{J}$

**Definition 18** (Compatibility factor)**.** Let $\mathbf{\Delta}^{(k+1)}$ be fixed. The compatibility factor $\kappa_T$ of a nonempty set $T \subseteq \{1, 2, \ldots, r\}$ is defined as

$$\kappa_T(\mathbf{\Delta}^{(k+1)}) = \inf_{\boldsymbol{\beta}\in\mathbb{R}^n} \left\{ \frac{\sqrt{|T|} \cdot \|\boldsymbol{\beta}\|_2}{\|(\mathbf{\Delta}^{(k+1)}\boldsymbol{\beta})_T\|_1} \right\}.$$

Below, we present a simple lower bound on $\kappa_T$:

**Proposition 5** (Bound on $\kappa_T$)**.** Let $d_{\max}$ be the maximal degree of the graph, then $\kappa_T$ is bounded for any $T$ and $d$ as

$$\kappa_T(\mathbf{\Delta}^{(k+1)}) \geq \frac{1}{(2d_{\max})^{\frac{k+1}{2}}}.$$

We have that,

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \mathbf{\Psi}\boldsymbol{\beta}\|_2^2 + \lambda\|\mathbf{\Delta}^{(k+1)}\boldsymbol{\beta}\|_1$$

As a result, by optimality we can write,

$$\|\mathbf{y} - \mathbf{\Psi}\hat{\boldsymbol{\beta}}\|_2^2 + \lambda\|\mathbf{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1 \leq \|\mathbf{y} - \mathbf{\Psi}\boldsymbol{\beta}^\star\|_2^2 + \lambda\|\mathbf{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 \tag{5.6}$$

By rearranging,

$$\|\mathbf{\Psi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_2^2 \leq 2(\mathbf{y} - \Psi\boldsymbol{\beta}^\star)^T\Psi(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star) + \lambda\|\mathbf{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 - \lambda\|\mathbf{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1 \tag{5.7}$$

Denoting $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star$ as $\boldsymbol{\gamma}$ for simplicity, we have that

$$\begin{aligned}
\|\Psi\boldsymbol{\gamma}\|_2^2 &\leq 2\boldsymbol{\epsilon}^T\Psi\boldsymbol{\gamma} + \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 - \lambda\|\boldsymbol{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1 \\
&\leq 2\boldsymbol{\epsilon}^T\Psi(\boldsymbol{\Delta}^{(k+1)\dagger}\boldsymbol{\Delta}^{(k+1)} + \frac{1}{n}\boldsymbol{J})\boldsymbol{\gamma} + \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 - \lambda\|\boldsymbol{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1 \\
&\leq \frac{2}{n}\boldsymbol{\epsilon}^T\Psi\boldsymbol{J}\boldsymbol{\gamma} + 2\boldsymbol{\epsilon}^T\Psi\boldsymbol{\Delta}^{(k+1)\dagger}\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\gamma} + \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 - \lambda\|\boldsymbol{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1 \\
&\leq \frac{2}{n}\|\boldsymbol{\epsilon}^T\Psi\boldsymbol{J}\|_2\|\boldsymbol{\gamma}\|_2 + 2\|\boldsymbol{\epsilon}^T\Psi\boldsymbol{\Delta}^{(k+1)\dagger}\|_\infty\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\gamma}\|_1 + \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 - \lambda\|\boldsymbol{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1
\end{aligned}$$

We can maximally bound $\frac{2}{n}\|\boldsymbol{\epsilon}^T\Psi\boldsymbol{J}\|_2 \leq C$ with probability atleast $1-\delta$ such that

$$C = 2\sigma\sqrt{2\frac{M}{N}\log(\frac{2}{\delta})}$$

Further, setting $\lambda \geq 4\|\boldsymbol{\epsilon}^T\Psi\boldsymbol{\Delta}^{(k+1)\dagger}\|_\infty$, we have that

$$\|\Psi\boldsymbol{\gamma}\|_2^2 \leq C\|\boldsymbol{\gamma}\|_2 + \frac{\lambda}{2}(\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\gamma}\|_1 + 2\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 - 2\|\boldsymbol{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1)$$

For any subset $T \subseteq [r]$, we can write

$$\begin{aligned}
&\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\gamma}\|_1 + 2\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 - 2\|\boldsymbol{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1 \\
&= 3\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\gamma})_T\|_1 - \|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\gamma})_{T_c}\|_1 + 4\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1
\end{aligned}$$

Further, since $\|\Psi\boldsymbol{\gamma}\|_2^2 \geq 0$, $\boldsymbol{\gamma}$ lies in the cone

$$\mathcal{C} = \{\boldsymbol{t} : \|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{t})_{T_c}\|_1 \leq 3\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{t})_T\|_1 + 4\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1 + \frac{2C}{\lambda}\|\boldsymbol{\gamma}\|_2\} \tag{5.8}$$

By the definition of $\kappa_T$, we have that for any T,

$$\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\gamma})_T\|_1 \leq \frac{\sqrt{|T|}\|\boldsymbol{\gamma}\|_2}{\kappa_T}$$

As a result, we have that

$$\begin{aligned}
\|\Psi\boldsymbol{\gamma}\|_2^2 &\leq C\|\boldsymbol{\gamma}\|_2 + \frac{\lambda}{2}(\frac{3\sqrt{|T|}\|\boldsymbol{\gamma}\|_2}{\kappa_T} - \|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\gamma})_{T_c} + 4\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1) \\
&\leq (C + \frac{3\lambda\sqrt{|T|}}{2\kappa_T})\|\boldsymbol{\gamma}\|_2 + 2\lambda\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1
\end{aligned}$$

**Definition 19.** We now introduce a restricted isometry property for all $\boldsymbol{\gamma} \in \mathcal{C}$ such that

$$\|\Psi\boldsymbol{\gamma}\|_2^2 \geq \frac{\|\boldsymbol{\gamma}\|_2^2}{\Phi}$$

where $\mathcal{C} = \{\boldsymbol{t} : \|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{t})_{T_c}\|_1 \leq 3\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{t})_T\|_1 + 4\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1 + \frac{2C}{\lambda}\|\boldsymbol{\gamma}\|_2$

such that

$$\|\boldsymbol{\gamma}\|_2^2 \leq \Phi(C + \frac{3\lambda\sqrt{|T|}}{2\kappa_T})\|\boldsymbol{\gamma}\|_2 + 2\Phi\lambda\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1 \tag{5.9}$$

If $x^2 - bx - c \leq 0$, then $x^2 \leq 4\max(b^2, |c|) \leq 4(b^2 + c)$, for $b \geq 0$ As a result, we have that

**Theorem 32** (Main Error Bound)**.**

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 \leq 4(\Phi(C + \frac{3\lambda\sqrt{|T|}}{2\kappa_T}))^2 + 8\Phi\lambda\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1 \tag{5.10}$$

Since this holds for all sets $T$,

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 \leq \min_T \Big[4(\Phi(C + \frac{3\lambda\sqrt{|T|}}{2\kappa_T}))^2 + 8\Phi\lambda\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1\Big] \tag{5.11}$$

If we take $T = supp(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)$, $\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1 = 0$, we have that

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 \leq 4\Phi^2(C + \frac{3\lambda\sqrt{|T|}}{2\kappa_T})^2 \tag{5.12}$$

while if $T = \phi$, $\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)_{T_c}\|_1 = \|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1$, and

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 \leq 4\Phi^2C^2 + 8\Phi\lambda\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star)\|_1 \tag{5.13}$$

**Theorem 33** (Weak Consistency Error bound of the GTF-S minimizer)**.** Let $\widehat{\boldsymbol{\beta}}$ to be the minimizer of (5.4), $r$ be the number of rows of $\boldsymbol{\Delta}^{(k+1)}$, $\zeta$ be the maximum $\ell_2$ norm of the columns of $\boldsymbol{\Psi}\boldsymbol{\Delta}^{(k+1)\dagger}$. Set $\lambda = \sigma\zeta\sqrt{2\log(\frac{r}{\delta})}$, then with probability at least $1 - 2\delta$, we have:

$$\|\boldsymbol{\Psi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_2^2 \leq \sigma^2\Big(1 + 2\sqrt{2\log(\frac{1}{\delta})}\Big) + 4\sigma\zeta\sqrt{2\log(\frac{r}{\delta})}\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1$$

*Proof.* Setting $\lambda$, we have that

$$\begin{aligned}\|\Psi\boldsymbol{\gamma}\|_2^2 &\leq C\|\boldsymbol{\gamma}\|_2 + \lambda\|\boldsymbol{\Delta}^{(k+1)}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_1 + \lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1 - \lambda\|\boldsymbol{\Delta}^{(k+1)}\hat{\boldsymbol{\beta}}\|_1 \\ &\leq C\|\boldsymbol{\gamma}\|_2 + 2\lambda\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1\end{aligned}$$

□

Hence, the GTF-S fit is consistent if $\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta}^\star\|_1$ grows at a rate slower than $\frac{1}{\sigma\zeta\sqrt{\log(\frac{r}{\delta})}}$. From[155], we can show that $\zeta \leq 1/\lambda_{\min}(\boldsymbol{\Delta}^{(2)})^{\frac{k+1}{2}}$, where $\lambda_{\min}(\boldsymbol{\Delta}^{(2)})$ is the smallest *nonzero* eigenvalue of the graph Laplacian matrix $\boldsymbol{\Delta}^{(2)}$ and quantifies the algebraic connectivity of the graph. We note that this result is consistent with basic error rates for graph trend filtering[30] as $M \to N$.

## 5.4 Active Sampling for Piecewise Smooth Signals on Graphs

As alluded to earlier, the abrupt discontinuities are highly localized. That is, these discontinuities are fundamentally dissociated from samples outside their locations on the graph unlike globally smooth signals. This makes the task passive sampling significantly harder or perhaps even ineffective. Consequently, in the passive sampling setting, without prior knowledge on the location of the discontinuities, we do not expect experimentally designed sampling to perform better than uniform random sampling. Hence, in the first part, we propose understanding how uniform random

sampling differs from experimentally designed sampling for piecewise-smooth signals on graphs.

Previous work[189,73,190] in a similar vein has studied fundamental performance limits of active learning in for continuous functions the context of regression under noisy conditions. Significantly faster rates of convergence are generally achievable in cases involving functions whose complexity is highly concentrated in small regions of space. Further, for piecewise constant functions, active learning methods can capitalize on the highly localized nature of the boundary by focusing the sampling process in the estimated vicinity of the boundary. In this work, we ask similar questions for piecewise signals supported on irregular structures. As before, we seek to understand how the underlying graph structure influences both these limitations and the performance of these algorithms.

In the smooth signal setting, where the chosen basis is the graph Fourier transform, we know the support of the coefficients of the basis transform since smooth signals are approximately bandlimited. However, for piecewise smooth signals, without apriori knowledge on the locations of the discontinuities, the support of the wavelet coefficients are difficult to detect. This leads us to suspect that like for piecewise smooth functions, active sampling can achieve substantial gains over passive sampling since feedback can helps us localize the discontinuities.

We aim to develop an active learning framework for the piecewise constant class of graph signals. While, we would like to eventually develop optimal sequential sampling algorithms, we propose to initially explore the following two-step procedure for adaptively sampling a piecewise-constant graph signal. A simple scheme is the following two-step approach is similar in flavor to the approach studied by Willett et al[189] and is based in part on the tree-structured estimators for passive learning. In the first step, called the preview step, a rough estimator of the signal $\mathbf{x}$ or its discontinuties is constructed using half our sampling budget, distributed over the graph. In the second step, called the refinement step, we use our remaining sampling budget near the estimated locations of the boundaries in the previous steps to find the separating constant regions and recover the piecewise-constant signal $\mathbf{x}$.

### 5.4.1 Active Sampling

We expect localized behavior that may be hard to detect to hamper the performance of passive sampling strategies. Consequently, in this section, we seek to employ active sampling strategies when the signal exhibits inhomogeneous behavior over the graph and contains discontinuities as in the case of piecewise-smooth signals. This gain in performance can be measured both in terms of the error rates and the sample complexity required to achieve a particular guarantee on the error. In spirit, our work follows previous work that studied the capabilities of passive and active sampling for recovering non-smooth functions from samples; the difference is that we consider a discrete setting and deal with irregular structures. For a smooth function, it has been shown that active sampling, experimentally designed sampling and uniform sampling have the same performance from a statistical perspective. For brevity, we only consider the piecewise-constant ($k = 0$) setting here, however the ideas and strategies presented here can easily be extended to general $k$. Given a sampling budget of $M$ samples (assume for simplicity that $M$ is even), we employ a two-step approach based in part on the active sampling procedures discussed in[73].

1. In the first step, called the preview step, we randomly sample $M/2$ samples uniformly distributed over the graph with $\tilde{\Psi}$ and use the GTF-S estimator (5.4) of the signal to get the

rough estimate $\tilde{\boldsymbol{\beta}}$.

2. In the second step, called the refinement step, we select the remaining half of our budget $M/2$ samples, near the perceived locations of the boundaries estimated in the preview step. Particularly, in this step we define a probability distribution over the nodes such that $\pi_i \propto \sum_{j \in \mathcal{N}(i)} |\tilde{\boldsymbol{\beta}}_i - \tilde{\boldsymbol{\beta}}_j|$ where $\mathcal{N}(i)$ denotes the neighborhood of $i$, the nodes it shares an edge with. We sample $M/2$ nodes with replacement such that in each of the $M/2$ rounds, the probability of the $i$-th node being selected is proportional to $\pi_i$. Consequently, at the end of this process we can construct a randomized sampling set represented by $\widehat{\Psi}$ such that the samples are largely concentrated in the vicinity of the boundary or discontinuities. We then use the GTF-S estimator (5.4) with the full set of $M$ samples with sampling operator $\Psi^T = \left[\tilde{\Psi}^T | \widehat{\Psi}^T\right]$ to get our final estimate $\widehat{\boldsymbol{\beta}}$.

This prescribed strategy is a natural way to take advantage of the idea that estimating the signal near the boundary is key to obtaining better reconstruction performance. We consider the even split of the sampling budget between the preview and refinement step only for simplicity. In addition, instead of a two-step procedure, one can reprise this idea, performing multiple refinement steps where in each step we acquire a new estimate of the boundary. However, for simplicity, we only consider the two-step procedure here.

## 5.5 Numerical Experiments

In this section, we perform numerical experiments on the synthetic piecewise-constant and piecewise-linear graph signals on the Minnesota road network with $N = 2642$ nodes and $m = 3304$ edges illustrated in Figures 5.2 and 5.3. We construct the piecewise-constant graph signal with 4 pieces by randomly choosing the location of 4 seed nodes and connecting every node to the closest seed by shortest path distance. We construct the piecewise-linear signal by randomly choosing the location of 50 discontinuities and solving the Poisson equation $\boldsymbol{\Delta}^{(2)}\boldsymbol{\beta} = \boldsymbol{b}$ where the non-zero entries in sparse vector $\boldsymbol{b}$ correspond to the discontinuities. We tune the hyperparameters $\rho$ and $\tau$ in Algorithm 7 by grid-search for the below experiments.

### 5.5.1 Passive Sampling

In this section, we study the performance of our proposed algorithm for different sampling densities and noise settings. We inject white gaussian noise such that the noisy signal has a specified SNR (5dB,10dB,15dB) before uniformly randomly sampling the signal and recovering it with Algorithm 7. The results are illustrated in Figure 5.4.

We see that we can only accurately reconstruct at moderately higher sampling densities. Note that we can reconstruct a piecewise-linear signal with better accuracy than a piecewise-constant signal with the same sample budget since the piecewise-linear graph signal is more homogenous and its key characteristics are less localized.

### 5.5.2 Active Sampling

We repeat the same experiment as that in Section 5.5.1 for the 5dB input SNR setting but additionally employ the active sampling strategy described in Section 5.4.1. The results are illus-

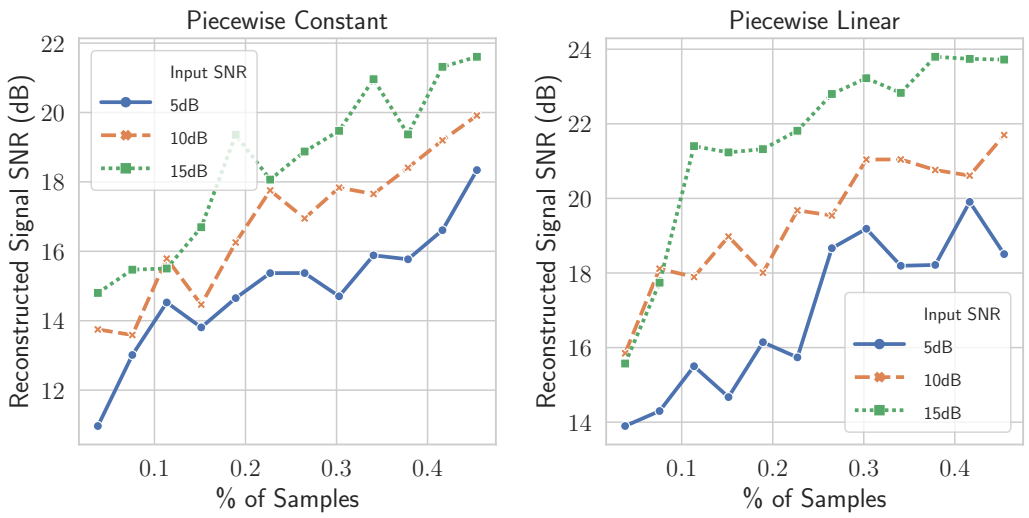



*Figure 5.4: Reconstructed signal SNR versus sampling density for different input SNR settings for both a piecewise-constant and a piecewise-linear graph signal*

trated in Figure 5.5. Note that for piecewise-linear signals, in the refinement step, we define $\pi_i \propto |\tilde{\boldsymbol{\beta}}_i - \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\tilde{\boldsymbol{\beta}}_j|$.

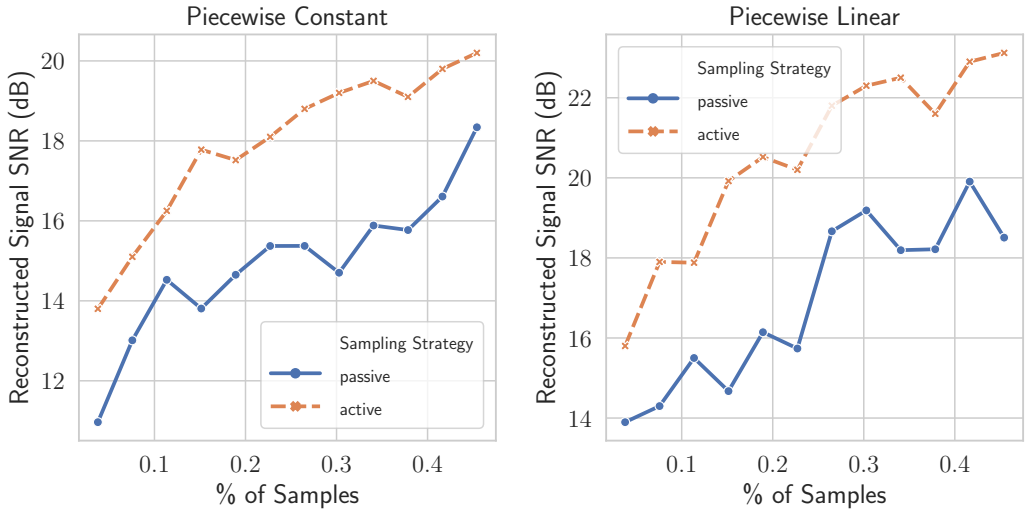



*Figure 5.5: Reconstructed signal SNR versus sampling density for passive and active sampling settings for both a piecewise-constant and a piecewise-linear graph signals*

We see that for both piecewise-constant and piecewise-linear graph signals, active sampling consistently and significantly outperforms passive sampling for the same sampling budget. This performance gain is particularly substantial at lower sampling densities.

## 5.6 Future Work and Extensions

We note that with approximate knowledge of the location of the discontinuities, we can estimate the support of the wavelet coefficients of the signal. We can then employ techniques developed for

experimental designed sampling[191] to design the samples in the second step similarly to the sampling scores derived in Section 3.4.4. It is then easy to see that the localization of the energy in the rows of the wavelet basis is key to understanding how the graph structure influences our performance. More specifically, we see how the uncertainty principle is insightful since wavelets at the same scale can have different degrees of localization. As a result, we expect that this interaction between the graph structure and the graph wavelet basis to play an important role in understanding these fundamental limits. If active sampling does indeed outperform passive sampling for this class of signals, then our aim is to show that the upper bound of this recovery algorithm is strictly smaller than the minimax lower bounds developed for passive sampling. As before, we seek to understand how the underlying graph structure influences both these limitations and the performance of these algorithms.

# 6

# Conclusion, Gaps and Future Work

## 6.1 Conclusion

In this thesis, we studied the tasks of (1) sampling signals and (2) reconstructing signals from noisy observations for each of the signal classes of (1) globally smooth and (2) piecewise smooth signals.

For smooth signals, we presented a sampling theory that guarantees perfect recovery, randomized sampling strategies and studied the fundamental limits of sampling in passive and active settings. That is, we showed that there are no fundamental gains from active sampling but there is. This is in contrast to sampling in a regular space. We showed how to efficiently sample on product graphs taking advantage of the structure. Further, we developed novel algorithms for smooth signal reconstruction on product graphs.

For piecewise-smooth signals that exhibit abrupt localized discontinuities over the graph, we presented the graph trend filtering estimator for signal reconstruction. Specifically, we showed that employing non-convex penalties has better reconstruction performance and support recovery. We also showed how to construct local-set multiresolution analysis and wavelet basis in order to approximate piecewise constant signals.

Moreover, we studied experimentally the limits of sampling piecewise-smooth graph signals under passive and active settings and explored designing adaptive sampling strategies that can outperform passive strategies.

Further, we outlined applications relevant to semi-supervised classification on graphs and problems in sensor networks that illustrate how the proposed tools and frameworks can provide solutions for many real-world applications of graph-structured data.

## 6.2 Future Work and Potential New Directions

### 6.2.1 Active Sampling

Studying the fundamental statistical limits of active learning versus passive learning on irregular spaces is still worth pursuing. Our largely experimental work on active sampling for piecewise smooth signals on graphs is a good first step before doing so. It also provides ample evidence that active sampling outperforms passive sampling in this setting.

For smooth signals, we have studied the fundamental minimax rates of active sampling and passive sampling, and shown that active sampling has no gain. However, active sampling still usually slightly out performs passive sampling in highly noisy settings. Hence, it is still worth developing active sampling algorithms even for smooth signals.

### 6.2.2 Localization and Network Motifs

Moving away from sampling and recovery, anomaly detection and event detection is a massive area of research. The uncertainty principle shows us the limiting effects of the graph Fourier transform in gleaning information since they possess simultaneous localization in vertex frequency space. While this area is still mostly pattern specific, whereby the pattern is swept across the graph, a more general solution is needed, perhaps an invariant translation operator.

Another alternative to using the graph Fourier transform which is also more well suited to analyze for example biological networks and social networks is network motifs. Network motifs are elementary subgraphs that repeat themselves in a complex network, which may reflect local, functional properties.

### 6.2.3 Graph Neural Networks and Graph Signal Processing

Recently developed sampling and recovery strategies have been based on two main frameworks: graph signal processing (GSP) and graph neural networks (GNNs). GSP provides a mathematically rigorous framework to analyze graph signals by generalizing the classical signal processing toolbox to the graph domain[6,172]. On the other hand, GNNs provide a unifying end-to-end learning framework by extending deep neural network techniques to the graph domain[192,193]. From the GSP perspective, the sampling and recovery operators $\Psi, \Phi$ are linear mappings, and the graph signal model is explicitly defined in advance. On the other hand, in the case of GNNs, the sampling and recovery operators $\Psi, \Phi$ could be nonlinear, and the graph signal model is implicitly learnt during the training process. GNNs have achieved impressive progress; however, a principled and mathematically rigorous approach to understand and design GNNs is greatly needed. GSP could potentially complement GNNs along this direction, which has not been fully explored yet.

Therefore, there is a niche for both signal processing and machine learning communities to collaborate and solve fundamental challenges on sampling and recovery of graph signals. This would not only benefit the development of both GSP and GNNs, but also a variety of applications, such as social network analysis, 3D point cloud processing, urban data analysis, sensor network localization, and many others.

However, we want to promote a combination between GSP-based algorithms and GNN-based algorithms to accelerate the research progress of sampling and recovery of graph signals. As emerg-

ing tools, GNNs enable an end-to-end learning framework to handle graph-related problems. With a huge amount of training data, GNNs would potentially provide powerful empirical performances; however, it is still an open issue to understand and design the architectures of GNNs. GSP could potentially guide the architecture design of GNNs by providing some theoretical insights. At the same time, GNNs could inspire more mathematically rigorous building blocks developed by GSP via a large amount of engineering attempts.

# A

## A.1 Proofs from Chapter 3

### A.1.1 Fano's Method

We aim to construct a typical set of vectors in $\mathcal{F}$, and use the Fano's method. Let $\mathcal{X}$ be a pruned hypercube and let

$$\mathcal{F}' = \{\mathbf{x}^{(\mathbf{w})} = \mathbf{V}\,\widehat{\mathbf{x}} \odot \mathbf{w} = \sum_{k=\kappa_0}^{2\kappa_0-1} w_k \psi_k, \quad \mathbf{w} \in \mathcal{X}\},$$

where $\kappa_0$ is no smaller than the bandwidth $K$,

$$\psi_k = \widehat{x}_k \mathbf{v}_k = (\pm)^k \sqrt{\frac{c\mu \left\|\mathbf{x}\right\|_2^2}{\kappa_0(1+k^{2\beta})}}\mathbf{v}_k,$$

and $0 < c < 1$. It is easy to check that $\mathcal{F}' \subseteq \mathcal{F}$. Let $d(\mathbf{x}, \mathbf{y}) = \left\|\mathbf{x} - \mathbf{y}\right\|_2$, we have

$$\begin{aligned}
d^2(\mathbf{x}^{(\mathbf{w})}, \mathbf{x}^{()}) &= \left\|\mathbf{V}\,\widehat{\mathbf{x}} \odot (\mathbf{w}-)^2\right\|_2^2 \\
&= \alpha_2 \sum_{k=\kappa_0}^{2\kappa_0-1} (w_k - u_k)^2 \widehat{x}_k^2 \\
&= \alpha_2 \sum_{k=\kappa_0}^{2\kappa_0-1} (w_k - u_k)^2 \frac{c\mu \left\|\mathbf{x}\right\|_2^2}{\kappa_0(1+k^{2\beta})} \\
&\overset{(a)}{\geq} \alpha_2 \sum_{k=\kappa_0}^{2\kappa_0-1} (w_k - u_k)^2 \cdot \frac{c\mu \left\|\mathbf{x}\right\|_2^2}{\kappa_0(1+(2\kappa_0)^{2\beta})} \\
&\overset{(b)}{\geq} \frac{\alpha_2\kappa_0}{8} \cdot \frac{c\mu \left\|\mathbf{x}\right\|_2^2}{\kappa_0(1+(2\kappa_0)^{2\beta})} \;\geq\; c_1 \mu \kappa_0^{-2\beta} \left\|\mathbf{x}\right\|_2^2,
\end{aligned}$$

where $(a)$ follows from $k \leq 2K$, and $(b)$ from the Varshamov-Gilbert lemma. To use the Fanno's method, we need to bound the KL divergence,

$$\begin{aligned}
KL(p_{\mathbf{w}}, p_{\mathbf{w}_0}|\mathcal{M}) &= \sum_{i\in\mathcal{M}} \mathbb{E}_{\mathbf{w}}\left[\log\frac{p(y_i - x_i^{(\mathbf{w})})}{p(y_i - x_i^{(\mathbf{w}_0)})}\right] \\
&\leq \sum_{i\in\mathcal{M}}\left[\frac{1}{2\sigma^2}(x_i^{(\mathbf{w})} - x_i^{(\mathbf{w}_0)})^2\right] = \sum_{i\in\mathcal{M}}\left[\frac{1}{2\sigma^2}\left(\mathbf{v}_i^*(\widehat{\mathbf{x}}\odot\mathbf{w})\right)^2\right] \\
&= \frac{1}{2\sigma^2}\sum_{i\in\mathcal{M}}\left(\sum_{k=\kappa_0}^{2\kappa_0-1}\widehat{x}_k w_k \mathbf{V}_{ik}\right)^2 \\
&= \frac{1}{2\sigma^2}\sum_{i\in\mathcal{M}}\left(\sum_{k=\kappa_0}^{2\kappa_0-1}(\widehat{x}_k w_k \mathbf{V}_{ik})^2 + \sum_{k,k'=\kappa_0,k\neq k'}^{2\kappa_0-1}\widehat{x}_k\widehat{x}_{k'} w_k w_{k'}\mathbf{V}_{ik}\mathbf{V}_{ik'}\right) \\
&\leq \frac{1}{2\sigma^2}\sum_{i\in\mathcal{M}}\left(\sum_{k=\kappa_0}^{2\kappa_0-1}\frac{c\mu\|\mathbf{x}\|_2^2 w_k^2\mathbf{V}_{ik}^2}{2\kappa_0(1+k^{2\beta})} + \sum_{k,k'=\kappa_0,k\neq k'}^{2\kappa_0-1}(-1)^{k+k'}\frac{c\mu\|\mathbf{x}\|_2^2 w_k w_{k'}\mathbf{V}_{ik}\mathbf{V}_{ik'}}{\kappa_0\sqrt{1+k^{2\beta}}\sqrt{1+k'^{2\beta}}}\right) \\
&\leq \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2}\sum_{i\in\mathcal{M}}\sum_{k=2\kappa_0}^{2\kappa_0-1}\frac{\mathbf{V}_{ik}^2}{\kappa_0(1+\kappa_0^{2\beta})} + \delta \\
&\asymp \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2}\kappa_0^{-(2\beta+1)}\sum_{i\in\mathcal{M}}\sum_{k=\kappa_0}^{2\kappa_0-1}\mathbf{V}_{ik}^2
\end{aligned}$$

where $\delta = \sum_{k,k'=\kappa_0,k\neq k'}^{2\kappa_0-1}(-1)^{k+k'}\frac{c\mu\|\mathbf{x}\|_2^2 w_k w_{k'}\mathbf{V}_{ik}\mathbf{V}_{ik'}}{\kappa_0\sqrt{1+k^{2\beta}}\sqrt{1+k'^{2\beta}}}$ has a lower order because of the cross signs. For random sampling, the sampling set $\mathcal{M}$ is chosen randomly, thus, we have

$$\begin{aligned}
KL(p_{\mathbf{w}}, p_{\mathbf{w}_0}) &= \mathbb{E}_{\mathcal{M}}\left[KL(p_{\mathbf{w}}, p_{\mathbf{w}_0}|\mathcal{M})\right] \\
&\leq \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2}\kappa_0^{-(2\beta+1)}\mathbb{E}_{\mathcal{M}}\left(\sum_{i\in\mathcal{M}}\sum_{k=\kappa_0}^{2\kappa_0-1}\mathbf{V}_{ik}^2\right) \\
&\overset{(a)}{=} \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2}K^{-(2\beta+1)}\mathbb{E}_i\left(|\mathcal{M}|\sum_{k=\kappa_0}^{2\kappa_0-1}\mathbf{V}_{ik}^2\right) \\
&= \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2}K^{-(2\beta+1)}|\mathcal{M}|\sum_{j=1}^{N}\sum_{k=\kappa_0}^{2\kappa_0-1}\mathbf{V}_{jk}^2\,\mathbb{P}(j=i) \\
&= \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2}\kappa_0^{-(2\beta+1)}\frac{|\mathcal{M}|}{N}\sum_{j=1}^{N}\sum_{k=\kappa_0}^{2\kappa_0-1}\mathbf{V}_{jk}^2 \leq \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2\kappa_0^{2\beta+1}N}\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2|\mathcal{M}|,
\end{aligned}$$

where $(a)$ follows from the independence of each sample, $\mathbb{P}(j=i)$ denotes the probability to sample the $i$th node that equals $j$, and $\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2 = \sum_{j=1}^{N}\sum_{k=\kappa_0}^{2\kappa_0-1}\mathbf{V}_{jk}^2$.

For experimentally designed sampling, we can choose the sampling set $\mathcal{M}$ to maximize $\sum_{i\in\mathcal{M}}\sum_{k=\kappa_0}^{2\kappa_0-1}\mathbf{V}_{ik}^2$, thus, we have

$$\begin{aligned}
KL(p_{\mathbf{w}}, p_{\mathbf{w}_0}) &\leq \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2}\kappa_0^{-(2\beta+1)}\max_{\mathcal{M}}\sum_{i\in\mathcal{M}}\sum_{k=\kappa_0}^{2\kappa_0-1}\mathbf{V}_{ik}^2 \\
&\leq \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2}\kappa_0^{-(2\beta+1)}\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{2,\infty}^2|\mathcal{M}|.
\end{aligned}$$

For active sampling, we cannot get more benefit from signal coefficients, so the KL divergence is the same of the experimentally designed sampling. By the Fanno's lemma, we finally have the following lower bounds for three sampling scenarios.

## Appendix B. Proof of Upper Bound for Random Sampling

We aim to bound the error by the bias-variance.

$$\begin{aligned}
\mathbb{E}\left\|\mathbf{x}^* - \mathbf{x}\right\|^2 &= \mathbb{E}\left\|\mathbf{x}^* - \mathbf{x}\right\|^2 \\
&= \mathbb{E}\left\|\mathbf{x}^* - \mathbb{E}\mathbf{x}^* + \mathbb{E}\mathbf{x}^* - \mathbf{x}\right\|^2 \\
&= \mathbb{E}\left(\left\|\mathbf{x}^* - \mathbb{E}\mathbf{x}^*\right\|^2 + \left\|\mathbb{E}\mathbf{x}^* - \mathbf{x}\right\|^2 + 2(\mathbf{x}^* - \mathbb{E}\mathbf{x}^*)^T(\mathbb{E}\mathbf{x}^* - \mathbf{x})\right) \\
&= \left\|\mathbb{E}\mathbf{x}^* - \mathbf{x}\right\|^2 + \mathbb{E}\left\|\mathbf{x}^* - \mathbb{E}\mathbf{x}^*\right\|^2.
\end{aligned}$$

The first term is bias and the second term is variance. For each element in the bias term, we have

$$\begin{aligned}
\mathbb{E}x_i^* - x_i &= \sum_{k<\kappa} \mathbf{V}_{ik} \frac{N}{|\mathcal{M}|} \mathbb{E}_{\mathcal{M},\epsilon}\left(\sum_{l\in\mathcal{M}} \mathbf{U}_{kl}(x_l + \epsilon_l)\right) - x_i \\
&\overset{(a)}{=} \sum_{k<\kappa} \mathbf{V}_{ik} \frac{N}{|\mathcal{M}|} |\mathcal{M}| \cdot \mathbb{E}_l\left(\mathbf{U}_{kl}\, x_l\right) - x_i \\
&= \sum_{k<\kappa} \mathbf{V}_{ik}\, N \sum_{j=1}^{N} \mathbf{U}_{kj}\, x_j \mathbb{P}(j = l) - x_i \\
&= \sum_{k<\kappa} \mathbf{V}_{ik} \sum_{j=1}^{N} \mathbf{U}_{kj}\, x_j - x_i \\
&= \sum_{k<\kappa} \mathbf{V}_{ik}\, \widehat{x}_k - x_i \;=\; -\sum_{k\geq\kappa} \mathbf{V}_{ik}\, \widehat{x}_k,
\end{aligned}$$

where $(a)$ follows from the independence of each sample, $\mathbb{P}(j = l)$ denotes the probability to sample the $l$th node that equals $j$. Since we sample randomly, $\mathbb{P}(j = l) = 1/N$. This leads to Lemma 3. We then bound the bias term based on Definition 10.

$$\begin{aligned}
\left\|\mathbb{E}\mathbf{x}^* - \mathbf{x}\right\|^2 &\overset{(a)}{\leq} \alpha_2 \sum_{k\geq\kappa} \widehat{x}_k^2 \\
&= \alpha_2 \sum_{k\geq\kappa} \frac{\widehat{x}_k^2(1 + k^{2\beta})}{1 + k^{2\beta}} \\
&\leq \frac{\alpha_2}{1 + \kappa^{2\beta}} \sum_{k\geq\kappa} \widehat{x}_k^2(1 + k^{2\beta}) \\
&\overset{(b)}{\leq} \frac{\alpha_2}{1 + \kappa^{2\beta}} \sum_{k\geq K} \widehat{x}_k^2(1 + k^{2\beta}) \\
&\overset{(c)}{\leq} \frac{\alpha_2}{1 + \kappa^{2\beta}} \mu \left\|\mathbf{x}\right\|_2^2
\end{aligned}$$

where $(a)$ follows from the assumption of $\mathbf{V}$ (3.1), $(b)$ from the assumption that $\kappa \geq K$, and $(c)$ from Definition 10.

We next bound the variance term by splitting into two parts, with and without noise. For each

element in the variance term, we have

$$\begin{aligned}
x_i^* - \mathbb{E}x_i^* &= \sum_{k<\kappa} \mathbf{V}_{ik} \left( \frac{N}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \mathbf{U}_{kl}\, y_l \right) - \sum_{k<\kappa} \mathbf{V}_{ik}\, \widehat{x}_k \\
&= \sum_{k<\kappa} \mathbf{V}_{ik} \left( \frac{N}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \mathbf{U}_{kl}\, \epsilon_l + \frac{N}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \mathbf{U}_{kl}\, x_l - \widehat{x}_k \right) \\
&= \frac{N}{|\mathcal{M}|} \sum_{k<\kappa} \mathbf{V}_{ik} \sum_{l \in \mathcal{M}} \mathbf{U}_{kl}\, \epsilon_l + \sum_{k<\kappa} \mathbf{V}_{ik} \left( \frac{N}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \mathbf{U}_{kl}\, x_l - \widehat{x}_k \right) \\
&= \Delta_i^{(1)} + \Delta_i^{(2)}.
\end{aligned}$$

To bound $\Delta_i^{(1)}$, we have

$$\begin{aligned}
\mathbb{E}||\Delta_i^{(1)}||^2 &= \mathbb{E}\left[ \frac{N^2}{|\mathcal{M}|^2} \left( \sum_{k<\kappa} \mathbf{V}_{ik} \sum_{l \in \mathcal{M}} \mathbf{U}_{kl}\, \epsilon_l \right) \left( \sum_{k'<\kappa} \mathbf{V}_{ik'} \sum_{l' \in \mathcal{M}} \mathbf{U}_{k'l'}\, \epsilon_{l'} \right) \right] \\
&= \mathbb{E}\left( \frac{N^2}{|\mathcal{M}|^2} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'} \sum_{l,l' \in \mathcal{M}} \mathbf{U}_{kl}\, \mathbf{U}_{k'l'}\, \epsilon_l \epsilon_{l'} \right) \\
&\overset{(a)}{=} \frac{N^2}{|\mathcal{M}|^2} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'}\, \mathbb{E}_{\mathcal{M}} \left( \sum_{l \in \mathcal{M}} \mathbf{U}_{kl}\, \mathbf{U}_{k'l}\, \mathbb{E}_{\epsilon}(\epsilon_l \epsilon_{l'}) \right) \\
&\overset{(b)}{=} \frac{N^2}{|\mathcal{M}|^2} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'} \cdot \mathbb{E}_l \left( |\mathcal{M}|\, \mathbf{U}_{kl}\, \mathbf{U}_{k'l}\, \sigma^2 \right) \\
&= \frac{N^2}{|\mathcal{M}|^2} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'}\, |\mathcal{M}| \sum_{j=1}^{N} \mathbf{U}_{kj}\, \mathbf{U}_{k'j}\, \sigma^2 \mathbb{P}(j = l) \\
&= \frac{\sigma^2 N}{|\mathcal{M}|} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'} \sum_{j=1}^{N} \mathbf{U}_{kj}\, \mathbf{U}_{k'j},
\end{aligned}$$

where $(a)$ follows from the independence of noise, $(b)$ from the independence of each sample.

To bound $\Delta_i^{(2)}$, we have

$$\begin{aligned}
\mathbb{E}\left\|\Delta_i^{(2)}\right\|^2 &= \mathbb{E}\left[\sum_{k<\kappa}\mathbf{V}_{ik}\left(\frac{N}{|\mathcal{M}|}\sum_{l\in\mathcal{M}}\mathbf{U}_{kl}\,x_l-\widehat{x}_k\right)\sum_{k'<\kappa}\mathbf{V}_{ik'}\left(\frac{N}{|\mathcal{M}|}\sum_{l'\in\mathcal{M}}\mathbf{U}_{k'l'}\,x_{l'}-\widehat{x}_{k'}\right)\right]\\
&= \mathbb{E}\left[\sum_{k,k'<K}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\left(\frac{N^2}{|\mathcal{M}|^2}\sum_{l,l'\in\mathcal{M}}\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}-\widehat{x}_k\widehat{x}_{k'}\right)\right]\\
&= \sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}[\frac{N^2}{|\mathcal{M}|^2}\mathbb{E}_{\mathcal{M}}\left(\sum_{l\neq l',l,l'\in\mathcal{M}}\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}\right)\\
&\qquad+\frac{N^2}{|\mathcal{M}|^2}\mathbb{E}_{\mathcal{M}}\left(\sum_{l=l',l,l'\in\mathcal{M}}\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}\right)-\widehat{x}_k\widehat{x}_{k'}]\\
&= \sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}[\frac{N^2}{|\mathcal{M}|^2}(|\mathcal{M}|^2-|\mathcal{M}|)\mathbb{E}_{l,l'}\left(\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}\right)+\frac{N^2}{|\mathcal{M}|}\mathbb{E}_l\left(\mathbf{U}_{kl}\,\mathbf{U}_{k'l}\,x_l^2\right)-\widehat{x}_k\widehat{x}_{k'}]\\
&= \sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\,[\frac{N^2}{|\mathcal{M}|^2}(|\mathcal{M}|^2-|\mathcal{M}|)\left(\sum_{j=1}^{N}\mathbf{U}_{kj}\,x_j\right)\left(\sum_{j'=1}^{N}\mathbf{U}_{kj'}\,x_{j'}\right)\mathbb{P}(j=l)\mathbb{P}(j'=l')\\
&\qquad+\frac{N^2}{|\mathcal{M}|}\sum_{j=1}^{N}\mathbf{U}_{kj}\,\mathbf{U}_{k'j}\,x_j^2\mathbb{P}(j=l)-\widehat{x}_k\widehat{x}_{k'}]\\
&= \frac{N}{|\mathcal{M}|}\sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\sum_{j=1}^{N}\mathbf{U}_{kj}\,\mathbf{U}_{k'j}\,x_j^2+\sum_{k,k'<K}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}(\frac{|\mathcal{M}|-1}{|\mathcal{M}|}-1)\widehat{x}_k\widehat{x}_{k'}\\
&\overset{(a)}{\leq} \frac{N}{|\mathcal{M}|}\sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\sum_{j=1}^{N}\mathbf{U}_{kj}\,\mathbf{U}_{k'j}\,x_j^2\\
&\leq \max_j x_j^2\frac{N}{|\mathcal{M}|}\sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\sum_{j=1}^{N}\mathbf{U}_{kj}\,\mathbf{U}_{k'j},
\end{aligned}$$

where $(a)$ follows from the fact that we ignore the second term.

We combine the bounds for both $\Delta_i^{(1)}$ and $\Delta_i^{(2)}$, and obtain the bounds for the variance term.

$$\begin{aligned}
\mathbb{E}\left\|\mathbf{x}^*-\mathbb{E}\mathbf{x}^*\right\|^2 &= \sum_{i=1}^{N}\mathbb{E}\left\|x_i^*-\mathbb{E}x_i^*\right\|\\
&= \sum_{i=1}^{N}\left(\mathbb{E}\left\|\Delta_i^{(1)}\right\|^2+\mathbb{E}\left\|\Delta_i^{(2)}\right\|^2\right)\\
&\leq (\max_j x_j^2+\sigma^2)\frac{N}{|\mathcal{M}|}\sum_{i=1}^{N}\sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\sum_{j=1}^{N}\mathbf{U}_{kj}\,\mathbf{U}_{k'j}\\
&= (\max_j x_j^2+\sigma^2)\frac{N}{|\mathcal{M}|}\left\|\mathbf{V}_{(\kappa)}\,\mathbf{U}_{(\kappa)}\right\|_F^2\\
&\leq (\max_j x_j^2+\sigma^2)\frac{\alpha_2 N}{|\mathcal{M}|}\left\|\mathbf{U}_{(\kappa)}\right\|_F^2.
\end{aligned}$$

Finally, we combine the bias term and the variance term, and obtain the bounds for the recovery error, as presented in Theorem 16.

## Appendix. C Proof of Upper Bound for Experimentally Designed Sampling

Similarly to Theorem 16, we split the error to the bias term and the variance term. For each element in the bias term, we have

$$\begin{aligned}
\mathbb{E}x_i^* - x_i &= \sum_{k<\kappa} \mathbf{V}_{ik} \frac{1}{|\mathcal{M}|} \mathbb{E}_{\mathcal{M},\epsilon} \left( \sum_{l \in \mathcal{M}} \frac{1}{w_l} \mathbf{U}_{kl} (x_l + \epsilon_l) \right) - x_i \\
&= \sum_{k<\kappa} \mathbf{V}_{ik} \frac{1}{|\mathcal{M}|} \mathbb{E}_l \left( |\mathcal{M}| \frac{1}{w_l} \mathbf{U}_{kl}\, x_l \right) - x_i \\
&\overset{(a)}{=} \sum_{k<\kappa} \mathbf{V}_{ik} \left( \sum_{j=1}^{N} \frac{1}{w_j} \mathbf{U}_{kj}\, x_j \mathbb{P}(j=l) \right) - x_i \\
&= \sum_{k<\kappa} \mathbf{V}_{ik} \sum_{j=1}^{N} \mathbf{U}_{kj}\, x_j - x_i \\
&= \sum_{k<\kappa} \mathbf{V}_{ik}\, \widehat{x}_k - x_i \;=\; -\sum_{k \geq \kappa} \mathbf{V}_{ik}\, \widehat{x}_k,
\end{aligned}$$

where $\mathbb{P}(j=l)$ denotes the probability to sample the $l$th node that equals $j$. Since we sample with a given weight, $\mathbb{P}(j=l) = 1/w_i$. This leads to Lemma 4, and the bias term is thus the same as presented in Theorem 16. We next bound the variance term by splitting into two parts, with and without noise. For each element in the variance term, we have

$$\begin{aligned}
x_i^* - \mathbb{E}x_i^* &= \sum_{k<\kappa} \mathbf{V}_{ik} \frac{1}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \frac{1}{w_l} \mathbf{U}_{kl}\, y_l - \sum_{k<\kappa} \mathbf{V}_{ik}\, \widehat{x}_k \\
&= \sum_{k<\kappa} \mathbf{V}_{ik} \left( \frac{1}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \frac{1}{w_l} \mathbf{U}_{kl}\, \epsilon_l + \frac{1}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \frac{1}{w_l} \mathbf{U}_{kl}\, x_l - \widehat{x}_k \right) \\
&= \frac{1}{|\mathcal{M}|} \sum_{k<\kappa} \mathbf{V}_{ik} \sum_{l \in \mathcal{M}} \frac{1}{w_l} \mathbf{U}_{kl}\, \epsilon_l + \sum_{k<\kappa} \mathbf{V}_{ik} \left( \frac{1}{|\mathcal{M}|} \sum_{l \in \mathcal{M}} \frac{1}{w_l} \mathbf{U}_{kl}\, x_l - \widehat{x}_k \right) \\
&= \Delta_i^{(1)} + \Delta_i^{(2)}.
\end{aligned}$$

To bound $\Delta_i^{(1)}$, we have

$$\begin{aligned}
\mathbb{E}||\Delta_i^{(1)}||^2 &= \mathbb{E} \left( \frac{1}{|\mathcal{M}|} \sum_{k<\kappa} \mathbf{V}_{ik} \sum_{l \in \mathcal{M}} \frac{1}{w_l} \mathbf{U}_{kl}\, \epsilon_l \right) \left( \frac{1}{|\mathcal{M}|} \sum_{k'<\kappa} \mathbf{V}_{ik'} \sum_{l' \in \mathcal{M}} \frac{1}{w_{l'}} \mathbf{U}_{k'l'}\, \epsilon_{l'} \right) \\
&= \mathbb{E} \left( \frac{1}{|\mathcal{M}|^2} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'} \sum_{l,l' \in \mathcal{M}} \frac{1}{w_l w_{l'}} \mathbf{U}_{kl}\, \mathbf{U}_{k'l'}\, \epsilon_l \epsilon_{l'} \right) \\
&= \frac{1}{|\mathcal{M}|^2} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'}\, \mathbb{E}_l \left( |\mathcal{M}| \frac{1}{w_l^2} \mathbf{U}_{kl}\, \mathbf{U}_{k'l}\, \epsilon_l^2 \right) \\
&= \frac{\sigma^2}{|\mathcal{M}|} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'} \sum_{j=1}^{N} \frac{1}{w_j^2} \mathbf{U}_{kj}\, \mathbf{U}_{k'j}\, \mathbb{P}(j=l) \\
&= \frac{\sigma^2}{|\mathcal{M}|} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'} \sum_{j=1}^{N} \frac{1}{w_j} \mathbf{U}_{kj}\, \mathbf{U}_{k'j}\,.
\end{aligned}$$

To bound $\Delta_i^{(2)}$, we have

$$
\begin{aligned}
\mathbb{E}\left\|\Delta_i^{(2)}\right\|^2 &= \mathbb{E}\left[\sum_{k<\kappa}\mathbf{V}_{ik}\left(\frac{1}{|\mathcal{M}|}\sum_{l\in\mathcal{M}}\frac{1}{w_l}\,\mathbf{U}_{kl}\,x_l-\widehat{x}_k\right)\sum_{k'<\kappa}\mathbf{V}_{ik'}\left(\frac{1}{|\mathcal{M}|}\sum_{l'\in\mathcal{M}}\frac{1}{w_{l'}}\,\mathbf{U}_{k'l'}\,x_{l'}-\widehat{x}_{k'}\right)\right]\\
&= \mathbb{E}\left[\sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\left(\frac{1}{|\mathcal{M}|^2}\sum_{l,l'\in\mathcal{M}}\frac{1}{w_l w_{l'}}\,\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}-\widehat{x}_k\widehat{x}_{k'}\right)\right]\\
&= \sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\Big[\frac{1}{|\mathcal{M}|^2}\mathbb{E}_{\mathcal{M}}\left(\sum_{l\neq l'}\frac{1}{w_l w_{l'}}\,\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}\right)\\
&\qquad+\frac{1}{|\mathcal{M}|^2}\mathbb{E}_{\mathcal{M}}\left(\sum_{l=l'}\frac{1}{w_l w_{l'}}\,\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}\right)-\widehat{x}_k\widehat{x}_{k'}\Big]\\
&= \sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\Big[\frac{1}{|\mathcal{M}|^2}\mathbb{E}_l\left((|\mathcal{M}|^2-|\mathcal{M}|)\frac{1}{w_l w_{l'}}\,\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}\right)\\
&\qquad+\frac{1}{|\mathcal{M}|^2}\mathbb{E}_l\left(|\mathcal{M}|\frac{1}{w_l w_{l'}}\,\mathbf{U}_{kl}\,\mathbf{U}_{k'l'}\,x_l x_{l'}\right)-\widehat{x}_k\widehat{x}_{k'}\Big]\\
&= \sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\Big[\frac{|\mathcal{M}|^2-|\mathcal{M}|}{|\mathcal{M}|^2}\left(\sum_{j=1}^{N}\frac{1}{w_j}\,\mathbf{U}_{kj}\,x_j\right)\left(\sum_{j'=1}^{N}\frac{1}{w_{j'}}\,\mathbf{U}_{kj'}\,x_{j'}\right)\mathbb{P}(j=l)\mathbb{P}(j'=l')\\
&\qquad+\frac{1}{|\mathcal{M}|}\sum_{j=1}^{N}\frac{1}{w_j}\,\mathbf{U}_{kj}\,\mathbf{U}_{k'j}\,x_j^2\mathbb{P}(j=l)-\widehat{x}_k\widehat{x}_{k'}\Big]\\
&= \frac{1}{|\mathcal{M}|}\sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\sum_{j=1}^{N}\frac{1}{w_j}\,\mathbf{U}_{kj}\,\mathbf{U}_{k'j}\,x_j^2+\sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\Big(\frac{|\mathcal{M}|-1}{|\mathcal{M}|}-1\Big)\widehat{x}_k\widehat{x}_{k'}\\
&\overset{(a)}{\leq} \frac{1}{|\mathcal{M}|}\sum_{k,k'\leq\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\sum_{j=1}^{N}\frac{1}{w_j}\,\mathbf{U}_{kj}\,\mathbf{U}_{k'j}\,x_j^2\\
&\leq \frac{\max_j x_j^2}{|\mathcal{M}|}\sum_{k,k'<\kappa}\mathbf{V}_{ik}\,\mathbf{V}_{ik'}\sum_{j=1}^{N}\frac{1}{w_j}\,\mathbf{U}_{kj}\,\mathbf{U}_{k'j},
\end{aligned}
$$

where $(a)$ follows from the fact that we ignore the second term.

We combine the bounds for both $\Delta_i^{(1)}$ and $\Delta_i^{(2)}$, and obtain the bounds for the variance term.

$$\begin{aligned}
\mathbb{E}\left\|\mathbf{x}^* - \mathbb{E}\mathbf{x}^*\right\|^2 &= \sum_{i=1}^{N} \mathbb{E}\left\|x_i^* - \mathbb{E}x_i^*\right\| \\
&= \sum_{i=1}^{N} \left( \mathbb{E}\left\|\Delta_i^{(1)}\right\|^2 + \mathbb{E}\left\|\Delta_i^{(2)}\right\|^2 \right) \\
&\leq (\max_j x_j^2 + \sigma^2) \frac{1}{|\mathcal{M}|} \sum_{i=1}^{N} \sum_{k,k'<\kappa} \mathbf{V}_{ik}\, \mathbf{V}_{ik'} \sum_{j=1}^{N} \frac{1}{w_j}\, \mathbf{U}_{kj}\, \mathbf{U}_{k'j} \\
&= (\max_j x_j^2 + \sigma^2) \frac{1}{|\mathcal{M}|} \mathrm{Tr}\left( \mathbf{U}_{(\kappa)}^T\, \mathbf{V}_{(\kappa)}^T\, \mathbf{V}_{(\kappa)}\, \mathbf{U}_{(\kappa)} \mathbf{W} \right) \\
&\leq (\max_j x_j^2 + \sigma^2) \frac{\alpha_2}{|\mathcal{M}|} \mathrm{Tr}\left( \mathbf{U}_{(\kappa)}\, \mathbf{W}\, \mathbf{U}_{(\kappa)}^T \right) \\
&= (\max_j x_j^2 + \sigma^2) \frac{\alpha_2}{|\mathcal{M}|} \sum_{k<\kappa} \sum_{i=1}^{N} \frac{\sum_{j=0}^{N-1} \sqrt{\sum_{k'<\kappa} \mathbf{U}_{k'j}^2}}{\sqrt{\sum_{k<\kappa} \mathbf{U}_{ki}^2}}\, \mathbf{U}_{kj}^2 \\
&= (\max_j x_j^2 + \sigma^2) \frac{\alpha_2}{|\mathcal{M}|} \left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2,
\end{aligned}$$

Finally, we combine the bias term and the variance term, and obtain the bounds for the recovery error, as presented in Theorem 17.

## Appendix. D Proof of Corollary 1

We start with the upper bound of Corollary 3. Based on Theorem 16, we have

$$\begin{aligned}
& \frac{\alpha_2 \mu \left\|\mathbf{x}\right\|_2^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2 + \sigma^2)\alpha_2}{|\mathcal{M}|} N \left\|\mathbf{U}_{(\kappa)}\right\|_F^2 \\
\overset{(a)}{=} & \frac{\alpha_2 \mu \left\|\mathbf{x}\right\|_2^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2 + \sigma^2)\alpha_2}{|\mathcal{M}|} N^2 \kappa (\frac{c}{\sqrt{N}})^2 \\
\leq & N \left( \frac{\alpha_2 \mu \max_i x_i^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2 + \sigma^2)\alpha_2 c^2}{|\mathcal{M}|} \kappa \right) \\
\asymp & C |\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},
\end{aligned}$$

where $(a)$ follows from Definition 12, $\kappa$ is set in the order of $|\mathcal{M}|^{\frac{1}{2\beta+1}}$, and $C > 0$ is some constant. Since at least Algorithm 2 satisfies this rate of convergence, we thus have

$$\inf_{(\mathbf{x}^*, \mathcal{M}) \in \Theta_{\mathrm{rand}}} \sup_{\mathbf{x} \in \mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}} \left( \left\|\mathbf{x}^* - \mathbf{x}\right\|_2^2 \right) \leq C |\mathcal{M}|^{-\frac{2\beta}{2\beta+1}}.$$

Based on Definition 12, $\left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2 = \left( N \sqrt{\kappa (\frac{c}{\sqrt{N}})^2} \right)^2 = c^2 N \kappa$, which is in the same order of $N \left\|\mathbf{U}_{(\kappa)}\right\|_F^2$, we thus have

$$\frac{\alpha_2 \mu \left\|\mathbf{x}\right\|_2^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2 + \sigma^2)\alpha_2}{|\mathcal{M}|} N \left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2 \asymp C |\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},$$

where $C > 0$. Since at least Algorithm 5 satisfies this rate of convergence. We thus have

$$\inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\exp}} \sup_{\mathbf{x}\in \mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \leq C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}}.$$

The above content proofs the upper bound in Corollary 3. We next show the lower bound. Based on Definition 12, $\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2 = N\kappa_0(\frac{c}{\sqrt{N}})^2 = c^2\kappa_0$, we thus have

$$\begin{aligned}
& \inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\mathrm{rand}}} \sup_{\mathbf{x}\in \mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \\
\geq & \frac{c_1\mu\|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}}\left(1 - \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2\kappa_0^{2\beta+2}}\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2|\mathcal{M}|\right), \\
= & \frac{c_1\mu\|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}}\left(1 - \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2\kappa_0^{2\beta+1}N}|\mathcal{M}|\right) \\
\geq & \frac{c_1\mu\|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}}\left(1 - \frac{c\mu\max_i x_i^2}{\sigma^2\kappa_0^{2\beta+1}}|\mathcal{M}|\right) \\
\asymp & c|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},
\end{aligned}$$

where $\kappa_0$ is set in the order of $|\mathcal{M}|^{\frac{1}{2\beta+1}}$. Based on Definition 12, $\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{\infty,2}^2 = \kappa_0(\frac{c}{\sqrt{N}})^2 = c^2\kappa_0/N$, which is in the same order of $\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2/N$, we thus have we thus have

$$\begin{aligned}
& \inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\exp}} \sup_{\mathbf{x}\in \mathrm{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\|\mathbf{x}^* - \mathbf{x}\|_2^2\right) \\
\geq & \frac{c_1\mu\|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}}\left(1 - \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2\kappa_0^{2\beta+2}}\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{\infty,2}^2|\mathcal{M}|\right), \\
= & \frac{c_1\mu\|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}}\left(1 - \frac{c\mu\|\mathbf{x}\|_2^2}{\sigma^2\kappa_0^{2\beta+1}N}|\mathcal{M}|\right) \\
\geq & \frac{c_1\mu\|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}}\left(1 - \frac{c\mu\max_i x_i^2}{\sigma^2\kappa_0^{2\beta+1}}|\mathcal{M}|\right) \\
\asymp & c|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},
\end{aligned}$$

where $K$ is set in the order of $|\mathcal{M}|^{\frac{1}{2\beta+1}}$.

## Appendix. E Proof of Corollary 2

We start with the upper bound of Corollary 5. Based on Theorem 17, we have

$$\begin{aligned}
& \frac{\alpha_2\mu\|\mathbf{x}\|_2^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2 + \sigma^2)\alpha_2}{|\mathcal{M}|}N\left\|\mathbf{U}_{(\kappa)}\right\|_F^2 \\
\overset{(a)}{=} & \frac{\alpha_2\mu\|\mathbf{x}\|_2^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2 + \sigma^2)\alpha_2}{|\mathcal{M}|}N\kappa \\
\leq & N\left(\frac{\alpha_2\mu\max_i x_i^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2 + \sigma^2)\alpha_2 c^2}{|\mathcal{M}|}\kappa\right) \\
\asymp & C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},
\end{aligned}$$

where $(a)$ follows from Definition 12, $\kappa$ is set in the order of $|\mathcal{M}|^{\frac{1}{2\beta+1}}$, and $C>0$ is some constant. Since at least Algorithm 2 satisfies this rate of convergence, we thus have

$$\inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\text{rand}}} \sup_{\mathbf{x}\in \text{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\|\mathbf{x}^*-\mathbf{x}\|_2^2\right) \leq C|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}}.$$

We next bound $\left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2$. Since $\kappa \geq K_0$, we have

$$\begin{aligned}
\left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2 &= \|h\|_1^2 = \left(\|h_T\|_1 + \|h_{T^c}\|_1\right)^2 \\
&\overset{(a)}{\leq} (1+c)^2\|h_T\|_1^2 \overset{(b)}{\leq} (1+c)^2\kappa\|h_T\|_2^2 \\
&\leq (1+c)^2\kappa\|h\|_1^2 = (1+c)^2\kappa^2,
\end{aligned}$$

where $(a)$ follows from Definition 13, and $(b)$ from the norm equivalence.

Based on Theorem 17, we thus have

$$\begin{aligned}
& \frac{\alpha_2\mu\|\mathbf{x}\|_2^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2+\sigma^2)\alpha_2}{|\mathcal{M}|}\left\|\mathbf{U}_{(\kappa)}\right\|_{2,1}^2 \\
\leq\ & \frac{\alpha_2\mu\|\mathbf{x}\|_2^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2+\sigma^2)\alpha_2 c}{|\mathcal{M}|}\kappa^2 \\
\leq\ & N\left(\frac{\alpha_2\mu\max_i x_i^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2+\sigma^2)\alpha_2 c^2}{|\mathcal{M}|N}\kappa^2\right) \\
\overset{(a)}{\leq}\ & N\left(\frac{\alpha_2\mu\max_i x_i^2}{\kappa^{2\beta}} + \frac{(\max_j x_j^2+\sigma^2)\alpha_2 c^2}{|\mathcal{M}|}\kappa^{2-\gamma}\right) \\
\asymp\ & C|\mathcal{M}|^{\frac{2\beta}{2\beta+2-\gamma}},
\end{aligned}$$

where $\gamma$ varies with $\kappa$ to satisfy $\kappa^\gamma \leq N$. The upper bound reaches the minimum when $\kappa$ is set in the order of $|\mathcal{M}|^{\frac{1}{2\beta+2-\gamma}}$, also, $\kappa \geq K$, thus,

$$\max\{1, 2\beta+2-\frac{\log|\mathcal{M}|}{\log K}\} \leq \gamma \leq \frac{(2\beta+2)\log N}{\log N+\log|\mathcal{M}|}.$$

Note that $\gamma=1$ corresponds to $N=\kappa$. Since at least Algorithm 5 satisfies this rate of convergence. We thus have

$$\inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\text{exp}}} \sup_{\mathbf{x}\in \text{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}}\left(\|\mathbf{x}^*-\mathbf{x}\|_2^2\right) \leq C|\mathcal{M}|^{-\frac{2\beta}{2\beta+2-\gamma}},$$

where $\max\{1, 2\beta+2-\frac{\log|\mathcal{M}|}{\log K}\} \leq \gamma \leq \frac{(2\beta+2)\log N}{\log N+\log|\mathcal{M}|}$..

The above content proofs the lower bound in Corollary 3. We next show the lower bound. Based

on Definition 12, $\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2 = N\kappa_0(\frac{1}{\sqrt{N}})^2 = \kappa_0$, we thus have

$$
\begin{aligned}
& \inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\text{rand}}} \sup_{\mathbf{x}\in \text{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}} \left( \|\mathbf{x}^* - \mathbf{x}\|_2^2 \right) \\
\geq & \frac{c_1\mu \|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}} \left( 1 - \frac{c\mu \|\mathbf{x}\|_2^2}{\sigma^2 \kappa_0^{2\beta+2}} \left\|\mathbf{V}_{(2,\kappa_0)}\right\|_F^2 |\mathcal{M}| \right), \\
= & \frac{c_1\mu \|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}} \left( 1 - \frac{c\mu \|\mathbf{x}\|_2^2}{\sigma^2 \kappa_0^{2\beta+1} N} |\mathcal{M}| \right) \\
= & \frac{c_1\mu \|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}} \left( 1 - \frac{c\mu \max_i x_i^2}{\sigma^2 \kappa_0^{2\beta+1}} |\mathcal{M}| \right) \\
\asymp & c|\mathcal{M}|^{-\frac{2\beta}{2\beta+1}},
\end{aligned}
$$

where $\kappa_0$ is set in the order of $|\mathcal{M}|^{\frac{1}{2\beta+1}}$. Based on Definition 12, $\left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{\infty,2}^2 = c$, we thus have we thus have

$$
\begin{aligned}
& \inf_{(\mathbf{x}^*,\mathcal{M})\in\Theta_{\text{exp}}} \sup_{\mathbf{x}\in \text{ABL}_{\mathbf{A}}(K,\beta,\mu)} \mathbb{E}_{\mathbf{x},\mathcal{M}} \left( \|\mathbf{x}^* - \mathbf{x}\|_2^2 \right) \\
\geq & \frac{c_1\mu \|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}} \left( 1 - \frac{c\mu \|\mathbf{x}\|_2^2}{\sigma^2 \kappa_0^{2\beta+2}} \left\|\mathbf{V}_{(2,\kappa_0)}\right\|_{\infty,2}^2 |\mathcal{M}| \right), \\
= & \frac{c_1\mu \|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}} \left( 1 - \frac{c\mu \|\mathbf{x}\|_2^2}{\sigma^2 \kappa_0^{2\beta+2}} |\mathcal{M}| \right) \\
= & \frac{c_1\mu \|\mathbf{x}\|_2^2}{\kappa_0^{2\beta}} \left( 1 - \frac{c\mu \max_i x_i^2}{\sigma^2 \kappa_0^{2\beta+2}} N|\mathcal{M}| \right) \\
\asymp & c|\mathcal{M}|^{-\frac{2\beta}{2\beta+2-\gamma}},
\end{aligned}
$$

where $\kappa_0$ is set in the order of $|\mathcal{M}|^{\frac{1}{2\beta+1}}$, and $\max\{1, 2\beta+2-\frac{\log|\mathcal{M}|}{\log K}\} \leq \gamma \leq \frac{(2\beta+2)\log N}{\log N + \log|\mathcal{M}|}$, because $\kappa_0 \leq K$, and $\kappa_0^\gamma \leq N$.

# B

## B.1 Proofs from Chapter 4

Denoting SCAD and MCP parameters $\gamma_{SCAD} \geq 2$ and $\gamma_{MCP} \geq 1$ and define the SCAD penalty functions as:

$$\rho_{SCAD}(t; \lambda, \gamma_{SCAD}) = \lambda \int_0^{|t|} \min(1, \frac{(\gamma_{SCAD} - u/\lambda)_+}{\gamma_{SCAD}}) du \tag{B.1}$$

and the MCP penalty function as

$$\rho_{MCP}(t; \lambda, \gamma_{MCP}) = \lambda \int_0^{|t|} (1 - \frac{u}{\lambda \gamma_{MCP}})_+ du \tag{B.2}$$

### B.1.1 Proof of Theorem 21

*Proof.* We denote $\mathcal{D} = \boldsymbol{\Delta}^{(k+1)}$. Define $R$ as the row space of $\mathcal{D}$, and $R^\perp$ the null space. Let $\mathcal{P}_{\mathrm{R}} = \boldsymbol{D}^\dagger \mathcal{D}$, the projection onto $R$, and $\|\mathbf{x}\|_{\mathrm{R}} = \|\mathcal{P}_{\mathrm{R}} \mathbf{x}\|_2$. Additionally, $\mathcal{P}_{\mathrm{R}^\perp} = \boldsymbol{I} - \boldsymbol{D}^\dagger \mathcal{D}$, the projection onto $R^\perp$. Now consider

$$\tilde{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \boldsymbol{\beta}\|_{\mathrm{R}}^2 + g(\mathcal{D}\boldsymbol{\beta}), \tag{B.3}$$

such that $\bar{\boldsymbol{\beta}} = \mathcal{P}_{\mathrm{R}^\perp} \mathbf{y} + \tilde{\boldsymbol{\beta}}$ and $\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 = \|\boldsymbol{\epsilon}\|_{\mathrm{R}^\perp}^2 + \|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2$. Recognizing that $\|\boldsymbol{\epsilon}\|_{\mathrm{R}^\perp}^2$ is a chi-squared random variable with $C_G$ degrees of freedom, we can then invoke the one-sided tail bound for chi-squared random variables (c.f. [194] Example 2.5) such that for any $0 \leq t \leq 1$,

$$P\big(\|\boldsymbol{\epsilon}\|_{\mathrm{R}^\perp}^2 \geq \sigma^2 C_G (1 + t)\big) \leq \exp\left(\frac{-C_G t^2}{8}\right).$$

Consequently, with probability at least $1 - \delta$,

$$\|\boldsymbol{\epsilon}\|_{\mathrm{R}^\perp}^2 \leq \sigma^2 \Big(C_G + 2\sqrt{2 C_G \log(1/\delta)}\Big). \tag{B.4}$$

We now consider the second term $\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2$. By the optimality of $\tilde{\boldsymbol{\beta}}$, we have

$$\frac{1}{2} \|\mathbf{y} - \tilde{\boldsymbol{\beta}}\|_{\mathrm{R}}^2 + g(\mathcal{D}\tilde{\boldsymbol{\beta}}) \leq \frac{1}{2} \|\mathbf{y} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2 + g(\mathcal{D}\boldsymbol{\beta}^\star).$$

Rearranging the terms and substituting (4.1) give us

$$\begin{aligned}
&\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2 \\
&\leq 2\boldsymbol{\epsilon}^\top \mathcal{P}_{\mathrm{R}}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star) - 2g(\mathcal{D}\tilde{\boldsymbol{\beta}}) + 2g(\mathcal{D}\boldsymbol{\beta}^\star) \\
&= 2\big\langle (\boldsymbol{D}^\dagger)^\top \boldsymbol{\epsilon}, \mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\big\rangle - 2g(\mathcal{D}\tilde{\boldsymbol{\beta}}) + 2g(\mathcal{D}\boldsymbol{\beta}^\star) \\
&\leq 2\big\|(\boldsymbol{D}^\dagger)^\top \boldsymbol{\epsilon}\big\|_\infty \big\|\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\big\|_1 - 2g(\mathcal{D}\tilde{\boldsymbol{\beta}}) + 2g(\mathcal{D}\boldsymbol{\beta}^\star),
\end{aligned} \tag{B.5}$$

where the last line follows from Hölder's inequality. By standard tail bounds for independent Gaussian random variables, we have

$$\|(\boldsymbol{D}^\dagger)^\top \boldsymbol{\epsilon}\|_\infty \leq \sigma\zeta_k \sqrt{2\log(\frac{er}{\delta})} \tag{B.6}$$

with probability at least $1-\delta$. Further note that the inequality (B.21) holds simultaneously with the inequality (B.20) with probability at least $1-2\delta$.

By setting $\lambda = \sigma\zeta_k\sqrt{2\log(\frac{er}{\delta})} \geq \|(\boldsymbol{D}^\dagger)^\top\boldsymbol{\epsilon}\|_\infty$, we continue bounding (B.5) as

$$\begin{aligned}
\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2 &\leq 2\lambda\big\|\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\big\|_1 - 2g(\mathcal{D}\tilde{\boldsymbol{\beta}}) + 2g(\mathcal{D}\boldsymbol{\beta}^\star) \\
&= \Big[2\lambda\big\|\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\big\|_1 - 2g\big(\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\big)\Big] \\
&\quad + \Big[2g\big(\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\big) - 2g(\mathcal{D}\tilde{\boldsymbol{\beta}}) + 2g(\mathcal{D}\boldsymbol{\beta}^\star)\Big].
\end{aligned} \tag{B.7}$$

Under Assumption 1, we have $\lambda\|\boldsymbol{x}\|_1 \leq g(\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{x}\|_2^2$ from[118] Lemma 4. Hence, the first two terms in (B.7) can be bounded as:

$$2\lambda\|\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_1 - 2g(\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)) \leq \mu\|\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_2^2.$$

For the next two terms in (B.7), by the subadditivity and symmetry of $g(\cdot)$[118],

$$\begin{aligned}
&2g\big(\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\big) - 2g(\mathcal{D}\tilde{\boldsymbol{\beta}}) \\
&\leq 2g\big((\mathcal{D}\tilde{\boldsymbol{\beta}} - \mathcal{D}\boldsymbol{\beta}^\star) - \mathcal{D}\tilde{\boldsymbol{\beta}}\big) = 2g(\mathcal{D}\boldsymbol{\beta}^\star).
\end{aligned}$$

Plugging the above two inequalities into (B.7), we have

$$\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2 \leq \mu\|\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_2^2 + 4g(\mathcal{D}\boldsymbol{\beta}^\star). \tag{B.8}$$

Note that

$$\begin{aligned}
\|\mathcal{D}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_2 &= \|\mathcal{D}\mathcal{P}_{\mathrm{R}}(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star)\|_2 \\
&\leq \|\mathcal{D}\| \cdot \|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}},
\end{aligned}$$

which, combined with (B.8), leads to

$$\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2 \leq \mu\|\mathcal{D}\|^2\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2 + 4g(\mathcal{D}\boldsymbol{\beta}^\star).$$

By the assumption $\mu\|\mathcal{D}\|^2 < 1$, we have

$$\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2 \leq \frac{4g(\mathcal{D}\boldsymbol{\beta}^\star)}{1-\mu\|\mathcal{D}\|^2}. \tag{B.9}$$

Since $\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 = \|\boldsymbol{\epsilon}\|_{\mathrm{R}^\perp}^2 + \|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_{\mathrm{R}}^2$, we conclude that

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 \leq \sigma^2\Big(C_G + 2\sqrt{2C_G\log(1/\delta)}\Big) + \frac{4g(\mathcal{D}\boldsymbol{\beta}^\star)}{1-\mu\|\mathcal{D}\|^2}$$

with probability at least $1-2\delta$.

The proof for vector-GTF (4.9) is a straightforward extension. Defining $\widetilde{\boldsymbol{B}}$ similarly to $\tilde{\boldsymbol{\beta}}$, we have $\|\overline{\boldsymbol{B}} - \boldsymbol{B}\|_{\mathrm{F}}^2 = \|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{E}\|_{\mathrm{F}}^2 + \|\widetilde{\boldsymbol{B}} - \boldsymbol{B}^\star\|_{\mathrm{R}}^2$. The first term can be bounded as $\|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{E}\|_{\mathrm{F}}^2 \leq d\sigma^2\Big(C_G + 2\sqrt{2C_G\log(d/\delta)}\Big)$ with probability at least $1-\delta$. The second term can be bounded as

$$\|\widetilde{\boldsymbol{B}} - \boldsymbol{B}^\star\|_{\mathrm{R}}^2 \leq 2\langle \boldsymbol{E}, \mathcal{P}_{\mathrm{R}}(\widetilde{\boldsymbol{B}} - \boldsymbol{B}^\star)\rangle - 2h(\mathcal{D}\widetilde{\boldsymbol{B}}) + 2h(\mathcal{D}\boldsymbol{B}^\star).$$

For the first term, using Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}
\langle \boldsymbol{E}, \mathcal{P}_{\mathrm{R}}(\widetilde{\boldsymbol{B}} - \boldsymbol{B}^\star)\rangle &= \langle (\boldsymbol{D}^\dagger)^\top \boldsymbol{E}, \mathcal{D}(\widetilde{\boldsymbol{B}} - \boldsymbol{B}^\star)\rangle \\
&\leq \sum_{\ell=1}^{r} \big\|((\boldsymbol{D}^\dagger)^\top \boldsymbol{E})_{\ell\cdot}\big\|_2 \big\|(\mathcal{D}(\widetilde{\boldsymbol{B}} - \boldsymbol{B}^\star))_{\ell\cdot}\big\|_2 \\
&\leq \sum_{\ell=1}^{r} \sqrt{d}\,\big\|((\boldsymbol{D}^\dagger)^\top \boldsymbol{E})_{\ell\cdot}\big\|_\infty \big\|(\mathcal{D}(\widetilde{\boldsymbol{B}} - \boldsymbol{B}^\star))_{\ell\cdot}\big\|_2 \\
&\leq \lambda \sum_{\ell=1}^{r} \big\|(\mathcal{D}(\widetilde{\boldsymbol{B}} - \boldsymbol{B}^\star))_{\ell\cdot}\big\|_2,
\end{aligned} \tag{B.10}$$

where the last line follows from the assumption on $\lambda$ as well as the tail bound (B.20). One can then continue to bound (B.10) in a similar way as (B.5). □

For SCAD/MCP, $\mu \geq \max(\frac{1}{\gamma_{SCAD}-1}, \frac{1}{\gamma_{MCP}})$ satisfies the inequality, and $\mu = 0$ does so trivially for $\ell_1$ [118].

**Remark 6.** This is a straightforward result using concentration bounds for sub-Gaussian random variables. Let $a_i = ((\boldsymbol{D}^\dagger)^\top\boldsymbol{\epsilon})_i = \boldsymbol{d}_i^\top\boldsymbol{\epsilon}$ where $\boldsymbol{d}_i$ is the $i$-th column of $\boldsymbol{D}^\dagger$ . It follows that if $\epsilon_i$ is $SG(\sigma^2)$, $a_i$ is $SG(\sigma^2\|\boldsymbol{d}_i\|_2^2)$. It also follows then that $\forall i$, $a_i$ is $SG(\sigma^2 \max_i \|\boldsymbol{d}_i\|_2^2) = SG(\sigma^2\zeta^2)$ such that

$$P(|a_i| \geq t) \leq 2\exp(-\frac{t^2}{2\sigma^2\zeta^2})$$

We can then apply the union bound to get the standard result for the maximum of sub-gaussian random variables:

$$P\Big(\max_{i=1,\cdots r}|a_i| \geq \sigma\zeta\sqrt{2\log(\frac{r}{\delta})}\Big) \leq 2\delta$$

### B.1.2 Proof of Proposition 2

*Proof.* We denote $\boldsymbol{\Delta}^{(k+1)\dagger} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]$, and the eigendecomposition of $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top$, where orthogonal matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$. Note that $\boldsymbol{L}^\dagger = \boldsymbol{U}\boldsymbol{\Sigma}^\dagger\boldsymbol{U}^\top$, and $\boldsymbol{L}^{(k+1)} = \boldsymbol{U}\boldsymbol{\Sigma}^{(k+1)}\boldsymbol{U}^\top$. Lastly, $\boldsymbol{\Delta}^\top = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m]$, where $\|\boldsymbol{d}_j\|_2 = \sqrt{2}$. We consider two cases:

- $k$ is odd. $\boldsymbol{\Delta}^{(k+1)\dagger} = (\boldsymbol{L}^{(k+1)})^{\dagger}\boldsymbol{\Delta}^{(k+1)^{\top}} = (\boldsymbol{L}^{(k+1)})^{\dagger}(\boldsymbol{L}^{(\frac{k+1}{2})})^{\top} = \boldsymbol{U}(\boldsymbol{\Sigma}^{(k+1)})^{\dagger}\boldsymbol{\Sigma}^{(\frac{k+1}{2})}\boldsymbol{U}^{\top}$. Then,

$$\|\boldsymbol{s}_j\|_2^2 = \sum_{i=2}^{n} \frac{1}{\lambda_i^{k+1}} \langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle^2 \leq \frac{1}{\lambda_2^{k+1}}$$

- $k$ is even. $\boldsymbol{\Delta}^{(k+1)\dagger} = (\boldsymbol{L}^{(k+1)})^{\dagger}(\boldsymbol{L}^{(\frac{k}{2})})^{\top}\boldsymbol{\Delta}^{\top} = \boldsymbol{U}\boldsymbol{\Sigma}^{(k+1)\dagger}\boldsymbol{\Sigma}^{(\frac{k}{2})}\boldsymbol{U}^{\top}\boldsymbol{\Delta}^{\top}$. Similarly,

$$\|\boldsymbol{s}_j\|_2^2 = \sum_{i=2}^{n} \frac{1}{\lambda_i^{k+2}} \langle \boldsymbol{u}_i, \boldsymbol{d}_j \rangle^2 \leq \frac{1}{\lambda_2^{k+2}} \|\boldsymbol{d}_j\|_2^2 = \frac{2}{\lambda_2^{k+2}}$$

□

*Proof.* We start from $\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta})_T\|_1 \leq \sqrt{|T|}\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta})_T\|_2$, and note that given two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $(\boldsymbol{AB})_T = (\boldsymbol{A})_T\boldsymbol{B}$ where $T$ is a subset of rows indices. Therefore we consider two cases:

- $k$ is even. $\lambda_{\max}(\boldsymbol{X})$ indicates the largest eigenvalue of matrix $\boldsymbol{X}$, and $d_i$ is the degree of node $i$.

$$\begin{aligned}
&\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta})_T\|_2 = \|(\boldsymbol{\Delta})_T\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 \\
&\leq \|(\boldsymbol{\Delta})_T\|\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 = \sqrt{\lambda_{\max}((\boldsymbol{\Delta})_T^{\top}(\boldsymbol{\Delta})_T))}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2
\end{aligned}$$

  Note that $(\boldsymbol{\Delta})_T$ is equivalent to the incidence matrix of a subgraph with only $T$ edges, which allows us to bound,

$$\begin{aligned}
&\sqrt{\lambda_{\max}((\boldsymbol{\Delta})_T^{\top}(\boldsymbol{\Delta})_T))}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 \\
&\leq \sqrt{\max_{(u,v)\in T}\{d_u + d_v\}}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 \leq \sqrt{2d_{\max}}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2
\end{aligned}$$

- $k$ is odd.

$$\begin{aligned}
&\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta})_T\|_2 = \|(\boldsymbol{\Delta}^{\top})_T\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 \\
&\leq \|(\boldsymbol{\Delta}^{\top})_T\|\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 = \sqrt{\lambda_{\max}(\boldsymbol{L}_{T\times T})}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2
\end{aligned}$$

  $\boldsymbol{L}_{T\times T} \in \mathbb{R}^{|T|\times|T|}$ is the principal submatrix of $\boldsymbol{L}$ indexed by $T$. From Cauchy's interlacing theorem, the maximum eigenvalue of the submatrix is upperbounded, so

$$\begin{aligned}
&\sqrt{\lambda_{\max}(\boldsymbol{L}_{T\times T})}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 \\
&\leq \sqrt{\lambda_{\max}(\boldsymbol{L})}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 \leq \sqrt{2d_{\max}}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2
\end{aligned}$$

To conclude the proof, we use ${\boldsymbol{\Delta}^{(k)}}^{\top}\boldsymbol{\Delta}^{(k)} = \boldsymbol{L}^{(k)}$, and that the eigenvalues of a power matrix is also raised to the power.

$$\begin{aligned}
&\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta})_T\|_1 \leq \sqrt{|T|}\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{\beta})_T\|_2 \\
&\leq \sqrt{|T|}\sqrt{2d_{\max}}\|\boldsymbol{\Delta}^{(k)}\boldsymbol{\beta}\|_2 \leq (2d_{\max})^{\frac{k+1}{2}}\sqrt{|T|}\|\boldsymbol{\beta}\|_2
\end{aligned}$$

$k = 0$.

$$\|(\Delta\boldsymbol{\beta})_T\|_1 \leq \sqrt{|T|}\sqrt{\sum_{(i,j)\in T}|\beta_i - \beta_j|^2} = \sqrt{|T|}\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2 \leq \sqrt{|T|}\|\boldsymbol{\beta}_i\|_2 + \sqrt{|T|}\|\boldsymbol{\beta}_j\|_2 \leq 2\sqrt{|T|}\sqrt{d}\|\boldsymbol{\beta}\|_2$$

$k = 1$.

$$\|(\Delta^\top \Delta \boldsymbol{\beta})_T\|_1 \leq \sqrt{|T|} \sqrt{\sum_{i \in T} |d_i \beta_i - \sum_{j:(i,j)\in\mathcal{E}} \beta_j|^2} = \sqrt{|T|} \|\boldsymbol{d}^\top \boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2 \tag{B.11}$$

$$\leq \sqrt{|T|} \|\boldsymbol{d}^\top \boldsymbol{\beta}_i\|_2 + \sqrt{|T|} \|\boldsymbol{\beta}_j\|_2 \leq 2\sqrt{|T|} d \|\boldsymbol{\beta}\|_2 \tag{B.12}$$

□

### B.1.3 Proof of Theorem 23

*Proof.* We denote $\mathcal{D} = \boldsymbol{\Delta}^{(k+1)}$. Define $R$ as the row space of $\mathcal{D}$, and $R^\perp$ the null space. Let $\mathcal{P}_{\mathrm{R}} = \boldsymbol{D}^\dagger \mathcal{D}$, the projection onto $R$, and $\|\mathbf{x}\|_{\mathrm{R}} = \|\mathcal{P}_{\mathrm{R}} \mathbf{x}\|_2$. Additionally, $\mathcal{P}_{\mathrm{R}^\perp} = \boldsymbol{I} - \boldsymbol{D}^\dagger \mathcal{D}$, the projection onto $R^\perp$. Since $\widehat{\boldsymbol{\beta}}$ is a stationary point of $f(\boldsymbol{\beta})$, it follows that

$$\mathbf{0} \in \nabla_{\boldsymbol{\beta}} f(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}} = (\widehat{\boldsymbol{\beta}} - \mathbf{y}) + \nabla_{\boldsymbol{\beta}} g(\mathcal{D}\boldsymbol{\beta})|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}}. \tag{B.13}$$

By the chain rule, $\nabla_{\boldsymbol{\beta}} g(\mathcal{D}\boldsymbol{\beta})|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}} = \{\mathcal{D}^\top \boldsymbol{z} : \boldsymbol{z} \in \nabla_{\boldsymbol{x}} g(\boldsymbol{x})|_{\boldsymbol{x}=\mathcal{D}\widehat{\boldsymbol{\beta}}}\}$. Then by (B.13), there exists $\boldsymbol{z} \in \nabla_{\boldsymbol{x}} g(\boldsymbol{x})|_{\boldsymbol{x}=\mathcal{D}\widehat{\boldsymbol{\beta}}}$, such that

$$\mathbf{0} = (\widehat{\boldsymbol{\beta}} - \mathbf{y}) + \mathcal{D}^\top \boldsymbol{z}.$$

In particular, $\forall \boldsymbol{\beta} \in \mathbb{R}^n$, we have

$$\boldsymbol{\beta}^\top (\mathbf{y} - \widehat{\boldsymbol{\beta}}) = (\mathcal{D}\boldsymbol{\beta})^\top \boldsymbol{z}, \tag{B.14}$$

and, specializing to $\widehat{\boldsymbol{\beta}}$,

$$\widehat{\boldsymbol{\beta}}^\top (\mathbf{y} - \widehat{\boldsymbol{\beta}}) = (\mathcal{D}\widehat{\boldsymbol{\beta}})^\top \boldsymbol{z}. \tag{B.15}$$

Subtract (B.15) from (B.14), and use the definition of subgradient to get $\forall \boldsymbol{\beta} \in \mathbb{R}^n$,

$$\begin{aligned} \boldsymbol{\beta}^\top (\mathbf{y} - \widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{\beta}}^\top (\mathbf{y} - \widehat{\boldsymbol{\beta}}) &= (\mathcal{D}\boldsymbol{\beta} - \mathcal{D}\widehat{\boldsymbol{\beta}})^\top \boldsymbol{z} \\ &\leq g(\mathcal{D}\boldsymbol{\beta}) - g(\mathcal{D}\widehat{\boldsymbol{\beta}}). \end{aligned} \tag{B.16}$$

By the measurement model $\mathbf{y} = \boldsymbol{\beta}^\star + \boldsymbol{\epsilon}$ and the polarization equality, i.e. $2\boldsymbol{a}^\top \boldsymbol{b} = \|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2 - \|\boldsymbol{a} - \boldsymbol{b}\|_2^2$, the left-hand side of (B.16) can be rewritten as

$$\begin{aligned} &\boldsymbol{\beta}^\top (\mathbf{y} - \widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{\beta}}^\top (\mathbf{y} - \widehat{\boldsymbol{\beta}}) \\ &= (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^\top (\boldsymbol{\beta}^\star - \widehat{\boldsymbol{\beta}}) + \boldsymbol{\epsilon}^\top (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}) \\ &= \frac{1}{2}\|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\|_2^2 + \frac{1}{2}\|\boldsymbol{\beta}^\star - \widehat{\boldsymbol{\beta}}\|_2^2 - \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}^\star\|_2^2 + \boldsymbol{\epsilon}^\top (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}). \end{aligned} \tag{B.17}$$

Combining (B.16) and (B.17) gives us $\forall \boldsymbol{\beta} \in \mathbb{R}^n$

$$\begin{aligned} &\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 + \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 \\ &\leq \|\boldsymbol{\beta} - \boldsymbol{\beta}^\star\|_2^2 + 2\boldsymbol{\epsilon}^\top (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) + 2g(\mathcal{D}\boldsymbol{\beta}) - 2g(\mathcal{D}\widehat{\boldsymbol{\beta}}). \end{aligned} \tag{B.18}$$

Let us first consider $\boldsymbol{\epsilon}^\top(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})$. From Hölder's inequality,

$$\begin{aligned}\boldsymbol{\epsilon}^\top(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) &= (\mathcal{D}^\dagger\mathcal{D}\boldsymbol{\epsilon})^\top(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) + (\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon})^\top(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \\ &\leq \|(\mathcal{D}^\dagger)^\top\boldsymbol{\epsilon}\|_\infty\|\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_1 + \|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2. \end{aligned} \tag{B.19}$$

By standard tail bounds for independent Gaussian random variables, we have with probability at least $1 - \delta$,

$$\|(\boldsymbol{D}^\dagger)^\top\boldsymbol{\epsilon}\|_\infty \leq \sigma\zeta_k\sqrt{2\log(\frac{er}{\delta})}. \tag{B.20}$$

Additionally, recognize that $\|\boldsymbol{\epsilon}\|_{\mathrm{R}^\perp}^2$ is a chi-squared random variable with $C_G$ degrees of freedom. We can then invoke the one-sided tail bound for chi-squared random variables (c.f. [194] Example 2.5) such that for any $0 \leq t \leq 1$,

$$P\big(\|\boldsymbol{\epsilon}\|_{\mathrm{R}^\perp}^2 \geq \sigma^2 C_G(1+t)\big) \leq \exp\left(\frac{-C_G t^2}{8}\right).$$

Consequently, with probability at least $1 - \delta$,

$$\|\boldsymbol{\epsilon}\|_{\mathrm{R}^\perp}^2 \leq \sigma^2\Big(C_G + 2\sqrt{2C_G\log(1/\delta)}\Big). \tag{B.21}$$

The inequalities (B.21) and (B.20) hold simultaneously with probability at least $1 - 2\delta$. Then, using $\lambda\|\boldsymbol{x}\|_1 \leq g(\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{x}\|_2^2$ and $\lambda = \sigma\zeta_k\sqrt{2\log(\frac{er}{\delta})} \geq \|(\mathcal{D}^\dagger)^\top\boldsymbol{\epsilon}\|_\infty$, we can bound (B.19) further as

$$\begin{aligned}\boldsymbol{\epsilon}^\top(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) &\leq \|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 + \lambda\|\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_1 \\ &\leq \|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 + g(\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})) + \frac{\mu}{2}\|\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_2^2.\end{aligned}$$

Together with $\|\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_2^2 \leq \|\mathcal{D}\|^2\|(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_2^2$, we can upper bound (B.18) as

$$\begin{aligned}&\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 + \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2^2 \\ &\leq \|\boldsymbol{\beta} - \boldsymbol{\beta}^\star\|_2^2 + 2\|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 + \mu\|\mathcal{D}\|^2\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 \\ &\quad + 2g(\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})) + 2g(\mathcal{D}\boldsymbol{\beta}) - 2g(\mathcal{D}\widehat{\boldsymbol{\beta}}).\end{aligned} \tag{B.22}$$

Note that for any set $T$, $g(\mathbf{x}) = \sum_{i\in T}\rho(x_i) + \sum_{j\in T^c}\rho(x_j) = g((\mathbf{x})_T) + g((\mathbf{x})_{T^c})$. Therefore, using the triangle inequality and subadditivity and symmetry of $\rho$,

$$\begin{aligned}&g(\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})) + g(\mathcal{D}\boldsymbol{\beta}) - g(\mathcal{D}\widehat{\boldsymbol{\beta}}) \\ &\leq g((\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}))_T) + g((\mathcal{D}\boldsymbol{\beta})_{T^c}) + g((\mathcal{D}\widehat{\boldsymbol{\beta}})_{T^c}) \\ &\quad + g(\mathcal{D}\boldsymbol{\beta}) - g((\mathcal{D}\widehat{\boldsymbol{\beta}})_T) - g((\mathcal{D}\widehat{\boldsymbol{\beta}})_{T^c}) \\ &= g((\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}))_T) + 2g((\mathcal{D}\boldsymbol{\beta})_{T^c}) + g((\mathcal{D}\boldsymbol{\beta})_T) - g((\mathcal{D}\widehat{\boldsymbol{\beta}})_T) \\ &\leq 2g((\mathcal{D}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}))_T) + 2g((\mathcal{D}\boldsymbol{\beta})_{T^c}).\end{aligned} \tag{B.23}$$

We bound (B.23) further by the compatibility factor,

$$\begin{aligned} g((\mathcal{D}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}))_T) &\leq \lambda \|(\mathcal{D}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}))_T\|_1 \\ &\leq \lambda\sqrt{|T|}\kappa_T^{-1}\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2. \end{aligned} \tag{B.24}$$

Now combining (B.22), (B.23), and (B.24), we then have

$$\begin{aligned} \|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2^2 + \|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}^\star\|_2^2 \leq \|\boldsymbol{\beta}-\boldsymbol{\beta}^\star\|_2^2 &+ 4g((\mathcal{D}\boldsymbol{\beta})_{T^c}) \\ &+ 2\left(\|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2 + 2\lambda\sqrt{|T|}\kappa_T^{-1}\right)\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2 \\ &+ \mu\|\mathcal{D}\|^2\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2^2. \end{aligned}$$

Apply Young's inequality, which is $2ab \leq a^2/\epsilon + \epsilon b^2$ for $\epsilon > 0$, with $a = \|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2 + 2\lambda\sqrt{|T|}\kappa_T^{-1}$, $b = \|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2$, and $\epsilon = 1-\mu\|\mathcal{D}\|^2 > 0$, we have

$$\begin{aligned} &2\left(\|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2 + 2\lambda\sqrt{|T|}\kappa_T^{-1}\right)\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2 \\ &\leq \frac{1}{\epsilon}\left(\|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2 + 2\lambda\sqrt{|T|}\kappa_T^{-1}\right)^2 + \epsilon\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2^2 \\ &\leq \frac{2}{(1-\mu\|\mathcal{D}\|^2)}\left(\|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2^2 + 4\lambda^2|T|\kappa_T^{-2}\right) \\ &\quad + (1-\mu\|\mathcal{D}\|^2)\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2^2. \end{aligned} \tag{B.25}$$

Therefore,

$$\begin{aligned} &\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|^2 + \|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}^\star\|_2^2 \\ &\leq \|\boldsymbol{\beta}-\boldsymbol{\beta}^\star\|_2^2 + 4g((\mathcal{D}\boldsymbol{\beta})_{T^c}) + \|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2^2 \\ &+ \frac{2}{(1-\mu\|\mathcal{D}\|^2)}\left(\|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{\epsilon}\|_2^2 + 4\lambda^2|T|\kappa_T^{-2}\right). \end{aligned} \tag{B.26}$$

Cancel $\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2^2$ on both sides, apply the infimum over $\boldsymbol{\beta}$ and plug in the bounds (B.21) to get

$$\begin{aligned} \|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}^\star\|_2^2 &\leq \inf_{\boldsymbol{\beta}}\left\{\|\boldsymbol{\beta}-\boldsymbol{\beta}^\star\|_2^2 + 4g((\mathcal{D}\boldsymbol{\beta})_{T^c})\right\} \\ &+ \frac{2\sigma^2}{(1-\mu\|\mathcal{D}\|^2)}\left(C_G + 2\sqrt{2C_G\log(\frac{1}{\delta})} + \frac{8\zeta_k^2|T|}{\kappa_T^2}\log(\frac{er}{\delta})\right). \end{aligned}$$

The proof extends for the vector-GTF (4.9) in a similar manner. We need to replace (B.19) by

$$\begin{aligned} \langle \boldsymbol{E}, \widehat{\boldsymbol{B}}-\boldsymbol{B}\rangle &= \langle \mathcal{D}^\dagger\mathcal{D}\boldsymbol{E}, \widehat{\boldsymbol{B}}-\boldsymbol{B}\rangle + \langle \mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{E}, \widehat{\boldsymbol{B}}-\boldsymbol{B}\rangle \\ &\leq \lambda\sum_{\ell=1}^{r}\left\|\mathcal{D}_{\ell\cdot}(\widehat{\boldsymbol{B}}-\boldsymbol{B})\right\|_2 + \|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{E}\|_{\mathrm{F}}\|\widehat{\boldsymbol{B}}-\boldsymbol{B}\|_{\mathrm{F}}, \end{aligned}$$

where $\|\mathcal{P}_{\mathrm{R}^\perp}\boldsymbol{E}\|_{\mathrm{F}}^2 \leq d\sigma^2\left(C_G + 2\sqrt{2C_G\log(d/\delta)}\right)$ with probability at least $1-\delta$. Similarly, for

(B.24), we use the generalized definition of the compatibility factor $\kappa_T$, given as

$$\begin{aligned} h((\mathcal{D}(\widehat{\boldsymbol{B}} - \boldsymbol{B}))_T) &\leq \lambda \sum_{\ell \in T} \|(\mathcal{D}(\widehat{\boldsymbol{B}} - \boldsymbol{B}))_{\ell \cdot}\|_2 \\ &\leq \lambda \sqrt{|T|} \kappa_T^{-1} \|\widehat{\boldsymbol{B}} - \boldsymbol{B}\|_{\mathrm{F}}, \end{aligned}$$

which will lead to the claimed bound in the theorem. □

### B.1.4 Proof of Proposition 5

*Proof.* By Cauchy-Schwartz inequality, we have

$$\sum_{\ell \in T} \|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_{\ell \cdot}\|_2 \leq \sqrt{|T|} \|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_T\|_{\mathrm{F}},$$

and note that given two matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, $(\boldsymbol{U}\boldsymbol{V})_T = (\boldsymbol{U})_T\boldsymbol{V}$ where $T$ is a subset of rows indices. We also use the fact that $\|\boldsymbol{U}\boldsymbol{V}\|_{\mathrm{F}} \leq \|\boldsymbol{U}\| \|\boldsymbol{V}\|_{\mathrm{F}}$. We consider two cases:

- For even $k$, we have

  $$\begin{aligned} &\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_T\|_{\mathrm{F}} = \|(\boldsymbol{\Delta})_T \boldsymbol{\Delta}^{(k)}\boldsymbol{B}\|_{\mathrm{F}} \\ &\leq \|(\boldsymbol{\Delta})_T\| \|\boldsymbol{\Delta}^{(k)}\boldsymbol{B}\|_{\mathrm{F}} = \sqrt{\lambda_{\max}((\boldsymbol{\Delta})_T^\top (\boldsymbol{\Delta})_T)} \|\boldsymbol{\Delta}^{(k)}\boldsymbol{B}\|_{\mathrm{F}}. \end{aligned}$$

  Note that $(\boldsymbol{\Delta})_T$ is equivalent to the incidence matrix of a subgraph with only $T$ edges, which allows us to bound,

  $$\lambda_{\max}((\boldsymbol{\Delta})_T^\top (\boldsymbol{\Delta})_T)) \leq \max_{(u,v) \in T} \{d_u + d_v\} \leq 2d_{\max}$$

  where $d_i$ is the degree of node $i$.

- For odd $k$, we have

  $$\begin{aligned} &\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_T\|_{\mathrm{F}} = \|(\boldsymbol{\Delta}^\top)_T \boldsymbol{\Delta}^{(k)}\boldsymbol{B}\|_{\mathrm{F}} \\ &\leq \|(\boldsymbol{\Delta}^\top)_T\| \|\boldsymbol{\Delta}^{(k)}\boldsymbol{B}\|_{\mathrm{F}} = \sqrt{\lambda_{\max}(\boldsymbol{\Delta}^{(2)}_{T\times T})} \|\boldsymbol{\Delta}^{(k)}\boldsymbol{B}\|_{\mathrm{F}}, \end{aligned}$$

  where $\boldsymbol{\Delta}^{(2)}_{T\times T} \in \mathbb{R}^{|T|\times|T|}$ is the principal submatrix of $\boldsymbol{\Delta}^{(2)}$ indexed by $T$. By Cauchy's interlacing theorem, the maximum eigenvalue of the submatrix is upper bounded, so

  $$\lambda_{\max}(\boldsymbol{\Delta}^{(2)}_{T\times T}) \leq \lambda_{\max}(\boldsymbol{\Delta}^{(2)}) \leq 2d_{\max}.$$

  Therefore, for all $k$, $\|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_T\|_{\mathrm{F}} \leq \sqrt{2d_{\max}} \|\boldsymbol{\Delta}^{(k)}\boldsymbol{B}\|_{\mathrm{F}}$. To conclude the proof, note

  $$\begin{aligned} &\sum_{\ell \in T} \|(\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B})_{\ell \cdot}\|_2 \leq \sqrt{|T|}\sqrt{2d_{\max}} \|\boldsymbol{\Delta}^{(k)}\boldsymbol{B}\|_{\mathrm{F}} \\ &\leq \sqrt{|T|}\sqrt{2d_{\max}} \|\boldsymbol{\Delta}^{(k)}\| \|\boldsymbol{B}\|_{\mathrm{F}} \leq (2d_{\max})^{\frac{k+1}{2}} \sqrt{|T|} \|\boldsymbol{B}\|_{\mathrm{F}}. \end{aligned}$$

□

### B.1.5 Proof of Theorem 25

We show the convergence of Alg. 5 by proving a modified version of[122] Proposition 1. The superscript $(m)$ denotes the values of $\boldsymbol{B}, \boldsymbol{Z}, \mathcal{U}$ at the $m$th iteration of the loop inside Alg. 5.

**Proposition 6** (Convergence to a feasible solution)**.** If $\tau \geq \mu$, then the primal residual $r^{(m)} = \|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B}^{(m)} - \boldsymbol{Z}^{(m)}\|_{\mathrm{F}}$ and the dual residual $s^{(m+1)} = \|\tau(\boldsymbol{\Delta}^{(k+1)})^\top(\boldsymbol{Z}^{(m+1)} - \boldsymbol{Z}^{(m)})\|_{\mathrm{F}}$ of Alg. 5 satisfy that $\lim_{m\to\infty} r^{(m)} = 0$ and $\lim_{m\to\infty} s^{(m)} = 0$.

*Proof.* Denote $\boldsymbol{D} = \boldsymbol{\Delta}^{(k+1)}$, and $\rho_\lambda(\cdot) = \rho(\cdot; \lambda, \gamma)$. Recall from Assumption 1 (c) that there exists $\mu > 0$ such that $\rho_\lambda(\|\mathbf{x}\|_2) + \frac{\mu}{2}\|\mathbf{x}\|_2^2$ is convex. Now consider the Lagrangian $\mathcal{L}(\boldsymbol{B}, \boldsymbol{Z}, \mathcal{U})$ with regard to the $\ell$-th row $\mathbf{z}_\ell$ of $\boldsymbol{Z} = [\mathbf{z}_1^\top, ...\mathbf{z}_r^\top]^\top$, assuming all other variables are fixed:

$$\begin{aligned}&\rho_\lambda(\|\mathbf{z}_\ell\|_2) + \frac{\tau}{2}\|\mathbf{z}_\ell - \boldsymbol{c}_1\|_2^2 + c_2\\ =&\rho_\lambda(\|\mathbf{z}_\ell\|_2) + \frac{\tau}{2}\|\mathbf{z}_\ell\|_2^2 - \tau\langle \mathbf{z}_\ell, \boldsymbol{c}_1\rangle + \frac{\tau}{2}\|\boldsymbol{c}_1\|_2^2 + c_2\end{aligned}$$

where $\boldsymbol{c}_1$ and $c_2$ represent terms of $\mathcal{L}(\boldsymbol{B}, \boldsymbol{Z}, \mathcal{U})$ that do not depend on $\mathbf{z}_\ell$. With our choice of $\tau \geq \mu$, then $\mathcal{L}(\boldsymbol{B}, \boldsymbol{Z}, \mathcal{U})$ is convex with regard to each of $\boldsymbol{B}$, $\mathcal{U}$, and for each row of $\boldsymbol{Z}$, allowing us to apply[195] Theorem 5.1. Therefore, Alg. 5 converges to limit points $\boldsymbol{B}^\star, \boldsymbol{Z}^\star, \mathcal{U}^\star$.

Then it follows that the dual residual $\lim_{m\to\infty} s^{(m)} = \|\tau\mathcal{D}^\top(\boldsymbol{Z}^\star - \boldsymbol{Z}^\star)\|_{\mathrm{F}} = 0$. For the primal residual, notice that the $\mathcal{U}$ update step in line 10 of Alg. 5 also shows that $\forall m, t \geq 0$,

$$\mathcal{U}^{(m+t)} = \mathcal{U}^{(m)} + \sum_{i=1}^{t}(\mathcal{D}\boldsymbol{B}^{(m+i)} - \boldsymbol{Z}^{(m+i)}).$$

Fixing $t$ and setting $m \to \infty$, we have

$$\mathcal{U}^\star = \mathcal{U}^\star + t(\mathcal{D}\boldsymbol{B}^\star - \boldsymbol{Z}^\star)$$

holds $\forall t \geq 0$. Hence, $\mathcal{D}\boldsymbol{B}^\star - \boldsymbol{Z}^\star = \mathbf{0}$, and therefore $\lim_{m\to\infty} r^{(m)} = \|\mathcal{D}\boldsymbol{B}^\star - \boldsymbol{Z}^\star\|_{\mathrm{F}} = 0$. □

This proposition shows that the algorithm in the limit achieves primal and dual feasibility, and that the Augmented Lagrangian in (4.25) with $\boldsymbol{Z}^\star$ and $\mathcal{U}^\star$ becomes the original GTF formulation in (4.9). $\boldsymbol{B}$ that is produced by every iteration of Alg. 5 is a stationary point of (4.25) with fixed $\boldsymbol{Z}$ and $\mathcal{U}$. As a result, $\boldsymbol{B}^\star$ is a stationary point of (4.9).

### B.1.6 Semi-Supervised Learning Algorithmic Details

Compared to the vector-valued GTF problem in (4.9), the semi-supervised learning problem in (4.26) has additional variables, such as $\boldsymbol{M}, \boldsymbol{R}$. Therefore, (4.26) has a different Augmented Lagrangian equation, which was then optimized using Alg. 8 which we present below for completeness.

$$\begin{aligned}\mathcal{L}'(\boldsymbol{B}, \boldsymbol{Z}, \boldsymbol{U}) = &\frac{1}{2}\|\boldsymbol{M}(\boldsymbol{Y} - \boldsymbol{B})\|_{\mathrm{F}}^2 + \epsilon\|\boldsymbol{R} - \boldsymbol{B}\|_{\mathrm{F}}^2 + h(\boldsymbol{Z}; \lambda, \gamma)\\ &+ \frac{\tau}{2}\|\boldsymbol{\Delta}^{(k+1)}\boldsymbol{B} - \boldsymbol{Z} + \boldsymbol{U}\|_{\mathrm{F}}^2 - \frac{\tau}{2}\|\boldsymbol{U}\|_{\mathrm{F}}^2\end{aligned}$$

We used the proximal operators of $\ell_1$, SCAD, MCP as derived in[160].

**Algorithm 8** ADMM for Semi-Supervised Learning

**Inputs:** $\boldsymbol{Y}, \boldsymbol{\Delta}^{(k+1)}$, $\boldsymbol{M}$, $\boldsymbol{R}$, and parameters $\lambda$, $\gamma$, $\tau$, $\epsilon$
**Initialize:**
    $\mathcal{D} \leftarrow \boldsymbol{\Delta}^{(k+1)}$, $\boldsymbol{Z} \leftarrow \mathcal{D}\boldsymbol{B}$, $\boldsymbol{U} \leftarrow \mathcal{D}\boldsymbol{B} - \boldsymbol{Z}$,
    $\boldsymbol{B} \leftarrow \boldsymbol{Y}$ or $\boldsymbol{B}_{\texttt{init}}$ if given.
**repeat**
    **for** $j \leftarrow$ `1 to num_cols(`$\boldsymbol{B}$`)` **do**
        $\boldsymbol{B}_{\cdot j} \leftarrow (\boldsymbol{M}^\top \boldsymbol{M} + \epsilon \boldsymbol{I} + \tau \mathcal{D}^\top \mathcal{D})^{-1} (\boldsymbol{M}^\top \boldsymbol{M} \boldsymbol{Y}_{\cdot j} + \epsilon \boldsymbol{R}_{\cdot j} + \tau \mathcal{D}^\top (\boldsymbol{Z}_{\cdot j} - \boldsymbol{U}_{\cdot j}))$
    **end for**
    **for** $\ell \leftarrow$ `1 to num_rows(`$\mathcal{D}\boldsymbol{B}$`)` **do**
        $\boldsymbol{Z}_{\ell \cdot} \leftarrow \texttt{Prox}_\rho(\|\mathcal{D}_{\ell \cdot}\boldsymbol{B} + \boldsymbol{U}_{\ell \cdot}\|_2; \lambda/\tau)$
    **end for**
    $\boldsymbol{U} \leftarrow \boldsymbol{U} + \mathcal{D}\boldsymbol{B} - \boldsymbol{Z}$
**until** termination

# References


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

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

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

[4] M. Puschel and J. M. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3572–3585, 2008.

[5] M. Püschel and J. M. Moura, "Algebraic signal processing theory," *arXiv preprint cs/0612077*, 2006.

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

[7] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, pp. 129–150, Mar. 2011.

[8] S. K. Narang, G. Shen, and A. Ortega, "Unidirectional graph-based wavelet transforms for efficient data gathering in sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Dallas, TX, Mar. 2010, pp. 2902–2905.

[9] I. Z. Pesenson, "Sampling in Paley-Wiener spaces on combinatorial graphs," *Trans. Amer. Math. Soc.*, vol. 360, no. 10, pp. 5603–5627, May 2008.

[10] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, May 2013, pp. 5445–5449.

[11] P. Liu, X. Wang, and Y. Gu, "Coarsening graph signal with spectral invariance," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Florence, May 2014, pp. 1075–1079.

[12] S. Chen, F. Cerda, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovačević, "Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2879–2893, Jun. 2014.

[13] A. Sandryhaila and J. M. F. Moura, "Classification via regularization on graphs," in *IEEE GlobalSIP*, Austin, TX, Dec. 2013, pp. 495–498.

[14] V. N. Ekambaram, B. A. G. Fanti, and K. Ramchandran, "Wavelet-regularized graph semi-supervised learning," in *Proc. IEEE Glob. Conf. Signal Information Process.*, Austin, TX, Dec. 2013, pp. 423 – 426.

[15] X. Zhang, X. Dong, and P. Frossard, "Learning of structured graph dictionaries," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Kyoto, Japan, 2012, pp. 3373–3376.

[16] D. Thanou, D. I. Shuman, and P. Frossard, "Parametric dictionary learning for graph signals," in *IEEE GlobalSIP*, Austin, TX, Dec. 2013, pp. 487–490.

[17] P.-Y. Chen and A. O. Hero, "Local Fiedler vector centrality for detection of deep and overlapping communities in networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Florence, 2014, pp. 1120 – 1124.

[18] A. Sandryhaila and J. M. F. Moura, "Big data processing with signal processing on graphs," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, 2014.

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

[20] R. Varma, S. Chen, and J. Kovačević, "Graph topology recovery for regular and irregular graphs," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2017, pp. 1–5.

[21] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[22] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3572–3585, Aug. 2008.

[23] ——, "Algebraic signal processing theory: 1-D space," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3586–3599, Aug. 2008.

[24] A. Sandryhaila and J. M. Moura, "Discrete Signal Processing on Graphs: Frequency Analysis." *IEEE Trans. Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[25] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of signal processing.* Cambridge University Press, 2014.

[26] S. Chen, R. Varma, A. Singh, and J. Kova\v{c}evi\'c, "Signal representations on graphs: Tools and applications," *arXiv preprint arXiv:1512.05406*, 2015.

[27] S. Chen, R. Varma, A. Singh, and J. Kovacević, "Signal recovery on graphs: Random versus experimentally designed sampling," in *2015 International Conference on Sampling Theory and Applications (SampTA)*, May 2015, pp. 337–341.

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

[29] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal Recovery on Graphs: Fundamental Limits of Sampling Strategies," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 539–554, Dec. 2016.

[30] Y.-X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani, "Trend filtering on graphs," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3651–3691, 2016.

[31] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[32] R. J. Tibshirani, *The solution path of the generalized lasso*. Stanford University, 2011.

[33] M. Elad and A. M. Bruckstein, "A generalized uncertainty principle and sparse representation in pairs of bases," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2558–2567, 2002.

[34] D. L. Donoho and P. B. Stark, "Uncertainty principles and signal recovery," *SIAM Journal on Applied Mathematics*, vol. 49, no. 3, pp. 906–931, 1989.

[35] A. Agaskar and Y. M. Lu, "A Spectral Graph Uncertainty Principle." *IEEE Trans. Information Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.

[36] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.

[37] O. Teke and P. P. Vaidyanathan, "Uncertainty principles and sparse eigenvectors of graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5406–5420, 2017.

[38] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge University Press, 2014, http://www.fourierandwavelets.org/. [Online]. Available: http://www.fourierandwavelets.org/

[39] J. Kovačević and M. Püschel, "Algebraic signal processing theory: Sampling for infinite and finite 1-D space," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 242–257, Jan. 2010.

[40] M. Unser, "Sampling – 50 years after Shannon," *Proc. IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.

[41] S. Chen, A. Sandryhaila, and J. Kovačević, "Sampling theory for graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brisbane, Australia, Apr. 2015.

[42] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2014, pp. 3864–3868.

[43] V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran., "Multiresolution graph signal processing via circulant structures," in *2013 IEEE DSP/SPE*, Napa, CA, Aug. 2013, pp. 112–117.

[44] E. J. Candès, "Compressive sampling," in *Int. Congr. Mathematicians*, Madrid, Spain, 2006.

[45] R. A. Horn and C. Johnson, *Matrix Analysis*. Cambridge: Cambridge University Press, 1985.

[46] M. Püschel and J. Kovačević, "Real, tight frames with maximal robustness to erasures," in *Proc. Data Compr. Conf.*, Snowbird, UT, Mar. 2005, pp. 63–72.

[47] A. Sandryhaila, A. Chebira, C. Milo, J. Kovačević, and M. Püschel, "Systematic construction of real lapped tight frame transforms," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2256–2567, May 2010.

[48] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[49] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[50] D. L. Donoho and M. Elad, "On the stability of basis pursuit in the presence of noise," *Signal Process.*, vol. 86, no. 3, pp. 511–532, Mar. 2006.

[51] E. J. Candès, Y. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis*, vol. 31, pp. 59–73, 2010.

[52] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, pp. 1207–1223, 2006.

[53] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 985–1042, 2010.

[54] A. Sandryhaila and J. M. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, 2014.

[55] P. M. Weichsel, "The Kronecker product of graphs," *Proceedings of the American mathematical society*, vol. 13, no. 1, pp. 47–52, 1962.

[56] R. A. Varma and J. Kovacevic, "Sampling Theory for Graph Signals on Product Graphs," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov. 2018, pp. 768–772.

[57] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 446–475, 2018.

[58] J. Kovačević and A. Chebira, "Life beyond bases: The advent of frames (Part I)," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 86–104, Jul. 2007.

[59] ——, "Life beyond bases: The advent of frames (Part II)," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 115–125, Sep. 2007.

[60] H. Avron and C. Boutsidis, "Faster subset selection for matrices and applications," *SIAM J. Matrix Analysis Applications*, vol. 34, no. 4, pp. 1464–1499, 2013.

[61] M. Jackson, *Social and Economic Networks*. Princeton University Press, 2008.

[62] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[63] L. V. Tran, V. H. Vu, and K. Wang, "Sparse random graphs: Eigenvalues and eigenvectors," *Random Struct. Algorithms*, vol. 42, no. 1, pp. 110–134, 2013.

[64] E. J. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, pp. 110–134, 2007.

[65] S. Chen, R. Varma, A. Singh, and J. Kovacević, "Signal recovery on graphs: Random versus experimentally designed sampling," in *2015 International Conference on Sampling Theory and Applications (SampTA)*, May 2015, pp. 337–341.

[66] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal recovery on graphs: Random versus experimentally designed sampling," in *SampTA*, Washington, DC, May 2015.

[67] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, Jun. 1998.

[68] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, Oct. 1999.

[69] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. LinkKDD*, 2005, pp. 36–43.

[70] S. Chen, A. Sandryhaila, G. Lederman, Z. Wang, J. M. F. Moura, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovačević, "Signal inpainting on graphs via total variation minimization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Florence, May 2014, pp. 8267–8271.

[71] 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.

[72] R. Castro, R. Willett, and R. Nowak, "Faster rates in regression via active learning," in *Proc. Neural Information Process. Syst.*, Vancouver, Dec. 2005.

[73] R. M. Castro and R. D. Nowak, "Minimax bounds for active learning," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2339–2353, 2008.

[74] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, 2015, to appear.

[75] J. Sharpnack and A. Singh, "Identifying graph-structured activation patterns in networks," in *Advances in Neural Information Processing Systems*, 2010, pp. 2137–2145.

[76] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, 2015, to appear.

[77] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, "Fast approximation of matrix coherence and statistical leverage," *Journal of Machine Learning Research*, vol. 13, no. Dec, pp. 3475–3506, 2012.

[78] X. Wang, M. Wang, and Y. Gu, "A distributed tracking algorithm for reconstruction of graph signals," *IEEE Journal of Selected Topics on Signal Processing*, vol. 9, no. 4, Jun. 2015.

[79] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Distributed algorithm for graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brisbane, Apr. 2015.

[80] A. Sandryhaila, J. Kovačević, and M. Püschel, "Algebraic signal processing theory: 1-D nearest-neighbor models," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2247–2259, May 2012.

[81] X. Zhu, "Semi-supervised learning literature survey," Univ. Wisconsin-Madison, Tech. Rep. 1530, 2005.

[82] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Governance in social media: A case study of the Wikipedia promotion process," in *Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[83] J. Dall and M. Christensen, "Random geometric graphs," *Physical review E*, vol. 66, no. 1, p. 016121, 2002.

[84] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14, New York, New York, USA, 2014, pp. 492–501.

[85] R. Varma, S. Chen, and J. Kovačević, "Spectrum-blind signal recovery on graphs," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2015, pp. 81–84.

[86] M. Rudelson and R. Vershynin, "On sparse reconstruction from Fourier and Gaussian measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 61, no. 8, pp. 1025–1045, 2008.

[87] E. J. Candès and T. Tao, "Near optimal signal recovery from random projections: universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[88] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.

[89] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes rendus mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[90] L. V. Tran, V. H. Vu, and K. Wang, "Sparse random graphs: Eigenvalues and eigenvectors," *Random Structures & Algorithms*, vol. 42, no. 1, pp. 110–134, 2013.

[91] F. Krahmer, H. Rauhut, and R. Ward, "Local coherence sampling in compressed sensing," in *Proceedings of the 10th International Conference on Sampling Theory and Applications*, 2013, pp. 476–480.

[92] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning theory and kernel machines*. Springer, 2003, pp. 144–158.

[93] R. K. Ando and T. Zhang, "Learning on graph with Laplacian regularization," in *Advances in neural information processing systems*, 2007, pp. 25–32.

[94] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[95] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[96] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[97] L. Grasedyck, D. Kressner, and C. Tobler, "A literature survey of low-rank tensor approximation techniques," *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 53–78, 2013.

[98] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[99] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor factorization using auxiliary information," *Data Mining and Knowledge Discovery*, vol. 25, no. 2, pp. 298–324, 2012.

[100] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, p. 025010, 2011.

[101] M. Yuan and C.-H. Zhang, "On tensor completion via nuclear norm minimization," *Foundations of Computational Mathematics*, vol. 16, no. 4, pp. 1031–1068, 2016.

[102] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

[103] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.

[104] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[105] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[106] S. Chen, A. Sandryhaila, and J. Kovačević, "Sampling theory for graph signals," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3392–3396.

[107] M. Mangia, R. Rovatti, and G. Setti, "Rakeness in the design of analog-to-information conversion of sparse and localized signals," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 1001–1014, 2012.

[108] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Low-complexity multiclass encryption by compressed sensing," *IEEE transactions on signal processing*, vol. 63, no. 9, pp. 2183–2195, 2015.

[109] E. van den Berg and M. P. Friedlander, *SPGL1: A solver for large-scale sparse reconstruction*. June, 2007.

[110] T. Ji, S. Chen, R. Varma, and J. Kovačević, "Energy-efficient route planning for autonomous aerial vehicles based on graph signal recovery," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2015, pp. 1414–1421.

[111] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing," *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.

[112] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *International Conference on Computational Learning Theory*. Springer, 2004, pp. 624–638.

[113] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[114] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "\ell_1 Trend Filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009.

[115] P. Bühlmann and S. Van De Geer, *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[116] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[117] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, vol. 38, no. 2, pp. 894–942, 2010.

[118] P.-L. Loh and M. J. Wainwright, "Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima," in *Advances in Neural Information Processing Systems*, 2013, pp. 476–484.

[119] P.-L. Loh, "Statistical consistency and asymptotic normality for high-dimensional robust $ M $-estimators," *The Annals of Statistics*, vol. 45, no. 2, pp. 866–896, 2017.

[120] C.-H. Zhang and T. Zhang, "A general theory of concave regularization for high-dimensional sparse estimation problems," *Statistical Science*, vol. 27, no. 4, pp. 576–593, 2012.

[121] P. Breheny and J. Huang, "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection," *The Annals of Applied Statistics*, vol. 5, no. 1, p. 232, 2011.

[122] S. Ma and J. Huang, "A concave pairwise fusion approach to subgroup analysis," *Journal of the American Statistical Association*, vol. 112, no. 517, pp. 410–423, 2017.

[123] D. Hallac, J. Leskovec, and S. Boyd, "Network lasso: Clustering and optimization in large graphs," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2015, pp. 387–396.

[124] A. Jung, A. O. Hero III, A. Mara, and S. Jahromi, "Semi-supervised learning via sparse label propagation," *arXiv preprint arXiv:1612.01414*, 2016.

[125] A. Jung, N. Tran, and A. Mara, "When is network lasso accurate?" *Frontiers in Applied Mathematics and Statistics*, vol. 3, p. 28, 2018.

[126] E. Mammen and S. van de Geer, "Locally adaptive regression splines," *The Annals of Statistics*, vol. 25, no. 1, pp. 387–413, 1997.

[127] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[128] T. F. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Transactions on Image processing*, vol. 10, no. 2, pp. 231–241, 2001.

[129] F. Mahmood, N. Shahid, U. Skoglund, and P. Vandergheynst, "Adaptive graph-based total variation for tomographic reconstructions," *IEEE Signal Processing Letters*, vol. 25, no. 5, pp. 700–704, 2018.

[130] R. J. Tibshirani, "Adaptive piecewise polynomial estimation via trend filtering," *The Annals of Statistics*, vol. 42, no. 1, pp. 285–323, 2014.

[131] J. Sharpnack, A. Singh, and A. Rinaldo, "Sparsistency of the edge lasso over graphs," in *Artificial Intelligence and Statistics*, 2012, pp. 1028–1036.

[132] Z. Harchaoui and C. Lévy-Leduc, "Multiple change-point estimation with a total variation penalty," *Journal of the American Statistical Association*, vol. 105, no. 492, pp. 1480–1493, 2010.

[133] J.-C. Hütter and P. Rigollet, "Optimal rates for total variation denoising," in *Conference on Learning Theory*, 2016, pp. 1115–1146.

[134] A. S. Dalalyan, M. Hebiri, and J. Lederer, "On the prediction performance of the lasso," *Bernoulli*, vol. 23, no. 1, pp. 552–581, 2017.

[135] K. Lin, J. Sharpnack, A. Rinaldo, and R. J. Tibshirani, "Approximate Recovery in Changepoint Problems, from $\ell_2$ Estimation Error Rates," *arXiv:1606.06746 [math, stat]*, Jun. 2016, arXiv: 1606.06746. [Online]. Available: http://arxiv.org/abs/1606.06746

[136] N. Tran, S. Basirian, and A. Jung, "When is network lasso accurate: The vector case," *arXiv preprint arXiv:1710.03942*, 2017.

[137] S. A. Van De Geer and P. Bühlmann, "On the conditions used to prove oracle results for the Lasso," *Electronic Journal of Statistics*, vol. 3, pp. 1360–1392, 2009.

[138] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.

[139] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Transactions on Signal processing*, vol. 54, no. 12, pp. 4634–4643, 2006.

[140] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, 2010.

[141] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010.

[142] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral Image Classification Using Dictionary-Based Sparse Representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[143] Y. Li and Y. Chi, "Off-the-Grid Line Spectrum Denoising and Estimation With Multiple Measurement Vectors," *IEEE Transactions on Signal Processing*, vol. 64, no. 5, pp. 1257–1269, Mar. 2016.

[144] Y. C. Eldar and M. Mishali, "Robust Recovery of Signals From a Structured Union of Subspaces," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[145] M. E. Davies and Y. C. Eldar, "Rank Awareness in Joint Sparse Recovery," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1135–1146, Feb. 2012.

[146] P.-L. Loh and M. J. Wainwright, "Support recovery without incoherence: A case for nonconvex regularization," *The Annals of Statistics*, vol. 45, no. 6, pp. 2455–2482, 2017.

[147] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, 2002, pp. 585–591.

[148] ——, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[149] P. P. Talukdar and K. Crammer, "New regularized algorithms for transductive learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 442–457.

[150] K. Q. Weinberger, F. Sha, Q. Zhu, and L. K. Saul, "Graph Laplacian regularization for large-scale semidefinite programming," in *Advances in neural information processing systems*, 2007, pp. 1489–1496.

[151] J. Pang, G. Cheung, A. Ortega, and O. C. Au, "Optimal graph Laplacian regularization for natural image denoising," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2294–2298.

[152] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, no. Nov, pp. 1817–1853, 2005.

[153] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[154] L. Chen and Y. Gu, "The convergence guarantees of a non-convex approach for sparse recovery," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3754–3767, 2014.

[155] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997.

[156] R. Varma, H. Lee, Y. Chi, and J. Kova\v{c}evi\'c, "Improving Graph Trend Filtering with Non-Convex Penalties," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2019. [Online]. Available: https://users.ece.cmu.edu/~yuejiec/papers/noncvxGTF.pdf

[157] F. Chung and M. Radcliffe, "On the spectra of general random graphs," *the electronic journal of combinatorics*, vol. 18, no. 1, p. 215, 2011.

[158] A. Blum, J. Hopcroft, and R. Kannan, "Foundations of data science," *Vorabversion eines Lehrbuchs*, 2016.

[159] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[160] J. Huang, P. Breheny, and S. Ma, "A Selective Review of Group Selection in High-Dimensional Models," *Statistical Science*, vol. 27, no. 4, pp. 481–499, Nov. 2012. [Online]. Available: https://projecteuclid.org/euclid.ss/1356098552

[161] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[162] M. Defferrard, L. Martin, R. Pena, and N. Perraudin, "Pygsp: Graph signal processing in python," *URL https://github. com/epfl-lts2/pygsp*, 2017.

[163] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using NetworkX," 2008.

[164] G. Boeing, "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, Sep. 2017, arXiv: 1611.01890. [Online]. Available: http://arxiv.org/abs/1611.01890

[165] A. Asuncion and D. Newman, *UCI Machine Learning Repository*, 2007.

[166] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53–94, 2006.

[167] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[168] M. Vetterli and J. Kovačević, *Wavelets and subband coding*. Prentice Hall Englewood Cliffs, 1995, vol. 995.

[169] U. V. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing.*, vol. 17, pp. 395–416, 2007.

[170] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, May 2015.

[171] A. K. J. Sharpnack and A. Singh, "Detecting activations over graphs using spanning tree wavelet bases," in *AISTATS*, Scottsdale, AZ, Apr. 2013.

[172] 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, pp. 83–98, May 2013.

[173] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmon. Anal.*, pp. 53–94, Jul. 2006.

[174] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Trans. Signal Process.*, vol. 60, pp. 2786–2799, Jun. 2012.

[175] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, February 2015, to appear.

[176] D. Thanou, D. I. Shuman, and P. Frossard, "Learning parametric dictionaries for signals on graphs," *IEEE Trans. Signal Process.*, vol. 62, pp. 3849–3862, Jun. 2014.

[177] M. Gavish, B. Nadler, and R. R. Coifman, "Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 367–374.

[178] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," in *Proc. IEEE INFOCOM*, vol. 3, Mar. 2013, p. 1848–1857.

[179] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[180] S. Chen, A. Singh, and J. Kovačević, "Multiresolution representations for piecewise-smooth signals on graphs," *arXiv preprint arXiv:1803.02944*, 2018.

[181] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Representations of piecewise smooth signals on graphs," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 6370–6374.

[182] R. Kondor, N. Teneva, and V. Garg, "Multiresolution matrix factorization," in *International Conference on Machine Learning*, 2014, pp. 1620–1628.

[183] D. Needell and R. Ward, "Stable image reconstruction using total variation minimization," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 1035–1058, 2013.

[184] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Signal, Syst. Comput.*, vol. 1, Pacific Grove, CA, Nov. 1993, pp. 40–44.

[185] D. Gleich. The MatlabBGL Matlab library, http://www.cs.purdue.edu/homes/dgleich/packages/matlab bgl/index.html.

[186] Y.-X. Wang, J. Sharpnack, A. Smola, and R. J. Tibshirani, "Trend filtering on graphs," in *AISTATS*, San Diego, CA, May 2015.

[187] J. Zhang and J. M. F. Moura, "Diffusion in social networks as sis epidemics: Beyond full mixing and complete graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 537–551, Aug. 2014.

[188] ——, "Accounting for topology in spreading contagion in non-complete networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 2681–2684.

[189] R. Willett, R. Nowak, and R. M. Castro, "Faster rates in regression via active learning," in *Advances in Neural Information Processing Systems*, 2006, pp. 179–186.

[190] J. Haupt, R. M. Castro, and R. Nowak, "Distilled sensing: Adaptive sampling for sparse detection and estimation," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6222–6235, 2011.

[191] Y. Wang, A. W. Yu, and A. Singh, "On computationally tractable selection of experiments in measurement-constrained regression models," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5238–5278, 2017.

[192] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, Jul. 2017, arXiv: 1611.08097. [Online]. Available: http://arxiv.org/abs/1611.08097

[193] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv:1609.02907 [cs, stat]*, Feb. 2017, arXiv: 1609.02907. [Online]. Available: http://arxiv.org/abs/1609.02907

[194] M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, Feb. 2019, google-Books-ID: IluHDwAAQBAJ.

[195] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.

This thesis was written by Rohan Anilkumar Varma