Yuichi Tanaka, Yonina C. Eldar,
Antonio Ortega, and Gene Cheung

Sampling Signals on Graphs

From theory to applications

©ISTOCKPHOTO.COM/ALISEFOX

The study of sampling signals on graphs, with the goal of building an analog of sampling for standard signals in the time and spatial domains, has attracted considerable attention recently. Beyond adding to the growing theory on graph signal processing (GSP), sampling on graphs has various promising applications. In this article, we review the current progress on sampling over graphs, focusing on theory and potential applications.

Although most methodologies used in graph signal sampling are designed to parallel those used in sampling for standard signals, sampling theory for graph signals significantly differs from the theory of Shannon–Nyquist and shift-invariant (SI) sampling. This is due, in part, to the fact that the definitions of several important properties, such as shift invariance and bandlimitedness, are different in GSP systems. Throughout this review, we discuss similarities and differences between standard and graph signal sampling and highlight open problems and challenges.

Introduction

Sampling is one of the fundamental tenets of digital signal processing (see [1] and the references therein). As such, it has been studied extensively for decades and continues to draw considerable research efforts. Standard sampling theory relies on concepts of frequency domain analysis, SI signals, and bandlimitedness [1]. The sampling of time and spatial domain signals in SI spaces is one of the most important building blocks of digital signal processing systems. However, in the big data era, the signals we need to process often have other types of connections and structure, such as network signals described by graphs.

This article provides a comprehensive overview of the theory and algorithms for the sampling of signals defined on graph domains, i.e., graph signals. GSP [2]–[4]—a fast developing field in the signal processing community—generalizes key signal processing ideas for signals defined on regular domains to discrete-time signals defined over irregular domains described abstractly by graphs. GSP has found numerous promising

Digital Object Identifier 10.1109/MSP.2020.3016908
Date of current version: 28 October 2020

applications across many engineering disciplines, including image processing, wireless communications, machine learning, and data mining [2], [3], [5].

Network data are pervasive and found in applications such as sensor, neuronal, transportation, and social networks. The number of nodes in such networks is often very large: processing and storing all of the data as is can require huge computation and storage resources, which may not be tolerable even in modern, high-performance communication and computer systems. Therefore, it is often of interest to reduce the amount of data while keeping the important information as much as possible. Sampling of graph signals addresses this issue: how one can reduce the number of samples on a graph and reconstruct the underlying signal, generalizing the standard sampling paradigm to graph signals.

GSP has found numerous promising applications across many engineering disciplines, including image processing, wireless communications, machine learning, and data mining.

Generalization of the sampling problem to GSP raises a number of challenges. First, for a given graph and graph operator, the notion of frequency for graph signals is mathematically straightforward, but the connection of these frequencies to the actual properties of signals of interest (and, thus, the practical meaning of concepts, such as bandlimitedness and smoothness) is still being investigated. Second, periodic sampling, widely used in traditional signal processing, is not applicable in the graph domain (e.g., it is unclear how to select "every other" sample). Although the theory of sampling on graphs and manifolds has been studied in the past (see [6], [7], and follow-up works), early works have not considered the problems inherent in applications, e.g., how to select the best set of nodes from a given graph. Therefore, developing practical techniques for sampling set selection that can adapt to local graph topology is very important. Third, the work to date has mostly focused on direct nodewise sampling, while there has been only limited work on developing more advanced forms of sampling, e.g., adapting SI sampling [1] to the graph setting [8]. Finally, graph signal sampling and reconstruction algorithms must be implemented efficiently to achieve a good tradeoff between accuracy and complexity.

To address these challenges, various graph sampling approaches have recently been developed (e.g., [7] and [9]–[14]) based on different notions of graph frequency, bandlimitedness, and shift invariance. For example, a common approach to define the graph frequency is based on the spectral decomposition of different variation operators, such as the adjacency matrix or variants of graph Laplacians. The proposed reconstruction procedures in the literature differ in their objective functions, leading to a tradeoff between accuracy and complexity. Our goal is to provide a broad overview of existing techniques, highlighting what is known to date to inspire further research on sampling over graphs and its use in a broad class of applications in signal processing and machine learning.

In what follows, we use boldfaced lowercase (uppercase) symbols to represent vectors (matrices); the ith element in a vector $\mathbf{x}$ is $x[i]$ or x_i; and the ith row, jth column of a matrix $\mathbf{X}$ is given by $[\mathbf{X}]_{ij}$. A subvector of $\mathbf{x}$ is denoted $\mathbf{x}_{\mathcal{S}}$ with indicator index set $\mathcal{S}$. Similarly, a submatrix of $\mathbf{X} \in \mathbb{R}^{N \times M}$ is denoted $\mathbf{X}_{\mathcal{RC}} \in \mathbb{R}^{|\mathcal{R}| \times |\mathcal{C}|}$, where indicator indices of its rows and columns are given by $\mathcal{R}$ and $\mathcal{C}$, respectively; $\mathbf{X}_{\mathcal{RR}}$ is simply written as $\mathbf{X}_{\mathcal{R}}$.

Review: GSP and sampling in Hilbert spaces

The basics of GSP

We denote by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a graph where $\mathcal{V}$ and $\mathcal{E}$ are the sets of vertices and edges, respectively. The number of vertices is $N = |\mathcal{V}|$ unless otherwise specified. We define an adjacency matrix $\mathbf{W}$ where entry $[\mathbf{W}]_{mn}$ represents the weight of the edge between vertices m and n; $[\mathbf{W}]_{mn} = 0$ for unconnected vertices. The degree matrix $\mathbf{D}$ is diagonal, with mth diagonal element $[\mathbf{D}]_{mm} = \Sigma_n [\mathbf{W}]_{mn}$. In this article, we consider undirected graphs without self-loops, i.e., $[\mathbf{W}]_{mn} = [\mathbf{W}]_{nm}$ and $[\mathbf{W}]_{nn} = 0$ for all m and n, but most theory and methods discussed can be extended to signals on directed graphs.

GSP uses different variation operators [2], [3] depending on the application and assumed signal and/or network models. Here, for concreteness, we focus on the graph Laplacian $\mathbf{L} := \mathbf{D} - \mathbf{W}$ or its symmetrically normalized version $\underline{\mathbf{L}} := \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. The extension to other variation operators (e.g., adjacency matrix) is possible with a proper modification of the basic operations discussed in this section. Since $\mathbf{L}$ is a real symmetric matrix, it always possesses an eigendecomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\top}$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ is an orthonormal matrix containing the eigenvectors $\mathbf{u}_i$, and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ consists of the eigenvalues λ_i. We refer to λ_i as the *graph frequency.*

A graph signal $x : \mathcal{V} \to \mathbb{R}$ is a function that assigns a value to each node. Graph signals can be written as vectors, $\mathbf{x}$, in which the nth element, $x[n]$, represents the signal value at node n. Note that any vertex labeling can be used, since a change in labeling simply results in row/column permutation of the various matrices, their corresponding eigenvectors, and the vectors representing graph signals. The graph Fourier transform (GFT) is defined as

$$\hat{x}[i] = \langle \mathbf{u}_i, \mathbf{x} \rangle = \sum_{n=0}^{N-1} u_i[n] x[n]. \tag{1}$$

Other GFT definitions can also be used without changing the framework. In this article, for simplicity, we assume real-valued signals. Although the GFT basis is real valued for undirected graphs, extensions to complex-valued GSP systems are straightforward.

A *linear graph filter* is defined by $\mathbf{G} \in \mathbb{R}^{N \times N}$, which, applied to $\mathbf{x}$, produces an output

$$\mathbf{y} = \mathbf{G}\mathbf{x}. \tag{2}$$

Vertex- and frequency-domain graph filter designs considered in the literature both lead to filters $\mathbf{G}$ that depend on the structure of the graph $\mathcal{G}$. *Vertex-domain filters* are defined as polynomials of the variation operator, i.e.,

$$\mathbf{y} = \mathbf{G}\mathbf{x} = \left(\sum_{p=0}^{P} c_p \mathbf{L}^p\right)\mathbf{x}, \tag{3}$$

where the output at each vertex is a linear combination of signal values in its P-hop neighborhood. In frequency-domain filter design, $\mathbf{G}$ is chosen to be diagonalized by $\mathbf{U}$ so that

$$\mathbf{y} = \mathbf{G}\mathbf{x} = \mathbf{U}\hat{g}(\boldsymbol{\Lambda})\mathbf{U}^{\top}\mathbf{x}, \tag{4}$$

where $\hat{g}(\boldsymbol{\Lambda}) := \operatorname{diag}(\hat{g}(\lambda_1), \ldots, \hat{g}(\lambda_N))$ is the graph frequency response. Filtering via (4) is analogous to filtering in the Fourier domain for conventional signals. When there exist repeated eigenvalues $\lambda_i = \lambda_j$, their graph frequency responses must be the same, i.e., $\hat{g}(\lambda_i) = \hat{g}(\lambda_j)$. If $\hat{g}(\lambda_i)$ is a Pth order polynomial, (4) coincides with vertex-domain filtering (3).

Generalized sampling in Hilbert spaces

We next briefly review generalized sampling in Hilbert spaces [1] (see Figure 1). We highlight generalized sampling with a known SI signal subspace [1], a generalization of Shannon–Nyquist sampling beyond bandlimited (BL) signals. Reviewing these cases will help us illustrate similarities and differences with respect to sampling and reconstruction in the graph setting.

Let x be a vector in a Hilbert space $\mathcal{H}$ and $c[n]$ be its nth sample, $c[n] = \langle s_n, x\rangle$, where $\{s_n\}$ is a Riesz basis, and $\langle \cdot, \cdot \rangle$ is an inner product. Denoting by S the set transformation corresponding to $\{s_n\}$, we can write the samples as $c = S^* x$, where $\cdot^*$ represents the adjoint. The subspace generated by $\{s_n\}$ is denoted by $\mathcal{S}$. In the SI setting, $s_n = s(t - nT)$ for a real function $s(t)$ and a given period T. The samples can then be expressed as

$$c[n] = \langle s(t-nT), x(t)\rangle = x(t) * s(-t)\,|_{t=nT}, \tag{5}$$

where $*$ denotes convolution. The continuous-time Fourier transform (CTFT) of $c[n]$, $C(\omega)$, can be written as

$$C(\omega) = R_{SX}(\omega), \tag{6}$$

where

$$R_{SX}(\omega) := \frac{1}{T}\sum_{k=-\infty}^{\infty} S^*\left(\frac{\omega - 2\pi k}{T}\right) X\left(\frac{\omega - 2\pi k}{T}\right) \tag{7}$$

is the sampled cross correlation. Thus, we may view sampling in the Fourier domain as multiplying the input spectrum by the filter's frequency response and subsequently aliasing the result with uniform intervals that depend on the sampling period. In BL sampling, $s(-t) = \operatorname{sinc}(t/T)$, where $\operatorname{sinc}(t) = \sin(\pi t)/(\pi t)$, while s(t) may be chosen more generally in the generalized sampling framework.

Recovery of the signal from its samples c is represented as

$$\tilde{x} = WHc = WH(S^* x), \tag{8}$$

where W is a set transformation corresponding to a basis $\{w_n\}$ for the reconstruction space, which spans a closed subspace $\mathcal{W}$ of $\mathcal{H}$. The transform H is called the *correction transformation* and operates on the samples c prior to recovery. The reconstruction problem is to choose H so that $\tilde{x}$ is either equal to x or as close as possible under a desired metric. Typically, solving this problem requires making assumptions about x, e.g., that it lies in a known subspace or is smooth.

In the SI setting, the recovery corresponding to (8) is given by

$$\tilde{x}(t) = \sum_{n\in\mathbb{Z}} (h[n] * c[n])\, w(t - nT), \tag{9}$$

where a discrete-time correction filter $h[n]$ is first applied to $c[n]$: the output $d[n] = h[n] * c[n]$ is interpolated by a filter $w(t)$ to produce the recovery $\tilde{x}(t)$.

Suppose that x lies in an arbitrary subspace $\mathcal{A}$ of $\mathcal{H}$, and assume that $\mathcal{A}$ is known. This is one of the well-studied signal models, i.e., signal priors, that have been considered in generalized sampling. With this subspace prior, x can be represented as

$$x = \sum d[n] a_n = Ad, \tag{10}$$

where $\{a_n\}$ is an orthonormal basis for $\mathcal{A}$, and $d[n]$ are the expansion coefficients of x. In the SI setting, $x(t)$ is written as

$$x(t) = \sum_{n\in\mathbb{Z}} d[n]\, a(t - nT) \tag{11}$$

for some sequence $d[n]$, where $a(t)$ is a real generator satisfying the Riesz condition. In the Fourier domain, (11) becomes

$$X(\omega) = D(e^{j\omega T}) A(\omega), \tag{12}$$

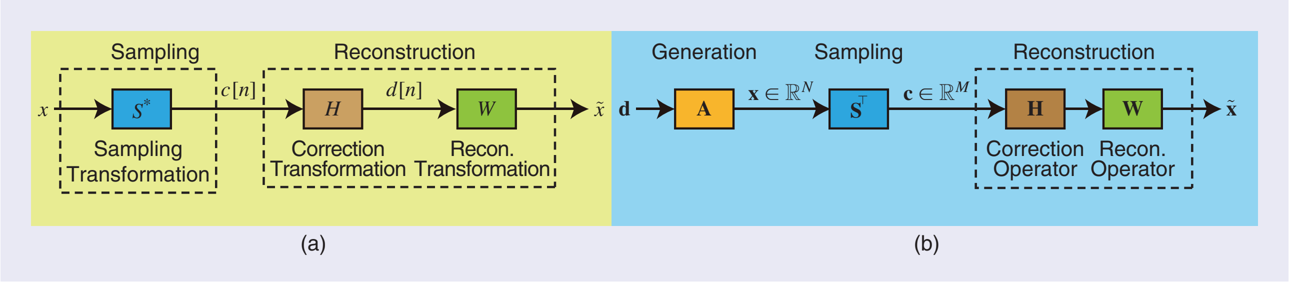


FIGURE 1. The generalized sampling frameworks for (a) sampling in Hilbert space and (b) its GSP counterpart.

where $A(\omega)$ is the CTFT of $a(t)$, and $D(e^{j\omega T})$ is the discrete-time Fourier transform of $d[n]$ and is $2\pi/T$ periodic.

In this article, we focus on generalized sampling for the unconstrained case, where an arbitrary transformation can be used as W. We can also consider generalized sampling for a predefined W (see [1] and the references therein). In the unconstrained setting, we may recover a signal in $\mathcal{A}$ by choosing $W = A$ in (8). If S^*A is invertible, then perfect recovery of any $x \in \mathcal{A}$ is possible by using $H = (S^*A)^{-1}$. Invertibility can be ensured by the direct-sum (DS) condition: $\mathcal{A}$ and $\mathcal{S}^{\perp}$ intersect only at the origin and span $\mathcal{H}$ jointly so that $\mathcal{H} = \mathcal{A} \oplus \mathcal{S}^{\perp}$. Under the DS condition, a unique recovery is obtained by an oblique projection operator onto $\mathcal{A}$ along $\mathcal{S}^{\perp}$ given by

$$\tilde{x} = A(S^*A)^{-1}S^*x = x. \tag{13}$$

In the SI setting, the frequency response of the correction filter is

$$H(\omega) = \frac{1}{R_{SA}(\omega)}, \tag{14}$$

where $R_{SA}(\omega)$ is given by (7).

If $\mathcal{A}$ and $\mathcal{S}^{\perp}$ intersect, then there is more than one signal in $\mathcal{A}$ that matches the sampled signal c. We may then consider several selection criteria to obtain an appropriate signal out of (infinitely) many candidates. Here, we consider the least-squares (LS) approach, but other methods, e.g., based on the minimax criterion, can be used as well [1]. The LS recovery is the minimum energy solution

$$\tilde{x} = \underset{x \in \mathcal{A},\, S^*x = c}{\operatorname{argmin}} \|x\|^2 \tag{15}$$

and is given by

$$\tilde{x} = A(S^*A)^{\dagger}S^*x. \tag{16}$$

The correction transformation is $H = (S^*A)^{\dagger}$, and $\cdot^{\dagger}$ represents the Moore–Penrose pseudoinverse. Its corresponding form in the SI setting is the same as (14), but $H(\omega) = 0$ if $R_{SA}(\omega) = 0$.

These results on sampling in Hilbert space are based on a signal model where $x \in \mathcal{A}$, with $\mathcal{A}$ assumed to be a known subspace. These results have also been extended to include various forms of smoothness on x as well as sparsity.

Graph sampling theory

In this section, we first describe a general graph signal sampling and reconstruction framework that is inspired by that of the "Generalized Sampling in Hilbert Spaces" section. Then, we discuss the graph signal subspaces proposed in the literature. Two definitions of graph signal sampling, which are generalizations of those studied in standard sampling, are also described. Finally, we present recovery experiments for BL and full-band signals, both of which can be perfectly reconstructed based on the proposed framework.

General sampling and recovery framework

We consider finite dimensional graphs and graph signals for which the generalized sampling in the "Generalized Sampling in Hilbert Spaces" section can be written in matrix form [15]. Similar to (10), we can assume any graph signal $\mathbf{x}$ is modeled by a known generator matrix $\mathbf{A} \in \mathbb{R}^{N \times K}$ $(K \leq N)$ and expansion coefficients $\mathbf{d} \in \mathbb{R}^K$ as follows:

$$\mathbf{x} := \mathbf{A}\mathbf{d}. \tag{17}$$

The graph sampling operator is a matrix $\mathbf{S} \in \mathbb{R}^{N \times M}$ $(M \leq N)$ that, without loss of generality, is assumed to have linearly independent columns that span a sampling space, $\mathcal{S} \subset \mathbb{R}^N$. The sampled signal is then given by

$$\mathbf{c} := \mathbf{S}^{\top}\mathbf{x} \in \mathbb{R}^M. \tag{18}$$

Since $\mathbf{A}$ is known, signal recovery may be given by using (16):

$$\tilde{\mathbf{x}} = \mathbf{A}\mathbf{H}\mathbf{c} = \mathbf{A}(\mathbf{S}^{\top}\mathbf{A})^{\dagger}\mathbf{S}^{\top}\mathbf{x}, \tag{19}$$

where the correction transform is given by $\mathbf{H} = (\mathbf{S}^{\top}\mathbf{A})^{\dagger}$. When the DS condition holds, $(\mathbf{S}^{\top}\mathbf{A})^{\dagger} = (\mathbf{S}^{\top}\mathbf{A})^{-1}$, and perfect recovery is obtained. In some cases, it may be better to select $\mathbf{W} \neq \mathbf{A}$, e.g., for more efficient implementation, so that the leftmost $\mathbf{A}$ in (19) would be replaced with $\mathbf{W}$ (as in Figure 1). Such predefined solutions have been studied in [8] and [15]. This is equivalent to the generalized sampling in Hilbert space described in the "Generalized Sampling in Hilbert Spaces" section.

Major challenges in graph signal sampling are selection and optimization of the generation and sampling matrices $\mathbf{A}$ and $\mathbf{S}$ as well as efficient implementation of the pseudoinverse described. In some cases, analogous to the SI setting in standard sampling, this inverse can be implemented by filtering in the graph Fourier domain, as we show in the next section. Next, we describe some typical graph signal models (i.e., specific $\mathbf{A}$'s) as well as two sampling approaches (i.e., choices of $\mathbf{S}$).

Graph signal models

The signal generation and recovery models of (17) and (19) are valid for any arbitrary signal subspace represented by $\mathbf{A}$. Here, we introduce several models of $\mathbf{A}$ proposed in the literature that are related to the specific graph $\mathcal{G}$ on which we wish to process data.

The most widely studied graph signal model is the BL signal model. This model corresponds to $\mathbf{A} = \mathbf{U}_{\mathcal{V}\mathcal{B}}$, where $\mathcal{B} := \{1, \ldots, K\}$. A BL signal is, thus, written by

$$\mathbf{x}_{\mathrm{BL}} = \sum_{i=1}^{K} d_i \mathbf{u}_i = \mathbf{U}_{\mathcal{V}\mathcal{B}}\mathbf{d}, \tag{20}$$

where $\omega := \lambda_K$ is called the *bandwidth* or *cutoff frequency* of the graph signal. The signal subspace of ω-BL graph signals on $\mathcal{G}$ is often called the *Paley–Wiener space* $PW_{\omega}(\mathcal{G})$ [7], [10]. In *spectral graph theory* [16], [17], it is known that eigenvectors corresponding to low graph frequencies are smooth within clusters, i.e., localized in the vertex domain.

A more general *frequency domain subspace model* could be obtained as

The Relationship Between Periodic Graph Spectrum and Shift-Invariant Signals

To connect signal generation models of graph signals to those of shift-invariant (SI) signals, the periodic graph spectrum (PGS) subspace has been proposed in [8].

Definition S1: PGS subspace

A K-dimensional PGS subspace, where $K \leq N$, of a given graph $\mathcal{G}$ is a space of graph signals that can be expressed as a graph Fourier transform (GFT) spectrum filtered by a given generator:

$$\mathcal{X}_{\text{PGS}} = \left\{ x[n] \,\middle|\, x[n] = \sum_{i=0}^{N-1} d_{i \bmod K}\, \hat{a}(\lambda_i)\, u_i[n] \right\}, \qquad \text{(S1)}$$

where $\hat{a}(\lambda_i)$ is the graph frequency-domain response of the generator, and d_i is an expansion coefficient. A signal in a PGS subspace can be represented in (22). Definition S1 assumes the graph spectrum is periodic.

The PGS subspace is related to the signal subspace of continuous-time SI signals in (12). Suppose that T in (12) is a positive integer; i.e., the spectrum $D(e^{j\omega T})$ is repeated T times within $\omega \in [0, 2\pi]$, and $A(\omega)$ in (12) has support $\omega \in [0, 2\pi]$. In this case, a sequence $X[i] = D(e^{j\omega T})A(\omega)|_{\omega = 2\pi i/N}$ $(i = 0, \ldots, N-1)$ corresponds to the discrete Fourier transform (DFT) spectrum of length N. Therefore, this $X[i]$ can be regarded as a graph signal spectrum in a PGS subspace if the GFT is identical to the DFT (by relaxing to a complex $\mathbf{U}$); e.g., the graph $\mathcal{G}$ is a cycle graph, i.e., a periodic graph consisting of a ring. This relationship is illustrated in Figure S1.

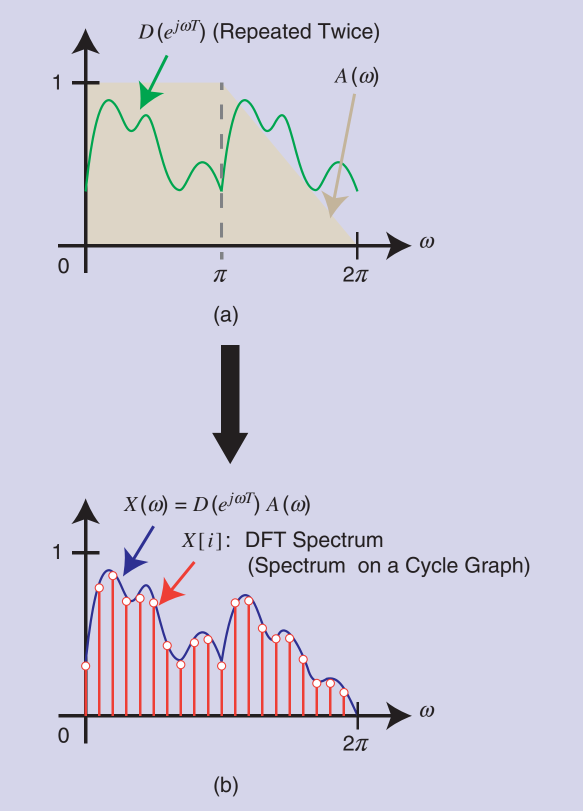


FIGURE S1. The relationship between PGS and SI signals for $T = 2$: (a) the signal generation model in SI space and (b) the spectrum of the sampled SI signal.

The correction filter $\mathbf{H} = (\mathbf{S}^{\top}\mathbf{A})^{\dagger}$ for signals in a PGS subspace mimics the frequency response of (14). Suppose that the graph frequency-domain sampling in the "Definition 2: Graph Frequency-Domain Sampling" section is used. The DS condition in this case implies $\tilde{R}_{ga}(\lambda_i) \neq 0$ for all $i = 1, \ldots, K$, where $\tilde{R}_{ga}(\lambda_i) := \sum_{\ell} \hat{g}(\lambda_{i+K\ell})\hat{a}(\lambda_{i+K\ell})$. If $\mathbf{x} \in \mathcal{X}_{\text{PGS}}$ and the DS condition holds, then the correction transform $\mathbf{H}$ is equivalent to filtering in the graph frequency domain with correction filter [8]

$$\hat{h}(\lambda_i) = \frac{1}{\tilde{R}_{ga}(\lambda_i)}, \qquad \text{(S2)}$$

which clearly parallels the SI expression (14).

$$\mathbf{x} = \sum_{i=1}^{K} d_i \sum_{j=1}^{N} \hat{a}_i(\lambda_j)\mathbf{u}_j = \mathbf{A}\mathbf{d}, \qquad (21)$$

where $\mathbf{d} \in \mathbb{R}^K$, and the ith column of A is $\Sigma_{j=1}^{N} \hat{a}_i(\lambda_j)\mathbf{u}_j$. In this case, each of the $\hat{a}_i(\lambda)$ imposes a certain spectral shape (e.g., exponential), and the parameter d_i controls how much weight is given to the ith spectral shape. It is clear that (21) includes (20) as a special case by choosing $\hat{a}_i(\lambda)$ appropriately.

Another special case of (21) has been proposed by assuming periodicity of the spectrum [8]. A signal in a periodic graph spectrum (PGS) subspace can be represented as

$$\mathbf{x}_{\text{PGS}} = \mathbf{U}\hat{a}(\boldsymbol{\Lambda})\mathbf{D}_{\text{samp}}^{\top}\mathbf{d}, \qquad (22)$$

where $\mathbf{D}_{\text{samp}}$ is the matrix for the GFT domain upsampling (details are given in the "Definition 2: Graph Frequency Domain Sampling" section). This model parallels those studied in the SI setting and leads to recovery methods based on filtering in the graph frequency domain, similar to (14). This relationship is described in "The Relationship Between Periodic Graph Spectrum and Shift-Invariant Signals."

The vertex-domain subspace model can also be considered. Let $\{\mathcal{T}_i\}$ $(i = 1, \ldots, K)$ be a partition of $\mathcal{V}$, where each node in $\mathcal{T}_i$ is locally connected within the cluster. A piecewise constant graph signal is then given by

$$\mathbf{x}_{\text{PC}} = \sum_{i=1}^{K} d_i \mathbf{1}_{\mathcal{T}_i} = [\mathbf{1}_{\mathcal{T}_1}, \ldots, \mathbf{1}_{\mathcal{T}_K}]\mathbf{d}, \qquad (23)$$

where $[\mathbf{1}_{\mathcal{T}_i}]_n = 1$ when the node n is in $\mathcal{T}_i$ and 0 otherwise [18]. In this case, $\mathbf{A} = [\mathbf{1}_{\mathcal{T}_1}, \ldots, \mathbf{1}_{\mathcal{T}_K}]$. Piecewise smooth graph signals can be similarly defined.

Graph signals parameterized by the various models introduced earlier can be perfectly recovered by (19) beyond the BL assumption in (20) as long as the class of signals to be reconstructed corresponds to a vector subspace of sufficiently low dimension relative to the sampling rate. While many studies have focused on the BL setting, the important point we stress here is that, when a proper signal model is given by prior information or by estimating/learning from data, recovery is often possible whether or not the signal is BL. Appropriate modeling of graph signal subspaces beyond those already shown is an interesting future research topic from both a theoretical and an application point of view.

Sampling methods

Similar to time- and frequency-domain sampling for signals in SI space, i.e., (5) and (6), graph signal sampling can be defined in both the vertex and spectral domains. For time-domain signals, there is a simple relationship between sampling in both domains, as can be seen from (11) and (12). In contrast, for general graphs, direct nodewise sampling in the vertex domain (i.e., selecting a subset of nodes) does not correspond to a spectrum folding operation in the graph frequency domain, and vice versa. Thus, we discuss vertex and frequency-domain graph sampling separately.

Vertex-domain sampling

Vertex-domain sampling is an analog of time-domain sampling. Samples are selected on a predetermined sampling set, $\mathcal{T} \in \mathcal{V}$, containing $|\mathcal{T}| = M$ nodes. Sampling set selection is described in the "Sampling Set Selection and Efficient Computation Methods" section. For a given $\mathcal{T}$, we define sampling as follows.

Definition 1: Vertex-domain sampling

Let $\mathbf{x} \in \mathbb{R}^N$ be a graph signal and $\mathbf{G} \in \mathbb{R}^{N \times N}$ be an arbitrary graph filter in (2). Suppose that the sampling set $\mathcal{T}$ is given a priori. The sampled graph signal $\mathbf{c} \in \mathbb{R}^M$ is defined by [9], [10]

$$\mathbf{c} = \mathbf{I}_{\mathcal{T}\mathcal{V}} \mathbf{G} \mathbf{x}, \tag{24}$$

where $\mathbf{I}_{\mathcal{T}\mathcal{V}}$ is a submatrix of the identity matrix whose rows are indicated by the sampling set $\mathcal{T}$.

The sampling matrix is, therefore, given by $\mathbf{S}^\top = \mathbf{I}_{\mathcal{T}\mathcal{V}} \mathbf{G}$. Though many methods in the literature consider direct sampling, where $\mathbf{G} = \mathbf{I}$, an arbitrary $\mathbf{G}$ can be used prior to nodewise sampling.

Graph frequency-domain sampling

Sampling in the graph frequency domain [12] parallels the Fourier domain sampling in (6): the graph Fourier-transformed input $\hat{\mathbf{x}}$ is first multiplied by a graph spectral filter $\hat{g}(\mathbf{\Lambda})$; the product is subsequently folded with period M, resulting in aliasing for full-band signals, which can be utilized for the design of graph wavelets/filter banks [19]. Graph frequency-domain sampling is defined as follows.

Definition 2: Graph frequency-domain sampling

Let $\hat{\mathbf{x}} \in \mathbb{R}^N$ be the original signal in the graph frequency domain, i.e., $\hat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$, and let $\hat{g}(\mathbf{\Lambda})$ be an arbitrary sampling filter expressed in the graph frequency domain. For a sampling ratio $M \in \mathbb{Z}$, where M is assumed to be a divisor of N for simplicity, the sampled graph signal in the graph frequency domain is given by

$$\mathbf{c} = \mathbf{D}_{\text{samp}} \hat{g}(\mathbf{\Lambda}) \hat{\mathbf{x}}, \tag{25}$$

where

$$\mathbf{D}_{\text{samp}} = [\mathbf{I}_M \;\; \mathbf{I}_M \;\; \ldots] \tag{26}$$

is the spectrum folding matrix.

The sampling matrix $\mathbf{S}^\top$ in the graph frequency domain is, thus, given by

$$\mathbf{S}^\top = \mathbf{D}_{\text{samp}} \hat{g}(\mathbf{\Lambda}) \mathbf{U}^\top. \tag{27}$$

While this definition is clearly an analog of the frequency-domain sampling in (6), in general, it cannot be written as an operator of the form of $\mathbf{I}_{\mathcal{T}\mathcal{V}} \mathbf{G}$, i.e., graph filtering, as defined in the "Review: GSP and Sampling in Hilbert Spaces" section, and vertex-domain sampling, except for some specific cases, such as cycle or bipartite graphs [8], [12], [19]. Therefore, graph frequency-domain sampling requires samples in all nodes to be available before performing sampling.

While this property may not be desirable for a direct application of sampling, e.g., obtaining samples at a subset of nodes in a sensor network, assuming all of the samples are available before sampling is reasonable for several applications. For example, graph filter banks require the whole signal prior to filtering and sampling (whether it is performed in the vertex or graph frequency domain), along with a strategy to downsample the filtered signals to achieve critical sampling.

It has been shown that a filter bank with frequency-domain downsampling can outperform one using vertex-domain sampling [19]. Graph frequency-domain sampling also outperforms that in the vertex domain with generalized graph signal sampling frameworks [8], [15]. See "An Illustrative Example of Sampling Procedures" for a comparison between graph signal sampling and conventional discrete-time signal sampling, which shows the lack of equivalence between vertex and frequency sampling.

Remarks on correction and reconstruction transforms

In the SI setting, signal recovery can be implemented in the time domain as (9) with counterparts in the Fourier domain as in (14). However, this is not the case for vertex-domain sampling: While the sampling matrix $\mathbf{S}$ in Definition 1 is designed to parallel sampling in the time domain, the correction matrix $\mathbf{H} = (\mathbf{S}^\top \mathbf{A})^\dagger$ does not have a diagonal graph frequency response in general. Refer to "Bandlimited Signal Recovery With Vertex-Domain Sampling" for an example with BL signals.

An Illustrative Example of Sampling Procedures

In Figure S2(a), standard discrete-time sampling is shown in both the time and frequency domains. Pointwise sampling in the time domain corresponds to folding of the discrete Fourier transform (DFT) spectrum [1]. Note that both sampling methods yield the same output after the inverse DFT of the frequency-sampled signal. Figure S2(b) illustrates graph signal sampling in the vertex and graph frequency domains (Definitions S1 and 2), which do not yield the same output, unlike their conventional signal counterparts of Figure S2(a).

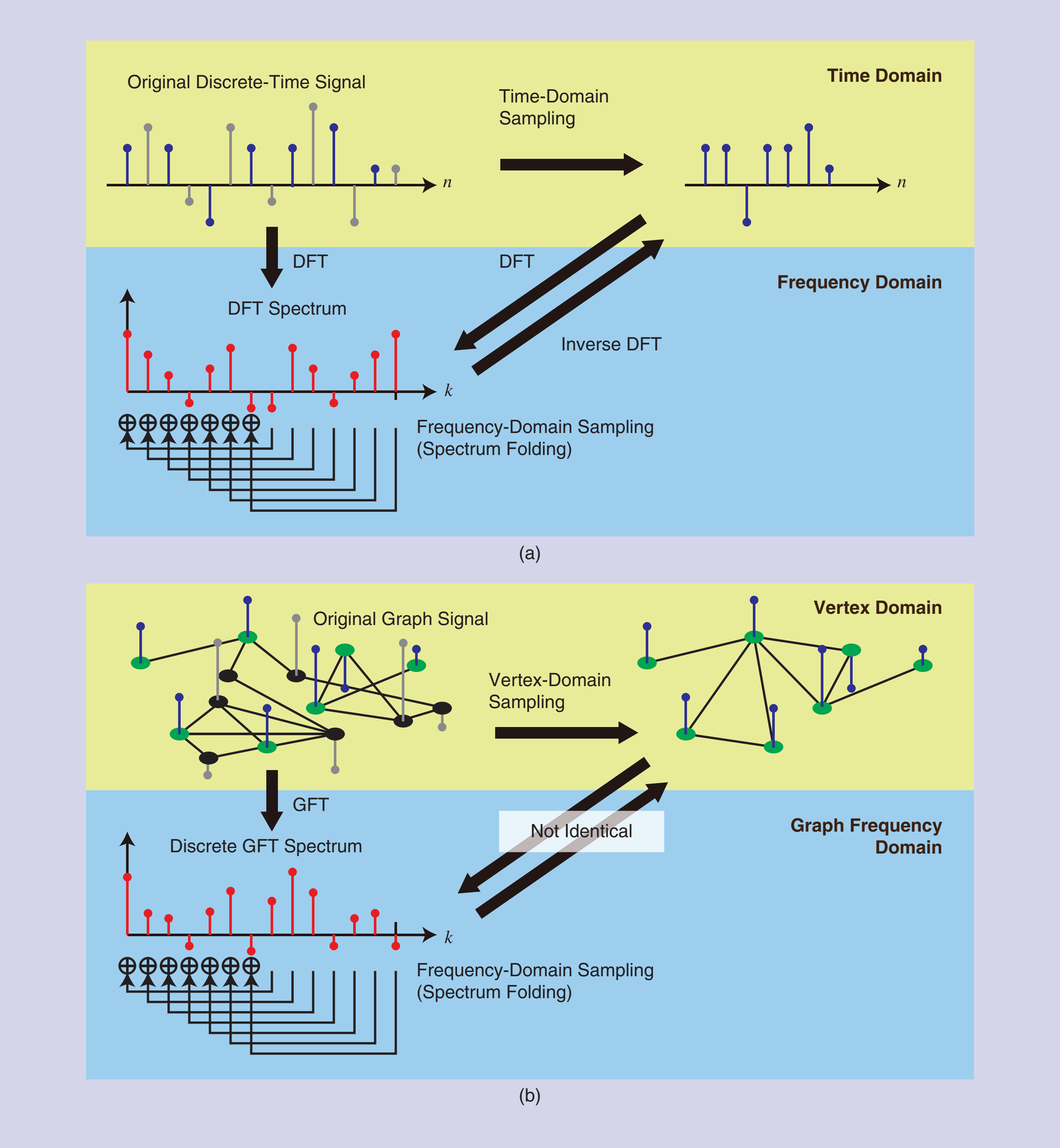


FIGURE S2. The sampling procedures for (a) standard and (b) graph signals. GFT: graph Fourier transform.

Bandlimited Signal Recovery With Vertex-Domain Sampling

Assume we have a bandlimited signal $\mathbf{x}_{\text{BL}}$ as defined in (20) and we use direct nodewise sampling, i.e., $\mathbf{G} = \mathbf{I}$. This is a well-studied setting for graph sampling theory. The DS condition in this case is often called the *uniqueness set condition* [7], [10] and requires a full-rank $\mathbf{I}_{\mathcal{TV}}\mathbf{U}_{\mathcal{VB}} = \mathbf{U}_{\mathcal{TB}} \in \mathbb{R}^{M \times M}$ [9], [10]. (We assume $M = K$.) In this case, we have $\mathbf{A} = \mathbf{U}_{\mathcal{VB}}$, and (19) is reduced to

$$\tilde{\mathbf{x}} = \mathbf{U}_{\mathcal{VB}}(\mathbf{U}_{\mathcal{TB}})^{\dagger}\mathbf{c}, \tag{S3}$$

where we assume $M = |\mathcal{T}| = |\mathcal{B}|$. In other words, the correction transform is given by $\mathbf{H} = (U_{\mathcal{TB}})^{\dagger}$. As a result, $\mathbf{H}$ cannot be written as a graph spectral filter having a diagonal frequency response. Even if the sampling filter $\mathbf{G}$ is applied before nodewise sampling, perfect or least-squares recovery is obtained by replacing $(\mathbf{U}_{\mathcal{TB}})^{\dagger}$ in (S3) with $(\mathbf{G}_{\mathcal{TV}}\mathbf{U}_{\mathcal{VB}})^{\dagger}$, while $\mathbf{A} = \mathbf{U}_{\mathcal{VB}}$ does not change. An approximate recovery of bandlimited graph signals is possible with an alternative approach, e.g., an iterative algorithm using polynomial filters and projection onto convex sets [S1].

Reference

[S1] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 5445–5449. doi: 10.1109/ICASSP.2013.6638704.

Different Reconstruction Operators

The reconstruction in (19) allows for perfect signal recovery under the discrete-sum condition, when the signal lies in a given subspace. However, we may not always have such a strong prior. For example, we may only know that our signal is smooth in some sense. A popular approach in this case is to consider the following recovery [1], [20]:

$$\tilde{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{S}^{\top}\mathbf{x} = \mathbf{c}} \|\mathbf{V}\mathbf{x}\|_p, \tag{S4}$$

where $\mathbf{V}$ is a matrix that measures smoothness, and $p \geq 1$. If there is noise, we can relax the goal of achieving a consistent solution, i.e., such that $\mathbf{S}^{\top}\mathbf{x} = \mathbf{S}^{\top}\tilde{\mathbf{x}}$, and instead solve the following problem:

$$\tilde{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{S}^{\top}\mathbf{x} - \mathbf{c}\|_2^2 + \gamma\|\mathbf{V}\mathbf{x}\|_p, \tag{S5}$$

where $\gamma > 0$ is a regularization parameter. Fast and efficient interpolation algorithms have also been studied in [S1] and [S2] based on generalizations of standard signal processing techniques to the graph setting.

Reference

[S2] A. Heimowitz and Y. C. Eldar, "Smooth graph signal interpolation for big data," 2018, arXiv:1806.03174.

In this section, for simplicity, we have considered the case where measurement, sampling, and recovery are noise-free. In the presence of noise, the reconstruction of (19) may be replaced by noise-robust methods. See sidebar "Different Reconstruction Operators" for an example. Note that the recovery procedures for the noisy cases have been well studied in the context of (generalized) sampling theory for standard signals [1] as well as compressed sensing [20]. Robustness against noisy measurements is also a major motivation to optimize the sampling set selection of graphs, which we discuss in the next section.

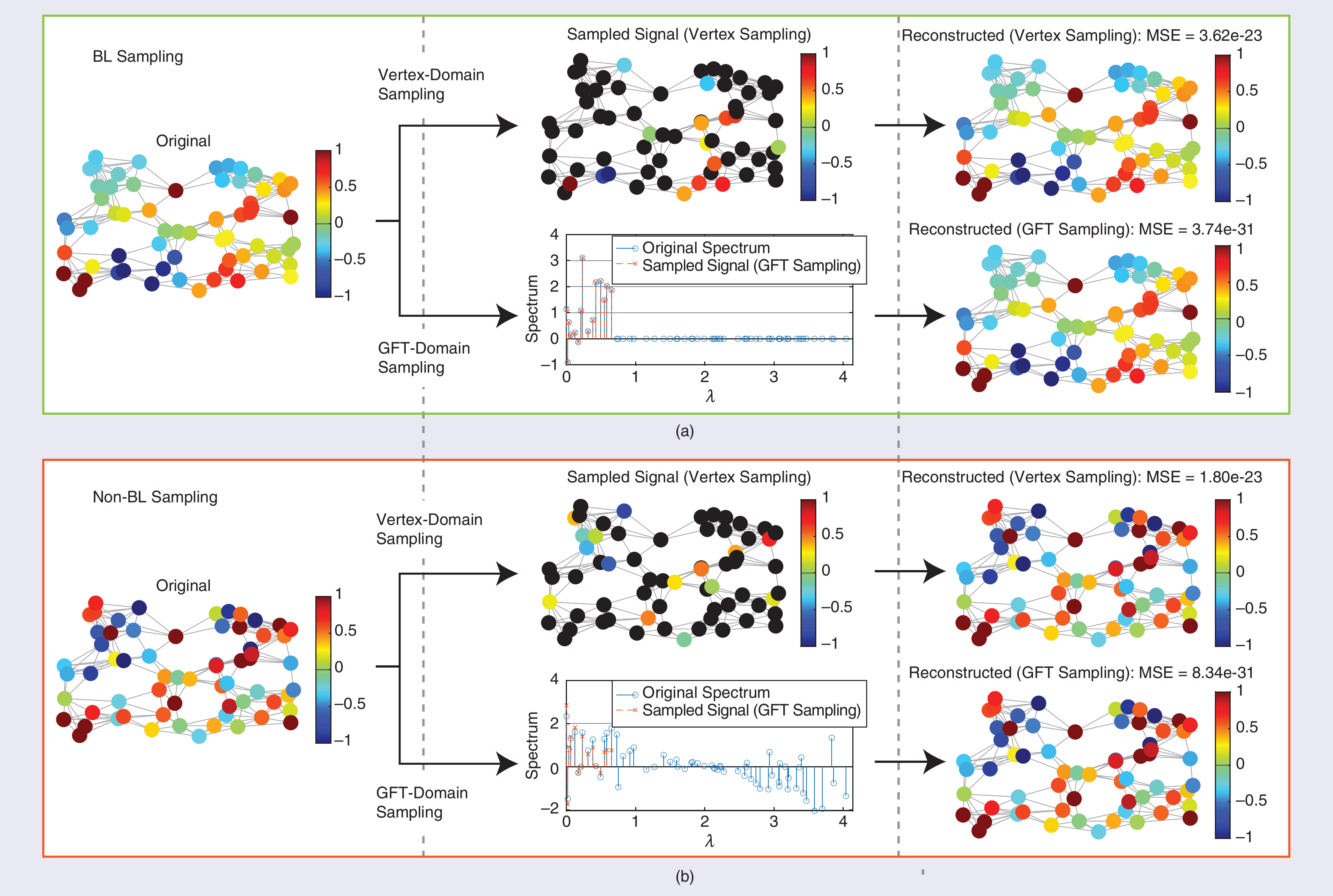
Graph signal sampling and recovery example for synthetic signals

To illustrate signal recovery both for BL and full-band settings, we consider the random sensor graph example of Figure 2, with $N = 64$ and $M = 15$. The first scenario is the well-known BL setting, where the signal is BL as in (20), with $K = 15$, and the sampling filter is the identity matrix, i.e., $\mathbf{G} = \mathbf{I}$. In the second scenario, we use the full-band generator in the graph frequency domain with the PGS model in (22) [8]. The generator function is $\hat{a}(\lambda_i) = 1 - 2\lambda_i/\lambda_{\max}$, and each element in $\mathbf{d} \in \mathbb{R}^M$ is drawn from $\mathcal{N}(1, 1)$. The sampling filter is also full-band where $\hat{g}(\lambda_i) = \exp(-\lambda_i/2)$.

As shown in Figure 2(a), both vertex and frequency sampling methods can recover the BL graph signal. Note that $\mathbf{c}$ is identical to $\mathbf{d}$ for graph frequency-domain sampling. In contrast, Figure 2(b) shows that the original signal oscillates in the vertex domain due to its full-band generator function. Also, $\mathbf{c}$ of graph frequency-domain sampling does not match the original spectrum due to aliasing and the sampling filter. However, even in that case, the original signal is perfectly recovered when the signal subspace is given.

Sampling set selection and efficient computation methods

The recovery method in (19) can be possible only if the signal subspace, e.g., cutoff frequency in the BL setting, is known perfectly a priori. However, in practice, the cutoff frequency is often unknown (and, thus, can at best be estimated), or the signal is smooth but not strictly BL in the first place. Furthermore, observed samples may be corrupted by additive noise. Thus,



(a)

(b)

FIGURE 2. Sampling examples for signals on a random sensor graph with $N=64$. The sample **c** has length $M=15$. (a) BL sampling and recovery, where the signal is BL with $K=15$ and the sampling filter is the identity matrix. (b) Sampling and recovery of the graph signal lying in a known subspace, where the original signal is generated by the PGS model [8] with generator function $\hat{a}(\lambda_i)=1-2\lambda_i/\lambda_{\max}$, and the sampling function is $\hat{g}(\lambda_i)=\exp(-\lambda_i/2)$. Even in this case, the original signal is perfectly recovered from **c** by using both vertex and graph frequency-domain sampling without changing the framework. Graph Signal Processing toolbox (GSPBOX) (https://epfl-lts2.github.io/gspbox-html/) is used for obtaining the random sensor graph. MSE: mean square error.

practical sampling set selection algorithms often aim at maximizing robustness to noise or imperfect knowledge of sampled signal characteristics. In this section, efficient sampling set selection methods for vertex-domain sampling are examined.

Along with signal reconstruction quality, computational complexity is another key concern when designing sampling algorithms since signals may reside on very large graphs. Often, one would like to avoid computing the eigendecomposition of the chosen graph variation operator, such as the graph Laplacian, which requires large computational cost [$O(N^3)$ in the general case]. In this section, we also provide an overview of fast and efficient sampling set selection methods.

Sampling set selection: Deterministic and random approaches

A list of representative sampling methods is given in Table 1. One of the first considerations when deciding on a sampling scheme is whether it is deterministic or random. Deterministic approaches [9]–[11], [13], [21]–[23] choose a fixed node subset to optimize a predetermined cost function. Since sampling set selection is, in general, combinatorial and NP-hard, many deterministic selection methods are greedy, adding one locally optimal node at a time until the sampling budget is exhausted. Though a greedy selection method is suboptimal in general, it gives a constant-factor approximation of the original combinatorial optimization problem if the cost function for the sampling set selection is maximization of a submodular function [14], [23]. Advantages of deterministic sampling set selection methods include the following:

- The "importance" of individual nodes is computed and totally ordered for greedy selection; if the sampling budget changes, one can add or remove nodes easily without rerunning the entire selection algorithm.
- The selected node subset remains fixed as long as the graph structure is the same.

In contrast, random methods [24]–[26] select nodes randomly according to a predetermined probability distribution. Typically, the distribution is designed so that more "important" nodes are selected with higher probabilities. One key merit of random methods is low computational cost. Once the probability distribution is determined, the selection itself can be realized quickly in a distributed manner. In practice, random sampling may perform well on average, but it often requires more samples than deterministic methods to achieve the same reconstruction quality, even if the signal is BL [25]. One may also combine deterministic and random selection methods in finding a sampling set.

Deterministic sampling set selection

Two main types of deterministic sampling set selection methods have been proposed in the literature. First, vertex-based methods have been studied extensively in machine learning and sensor network communities as a sensor placement problem. (See further discussion on applications in the "Applications" section). Second, spectrum-based methods—selection schemes grounded in graph frequency assumptions—represent a relatively new approach and have been studied in the context of graph sampling theory. We focus on the latter approach due to space limitations. See [11] for a summary of existing vertex-based methods.

The exact BL case

For simplicity, suppose we directly observe the samples, i.e., $\mathbf{G} = \mathbf{I}$, and choose a BL signal model in (20). To optimize the sampling set, we can define an objective function to quantify reconstruction error in the presence of noise. The sampled signal $\mathbf{y} \in \mathbb{R}^M$ is then $\mathbf{y} = \mathbf{c} + \mathbf{n}$, where $\mathbf{n}$ is an independent identically distributed additive noise introduced during the measurement or sampling process. Using the LS recovery (13), the reconstructed signal $\tilde{\mathbf{x}}$ is then given by

$$\tilde{\mathbf{x}} = \mathbf{U}_{\mathcal{VB}}\mathbf{U}^{\dagger}_{\mathcal{TB}}\,\mathbf{y} = \mathbf{U}_{\mathcal{VB}}\mathbf{U}^{\dagger}_{\mathcal{TB}}\mathbf{c} + \mathbf{U}_{\mathcal{VB}}\mathbf{U}^{\dagger}_{\mathcal{TB}}\mathbf{n}. \tag{28}$$

The LS reconstruction error thus becomes $\mathbf{e} := \tilde{\mathbf{x}} - \mathbf{x} = \mathbf{U}_{\mathcal{VB}}\mathbf{U}^{\dagger}_{\mathcal{TB}}\mathbf{n}$. Many deterministic methods consider an optimization objective based on the error covariance matrix:

$$\mathbf{E} := \mathbb{E}[\mathbf{e}\mathbf{e}^{\top}] = \mathbf{U}_{\mathcal{VB}}(\mathbf{U}^{\top}_{\mathcal{TB}}\mathbf{U}_{\mathcal{TB}})^{-1}\mathbf{U}^{\top}_{\mathcal{VB}}. \tag{29}$$

Given (29), one can choose different optimization criteria based on the optimal design of experiments [27]. For example, the A-optimality criterion minimizes the average errors by seeking $\mathcal{T}$, which minimizes the trace of the matrix inverse [9], [11], [14]:

$$\min_{\mathcal{T}\,:\,|\mathcal{T}|=M} \ \mathrm{Tr}\big((\mathbf{U}^{\top}_{\mathcal{TB}}\mathbf{U}_{\mathcal{TB}})^{-1}\big), \tag{30}$$

while E-optimality minimizes the worst-case errors by maximizing the smallest eigenvalue of the information matrix $\mathbf{U}^{\top}_{\mathcal{TB}}\mathbf{U}_{\mathcal{TB}}$ [9], [11], [14]:

Table 1. A comparison of graph spectrum-based sampling set selection methods.

Methods	Deterministic/ Random	Kernel	Localization in Vertex Domain	Localization in Graph Frequency Domain
Maximizing the cutoff frequency [10]	Deterministic	$\lambda^k\,(k \in \mathbb{Z}_+)$		✓
Error covariance [9], [14]	Deterministic	Ideal		✓
Approximate supermodularity [25]	Deterministic	Ideal		✓
Localized operator [11]	Deterministic	Arbitrary	✓	✓
Neumann series [24]	Deterministic	Ideal	✓*	✓
Gershgorin disk alignment [13]	Deterministic	λ	✓	
Cumulative coherence [27]	Random	Ideal	✓*	✓
Global/local uncertainty [28]	Random	Arbitrary	✓	✓

*Localized in the vertex domain only if the ideal kernel is approximated by a polynomial.

$$\max_{\mathcal{T} \,||\mathcal{T}|=M} \lambda_{\min}(\mathbf{U}_{\mathcal{T}\mathcal{B}}^{\top}\mathbf{U}_{\mathcal{T}\mathcal{B}}). \tag{31}$$

In either case, sampling set selection based on the error covariance matrix (29) requires (partial) singular value decomposition (SVD) of an $M \times M$ matrix, even when the GFT matrix $\mathbf{U}$ is given a priori. This results in a large computational cost. To alleviate this burden, greedy sampling without performing SVD has been recently proposed. This category includes methods using spectral proxies, which approximately maximize cutoff frequency [10], a graph filter submatrix that avoids SVD by utilizing a fast GFT and block matrix inversion [22], and a polynomial filtering-based approach that maximizes a vertex domain support of graph spectral filters [11].

Smooth signals

Instead of a strict BL assumption, one can assume the target signal $\mathbf{x}$ is smooth with respect to the underlying graph, where smoothness is measured via an operator $\mathbf{V}$. One can thus reconstruct via a regularization-based optimization in (S5); in [13], $\mathbf{V} = \mathbf{L}^{1/2}$, and the reconstruction becomes

$$\min_{\mathbf{x}} \| \mathbf{S}^{\top}\mathbf{x} - \mathbf{y} \|_2^2 + \gamma \mathbf{x}^{\top}\mathbf{L}\mathbf{x}. \tag{32}$$

Problem (32) has a closed-form solution $\mathbf{x}^*$:

$$\mathbf{x}^* = (\mathbf{S}\mathbf{S}^{\top} + \gamma\mathbf{L})^{-1}\mathbf{S}\mathbf{y}. \tag{33}$$

The authors in [13] choose the sampling matrix $\mathbf{S}^{\top}$ to maximize the smallest eigenvalue $\lambda_{\min}$ of the coefficient matrix $\mathbf{S}\mathbf{S}^{\top} + \gamma\mathbf{L}$ in (33)—corresponding to the E-optimality criterion. This is done without eigendecomposition via the well-known Gershgorin circle theorem.

The relationship between various methods based on the localized operator

Vertex- and spectrum-based methods have been proposed separately in different research fields. Interestingly, many of them can be described in a unified manner by utilizing a graph localization operator [11]. A graph localization operator is a vertex-domain expression of a spectral filter kernel $\hat{g}(\lambda)$ centered at the node i [26]:

$$\psi_{g,i}[n] := \sqrt{N} \sum_{k=1}^{N} \hat{g}(\lambda_k) u_k[i] u_k[n], \tag{34}$$

which can be viewed as the "impulse response" of a graph filter by rewriting (34) in vector form as

$$\psi_{g,i} = \mathbf{U}\hat{g}(\mathbf{\Lambda})\mathbf{U}^{\top}\delta_i, \tag{35}$$

where $\boldsymbol{\delta}_i$ is an indicator vector for the ith node, i.e., the unit impulse. In [11], it has been shown that many proposed cost functions can be interpreted as having the form of (35) for different kernels.

Random sampling set selection

Random selection methods can be classified into two categories. First, graph-independent approaches select nodes randomly without taking into account the underlying graph [9], [24], [25], which results in very low computational cost. However, theoretical results based on studies on compressed sensing [25] have shown that the number of required nodes for recovery of BL graph signals tends to be larger than for graph-dependent deterministic selections [25].

Second, graph-dependent random selection methods [24]–[26] assume that node importance varies according to the underlying graph, e.g., important nodes are connected to many other nodes with large edge weights. In these approaches, a sampling probability distribution $\mathbf{p} \in \mathbb{R}^N$, where $p[i] \geq 0$ for all $i \, (i = 0, \ldots, N-1)$ and $\Sigma_i p[i] = 1$, is first obtained prior to running a random selection algorithm. They assume $\mathcal{G}$ is given a priori. In addition to graph information, we can incorporate information about observations (samples) to obtain $\mathbf{p}$ [24]. The sampling set is then randomly chosen based on $\mathbf{p}$.

As an example, in the graph coherence-based random selection method for ω-BL graph signals of [25], the sampling distribution is given as $p[i] := \| \mathbf{U}_{\mathcal{V}\mathcal{B}}^{\top} \delta_i \|_2^2 / K$, where the numerator is the same as $\| \psi_{g,i} \|_2^2$ in (35), with $\hat{g}(\lambda)$ being the bandlimiting filter. To avoid eigendecomposition, a polynomial approximation for the filter can be applied, and the calculation cost can be further reduced by filtering random signals instead of $\delta_i, i = 0, \ldots, N-1$. A similar approach using an arbitrary filter kernel $\hat{g}(\lambda)$ has also been proposed [26].

Statistical analysis among random sampling strategies is performed in [24] for approximate BL signals. It has been shown that using previously observed samples in addition to graph information does not statistically outperform using graph topology only, in the sense of the worst-case reconstruction error.

Sampling set selection examples

We next consider several examples. A first example is shown in Figure 3(a), where sampling sets of size 10 are selected for a community graph with $N = 256$. The cluster sizes are different and range from 4 to 64, shown in a clockwise direction. The following methods are compared: 1) a deterministic method based on the localized operator [11], 2) a graph-dependent random selection method using cumulative coherence [25], and 3) a traditional entropy-based sensor selection method [28]. The random approach does not choose any nodes in the rightmost cluster in this realization. The entropy-based methods select many nodes in small clusters because they tend to favor low-degree nodes. In contrast, the deterministic approach selects nodes that are more uniformly distributed across clusters.

In the second example, Figure 3(b), we use graph signal sampling to select pixels in an image. Each graph node corresponds to a pixel. Sampling set selection is based on maximizing the cutoff frequency [10]. We use two variation operators, the combinatorial and symmetric normalized graph Laplacians, leading to very different sampling sets. When using the combinatorial Laplacian, the selected pixels tend to be closer to image contours or the image boundary. In contrast, pixels selected using the normalized Laplacian are more uniformly

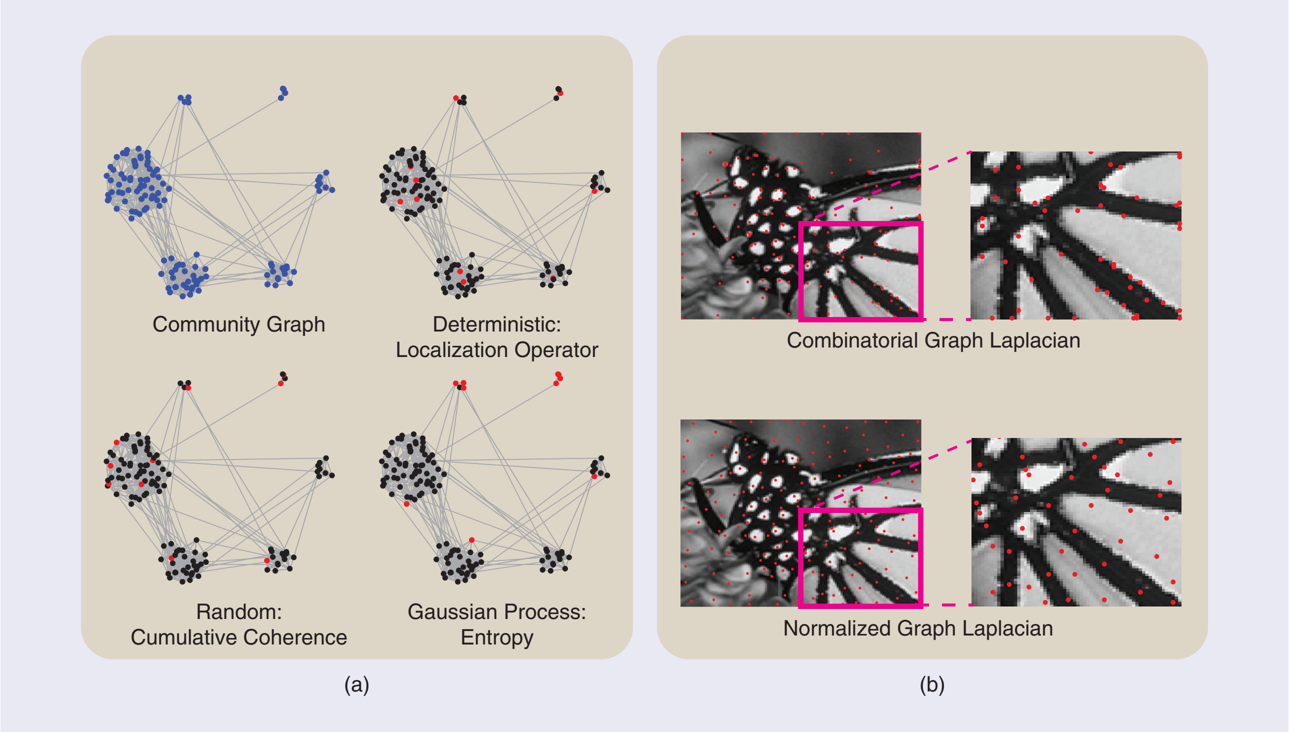


FIGURE 3. A comparison of sampling sets. (a) The sampling sets for a community graph with $N = 256$, where 10 nodes are selected. The sizes of the clusters are $\{4, 4, 8, 16, 32, 64\}$ from the top left cluster in a clockwise direction. At the top are the original graph and sampling set by a deterministic approach [11]. At the bottom are a sampling set of one realization by a random approach [27] and a sampling set by a Gaussian process-based method [30]. GSPBOX (https://epfl-lts2.github.io/gspbox-html/) is used for obtaining the cluster graph. (b) The sampling sets for image pixels. Each pixel is a node, and edge weights are chosen based on the method in https://github.com/STAC-USC/NNK_Image_graph. A sampling set selection based on maximizing the cutoff frequency [10] is used. The enlarged portions are shown for better visualization. The sampling sets use the (top) combinatorial and (bottom) normalized graph Laplacians, respectively.

distributed within the image. See also "To Normalize or Not to Normalize" for a comparison of variation operators.

Computational complexity

Selecting a sampling set can be divided into two phases: 1) preparation, which includes computing required prior information, e.g., the eigendecomposition of the graph operator, and 2) selection, the main routine that selects nodes for sampling, e.g., calculating the cost function for candidate nodes in each iteration. Computational complexities of different methods are summarized next.

Deterministic selection

Deterministic selection methods, studied in the context of graph sampling theory, basically need to calculate eigenpairs of (a part of) the variation operator in the selection phase. Their computational costs mostly depend on the number of edges in the graph and the assumed bandwidth. A recent trend is to investigate eigendecomposition-free sampling set selection algorithms [11], [13], [22]. These recent methods approximate a graph spectral cost function with vertex-domain processing, like polynomial approximation of the filter kernel. Table 2 shows that the computational complexities of these eigendecomposition-free methods compare with previous sampling methods [9], [10] that require computation of multiple eigenpairs.

Table 2. Computational complexities of GSP-based deterministic sampling set selection.

Method	Preparation	Selection
Maximizing the cutoff frequency [10]	$O(k\lvert\mathcal{E}\rvert MT(k))$	$O(NM)$
Error covariance: E-optimal [9]	$O((\lvert\mathcal{E}\rvert M + CM^3)T_e)$	$O(NM^4)$
Error covariance: A-optimal [9], [14]		$O(NM^4)$
Error covariance: T-optimal [14]		$O(NM)$
Error covariance: D-optimal [14]		$O(M^3)$
Approximate supermodularity [25]		$O(NM^2)$
Localized operator [11]	$O((\lvert\mathcal{E}\rvert + N)P + J)$	$O(JM)$
Neumann series [24]	$O(N^2 \log^2 N)$	$O(NM^3)$
Gershgorin disk alignment [13]	$O(J\log_2(1/\eta))$	$O(MJ\log_2(1/\eta))$

We assume $M = K$ for simplicity. Parameters: $T(k)$: the average number of iterations required for convergence of a single eigenpair, where k is a tradeoff factor between performance and complexity; Te: the number of iterations to convergence for the eigenpair computations using a block version of Rayleigh quotient minimization; C: constant; P: the approximation order of the Chebyshev polynomial approximation; J: the number of nonzero elements in the localization operator; η: the numerical precision to terminate the binary search in Gershgorin disk alignment.

Random selection

Random selection methods typically entail a much smaller computational cost in the selection phase than their deterministic

To Normalize or Not to Normalize

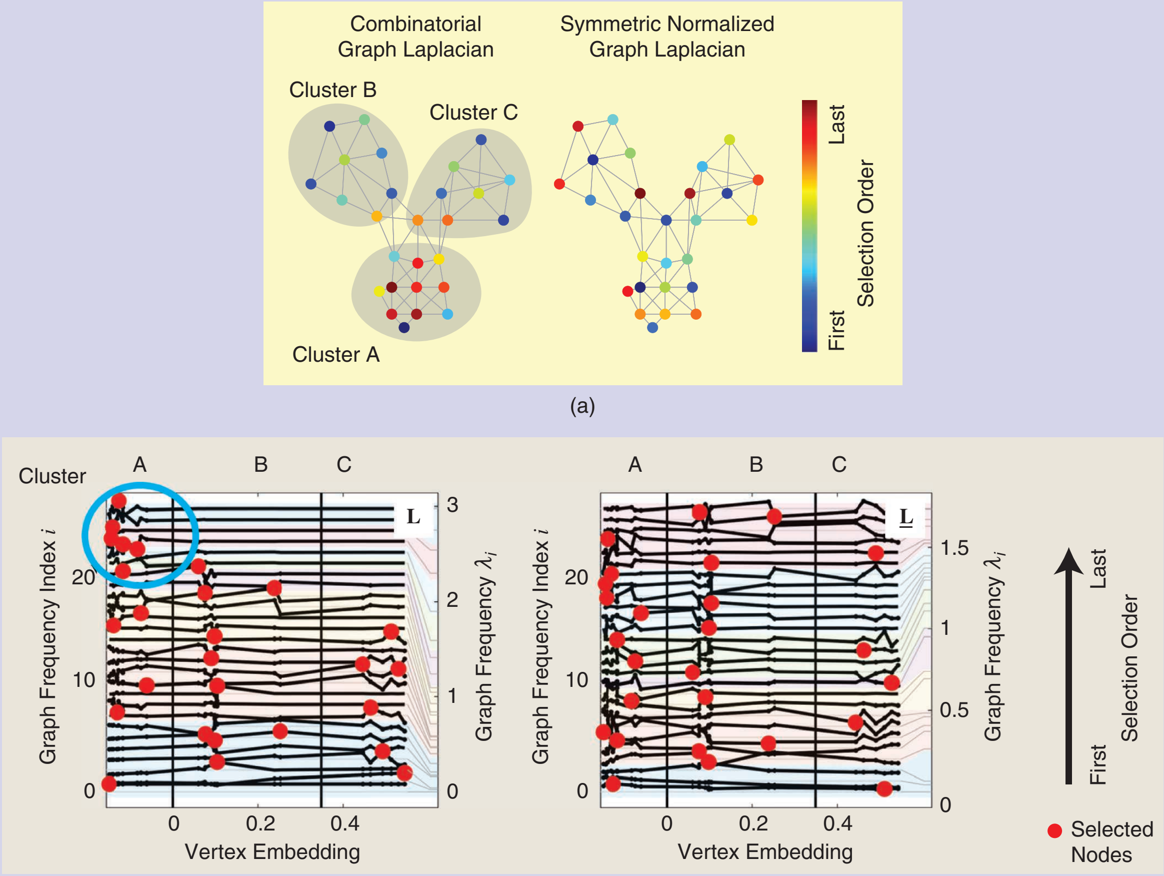
Different graph variation operators lead to different sampling sets, as shown in Figure 3. This difference in behavior is due to normalization. As an example, consider a three-cluster graph with $N = 27$ [S3] and compare the combinatorial graph Laplacian and its symmetric normalized version, with sampling set selection based on maximizing the cutoff frequency [10], as seen in Figure S3.

The vertex-domain expressions in Figure S3(a) represent colors as the node selection orders (the first-chosen nodes are blue, while the last-chosen ones are red). Observe that, for the combinatorial Laplacian, most nodes in cluster A are selected in the last stage, while for the normalized graph Laplacian, the nodes in cluster A (and the other clusters) are selected at all stages. This is due to the localization of the graph Fourier transform (GFT) bases: eigenvectors of the combinatorial graph Laplacian are localized in the vertex domain compared to the normalized one. This is illustrated by the spectral representations in Figure S3(b), using the visualization technique in [S3]. In this visualization, graph nodes are embedded into the 1D real line (i.e., the horizontal axis of the figure), and the GFT bases are shown as a series of 1D signals stacked vertically (lowest frequency at the bottom and highest at the top), with values only in points on the 1D line corresponding to nodes. Nodes in clusters A, B, and C are now grouped in the left, middle, and right regions, respectively, on the 1D line.

The sampling order is represented with red circles: the first selected node is shown with a red circle in the lowest-frequency basis (bottom signal), while the last one chosen appears in the highest-frequency GFT basis (top signal). Since high-frequency eigenvectors of the combinatorial graph Laplacian are highly localized in cluster A (see the light blue circle), the method in [10] more likely selects nodes in cluster A in its last stage. In contrast, for the normalized version, the eigenvectors are less localized, so selected nodes are more balanced among clusters.

Reference

[S3] B. Girault and A. Ortega, "What's in a frequency: New tools for graph Fourier transform visualization," 2019, arXiv:1903.08827.



(a)

(b)

FIGURE S3. The selection orders for different variation operators and oscillations of GFT bases, shown in (a) the vertex domain and (b) the graph frequency domain, along with oscillations of the GFT bases. GFT: graph Fourier transform.

counterparts. As discussed, given a sampling probability distribution $\mathbf{p}$, all sampled nodes can be chosen quickly and in parallel using $\mathbf{p}$. Hence, only the preparation phase needs to be considered. For graph-independent random selection, $p[i] = 1/N$ for all i, and its computational cost is negligible. Many graph-dependent approaches require repeated calculations of $\mathbf{U}\hat{g}(\boldsymbol{\Lambda})\mathbf{U}^\top\mathbf{v}$, where $\mathbf{v} \in \mathbb{R}^N$ is a random vector or $\boldsymbol{\delta}_i$. While the naive implementation still requires eigendecomposition, the graph filter response $\hat{g}(\lambda)$ is often approximated by a polynomial: the preparation phase requires iterative vertex domain processing. Typically, $\mathbf{p}$ is estimated after L filtering operations of random vectors (typically $L = 2\log N$ [25]), which leads to $O(LP(|\mathcal{E}| + N))$ complexity [25].

Applications

Graph signal sampling has been used across a wide range of applications, such as wireless communications, data mining, and 3D imaging. We select a few interesting applications for in-depth discussion in this section.

Sensor placement

Sensor placement [28], [29] has long been studied in the wireless communication community. The basic problem is to choose a subset of locations from a discrete feasible set to place sensors to monitor a physical phenomenon, such as temperature or radiation, over a large geographical area of interest. Commonly, the field signal is assumed to be represented by a low-dimensional parameter vector with a measurement matrix $\boldsymbol{\Phi}$ generated by a Gaussian process. Different criteria have been proposed to optimize the corresponding error covariance matrix, including A-optimality, E-optimality, D-optimality, and frame potential [29].

As a concrete example, one formulation is to maximize the smallest eigenvalue $\lambda_{\min}$ of the inverse error covariance matrix (information matrix) via selection of a sensor subset $\mathcal{T}$, with $|\mathcal{T}| = M$:

$$\max_{\mathcal{T} \,|\, |\mathcal{T}|=M} \lambda_{\min}(\boldsymbol{\Phi}_{\mathcal{TV}}^\top \boldsymbol{\Phi}_{\mathcal{TV}}), \tag{36}$$

where $\boldsymbol{\Phi}_{\mathcal{TV}}$ is a submatrix of $\boldsymbol{\Phi}$ with selected rows indicated by set $\mathcal{T}$, and maximization leads to E-optimality [27] as mentioned in the "Deterministic Sampling Set Selection" section.

If the measurement matrix $\boldsymbol{\Phi}$ is the matrix $\mathbf{U}_M$ containing the first M eigenvectors of a graph Laplacian matrix $\mathbf{L}$, then we can interpret (36) as a graph signal sampling problem under an M-BL assumption. Sampling set selection methods described in the "Sampling Set Selection and Efficient Computation Methods" section can thus be used to solve (36). Specifically, recent fast graph sampling schemes [11] have been used for sensor selection with improved execution speed and reconstruction quality compared to Gaussian process-based methods.

Sampling for matrix completion

Matrix completion is the problem of filling or interpolating missing values in a partially observable matrix signal $\mathbf{X} \in \mathbb{R}^{N_r \times N_c}$, where N_r and N_c are often very large. One well-known example is the Netflix challenge (https://en.wikipedia.org/wiki/Netflix_Prize): to recommend movies to viewers, missing movie ratings in a large matrix, with viewers and movies as rows and columns, respectively, are estimated based on a small subset of available viewer ratings. As an ill-posed problem, signal priors are required for regularization. One popular prior is the low-rank prior: target matrix signal $\mathbf{X}$ should be of low dimensionality and, thus, low rank. However, $\operatorname{rank}(\mathbf{X})$ is nonconvex, and convexifying it to the nuclear norm $\|\mathbf{X}\|_*$ (the sum of singular values) still requires computing the SVD per iteration in a proximal gradient method, which is expensive.

The underlying assumption of a low-rank prior is that the items along the rows and columns are similar. One can thus alternatively model these pairwise similarity relations using two graphs [30], [31]. Specifically, columns of $\mathbf{X}$ are assumed to be smooth with respect to an undirected weighted row graph $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r, \mathbf{W}_r)$ with vertices $\mathcal{V}_r = \{1, \dots, m\}$ and edges $\mathcal{E}_r \subseteq \mathcal{V}_r \times \mathcal{V}_r$. Weight matrix $\mathbf{W}_r$ specifies pairwise similarities among vertices in $\mathcal{G}_r$. The combinatorial graph Laplacian matrix of $\mathcal{G}_r$ is $\mathbf{L}_r = \mathbf{D}_r - \mathbf{W}_r$, where the degree matrix $\mathbf{D}_r$ is diagonal with entries $[\mathbf{D}_r]_{ii} = \sum_j [\mathbf{W}_r]_{ij}$. The work in [31] assumes that all columns of the matrix signal $\mathbf{X}$ are BL with respect to the graph frequencies defined using $\mathbf{L}_r$. As an alternative to strict bandlimitedness, [30] assumes that the columns of $\mathbf{X}$ are smooth with respect to $\mathbf{L}_r$, resulting in a small $\operatorname{Tr}(\mathbf{X}^\top \mathbf{L}_r \mathbf{X})$.

Similarly, one can define a column graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c, \mathbf{W}_c)$, with vertices $\mathcal{V}_c, = \{1, \dots, n\}$, edges $\mathcal{E}_c \subseteq \mathcal{V}_c \times \mathcal{V}_c$, and weight matrix $\mathbf{W}_c$, for the rows of $\mathbf{X}$. One can thus assume bandlimitedness for the rows of $\mathbf{X}$ with respect to the corresponding Laplacian $\mathbf{L}_c$ [31] or, simply, that the rows of $\mathbf{X}$ are smooth with respect to $\mathbf{L}_c$ [30].

Given a sampling set $\boldsymbol{\Omega} = \{(i,j) \,|\, i \in \{1, \dots, N_r\}, j \in \{1, \dots, N_c\}\}$, denote by $\mathbf{A}_\Omega$ the sampling matrix:

$$[\mathbf{A}_\Omega]_{ij} = \begin{cases} 1, & \text{if } (i,j) \in \boldsymbol{\Omega}; \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

We can now formulate the matrix completion problem with double graph Laplacian regularization as follows [32]:

$$\begin{aligned} \min_{\mathbf{X}} f(\mathbf{X}) = \frac{1}{2}\|\mathbf{A}_\Omega \circ (\mathbf{X} - \mathbf{Y})\|_F^2 + \frac{\alpha}{2}\operatorname{Tr}(\mathbf{X}^\top \mathbf{L}_r \mathbf{X}) \\ + \frac{\beta}{2}\operatorname{Tr}(\mathbf{X}\mathbf{L}_c\mathbf{X}^\top), \end{aligned} \tag{38}$$

where α and β are weight parameters. To solve the unconstrained quasi-peak problem (38), one can take the derivative with respect to $\mathbf{X}$, set it to zero, and solve for $\mathbf{X}$, resulting in a system of linear equations for the unknown vectorized $\operatorname{vec}(\mathbf{X}^*)$:

$$(\tilde{\mathbf{A}}_\Omega + \alpha \mathbf{I}_n \otimes \mathbf{L}_r + \beta \mathbf{L}_c \otimes \mathbf{I}_m)\operatorname{vec}(\mathbf{X}^*) = \operatorname{vec}(\mathbf{A}_\Omega \circ \mathbf{Y}), \tag{39}$$

where $\tilde{\mathbf{A}}_\Omega = \operatorname{diag}(\operatorname{vec}(\mathbf{A}_\Omega))$, $\operatorname{vec}(\cdot)$, means a vector form of a matrix by stacking its columns, and $\operatorname{diag}(\cdot)$ creates a diagonal matrix with the input vector as its diagonal elements.

A solution to (39) can be efficiently found, e.g., by using the conjugate gradient method.

There are practical scenarios where the available observed entries $\mathbf{A}_\Omega \circ \mathbf{Y}$ in a matrix $\mathbf{X}$ are not provided a priori but must be actively sampled first. This problem of how to best choose matrix entries for later completion given a sampling budget is called *active matrix completion*—a popular research topic in the machine learning community. Extending sampling algorithms for signals in single graphs, as discussed in previous sections, the authors in [31] and [32] propose sampling algorithms to select matrix entries, assuming that the target signal $\mathbf{X}$ is BL or smooth over both row and column graphs, respectively.

In a nutshell, the approach in [31] first selects rows and columns separately based on BL assumptions on row and column graphs and then chooses matrix entries that are indexed by the selected rows and columns. In contrast, [32] greedily selects one matrix entry at a time by considering the row and column graph smoothness alternately, where each greedy selection seeks to maximize the smallest eigenvalue of the coefficient matrix in (39)—the E-optimality criterion. In Figure 4, we see an example of a low-rank matrix and sampling performance (in root-mean-square error) of [31] (BL) under different bandwidth assumptions for row and column graphs and [32] iterative Gershgorin circle shift (IGCS). We see that BL and IGCS perform comparably for a large sample budget, but BL is sensitive to the assumed row and column graph bandwidths.

Subsampling of 3D point cloud

A point cloud (PC)—a collection of discrete geometric samples (i.e., 3D coordinates) of a physical object in 3D space—is a popular visual signal representation for 3D imaging applications, such as virtual reality. PCs can be very large, with millions of points, making subsequent image processing tasks, such as viewpoint rendering or object detection/recognition, very computation intensive. To lighten this computation load, one can perform PC subsampling: selection of a representative 3D point subset such that the salient geometric features of the original PC are well preserved.

Previous works in PC subsampling either employ a random or regular sampling approach that does not preserve shape characteristics proactively [33], [34] or maintain only obvious features like sharp corners and edges [36]. Instead, leveraging on a recent fast eigendecomposition-free graph sampling algorithm [13], [35], PC sampling is performed that preserves the overall shape of the original PC in a worst-case reconstruction sense. After connecting 3D points in a PC into a k-nearest-neighbor graph, a postprocessing procedure is first assumed to superresolve a subsampled PC to full resolution based on a variant of a graph-based regularization, similar in form to (32), that expects the sought signal to be smooth with respect to the constructed graph. Like (32), the superresolution procedure amounts to solving a system of linear equations, where the coefficient matrix $\mathbf{B}$ is a function of the PC sampling matrix $\mathbf{H}$. They then derive a sampling objective that maximizes the smallest eigenvalue $\lambda_{\min}(\mathbf{B})$ of $\mathbf{B}$—the E-optimality criterion—through the selection of $\mathbf{H}$. In Figure 5, one can observe that PCs subsampled via graph sampling [35] (denoted by "GS") outperformed other schemes [33], [34] in reconstruction quality after subsequent PC superresolution.

Closing remarks

In this article, we overview sampling on graphs from theory to applications. The graph sampling framework is similar to sampling for standard signals; however, its realization is completely different due to the irregular nature of the graph domain. Current methods have found several interesting applications. At the same time, the following issues, both theoretical and practical aspects, are still open:

- *Interconnection between vertex and spectral representations of sampling*: As shown in the "Sampling Methods" section,

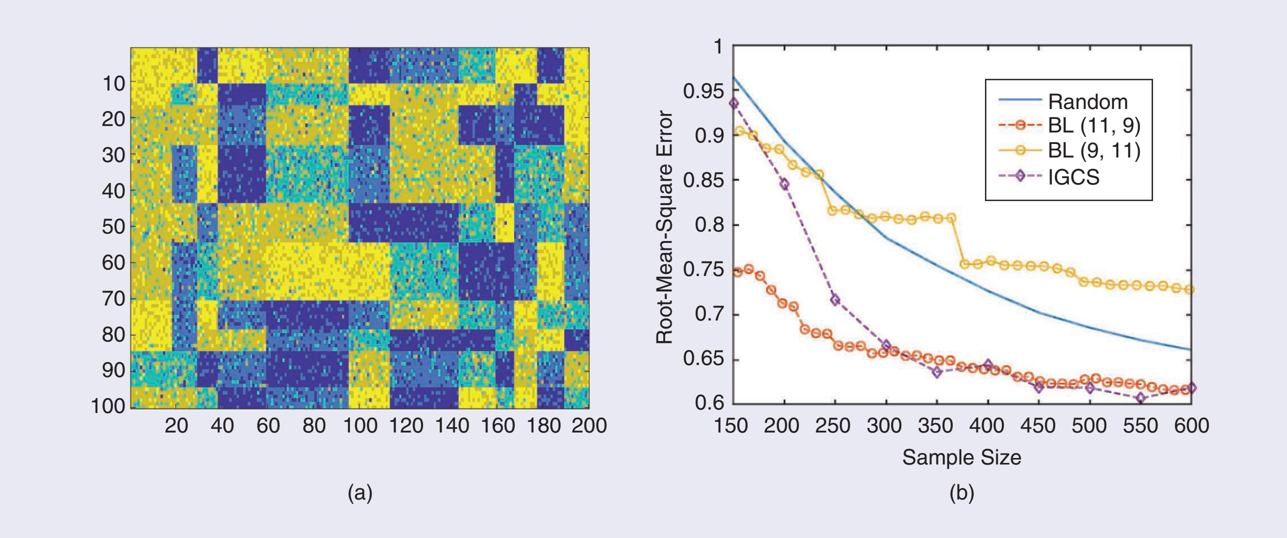


FIGURE 4. A performance comparison of the graph sampling algorithms (random, [31], [32]) for matrix completion: (a) an example of a low-rank matrix and (b) root-mean-square error comparison.

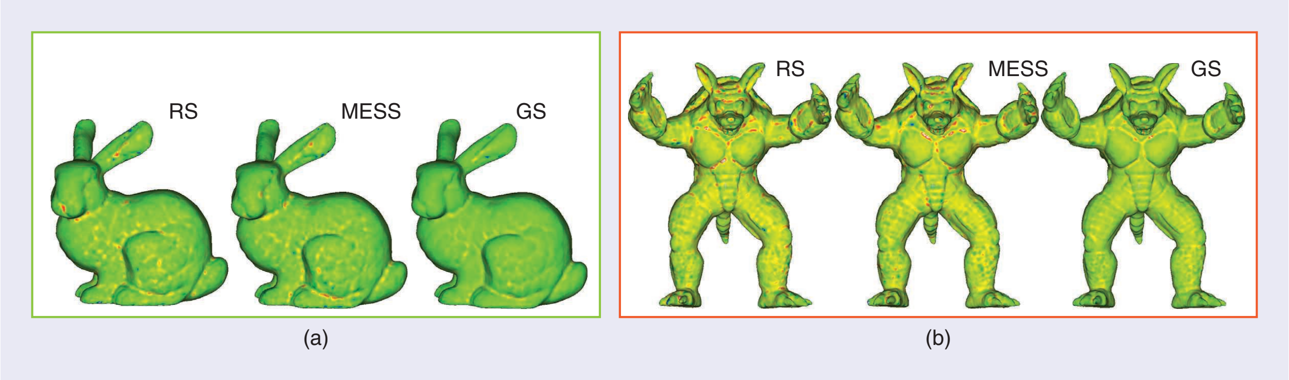


FIGURE 5. The reconstruction results from different 3D PC subsampling methods—RS [33], MESS [34], and GS [36]—for (a) bunny and (b) armadillo models. The surface is colorized by the distance from the ground-truth surface: green/yellow means smaller errors, and blue/red means larger errors. RS: random sampling; MESS: mesh element subsampling; GS: graph sampling.

two definitions can be possible for graph signal sampling. Can these sampling approaches be described in a more unified way beyond a few known special cases? This may lead to a more intuitive understanding of graph signal sampling.

- *Studies beyond BL graph signals*: Most studies in graph signal sampling are based on sampling and reconstruction of BL (or smooth) graph signals. However, as shown in the "Generalized Sampling in Hilbert Spaces" section, sampling methods beyond the BL setting have been studied in standard sampling. Investigating GSP systems beyond the BL assumption will be beneficial for many practical applications since real data are often not BL. Such examples include generalized graph signal sampling [15] and PGS sampling [8].
- *Fast and efficient deterministic sampling*: Eigendecomposition-free methods are a current trend for graph signal sampling, as seen in the "Deterministic Sampling Set Selection" section, but their computational complexities are still high compared to random methods. Furthermore, current deterministic approaches are mostly based on greedy sampling. Faster deterministic graph sampling methods are required that will be tractable for graphs with millions and even billions of nodes.
- *Fast and distributed reconstruction*: Similar to sampling, the reconstruction step also requires an eigendecomposition-free interpolation algorithm. Such an algorithm is expected to be implemented in a distributed manner. While fast filtering methods have been studied, as briefly introduced in the "Graph-Sampling Theory" section, fast and more accurate interpolation methods of signals on a graph are still required.
- *Applications*: Some direct applications of graph signal sampling have been introduced in the "Applications" section. Note that sampling itself is ubiquitous in signal processing and machine learning: many applications can apply graph signal sampling as an important ingredient. For example, graph neural networks and PC processing are potential areas of application because it is often convenient to treat available data as signals on a structured graph. Continued discussions with domain experts in different areas can facilitate applications of graph sampling theory and algorithms to the wider field of data science.

Acknowledgments

We thank Sarath Shekkizhar, Benjamin Girault, Fen Wang, and Chinthaka Dinesh for preparing the figures. This work was supported in part by the Japan Science and Technology Agency Precursory Research for Embryonic Science and Technology under grant JPMJPR1935, the Japan Society for the Promotion of Science Grants-in-Aid for Scientific Research under grants 19K22864 and 20H02145, the European Union's Horizon 2020 Research and Innovation Program under grant 646804-ERC-COG-BNYQ, the U.S. National Science Foundation under grants CCF-1410009 and CCF-1527874, and Natural Sciences and Engineering Research Council grants RGPIN-2019–06271 and RGPAS-2019-00110.

Authors

Yuichi Tanaka (ytnk@cc.tuat.ac.jp) is an associate professor in the Department of Electrical Engineering and Computer Science, Tokyo University of Agriculture and Technology, Japan. Currently, he has a cross appointment as a PRESTO Researcher with the Japan Science and Technology Agency. He served as an associate editor of *IEEE Transactions on Signal Processing* from 2016 to 2020. His current research interests are in the field of high-dimensional signal processing and machine learning, which includes graph signal processing, geometric deep learning, sensor networks, image/video processing in extreme situations, biomedical signal processing, and remote sensing. He is a Senior Member of IEEE.

Yonina C. Eldar (yonina.eldar@weizmann.ac.il) is a professor in the Department of Math and Computer Science at the Weizmann Institute of Science, Rehovot, Israel, where she heads the center for Biomedical Engineering and Signal Processing. She is also a visiting professor at the Massachusetts Institute of Technology and at the Broad Institute and an adjunct professor at Duke University, and she was a visiting professor at Stanford University. She is the editor-in-chief of *Foundations and Trends in Signal Processing* and serves the IEEE on several technical and award committees. She has received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award, IEEE Aerospace and Electronic Systems

Society Fred Nathanson Memorial Radar Award, and IEEE Kiyo Tomiyasu Award. She is a member of the Israel Academy of Sciences and Humanities and a fellow of the European Association for Signal Processing. She is a Fellow of IEEE.

Antonio Ortega (antonio.ortega@sipi.usc.edu) is a professor of Electrical and Computer Engineering at the University of Southern California. He is the editor-in-chief of *IEEE Transactions on Signal and Information Processing Over Networks* and recently served as a member of the board of governors of the IEEE Signal Processing Society. He has received several paper awards, including the 2016 Signal Processing Magazine Award. His research interests include graph signal processing, machine learning, multimedia compression, and wireless sensor networks. He is a fellow of the European Association for Signal Processing and a member of the Association for Computing Machinery and Asia-Pacific Signal and Information Processing Association. He is a Fellow of IEEE.

Gene Cheung (genec@yorku.ca) received his Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 2000. He is an associate professor at York University, Toronto, Canada. He has served as associate editor for multiple journals, including *IEEE Transactions on Multimedia* (2007–2011) and *IEEE Transactions on Image Processing* (2015–2019). He is a coauthor of several papers that received awards, including the Best Student Paper Award at IEEE ICIP 2013; ICIP 2017; and IEEE Image, Video, and Multidimensional Signal Processing Workshop 2016. He is a recipient of the Canadian Natural Sciences and Engineering Research Council Discovery Accelerator Supplement 2019. His research interests include 3D imaging and graph signal processing.

References

[1] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems.* Cambridge, U.K.: Cambridge Univ. Press, 2015.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, Oct. 2013. doi: 10.1109/MSP.2012.2235192.

[3] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018. doi: 10.1109/JPROC.2018.2820126.

[4] 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. doi: 10.1109/TSP.2013.2238935.

[5] G. Cheung, E. Magli, Y. Tanaka, and M. Ng, "Graph spectral image processing," *Proc. IEEE*, vol. 106, no. 5, pp. 907–930, May 2018. doi: 10.1109/JPROC.2018.2799702.

[6] I. Pesenson, "A sampling theorem on homogeneous manifolds," *Trans. Amer. Math. Soc.*, vol. 352, no. 9, pp. 4257–4269, 2000. doi: 10.1090/S0002-9947-00-02592-7.

[7] I. Pesenson, "Sampling in Paley–Wiener spaces on combinatorial graphs," *Trans. Amer. Math. Soc.*, vol. 360, no. 10, pp. 5603–5627, 2008. doi: 10.1090/S0002-9947-08-04511-X.

[8] Y. Tanaka and Y. C. Eldar, "Generalized sampling on graphs with subspace and smoothness priors," *IEEE Trans. Signal Process.*, vol. 68, pp. 2272–2286, Mar. 2020. doi: 10.1109/TSP.2020.2982325.

[9] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015. doi: 10.1109/TSP.2015.2469645.

[10] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, July 2016. doi: 10.1109/TSP.2016.2546233.

[11] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, "Eigendecomposition-free sampling set selection for graph signals," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2679–2692, May 2019. doi: 10.1109/TSP.2019.2908129.

[12] Y. Tanaka, "Spectral domain sampling of graph signals," *IEEE Trans. Signal Process.*, vol. 66, no. 14, pp. 3752–3767, July 2018. doi: 10.1109/TSP.2018.2839620.

[13] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, and W. Gao, "Fast graph sampling set selection using Gershgorin disc alignment," *IEEE Trans. Signal Process.*, vol. 68, pp. 2419–2434, 2020. doi: 10.1109/TSP.2020.2981202.

[14] P. D. Lorenzo, S. Barbarossa, and P. Banelli, "Sampling and recovery of graph signals," in *Cooperative and Graph Signal Processing*, P. M. Djurić and C. Richard, Eds. Amsterdam, The Netherlands: Elsevier, 2018, pp. 261–282.

[15] S. P. Chepuri, Y. C. Eldar, and G. Leus, "Graph sampling with and without input priors," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing (ICASSP)*, 2018, pp. 4564–4568.

[16] F. R. K. Chung, *Spectral Graph Theory* (CBMS Regional Conference Series in Mathematics, No. 92). American Mathematical Society, 1997.

[17] A. M. Duval and V. Reiner, "Perron–Frobenius type results and discrete versions of nodal domain theorems," *Linear Algebra Appl.*, vol. 294, no. 1–3, pp. 259–268, 1999. doi: 10.1016/S0024-3795(99)00090-7.

[18] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Representations of piecewise smooth signals on graphs," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6370–6374. doi: 10.1109/ICASSP.2016.7472903.

[19] A. Sakiyama, K. Watanabe, Y. Tanaka, and A. Ortega, "Two-channel critically-sampled graph filter banks with spectral domain sampling," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1447–1460, Mar. 2019. doi: 10.1109/TSP.2019.2892033.

[20] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications.* Cambridge, U.K.: Cambridge university press, 2012.

[21] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery Data Min*ing, 2014, pp. 492–501. doi: 10.1145/2623330.2623760.

[22] F. Wang, G. Cheung, and Y. Wang, "Low-complexity graph sampling with noise and signal reconstruction via Neumann series," *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5511–5526, 2019. doi: 10.1109/TSP.2019.2940129.

[23] L. F. O. Chamon and A. Ribeiro, "Greedy sampling of graph signals," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 34–47, 2017. doi: 10.1109/TSP.2017.2755586.

[24] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal recovery on graphs: Fundamental limits of sampling strategies," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 539–554, 2016. doi: 10.1109/TSIPN.2016.2614903.

[25] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Appl. Comput. Harmon. Anal.*, vol. 44, no. 2, pp. 446–475, Mar. 2018. doi: 10.1016/j.acha.2016.05.005.

[26] N. Perraudin, B. Ricaud, D. I. Shuman, and P. Vandergheynst, "Global and local uncertainty principles for signals on graphs," *APSIPA Trans. Signal Inform. Process.*, vol. 7, p. e3, Apr. 2018. doi: 10.1017/ATSIP.2018.2.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[28] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, Feb. 2008.

[29] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1135–1146, 2014. doi: 10.1109/TSP.2014.2299518.

[30] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs," 2014, arXiv:1408.1717.

[31] G. Ortiz-Jiménez, M. Coutino, S. P. Chepuri, and G. Leus, "Sparse sampling for inverse problems with tensors," *IEEE Trans. Signal Process.*, vol. 67, no. 12, pp. 3272–3286, 2019. doi: 10.1109/TSP.2019.2914879.

[32] F. Wang, Y. Wang, G. Cheung, and C. Yang, "Graph sampling for matrix completion using recurrent Gershgorin disc shift," *IEEE Trans. Signal Process.*, vol. 68, pp. 2814–2829, Apr. 2020. doi: 10.1109/TSP.2020.2988784.

[33] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auton. Robots*, vol. 34, no. 3, pp. 133–148, 2013. doi: 10.1007/s10514-013-9327-2.

[34] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: An open-source mesh processing tool," in *Proc. Eurographics Italian Chapter Conf.*, 2008, pp. 129–136.

[35] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, "Fast resampling of three-dimensional point clouds via graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 666–681, Feb. 2018. doi: 10.1109/TSP.2017.2771730.

[36] C. Dinesh, G. Cheung, F. Wang, and I. Bajić, "Sampling of 3D point cloud via Gershgorin disc alignment," in *Proc. IEEE Conf. Image Processing*, to be published.