Learning Stochastic Graph Neural Networks With Constrained Variance

Gao, Zhan ; Isufi, Elvin

DOI
10.1109/TSP.2023.3244101

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Signal Processing

Learning Stochastic Graph Neural Networks With Constrained Variance

Zhan Gao, *Graduate Student Member, IEEE*, and Elvin Isufi, *Member, IEEE*

***Abstract*—Stochastic graph neural networks (SGNNs) are information processing architectures that learn representations from data over random graphs. SGNNs are trained with respect to the expected performance, which comes with no guarantee about deviations of particular output realizations around the optimal expectation. To overcome this issue, we propose a variance-constrained optimization problem for SGNNs, balancing the expected performance and the stochastic deviation. An alternating primal-dual learning procedure is undertaken that solves the problem by updating the SGNN parameters with gradient descent and the dual variable with gradient ascent. To characterize the explicit effect of the variance-constrained learning, we analyze theoretically the variance of the SGNN output and identify a trade-off between the stochastic robustness and the discrimination power. We further analyze the duality gap of the variance-constrained optimization problem and the converging behavior of the primal-dual learning procedure. The former indicates the optimality loss induced by the dual transformation and the latter characterizes the limiting error of the iterative algorithm, both of which guarantee the performance of the variance-constrained learning. Through numerical simulations, we corroborate our theoretical findings and observe a strong expected performance with a controllable variance.**

***Index Terms*—Stochastic graph neural networks, variance constraint, primal-dual learning, duality gap, convergence.**

I. Introduction

NETWORKED data exhibits an irregular structure inherent in its underlying topology and can be represented as signals residing on the nodes of a graph [2]. Graph neural networks (GNNs) exploit this structural information to model task-relevant representations from graph signals [3], [4], [5], [6], which have found applications in recommender systems [7], [8], multi-agent coordination [9], [10], and wireless communications [11], [12], [13]. The success of GNNs can be attributed to their ability to leverage the coupling between the signal and the graph, but the latter may be perturbed due to adversarial attacks, link losses in distributed communications, or topological estimation errors. In this setting, the graph encountered during testing differs from the one used during training; hence, questioning the stability to such perturbations.

The stability of GNNs has been investigated in [14], [15], [16], [17], [18]. The work in [14] showed GNNs can be both stable to small topological perturbations and discriminative at high graph frequencies. The works in [15], [16] analyzed the stability of graph filters –the linear inner working mechanism of GNNs that captures the graph-data coupling [19]– and GNNs under structural perturbations and provided interpretable stability bounds. The work in [17] established GNNs can extract similar representations on graphs that describe the same phenomenon, while [18] extended the stability results to algebraic neural networks where GNNs can be seen as a particular case. The aforementioned works discuss the GNN stability w.r.t. small deterministic perturbations. However, the graph often changes randomly, resulting in stochastic perturbations that cannot be addressed with the above analysis.

Stochastic perturbations appear when GNNs are implemented distributively on physical networks [20], [21], [22], where communication links fall with a certain probability due to channel fading effects, leading to random communication graphs [23], [24], [25]. Other cases, in which GNNs operate on stochastic graphs, involve recommender systems, where the graph stochasticity is introduced to improve the recommendation diversity [26], [27], [28]. The impact of stochastic perturbations on graph filters has been analyzed in [20], while [29] extended the analysis to scenarios with both graph randomness and quantization effects. The work in [30] studied the stability of low pass graph filters to edge rewiring on the stochastic block model. Authors in [31] characterized the stability of GNNs to stochastic perturbations and identified the role played by the filter, nonlinearity, and architecture.

The work in [32] proposed stochastic graph neural networks (SGNNs) that account for the graph stochasticity during training to alleviate the performance degradation due to stochastic perturbations. Learning with uncertainty makes the trained model robust to perturbations encountered during testing, and thus endows the SGNN with robust transference properties. The graph stochasticity has also been considered during training as a regularization technique to prevent over-smoothing [33] or as a data augmentation technique to avoid over-fitting [34], [35]. While improving the stability to perturbations, training an SGNN implies optimizing the expected performance w.r.t. the random topology in an empirical risk minimization framework [32].

Manuscript received 5 August 2022; revised 23 December 2022; accepted 20 January 2023. Date of publication 10 February 2023; date of current version 27 February 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Cihan Tepedelenlioglu. The work of Elvin Isufi was supported by the TU Delft AI Labs Programme. An earlier version of this paper was presented at the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) [DOI: 10.1109/ICASSP39728.2021.9413751]. *(Corresponding author: Zhan Gao.)*

Zhan Gao is with the Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, U.K. (e-mail: zg292@cam.ac.uk).

Elvin Isufi is with the Department of Intelligent Systems, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: e.isufi-1@tudelft.nl).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2023.3244101, provided by the authors.

Digital Object Identifier 10.1109/TSP.2023.3244101

However, such a strategy does not provide any guarantee about the deviation of a single SGNN realization around the optimal expectation; hence, an undesirable performance may appear in individual realizations, even when the expected performance is satisfactory.

Variance reduction techniques for GNNs have also been developed for graph sampling methods to reduce the computational cost during training. Specifically, the work in [36] reduced the mini-batch variance by maintaining the historical embedding of the previous layer and assuming the embedding would be close to its history. The works in [37], [38] reduced the number of sampled nodes at each layer to keep the variance low, while [39], [40] used an adaptive sampling and trained the sampling distribution towards minimum sampling variance. The works in [41], [42] optimized the sampling strategy based on task performance by using the history of node embedding and gradient information for variance reduction. The aforementioned works impose the topological stochasticity during training to reduce computation, but implement the architecture over deterministic graph during inference. The latter results in a mismatch between training and testing, and may suffer from performance degradation under stochastic perturbations during implementation or when the sampling during training is large [31]. Moreover, these works reduce the variance by changing the graph sampling strategies, e.g., the sampling probability and the neighborhood size. However, stochastic perturbations are typically determined by external factors such as channel fading effects in communication networks and adversarial attacks, which cannot be changed during training.

In this work, we propose a variance-constrained learning strategy for SGNNs that does not control the sampling strategy and is tailored to stochastic graphs during inference. The proposed strategy adheres to solving a stochastic optimization problem w.r.t. the expected performance subject to a variance constraint. This is a challenging problem because of the constraint, the stochastic nature of the topology, and the non-convexity of the SGNN. Following recent advances in constrained learning [43], we adopt a primal-dual learning procedure to solve the problem. To study the effect of such strategy on the SGNN learning capacity, we characterize theoretically its output variance and identify a trade-off between the improved deviation robustness and the degraded discrimination power. Our detailed contribution is threefold:

1) *Variance-constrained learning (Section III):* We formulate a constrained stochastic optimization problem that balances the expected performance with the stochastic deviation. We solve this problem via a primal-dual learning procedure that updates alternatively the primal SGNN parameters with gradient descent and the dual variable with gradient ascent. We show this strategy acts as a self-learning variance regularizer.
2) *Variance and discrimination (Section IV):* We analyze theoretically the variance of the SGNN output and identify the effect of the filter property, graph stochasticity and architecture. The variance-constrained learning restricts the variance by allowing less variability of the filter frequency response; ultimately, leading to a trade-off between the stochastic deviation robustness and the SGNN discrimination power.
3) *Duality gap and convergence (Sections V–VI):* We analyze the optimality loss of the variance-constrained learning by characterizing the duality gap of the formulated optimization problem and the converging behavior of the proposed primal-dual algorithm. The sub-optimality is bounded proportionally by the representation capacity of the SGNN, the gradient descent approximation at the primal phase, and the gradient ascent step-size at the dual phase. These findings validate the effectiveness of the variance-constrained learning and identify our handle to obtain near-optimal solutions.

This paper contains one additional minor contribution. It conducts theoretical analysis of stochastic graph filters and SGNNs with a more general stochastic graph model than earlier works [20], [32], where a subset of edges are dropped with a probability p and another subset are added with another probability q [Def. 1]. The theoretical findings of this work are not presented in the preliminary version [1], which focused on the algorithm. Numerical simulations on source localization and recommender systems corroborate the theoretical findings in Section VII. The conclusions are drawn in Section VIII. All proofs and lemmas used in these proofs are collected in the supplementary material.

II. Stochastic Graph Neural Network

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{S})$ be a graph with node set $\mathcal{V} = \{1, \ldots, n\}$, edge set $\mathcal{E} = \{(i, j)\} \subseteq \mathcal{V} \times \mathcal{V}$, and graph shift operator $\mathbf{S} \in \mathbb{R}^{n \times n}$, e.g., the adjacency matrix $\mathbf{A}$ or the Laplacian matrix $\mathbf{L}$. Let also $\mathbf{x} = [x_1, \ldots, x_n]^\top \in \mathbb{R}^n$ be a graph signal with component x_i the signal value associated to node i [44], [45], [46], [47]. For example, in a recommender system nodes are movies, edges are similarities between them, and the graph signal is the ratings given by a user to these movies. We are interested in learning representations from the tuple $(\mathcal{G}, \mathbf{x})$ for tasks such as inferring user missing ratings, while we aim to keep these representations robust w.r.t. random topological changes on the nominal graph. These random changes may be due to different factors such as adversarial attacks [48], communication link outage [49], and edge rewiring in collaborative filtering to improve diversity [50]. In these cases, existing edges may be lost and new edges may be added, resulting in random topologies. We characterize the latter with the generalized random edge sampling (GRES) model.

Definition 1 (GRES(p, q) model): Consider the nominal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{E}_d \subseteq \mathcal{E}$ be a set of M_d existing edges that may be dropped and $\mathcal{E}_a \nsubseteq \mathcal{E}$ a set of M_a new edges that may be added. A GRES graph realization $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ of $\mathcal{G}$ comprises the same node set $\mathcal{V}$ and the edge set $\mathcal{E}_k$ where the edges in $\mathcal{E}_d$ are dropped independently with a probability $0 \leq p < 1$ and the edges in $\mathcal{E}_a$ are added independently with a probability $0 \leq q < 1$.

We denote by $\mathbf{S}_k$ the random shift operator of the GRES(p, q) graph $\mathcal{G}_k$ with $2^{M_d + M_a}$ possible realizations.

Stochastic graph neural network (SGNN) [32]: An SGNN is a graph neural network that learns representations over random

topologies. The key of this architecture is the *stochastic graph filter*. When applied to a graph signal $\mathbf{x}$, the output of a stochastic graph filter over a sequence of K GRES(p, q) graph realizations $\{\mathbf{S}_k\}_{k=0}^K$ can be written as

$$\mathbf{H}(\mathbf{S}_{K:0})\mathbf{x} := \sum_{k=0}^{K} h_k \mathbf{S}_k \dots \mathbf{S}_1 \mathbf{S}_0 \mathbf{x} = \sum_{k=0}^{K} h_k \prod_{i=0}^{k} \mathbf{S}_i \mathbf{x} \quad (1)$$

with $\{h_k\}_{k=0}^K$ the filter coefficients and $\mathbf{S}_0 = \mathbf{I}$ the identity matrix [32]. In the filter output (1), the first shift $\mathbf{S}_1\mathbf{x}$ collects at each node the information from its immediate neighbors and the successive k-shifts $\prod_{i=0}^{k} \mathbf{S}_i\mathbf{x}$ collect information from k-hop neighbors that can be reached via the randomly present edges in $\mathbf{S}_1, \dots, \mathbf{S}_k$. The stochastic graph filter aggregates these shifted signals $\{\prod_{i=0}^{k} \mathbf{S}_i\mathbf{x}\}_{k=0}^K$ and weighs them with coefficients $\{h_k\}_{k=0}^K$; ultimately, allowing for a distributed implementation – see also [19], [22], [45].

An SGNN is a layered architecture, in which each layer comprises a bank of stochastic graph filters followed by a pointwise nonlinearity. At layer $\ell = 1, \dots, L$, the input is a collection of F graph signal features $\{\mathbf{x}_{\ell-1}^g\}_{g=1}^F$ generated at the former layer $\ell - 1$. These features are processed by a bank of F^2 stochastic graph filters $\{\mathbf{H}_\ell^{fg}(\mathbf{S}_{K:0})\}_{fg}$ [cf. (1)], aggregated over the input index g, and finally passed through a nonlinearity $\sigma(\cdot)$ to generate F output features

$$\mathbf{x}_\ell^f = \sigma\left(\sum_{g=1}^{F} \mathbf{u}_\ell^{fg}\right) = \sigma\left(\sum_{g=1}^{F} \mathbf{H}_\ell^{fg}(\mathbf{S}_{K:0})\mathbf{x}_{\ell-1}^g\right), \text{ for } f = 1, \dots, F. \quad (2)$$

To ease exposition, we consider a single input $\mathbf{x}_0^1 = \mathbf{x}$ and output $\mathbf{x}_L^1$. We represent the SGNN as the nonlinear map $\mathbf{\Phi}(\cdot; \mathbf{S}_{P:1}, \mathcal{H}) : \mathbb{R}^n \to \mathbb{R}^n$, which applies on the input $\mathbf{x}$ and generates the output $\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H}) := \mathbf{x}_L^1$. The set $\mathcal{H} = \{h_{0\ell}^{fg}, \dots . h_{K\ell}^{fg}\}_{fg\ell}$ collects all filter coefficients and $\mathbf{S}_{P:1}$ indicates the sequence of all $P = K[2F + (L-2)F^2]$ shift operators in the SGNN.

Problem motivation: The SGNN output $\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})$ is random because of the graph stochasticity and the data distribution. Given a training set $\mathcal{T} = \{(\mathbf{x}, \mathbf{y})\}$ and a loss function $\mathcal{C}(\cdot, \cdot)$, we train the SGNN with stochastic gradient descent, which is shown equivalent to solving an unconstrained stochastic optimization problem over the graph and the data distributions [32]. I.e.,

$$\mathbb{P}_{\text{un}} := \min_{\mathcal{H}} \mathbb{E}_{\mathcal{M}}[\mathcal{C}_{\mathcal{T}}(\mathbf{y}, \mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H}))] \quad (3)$$

where $\mathcal{C}_{\mathcal{T}}(\mathbf{y}, \mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})) := \mathbb{E}_{\mathcal{T}}[\mathcal{C}(\mathbf{y}, \mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H}))]$ is the expected cost over the data distribution and $\mathcal{M}$ is the discrete set of the shift operator sequences $\mathbf{S}_{P:1}$, which contains $2^{P(M_d+M_a)}$ elements. The expectation of a function $f(\mathbf{x}; \mathbf{S}_{P:1})$ over the discrete set $\mathcal{M}$ is $\mathbb{E}_{\mathcal{M}} = \sum_{\mathbf{S}_{P:1} \in \mathcal{M}} f(\mathbf{x}; \mathbf{S}_{P:1}) \mu(\mathbf{S}_{P:1})$ where $\mu(\cdot)$ is the probability measure over $\mathcal{M}$ such that $\mu(\mathbf{S}_{P:1}) = 1/2^{P(M_d+M_a)}$ for each $\mathbf{S}_{P:1} \in \mathcal{M}$. The training converges to a stationary solution of (3), which accounts for the graph stochasticity and makes the SGNN robust when tested with random graphs [32]. However, problem (3) only guarantees robustness w.r.t. the expected performance but ignores stochastic deviations around it. The latter may lead to a single SGNN output far from the optimal expectation and be problematic when uncertainty must be controlled.

To overcome this issue, we propose a variance-constrained learning strategy to balance the expected performance with stochastic deviations. Specifically, we propose the constrained stochastic optimization problem as

$$\mathbb{P}_{\text{con}} := \min_{\mathcal{H}} \mathbb{E}_{\mathcal{M}}[\mathcal{C}_{\mathcal{T}}(\mathbf{y}, \mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H}))]$$
$$\text{s.t.} \quad \text{Var}[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{\text{P:1}}, \mathcal{H})] \leq \text{C}_\text{v} \quad (4)$$

where $\text{Var}[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{\text{P:1}}, \mathcal{H})]$ is a variance measure that characterizes stochastic deviations of the SGNN output and C_v is a variance bound we can tolerate. Problem (4) is challenging because of the non-convexity of the SGNN, the stochasticity of the GRES(p, q) model, and the variance constraint. We solve the problem via a primal-dual learning method in Section III. Since the proposed variance-constrained learning trades the variance with the discrimination power, we characterize this trade-off explicitly and show the role played by different factors in Section IV. We further analyze the optimality loss induced by the primal-dual method in Section V and prove this learning procedure converges to a neighborhood of the saddle point solution in Section VI.

III. Variance-Constrained Learning

We consider the average variance experienced over all nodes

$$\text{Var}[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{\text{P:1}}, \mathcal{H})] := \frac{1}{\text{n}} \sum_{\text{i=1}}^{\text{n}} \text{Var}[[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{\text{P:1}}, \mathcal{H})]_\text{i}]$$
$$= \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{E}_{\mathcal{M}}\left[[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})]_i^2\right] - \mathbb{E}_{\mathcal{M}}[[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})]_i]^2 \right). \quad (5)$$

This expression measures how individual node outputs $\{[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})]_i\}_{i=1}^n$ deviate from their expectations. It is a standard criterion used in multi-dimensional systems and is related to the A-optimality of the confidence ellipsoid [51]. In what follows, we use (5) as the variance measure in (4) and solve the latter problem with a primal-dual learning procedure. We further show how this learning strategy behaves as a *self-learning variance regularizer* that provides explicit theoretical guarantees about stochastic deviations.

Since problem (4) is a constrained optimization problem, we solve it in the dual domain. However, the variance constraint is a *non-convex* function of $\mathbb{E}_{\mathcal{M}}[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})]$ and $\mathbb{E}_{\mathcal{M}}[\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})^2]$. The latter makes it difficult to analyze the duality gap, which quantifies the optimality loss of the solution in the dual domain; consequently, there is no performance guarantee for any dual method solving (4) as we shall detail in Section V. To provide theoretical guarantees, we consider the surrogate problem where we constrain separately the first and second order moments in (5), i.e.,

$$\mathbb{P} := \min_{\mathcal{H}} \mathbb{E}_{\mathcal{M}}[\mathcal{C}_{\mathcal{T}}(\mathbf{y}, \mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H}))] \quad (6)$$

$$\text{s.t.}\frac{1}{n}\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]_i\right]\geq C_f,$$

$$\frac{1}{n}\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]_i^2\right]\leq C_s.$$

The constraints of (6) are *convex* functions (the outer function not the composed function with the SGNN) of the first order moment $\mathbb{E}_{\mathcal{M}}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]$ and of the second order moment $\mathbb{E}_{\mathcal{M}}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})^2]$, respectively.[1] Through scalar $C_f\geq 0$ we lower bound the expected output and through scalar $C_s\geq 0$ we upper bound the output autocorrelation. The latter are related to the variance (5); hence, we bound the variance as

$$\text{Var}\left[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})\right]\leq C_s-C_f^2. \tag{7}$$

Since there always exist C_f and C_s such that $C_s-C_f^2=C_v$, e.g., $C_f=0$ and $C_s=C_v$, the surrogate problem (6) restricts the SGNN output and balances the expected performance with the stochastic deviation as the original problem (4). However, C_f and C_s introduce certain bias on the first-order moment (expectation) and the second-order moment (energy) of the SGNN output, which may affect the learning performance. Specifically, the first-order constraint with C_f affects the space of feasible solutions, i.e., a larger C_f denotes a smaller space and reduces the representational capacity of feasible SGNNs. The second-order constraint with C_s affects the energy of output signals, i.e., a smaller C_s denotes a lower energy and increases the information loss. These factors need taking into consideration when selecting constraint constants for experiments – as we will discuss in Section VII.

A. Primal-Dual Learning

By introducing the non-negative dual variable $\boldsymbol{\gamma}=[\gamma_1,\gamma_2]\in\mathbb{R}_+^2$, we define the Lagrangian $\mathcal{L}(\mathcal{H},\boldsymbol{\gamma})$ of (6) as

$$\begin{aligned}\mathcal{L}(\mathcal{H},\boldsymbol{\gamma})=\mathbb{E}_{\mathcal{M}}[\mathcal{C}_{\mathcal{T}}(\mathbf{y},\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H}))]\\ +\gamma_1\left(C_f-\frac{1}{n}\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]_i\right]\right)\\ -\gamma_2\left(C_s-\frac{1}{n}\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]_i^2\right]\right).\end{aligned} \tag{8}$$

Given the dual function $\mathcal{D}(\boldsymbol{\gamma})=\min_{\mathcal{H}}\mathcal{L}(\mathcal{H},\boldsymbol{\gamma})$, it holds that $\mathcal{D}(\boldsymbol{\gamma})\leq\mathbb{P}$ for any $\boldsymbol{\gamma}$ [52]. The goal now is to find the optimal dual variable $\boldsymbol{\gamma}^*$ that maximizes the dual function as

$$\mathbb{D}=\max_{\boldsymbol{\gamma}}\mathcal{D}(\boldsymbol{\gamma}):=\max_{\boldsymbol{\gamma}}\min_{\mathcal{H}}\mathcal{L}(\mathcal{H},\boldsymbol{\gamma}). \tag{9}$$

That is, search for an optimal primal-dual pair $(\mathcal{H}^*,\boldsymbol{\gamma}^*)$ satisfying the saddle-point relationship $\mathcal{L}(\mathcal{H}^*,\boldsymbol{\gamma})\leq\mathcal{L}(\mathcal{H}^*,\boldsymbol{\gamma}^*)\leq\mathcal{L}(\mathcal{H},\boldsymbol{\gamma}^*)$ for any $\mathcal{H}$ and $\boldsymbol{\gamma}$ in the neighborhood of the optimal solution.

[1]A more intuitive constraint for the first order moment is to lower bound its absolute value $\left|\mathbb{E}_{\mathcal{M}}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]\right|\geq C_f$, i.e., $C_f-\left|\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]_i\right]/n\right|\leq 0$. However, the latter is still a non-convex function and thus does not allow for the duality gap analysis as (6).

We approach the dual problem (9) by alternatively updating the primal variable $\mathcal{H}$ with stochastic gradient descent and the dual variable $\boldsymbol{\gamma}$ with stochastic gradient ascent.

Primal phase: At iteration t, given the primal variable $\mathcal{H}_t$ and the dual variable $\boldsymbol{\gamma}_t$, we set $\mathcal{H}_t^{(0)}=\mathcal{H}_t$ and update the primal variable with gradient descent for Γ steps as

$$\mathcal{H}_t^{(\tau)}=\mathcal{H}_t^{(\tau-1)}-\eta_{\mathcal{H}}\nabla_{\mathcal{H}}\mathcal{L}(\mathcal{H}_t^{(\tau-1)},\boldsymbol{\gamma}_t),\ \text{for}\ \tau=1,\ldots,\Gamma, \tag{10a}$$

$$\mathcal{H}_{t+1}:=\mathcal{H}_t^{(\Gamma)} \tag{10b}$$

where $\eta_{\mathcal{H}}>0$ is the primal step-size. The challenge in (10) is to compute the gradient $\nabla_{\mathcal{H}}\mathcal{L}(\mathcal{H}_t^{(\tau-1)},\boldsymbol{\gamma}_t)$, which requires evaluating the expectation $\mathbb{E}_{\mathcal{M}}[\cdot]$. The latter needs to be estimated over $2^{P(M_d+M_a)}$ realizations resulting in an expensive computation. To overcome this issue, we approximate the expectation with empirical alternatives over N sampled realizations $\{\mathbf{S}_{P:1}^{(j)}\}_{j=1}^{N}$ as

$$\mathbb{E}_{\mathcal{M}}[\mathcal{C}_{\mathcal{T}}(\mathbf{y},\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H}))]\approx\frac{1}{N}\sum_{j=1}^{N}\mathcal{C}_{\mathcal{T}}(\mathbf{y},\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1}^{(j)},\mathcal{H})), \tag{11a}$$

$$\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]_i\right]\approx\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1}^{(j)},\mathcal{H})]_i, \tag{11b}$$

$$\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})]_i^2\right]\approx\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1}^{(j)},\mathcal{H})]_i^2. \tag{11c}$$

The sampling average is a standard procedure in stochastic optimization methods, such as Monte-Carlo simulation [53] and stochastic gradient descent [54]. A larger N leads to better approximations but needs more computations, which yields a trade-off between performance and complexity. The selection of N depends on the number of graph shift operators P (i.e., the architecture width F and depth L) and the graph size n because these factors affect the randomness of the SGNN output. We shall show in Section VII that for a graph of 50 nodes, an $N\geq 10$ suffices.

Dual phase: Given the updated primal variable $\mathcal{H}_{t+1}$, the dual variable is updated with a single step gradient ascent as

$$\gamma_{1,t+1}=\left[\gamma_{1,t}+\eta_\gamma\left(C_f-\frac{1}{n}\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H}_{t+1})]_i\right]\right)\right]_+, \tag{12a}$$

$$\gamma_{2,t+1}=\left[\gamma_{2,t}-\eta_\gamma\left(C_s-\frac{1}{n}\mathbb{E}_{\mathcal{M}}\left[\sum_{i=1}^{n}[\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H}_{t+1})]_i^2\right]\right)\right]_+ \tag{12b}$$

where $\eta_\gamma>0$ is the dual step-size and $[\cdot]_+$ is the non-negative projection since $\gamma_1,\gamma_2\geq 0$. In (12a) and (12b), we substitute the expectations with their empirical alternatives as in (11b) and (11c). These stochastic approximations allow updating the dual step and completing the iteration t. The algorithm is stopped either after a maximum number of iterations T or when a

Algorithm 1: Primal-Dual Learning Procedure.

Input: Training set $\mathcal{T}$, loss function $\mathcal{C}(\cdot,\cdot)$, initial primal variable $\mathcal{H}_0$, initial dual variable $\boldsymbol{\gamma}_0$, bounds C_f and C_s, primal step-size $\eta_{\mathcal{H}}$, and dual step-size η_{γ}
Establish the Lagrangian (8) and the dual problem (9)
for $t = 0, 1, 2, \ldots$ **do**
  Primal phase. Given $\mathcal{H}_t$ and $\boldsymbol{\gamma}_t$, update the primal variable with gradient descent for Γ steps [cf. (10)]
  Approximate $\mathcal{L}(\mathcal{H}_t^{(\tau-1)}, \boldsymbol{\gamma}_t)$ stochastically [cf. (11)]
  Dual phase. Given $\mathcal{H}_{t+1}$ and $\boldsymbol{\gamma}_t$, update the dual variable with stochastic gradient ascent [cf. (12)]
end for

tolerance on the gradient norm is reached. Algorithm 1 recaps this procedure.

Remark 1: Algorithm 1 is applicable to both the original problem (4) and the surrogate problem (6). We focus on the surrogate problem (6) because it allows to analyze its duality gap in Section V; hence, providing a unified exposition throughout the paper. However, if the duality analysis is not of interest and any local minima is acceptable, we can work with the original problem (4) directly. All the other theoretical findings – the above primal-dual learning, the discrimination analysis in Section IV and the convergence analysis in Section VI – apply to the original problem as well.

Remark 2: Any stochastic optimization algorithm can be used at the primal phase to solve the dual function $\min_{\mathcal{H}} \mathcal{L}(\mathcal{H}, \boldsymbol{\gamma})$ [cf. (9)]. We apply the stochastic gradient descent in (10) as a baseline method to ease the exposition. Other choices include the ADAM method, the quasi-Newton method, etc.

B. Self-Learning Variance Regularizer

An intuitive alternative to the variance-constrained problem (4) is to consider the variance as a regularizer for problem (3), i.e.,

$$\min_{\mathcal{H}} \mathbb{E}_{\mathcal{M}}[\mathcal{C}_{\mathcal{T}}(\mathbf{y}, \boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H}))] + \beta \mathrm{Var}[\boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{\mathrm{P}:1}, \mathcal{H})] \quad (13)$$

where $\beta > 0$ is the regularization parameter. The regularization term $\beta \mathrm{Var}[\boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{\mathrm{P}:1}, \mathcal{H})]$ incentivizes the SGNN output to have a small variance by forcing its parameters to trade between the expected cost and the variance. Problem (13) can be solved directly with stochastic gradient descent. However, we find it limiting in two aspects: (i) It does not provide theoretical guarantees for stochastic deviations. The explicit relation between the regularization term and the stochastic deviation is unclear, thus little insight or implication can be obtained; (ii) It is difficult to select a suitable regularization parameter β that well balances the expected performance and the variance. If β is too large, the SGNN would only restrict the variance but sacrifice the performance; if β is too small, the SGNN may generate outputs with a large variance. Deciding the value of β requires extensive cross-validation and could be computationally demanding.

Differently, the variance-constrained learning not only optimizes the SGNN parameters $\mathcal{H}$ akin to the variance regularized objective, but also learns the regularization parameter $\boldsymbol{\gamma}$ based on the variance bound. To see this, recall that minimizing the Lagrangian (8) at the primal phase is equivalent to solving

$$\min_{\mathcal{H}} \mathbb{E}_{\mathcal{M}}[\mathcal{C}_{\mathcal{T}}(\mathbf{y}, \boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H}))] - \frac{\gamma_1}{n} \mathbb{E}_{\mathcal{M}} \left[\sum_{i=1}^{n} [\boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})]_i \right] + \frac{\gamma_2}{n} \mathbb{E}_{\mathcal{M}} \left[\sum_{i=1}^{n} [\boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})]_i^2 \right]. \quad (14)$$

This is similar to the variance regularizer in (13), where the dual variable $\boldsymbol{\gamma} = [\gamma_1, \gamma_2]$ is the regularization parameter and the primal variable $\mathcal{H}$ is updated in the direction that reduces the variance [cf. (5)]. However, instead of hand-fixing $\boldsymbol{\gamma}$ at the outset, the variance-constrained learning updates $\boldsymbol{\gamma}$ at the dual phase based on the bounds of the first and second order moments C_f, C_s [cf. (12)]; ultimately, based on the variance bound C_v [cf. (7)]. Hence, we can consider the latter as a *self-learning variance regularizer*, where the regularization parameter is learned based on the variance bound C_v.

More importantly, feasible solutions of the variance-constrained problem provide explicit theoretical guarantees about stochastic deviations of the SGNN output around its expectation. The following proposition establishes the probability contraction bound for the SGNN output and the role of the variance constraint.

Proposition 1: Consider the variance-constrained problem (4). Let $\mathcal{H}$ be a feasible solution that satisfies the variance constraint. Then, for any $\varepsilon > 0$, it holds that

$$Pr\left(\frac{1}{n} \left\| \boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H}) - \mathbb{E}_{\mathcal{M}}[\boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})] \right\|^2 \leq \varepsilon \right) \geq 1 - \frac{C_v}{\varepsilon}.$$

Proof: See Appendix A in the supplementary material. ■

That is, the probability that an SGNN realization deviates from its expectation by at most ε is no more than a fraction of C_v/ε. When the variance constraint is strict, i.e., $C_v \to 0$, the bound approaches one and stochastic deviations are well-controlled, but it may be challenging to find a feasible solution. The result shows an explicit relation between the variance constraint and random SGNN behavior, which cannot be established by the regularizer in (13).

IV. Variance and Discrimination

Compared to the unconstrained problem (3), problem (4) trades the bounded variance with the expected performance. However, the explicit trade-off is unclear, i.e., how the imposed constraint affects the overall performance. To address the latter, we characterize theoretically the variance of the SGNN output and show that the variance-constrained learning improves the robustness to stochastic deviations by shrinking the frequency response of stochastic graph filters [cf. (1)] within the SGNN; thus, reducing the discrimination power. To obtain this result, we analyze next the SGNN behavior in the graph spectral domain.

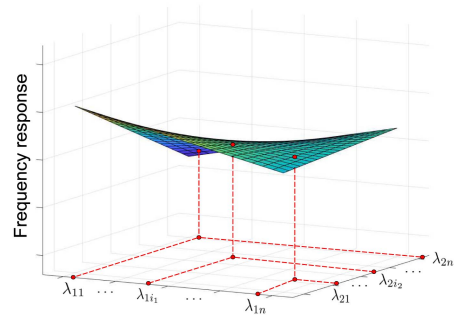


Fig. 1. The 2-dimensional frequency response of a stochastic graph filter. Function $h(\boldsymbol{\lambda})$ is independent of graph realizations and it is completely defined by parameters $\{h_k\}_{k=0}^K$ [cf. (17)]. For a specific chain of graph realizations $\{\mathbf{S}_1, \mathbf{S}_2\}$, $h(\boldsymbol{\lambda})$ is instantiated on specific eigenvalues $\{\lambda_{11}, \ldots, \lambda_{1n}\}$ determined by $\mathbf{S}_1$ and $\{\lambda_{21}, \ldots, \lambda_{2n}\}$ determined by $\mathbf{S}_2$.

A. Frequency Response of Stochastic Graph Filter

Consider the shift operator eigendecomposition $\mathbf{S}_k = \mathbf{V}_k \boldsymbol{\Lambda}_k \mathbf{V}_k^{\top}$ with eigenvectors $\mathbf{V}_k = [\mathbf{v}_{k1}, \ldots, \mathbf{v}_{kn}]$ and eigenvalues $\boldsymbol{\Lambda}_k = \text{diag}([\lambda_{k1}, \ldots, \lambda_{kn}])$. The graph Fourier transform (GFT) is the projection of signal $\mathbf{x}$ onto $\mathbf{V}_k$, i.e., $\mathbf{x} = \sum_{i=1}^n \hat{x}_i \mathbf{v}_{ki}$, where $\hat{\mathbf{x}} = [\hat{x}_1, \ldots, \hat{x}_n]^{\top}$ are the Fourier coefficients [46]. Given the eigendecompositions of $k+1$ successive shift operators $\mathbf{S}_0, \ldots, \mathbf{S}_k$, we can perform a chain of $k+1$ GFTs on $\mathbf{x}$ as

$$\prod_{i=0}^{k} \mathbf{S}_i \mathbf{x} = \sum_{i_0=1}^{n} \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} \hat{x}_{0i_0} \hat{x}_{1i_0 i_1} \cdots \hat{x}_{ki_{k-1}i_k} \prod_{j=0}^{k} \lambda_{ji_j} \mathbf{v}_{ki_k} \tag{15}$$

for all $k = 0, \ldots, K$, where we first perform the GFT over $\mathbf{S}_0$, then over $\mathbf{S}_1$, and so on. Here, $\{\hat{x}_{0i_0}\}_{i_0=1}^n$, $\{\hat{x}_{ji_{j-1}i_j}\}_{j=1}^k$ are the Fourier coefficients of expanding $\mathbf{x}$ on the chain of $\mathbf{S}_0, \ldots, \mathbf{S}_k$ – see also [31], [32]. Thus, we can represent the filter output $\mathbf{u} = \mathbf{H}(\mathbf{S}_{K:0})\mathbf{x}$ as

$$\mathbf{u} = \sum_{i_0=1}^{n} \sum_{i_1=1}^{n} \cdots \sum_{i_K=1}^{n} \hat{x}_{0i_0} \hat{x}_{1i_0 i_1} \cdots \hat{x}_{Ki_{K-1}i_K} \sum_{k=0}^{K} h_k \prod_{j=0}^{k} \lambda_{ji_j} \mathbf{v}_{Ki_K}. \tag{16}$$

As it follows from (16), the input-output relation of the filter in the spectral domain is determined by the eigenvalues $\boldsymbol{\Lambda}_K, \ldots, \boldsymbol{\Lambda}_1$ and eigenvectors $\mathbf{V}_K, \ldots, \mathbf{V}_1$. We can then define the *frequency response of the stochastic graph filter* as

$$h(\boldsymbol{\lambda}) := \sum_{k=0}^{K} h_k \prod_{j=0}^{k} \lambda_j \tag{17}$$

which is a K-dimensional analytic function of the generic frequency vector variable $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]^{\top} \in \mathbb{R}^K$ with $\lambda_0 = 1$ by default (i.e., $\mathbf{S}_0 = \mathbf{I}$) [32]. The frequency response $h(\boldsymbol{\lambda})$ is a multivariate function of a K-dimensional vector variable $\boldsymbol{\lambda}$, where the kth entry λ_k is the analytic variable corresponding to the kth shift operator $\mathbf{S}_k$. The shape of $h(\boldsymbol{\lambda})$ is determined by the coefficients $\{h_k\}_{k=0}^K$, while a specific chain of $\mathbf{S}_K, \ldots, \mathbf{S}_1$ only instantiates the eigenvalues $\{\lambda_{Ki}\}_{i=1}^n, \ldots, \{\lambda_{1i}\}_{i=1}^n$ on the K-dimensional variable $\boldsymbol{\lambda}$ – see Fig. 1 for an example.

B. Variance Analysis

Given the filter frequency response over stochastic graphs, we make the following conventional assumptions.

Assumption 1: Let $h(\boldsymbol{\lambda})$ be the filter frequency response [cf. (17)] of the K-dimensional variable $\boldsymbol{\lambda}$ satisfying $|h(\boldsymbol{\lambda})| \leq 1$. The stochastic graph filter is Lipschitz, i.e., there exists a constant C_L such that

$$|h(\boldsymbol{\lambda}_1) - h(\boldsymbol{\lambda}_2)| \leq C_L \|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|, \forall \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \Lambda^K \tag{18}$$

where Λ^K is the considered K-dimensional domain.

Assumption 2: The nonlinearity $\sigma(\cdot)$ satisfies $\sigma(0) = 0$ and it is Lipschitz, i.e., there exists a constant C_σ such that

$$|\sigma(x) - \sigma(y)| \leq C_\sigma |x - y|, \forall x, y \in \mathbb{R}. \tag{19}$$

Assumption 3: The nonlinearity $\sigma(\cdot)$ is variance non-increasing, i.e., for any real random variable x, it holds that $\text{Var}[\sigma(\text{x})] \leq \text{Var}[\text{x}]$.

Assumption 1 implies that the frequency response $h(\boldsymbol{\lambda})$ does not change faster than linear in any frequency direction of $\boldsymbol{\lambda}$, which is standard in the stability analysis of GNNs [14]. It holds for filter coefficients $\{h_k\}_{k=0}^K$ and graph eigenvalues $\boldsymbol{\lambda}$ of finite values because $h(\boldsymbol{\lambda})$ is a finite-order polynomial, such that it is bounded and Lipschitz for some $C_L < \infty$. Given $\{h_k\}_{k=0}^K$, we can express $h(\boldsymbol{\lambda})$ and estimate C_L as the maximal finite difference in the considered domain. Assumptions 2 and 3 hold for popular nonlinearities such as the ReLU and the absolute value [32, Lemma 1]. The following theorem then formalizes the SGNN output variance.

Theorem 1: Consider the SGNN in (2) of L layers, F features, and filter order K over the GRES(p, q) model with M_d dropping edges and M_a adding edges [Def. 1]. Let the stochastic graph filters with the frequency responses (17) satisfy Assumption 1 with C_L and the nonlinearity $\sigma(\cdot)$ satisfy Assumptions 2–3 with C_σ. Then, for any input graph signal $\mathbf{x}$, the variance of the SGNN output is upper bounded as

$$\text{Var}[\boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}_{\text{P:1}}, \mathcal{H})] \leq C_L^2 \left(M_d p(1-p) + M_a q(1-q)\right) C \|\mathbf{x}\|^2 + \mathcal{O}(p^2(1-p)^2) + \mathcal{O}(q^2(1-q)^2) \tag{20}$$

where $C = 4K \sum_{\ell=1}^{L} F^{2L-3} C_\sigma^{2\ell-2}/n$ is a constant.

Proof: See Appendix B in the supplementary material. ■

Theorem 1 states that the SGNN output variance is upper bounded proportionally to the Lipschitz term C_L^2 and quadratically to the edge dropping\adding probability p\q. The result guarantees the deviation of the SGNN output from the optimal expectation is finite and bounded, and the constant C embeds the role of the architecture hyper-parameters, i.e., the number of features F, the number of layers L and the Lipschitz constant of nonlinearity C_σ. Three explicit factors that affect the variance are identified:

1) *Filter property:* The term C_L^2 captures the variation of the filter frequency response $h(\boldsymbol{\lambda})$. The variance decreases with the Lipschitz constant C_L, which is determined by parameters $\mathcal{H}$. A smaller C_L implies the frequency response $h(\boldsymbol{\lambda})$ changes slower in the spectral domain; thus, it is more stable to frequency deviations induced by the graph

stochasticity and leads to a lower variance. However, this flatter response reduces the filter capacity to discriminate between nearby spectral features, i.e., the filter has similar responses for graph frequencies that are close to each other. The latter indicates an implicit trade-off between decreasing the variance and increasing the discrimination power.

2) *Graph stochasticity:* The term $M_d p(1-p) + M_a q(1-q)$ represents the impact of the graph stochasticity. The variance decreases when the number of dropping edges M_d or adding edges M_a is small. The variance decreases also when edges are stable ($p \backslash q \to 0$) or highly unstable ($p \backslash q \to 1$). The latter is because the maximal uncertainty on an edge is for $p = q = 0.5$. Such a graph stochasticity depends typically on external factors (e.g., interference, attacks) or design choices (e.g., graph dropout).
3) *SGNN architecture:* The term $4K \sum_{\ell=1}^{L} F^{2L-3} C_{\sigma}^{2\ell-2}/n$ indicates the effect of the SGNN architecture. The variance increases exponentially with the number of features F and the Lipschitz constant C_σ with exponents controlled by the number of layers L, i.e., the wider/deeper an SGNN, the larger the variance. This is the consequence of the graph stochasticity propagating through the architecture. That is, an architecture with more filters/layers contains more random components, generates more intermediate features, and ultimately results in a larger variance, which indicates a trade-off between the representational capacity and the stochastic deviation of the SGNN. The Lipschitz constant C_σ is typically one implying the non-expansivity of the nonlinearity such as the ReLU or the absolute value.

The aforementioned analysis indicates that we can constrain the variance in three ways: (1) reducing the Lipschitz constant C_L; (2) reducing the number of random edges M_d, M_a or edge probabilities p, q; (3) reducing the architecture width F and depth L. However, (2) and (3) are determined at the outset and cannot be controlled during training. This implies that the variance-constrained learning keeps the variance bounded by tuning parameters $\mathcal{H}$ to lower the Lipschitz constant C_L of the stochastic graph filters [As. 1]. Consequently, the stochastic graph filters exhibit flatter frequency responses and restricting the variance comes at the expense of the discrimination power. From this perspective, the variance bound C_v cannot be set too small; i.e., if C_v is small, C_L decreases yielding a flatter frequency response; hence, a lower discrimination power. This is an implicit trade-off we have to cope with for improving the SGNN robustness to stochastic deviations. We also note that Theorem 1 extends the variance analysis in [32], which is the particular case when all edges are only dropped with a probability p.

Remark 3: The bound in (20) may be loose when M_d, M_a are large and the graph changes dramatically, i.e., $p \backslash q$ are around 0.5, essentially because this bound holds uniformly for all graphs. However, this result still shows that the SGNN output variance is bounded and there is a trade-off between robustness to stochasticity and discrimination power. In turn, this indicates how the variance-constrained learning affects the performance, which mechanisms in the SGNN are mostly responsible, and which are our handle to reduce this bound (potentially the output variance).

V. Duality Gap

We solved problem (6) in the dual domain, where there exists a duality gap $\mathbb{P} - \mathbb{D}$ between the primal and dual solutions. The null duality gap can be achieved for convex problems, while problem (6) is typically non-convex. The latter makes it unclear how close is the dual solution $\mathbb{D}$ of (9) to the primal solution $\mathbb{P}$ of (6). In this section, we argue that the formulated problem could have a small duality gap despite its nonconvexity, which guarantees a small optimality loss caused by the dual transformation. To show such a result, we first consider a more general version of (6), where we generalize the SGNN to an unparameterized function and the discrete set of shift operator sequences to a continuous set. Upon proving this generalized setting has a null duality gap, we then analyze the duality deviation induced by two generalizations and characterize the duality gap of problem (6).

A. Problem Generalization

We consider the SGNN $\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})$ as a parameterized model of a function $f(\mathbf{x}; \mathbf{S}_{P:1})$ that takes as inputs a graph signal $\mathbf{x}$ and a discrete sequence of shift operators $\mathbf{S}_{P:1} \in \mathcal{M}$ and generates representational features as outputs. Problem (6) considers the expected objective and constraints over the discrete set $\mathcal{M}$. The latter can be extended to a continuous set $\widetilde{\mathcal{M}}$ via the following ε-Borel set [55].

Definition 2 (ε-Borel set): For a shift operator $\mathbf{S}_k$, the ε-Borel set of $\mathbf{S}_k$ is

$$\mathcal{B}_\varepsilon(\mathbf{S}_k) := \{\widetilde{\mathbf{S}}_k \in \mathbb{R}^{n\times n} : \|\widetilde{\mathbf{S}}_k - \mathbf{S}_k\| \leq \varepsilon\}, \text{ for } k = 1, \ldots, P \quad (21)$$

where $\|\cdot\|$ is the ℓ_2-norm.

The ε-Borel set $\mathcal{B}_\varepsilon(\mathbf{S}_k)$ is a continuous set of shift operators $\widetilde{\mathbf{S}}_k$ and has countless points [cf. (21)]. For each sequence of the shift operators $\mathbf{S}_{P:1} = \{\mathbf{S}_1, \ldots, \mathbf{S}_P\} \in \mathcal{M}$, we can construct the corresponding sequence of the ε-Borel sets $\mathcal{B}_\varepsilon(\mathbf{S}_{P:1}) = \{\mathcal{B}_\varepsilon(\mathbf{S}_1), \ldots, \mathcal{B}_\varepsilon(\mathbf{S}_P)\}$. The latter is a set of shift operator sequences $\widetilde{\mathbf{S}}_{P:1} = \{\widetilde{\mathbf{S}}_1, \ldots, \widetilde{\mathbf{S}}_P\}$ with each $\widetilde{\mathbf{S}}_k \in \mathcal{B}_\varepsilon(\mathbf{S}_k)$ for $k = 1, \ldots, P$ and contains also countless points. Given two discrete sequences $\mathbf{S}_{P:1}^{(i)}, \mathbf{S}_{P:1}^{(j)} \in \mathcal{M}$, the union of the respective ε-Borel set sequences $\{\mathcal{B}_\varepsilon(\mathbf{S}_1^{(i)}), \ldots, \mathcal{B}_\varepsilon(\mathbf{S}_P^{(i)})\}$ and $\{\mathcal{B}_\varepsilon(\mathbf{S}_1^{(j)}), \ldots, \mathcal{B}_\varepsilon(\mathbf{S}_P^{(j)})\}$ is defined as

$$\bigcup_{\mathbf{S}_{P:1} \in \left\{\mathbf{S}_{P:1}^{(i)}, \mathbf{S}_{P:1}^{(j)}\right\}} \{\mathcal{B}_\varepsilon(\mathbf{S}_1), \ldots, \mathcal{B}_\varepsilon(\mathbf{S}_P)\}$$
$$:= \left\{\mathcal{B}_\varepsilon\left(\mathbf{S}_1^{(i)}\right) \cup \mathcal{B}_\varepsilon\left(\mathbf{S}_1^{(j)}\right), \ldots, \mathcal{B}_\varepsilon\left(\mathbf{S}_P^{(i)}\right) \cup \mathcal{B}_\varepsilon\left(\mathbf{S}_P^{(j)}\right)\right\} \quad (22)$$

which is also a set of shift operator sequences $\widetilde{\mathbf{S}}_{P:1} = \{\widetilde{\mathbf{S}}_1, \ldots, \widetilde{\mathbf{S}}_P\}$ with each $\widetilde{\mathbf{S}}_k \in \mathcal{B}_\varepsilon(\mathbf{S}_k^{(i)}) \cup \mathcal{B}_\varepsilon(\mathbf{S}_k^{(j)})$ for $k =$

$1, \ldots, P$. This union contains all possible shift operator sequences that belong to the constituted ε-Borel set sequences. We then define the ε-Borel generalization $\widetilde{\mathcal{M}}$ as follows.

Definition 3 (ε-Borel generalization): The ε-Borel generalization of the discrete set $\mathcal{M}$ with shift operator sequences $\mathbf{S}_{P:1}$ is defined as the union of the ε-Borel set sequences

$$\widetilde{\mathcal{M}} := \bigcup_{\mathbf{S}_{P:1} \in \mathcal{M}} \{\mathcal{B}_\varepsilon(\mathbf{S}_1), \ldots, \mathcal{B}_\varepsilon(\mathbf{S}_P)\}. \tag{23}$$

where $\bigcup_{\mathbf{S}_{P:1} \in \mathcal{M}}$ stands for the union of all ε-Borel set sequences w.r.t. all sequences $\mathbf{S}_{P:1} \in \mathcal{M}$ [cf. (22)].

The ε-Borel generalization $\widetilde{\mathcal{M}}$ contains countless points $\widetilde{\mathbf{S}}_{P:1} = \{\widetilde{\mathbf{S}}_1, \ldots, \widetilde{\mathbf{S}}_P\}$ and the expectation of any function $\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})$ over $\widetilde{\mathcal{M}}$ is

$$\mathbb{E}[\mathbf{\Phi}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}, \mathcal{H})] = \int_{\widetilde{\mathbf{S}}_{P:1} \in \widetilde{\mathcal{M}}} \widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}) d\mu(\widetilde{\mathbf{S}}_{P:1}) \tag{24}$$

where $\mu(\cdot)$ is the probability measure over $\widetilde{\mathcal{M}}$. Such a probability measure is non-atomic, i.e., for any set $\mathcal{A} \in \widetilde{\mathcal{M}}$ with positive measure $\mu(\mathcal{A}) > 0$, there always exists a subset $\mathcal{A}' \subset \mathcal{A}$ such that $0 < \mu(\mathcal{A}') < \mu(\mathcal{A})$. Given the function $f(\mathbf{x}; \mathbf{S}_{P:1})$ and the ε-Borel generalization $\widetilde{\mathcal{M}}$, problem (6) can be seen as a particular instance of

$$\begin{aligned} \widetilde{\mathbb{P}} := \min_{\widetilde{f}} \; & \mathbb{E}_{\widetilde{\mathcal{M}}} \left[\mathbb{E}_{\mathcal{T}} \left[\mathcal{C}(\mathbf{y}, \widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})) \right] \right] \\ \text{s.t.} \; & \frac{1}{n} \mathbb{E}_{\widetilde{\mathcal{M}}} \left[\sum_{i=1}^{n} [\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})]_i \right] \geq C_f, \\ & \frac{1}{n} \mathbb{E}_{\widetilde{\mathcal{M}}} \left[\sum_{i=1}^{n} [\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})]_i^2 \right] \leq C_s \end{aligned} \tag{25}$$

where $\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})$ is the function defined on $\widetilde{\mathcal{M}}$ and $\widetilde{\mathbf{S}}_{P:1}$ is a sequence of random shift operators in $\widetilde{\mathcal{M}}$. We now establish the strong duality for problem (25).

Proposition 2: Suppose there exists a feasible solution $\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})$ satisfying the constraints in (25) with strict inequality. Then, problem (25) has a null duality gap $\widetilde{\mathbb{P}} = \widetilde{\mathbb{D}}$.

Proof: Define $\mathbf{z}_1 = \mathbb{E}_{\widetilde{\mathcal{M}}}[\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})], \mathbf{z}_2 = \mathbb{E}_{\widetilde{\mathcal{M}}}[\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})^2]$ where $(\cdot)^2$ is the pointwise square operation, and $\mathbf{z} = [\mathbf{z}_1^\top, \mathbf{z}_2^\top]^\top$. Let $g_1(\mathbf{z}) = \sum_{i=1}^n [\mathbf{z}_1]_i$, $g_2(\mathbf{z}) = \sum_{i=1}^n [\mathbf{z}_2]_i$ be functions of $\mathbf{z}$. Substituting these representations into problem (25) yields

$$\begin{aligned} \widetilde{\mathbb{P}} := \min_{\widetilde{f}} \; & \mathbb{E}_{\widetilde{\mathcal{M}}} \left[\mathbb{E}_{\mathcal{T}} \left[\mathcal{C}(\mathbf{y}, \widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})) \right] \right], \\ \text{s.t.} \; & -g_1(\mathbf{z}) + C_f \leq 0, g_2(\mathbf{z}) - C_s \leq 0, \\ & \mathbf{z} = \left[\mathbb{E}_{\widetilde{\mathcal{M}}} \left[\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}) \right]^\top, \mathbb{E}_{\widetilde{\mathcal{M}}} \left[\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})^2 \right]^\top \right]. \end{aligned} \tag{26}$$

Since $-g_1(\mathbf{z})$ and $g_2(\mathbf{z})$ are convex functions of $\mathbf{z}$, problem (26) can be considered as a sparse functional program [43]. By using [43, Theorem 1], we prove the strong duality $\widetilde{\mathbb{P}} = \widetilde{\mathbb{D}}$. Note that $-g_1(\mathbf{z})$ and $g_2(\mathbf{z})$ are also composite functions of $(\mathbf{x}, \widetilde{\mathbf{S}}_{P:1})$, which integrally may not be convex. But from the condition in [43], we need only the outer form convex but not the composite form. ■

That is, problem (25) can be solved in the dual domain without loss of optimality. We leverage this results to characterize the duality gap of problem (6) where the SGNN $\mathbf{\Phi}(\mathbf{x}; \mathbf{S}_{P:1}, \mathcal{H})$ operates over a discrete set $\mathcal{M}$.

Remark 4: Proposition 2 proves the null duality gap for the general version of the surrogate problem (6). If we were to consider the general version of the original problem (4), we would have not proven such strong duality. This is because the variance constraint in problem (4) takes the form

$$g(\mathbf{z}) - C_v \leq 0 \text{ with } g(\mathbf{z}) = \sum_{i=1}^{n} [\mathbf{z}_2]_i - [\mathbf{z}_1]_i^2. \tag{27}$$

Since $g(\mathbf{z})$ is a non-convex function of $\mathbf{z}$, the conditions of [43, Theorem 1] do not apply.

B. Duality Analysis

We now analyze the duality deviation induced by the problem generalization. First, we particularize the function $\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})$ to the SGNN $\mathbf{\Phi}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}, \mathcal{H})$ via the *ϵ-universal parameterization*.

Definition 4 (ϵ-universal parameterization): A parameterization[2] $\mathbf{\Phi}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}, \mathcal{H})$ is ϵ-universal if for any function $\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})$ in the considered domain, there exist a set of parameters $\mathcal{H}$ such that

$$\mathbb{E}_{\widetilde{\mathcal{M}}} \left[\| \widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}) - \mathbf{\Phi}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}, \mathcal{H}) \|^2 \right] \leq \epsilon^2 \tag{28}$$

where the expectation $\mathbb{E}_{\widetilde{\mathcal{M}}}[\cdot]$ is over the generalized set $\widetilde{\mathcal{M}}$ of the shift operator sequence $\widetilde{\mathbf{S}}_{P:1}$.

An ϵ-universal parametrization can model any function in the considered domain within some accuracy ϵ. Such a property holds for a number of machine learning architectures, including radial basis function networks [56], reproducing kernel Hilbert spaces [57], and deep neural networks [58].

Assumption 4: For a given SGNN $\mathbf{\Phi}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}, \mathcal{H})$, there exists a finite accuracy $\epsilon > 0$ such that the SGNN is an ϵ-universal parameterization w.r.t. the generalized set $\widetilde{\mathcal{M}}$.

Assumption 4 implies that for the considered SGNN, there exists some finite $\epsilon > 0$ to make it an ϵ-universal parameterization. The value of ϵ depends on the representational capacity of the considered SGNN, i.e., a deeper (high L) and wider (high F) SGNN may have a higher representational capacity and we may choose a smaller ϵ for it w.r.t. a stronger ϵ-universal parameterization. This property characterizes the deviation induced by particularizing $\widetilde{f}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1})$ to $\mathbf{\Phi}(\mathbf{x}; \widetilde{\mathbf{S}}_{P:1}, \mathcal{H})$ and will be reflected in the duality gap. Second, we particularize the continuous set $\widetilde{\mathcal{M}}$ to the discrete set $\mathcal{M}$. The relation between these two sets is characterized by the ε-Borel set [Def. 2]. To proceed the analysis, we assume the following.

Assumption 5: The loss $\mathcal{C}(\cdot, \cdot)$ is Lipschitz over $\mathcal{T} = \{(\mathbf{x}, \mathbf{y})\}$, i.e., for any $\mathbf{y}_1$ and $\mathbf{y}_2$, there exists a constant C_ℓ such

[2]A parameterization is defined as a mathematical model that represents some mapping as a function of some independent parameters.

that

$$\left|\mathbb{E}_{\mathcal{T}}[\mathcal{C}(\mathbf{y},\mathbf{y}_1)]-\mathbb{E}_{\mathcal{T}}[\mathcal{C}(\mathbf{y},\mathbf{y}_2)]\right|\leq C_\ell\|\mathbf{y}_1-\mathbf{y}_2\|. \tag{29}$$

Assumption 6: The SGNN output $\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})$ is bounded, i.e., there exists a constant C_y s.t. $\|\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})\|\leq C_y$.

Assumption 5 is a continuity statement on the loss $\mathcal{C}(\cdot,\cdot)$, which is common in optimization theory [59] and holds for popular classification and regression losses. Assumption 6 considers the SGNN output bounded by a constant C_y independent of the filter coefficients, which has been proven for the SGNN in Lemma 2 of the supplementary material.

The following theorem quantifies the duality gap of problem (6).

Theorem 2: Consider problem (6) with primal and dual solutions $\mathbb{P}$ and $\mathbb{D}$, respectively. Let the SGNN $\mathbf{\Phi}(\mathbf{x};\mathbf{S}_{P:1},\mathcal{H})$ be of L layers comprising F filters of order K. Let the frequency responses (17) of these filters satisfy Assumption 1 with C_L and the nonlinearity $\sigma(\cdot)$ satisfy Assumptions 2-3 with C_σ. Let also the SGNN satisfy Assumption 4 w.r.t. the ε-Borel generalization $\widetilde{\mathcal{M}}$ with ϵ, its output be bounded according to Assumption 6 with C_y, and the cost function $\mathcal{C}(\cdot,\cdot)$ satisfy Assumption 5 with C_ℓ. Then, the duality gap of problem (6) is bounded by

$$|\mathbb{P}-\mathbb{D}|\leq \left(C_\ell+\frac{\widetilde{\gamma}_1^*}{\sqrt{n}}+\widetilde{\gamma}_2^*\left(\frac{2C_y}{\sqrt{n}}+\frac{\epsilon}{n}\right)\right)\epsilon+C\varepsilon+\mathcal{O}(\varepsilon^2) \tag{30}$$

where $\widetilde{\boldsymbol{\gamma}}^*=[\widetilde{\gamma}_1^*,\widetilde{\gamma}_2^*]^\top$ is the optimal dual variable of problem (25) and C is a constant related to the SGNN architectural properties – see the expression of C in (S.28) of the supplementary material.

Proof: See Appendix C in the supplementary material. ■

The duality gap is induced by two types of errors: the parameterization error ϵ of the SGNN [Def. 4] and the generalization error ε of the set [Def. 3]. The parameterization error is present in the first term of (30), which is small when the SGNN exhibits a strong representational capacity to approximate unparameterized functions. This is an irreducible error that indicates how well the SGNN covers the function space and exists for any GNN solutions. The generalization error is present in the second term of (30), which can be sufficiently small by considering small Borel sets that satisfy Assumption 4. A small duality gap indicates that solving the problem in the dual domain comes with a contained optimality loss, compared to solving it directly in the primal domain, which justifies the primal-dual learning procedure.

Theorem 2 discusses the duality gap induced by solving problem (6) in the dual domain exactly. However, *it is still unclear if the primal-dual learning procedure [Alg. 1] converges to a neighboorhood of the dual solution* $\mathbb{D}$. In the next section, we answer this question affirmative and combine the convergence error with the duality gap to provide a unified performance analysis.

VI. Convergence

The main challenge to prove the convergence of the primal-dual learning, stands in the fact that we approximate the minimization at the primal phase with stochastic gradient descent [cf. (9)] and every Γ primal updates we run a single dual update. To characterize this convergence, we make the following mild assumption.

Assumption 7: Let $\mathcal{H}^*$ be the minima of the Lagrangian $\mathcal{L}(\mathcal{H},\boldsymbol{\gamma})$ [cf. (8)] and $\mathcal{H}^{(\Gamma)}$ the approximate solution obtained by the primal phase with gradient descent [cf. (10)]. There exists a constant $\xi\geq 0$ such that for any dual variable $\boldsymbol{\gamma}\in\mathbb{R}_+$, it holds that

$$|\mathcal{L}(\mathcal{H}^*,\boldsymbol{\gamma})-\mathcal{L}(\mathcal{H}^{(\Gamma)},\boldsymbol{\gamma})|\leq\xi. \tag{31}$$

That is, the gradient descent applied in the primal phase solves the dual function $\mathcal{D}(\boldsymbol{\gamma})=\min_{\mathcal{H}}\mathcal{L}(\mathcal{H},\boldsymbol{\gamma})$ within an error neighborhood ξ. The value of ξ depends on the performance of the gradient descent and on the steps Γ, which has exhibited success in a wide array of optimization problems [60]. The following theorem then establishes the convergence result.

Theorem 3: Consider the primal-dual learning for problem (6) [Alg. 1]. Let the SGNN output satisfy Assumption 6 with C_y and the primal phase satisfy Assumption 7 with ξ. Then, for an accuracy $\delta>0$, Algorithm 1 converges to an error neighborhood of the dual solution $\mathbb{D}$ of problem (6) as

$$|\mathcal{L}(\mathcal{H}_T^{(\Gamma)},\boldsymbol{\gamma}_T)-\mathbb{D}|\leq 2\xi+\frac{\left(\left(C_f+\frac{C_y}{\sqrt{n}}\right)^2+\left(C_s+\frac{C_y^2}{n}\right)^2\right)}{2}\eta_\gamma+\delta \tag{32}$$

in at most T iterations with $T\leq\|\boldsymbol{\gamma}_0-\boldsymbol{\gamma}^*\|^2/(2\eta_\gamma\delta)$, where $\boldsymbol{\gamma}_0$ and $\boldsymbol{\gamma}^*$ are the initial and optimal dual variables for the dual problem [cf. (9)], and η_γ is the dual step-size.

Proof: See Appendix D in the supplementary material. ■

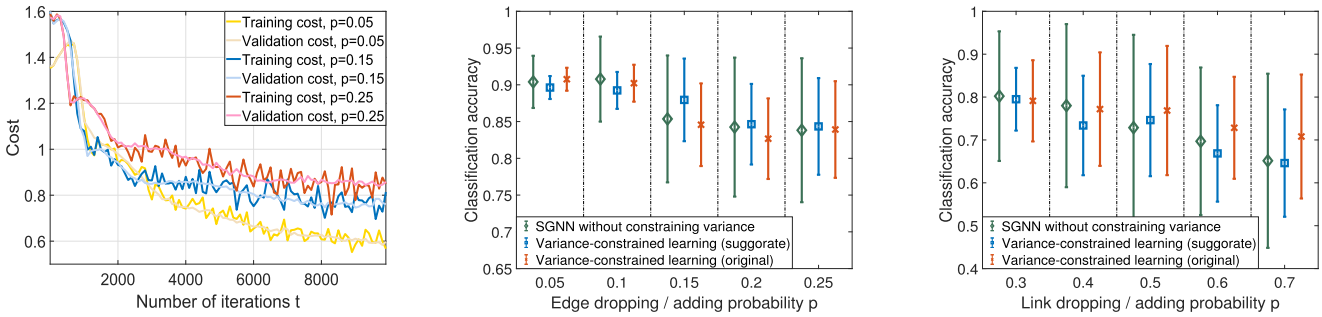
Theorem 3 states that the primal-dual learning converges to an error neighborhood of the dual solution within a finite number of iterations that is inversely proportional to the desirable accuracy δ. The error size depends on the suboptimality of the solution of the primal phase and the step-size of the dual phase. Inspecting (32), the error size consists of three terms:

1) The term 2ξ decreases when we perform sufficient gradient steps at the primal phase and the parameters $\mathcal{H}_t^{(\Gamma)}$ [cf. (10)] are close to the optimal $\mathcal{H}_t^*$ at iteration t.
2) The term is proportional to the dual step-size η_γ, which could be set sufficiently small [cf. (12)].
3) The term δ is inversely proportional to the number of iterations T, which decreases if we run the primal-dual learning for more iterations.

By combining Theorems 2-3, we can characterize completely the solution suboptimality of the primal-dual learning procedure w.r.t. both the duality gap and the iterative method.

Corollary 1: Under the same settings of Theorems 2–3, the suboptimality of solving problem (6) with the primal-dual learning procedure can be bounded as

$$|\mathcal{L}(\mathcal{H}_T^{(\Gamma)},\boldsymbol{\gamma}_T)-\mathbb{P}|\leq C_1\epsilon+2\xi+C_2\eta_\gamma+\sigma+C_3\varepsilon+\mathcal{O}(\varepsilon^2) \tag{33}$$



(a) (b) (c)

Fig. 2. (a) Convergence of the cost with different edge dropping probabilities. (b) Expected classification accuracy and standard deviation with and without (w/o) the variance-constrained learning for source localization. (c) Performance with large edge dropping probabilities p.

where constants C_1, C_2 are specified in (30) and constant C_3 is specified in (32).

This result indicates that the proposed variance-constrained learning converges to a solution $\mathcal{L}(\mathcal{H}_T^{(\Gamma)}, \boldsymbol{\gamma}_T)$ in the dual domain within a finite number of iterations, which is close to the optimal solution $\mathbb{P}$ of problem (6).

Remark 5: The convergence result (32) holds when the primal phase obtains parameters in a neighborhood of the global solution [cf. (31)]. Since working with neural networks is in a non-convex setting, it is likely to obtain parameters close to a local minima. In this context, (32) indicates what can be achieved at best via the primal-dual learning and what is our handle to control it. We corroborate next that the variance-constrained learning converges satisfactorily in numerical simulations.

VII. Numerical Results

We compare the variance-constrained learning with the vanilla GNN and the SGNN using synthetic data from source localization and real data from recommender systems [61]. The vanilla GNN is the standard GNN trained over the deterministic underlying graph [5] and has the same architecture hyper-parameters as the SGNN. In the stochastic setting, the vanilla GNN has shown a lower performance compared with the SGNN [32] and thus, we focus principally on the comparison with the latter and report the performance of the vanilla GNN as a baseline. For all architectures we tested both the stochastic gradient descent and the ADAM optimizer [62] for training, while used the latter because it has shown consistently a better performance. The learning rate is $\mu = 10^{-3}$ and decaying factors are $\beta_1 = 0.9, \beta_2 = 0.999$. The assumptions made in Section IV–VI typically hold for these practical applications, where the graph signals, graph eigenvalues and architecture parameters are of finite values, while the assumption constants depend on specific problem settings that vary among different applications. We consider the latter hold in our experiments for some finite constants. It is also worth mentioning that these are assumed properties for theoretical analysis to shed insights on the proposed algorithm but are not necessary for the algorithm implementation.

A. Source Localization

We consider a diffusion process over a stochastic block model (SBM) graph of 50 nodes divided into 5 communities, with the intra- and inter-community edge probabilities 0.8 and 0.2 respectively. The goal is to find the community originating the diffusion distributively at a node. The initial graph signal is a Kronecker delta $\boldsymbol{\delta}_s \in \mathbb{R}^{50}$ originated at a source node $s \in \{s_1, \ldots, s_5\}$ of a community, where $\{s_1, \ldots, s_5\}$ are the five source nodes of five communities respectively. The signal at time t is $\mathbf{x}_s^{(t)} = \mathbf{S}^t \boldsymbol{\delta}_s + \mathbf{n}$ with $\mathbf{n} \in \mathbb{R}^{50}$ a zero-mean Gaussian noise. We generate 15000 samples by randomly selecting a source node s and a diffused time $t \in [0, 50]$, which are split into 10000, 2500, and 2500 samples for training, validation, and testing, respectively. We consider all edges of the nominal graph may fall with a probability p due to channel fading effects and no edges are added during testing, according to the GRES(p, q) model with $q = 0$. The SGNN has two layers, each with $F = 32$ filters of order $K = 8$ and the ReLU nonlinearity. The mini-batch contains 50 samples and the cost function is the cross entropy. We set $C_f = 0$ to maximize the space of feasible solutions and $C_s = 0.5$ to balance the variance and the information loss, i.e., $C_v = 0.5$ according to (7). The performance is measured by the classification accuracy and the results are averaged over 10 SBM graph realizations, conditioned on which different graph stochasticity scenarios are investigated.

Convergence: First, we corroborate the convergence of the variance-constrained learning. Fig. 2(a) displays the primal-dual learning procedure over 10000 iterations with the edge dropping probability $p = 0.05, 0.15$ and 0.25. The expected cost decreases with the number of iterations, while the decreasing rate reduces gradually; ultimately, approaching a stationary point in all cases. The expected cost of $p = 0.05$ converges slightly later than that of $p = 0.15, 0.25$ because $p = 0.05$ yields a more stable graph with a better performance, so that it takes more iterations to reach a lower cost. The convergent value increases with the edge dropping probability p because of the increased graph randomness. Moreover, the convergence curves fluctuate with iterations because the graph stochasticity and the mini-batch sampling render the SGNN output random. The fluctuation reduces from $p = 0.25$ to $p = 0.05$, which can be explained by the decrease of the graph stochasticity as indicated in Theorem 1.

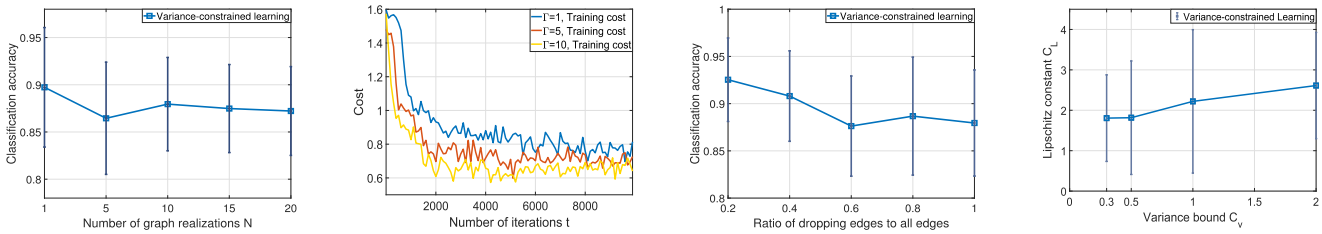


Fig. 3. (a) Expected classification accuracy and standard deviation with different numbers of GRES(p, q) realizations N. (b) Convergence of the cost with different numbers of gradient steps Γ at the primal phase. (c) Expected classification accuracy and standard deviation with different numbers of dropping edges M_d. (d) Expected Lipschitz constant C_L with different variance bounds C_v.

Comparison: We compare the performance of the SGNN w/o the variance-constrained learning w.r.t. both the surrogate problem (6) and the original problem (4). Fig. 2(b) and (c) shows the classification accuracy under different edge dropping probabilities: (mild) $p \in [0.05, 0.25]$ and (harsh) $p \in [0.3, 0.7]$. The proposed variance-constrained learning exhibits a better performance with a comparable expected value and a lower standard deviation. The latter is emphasized when p increases, i.e., when more edges are dropped. The expected performance degrades as p increases, which can be explained by the increased graph variation. The variance-constrained learning maintains a smaller standard deviation, while the unconstrained training increases the standard deviation inevitably. For small probabilities p, the variance-constrained learning w.r.t. the surrogate problem (6) performs comparably to that w.r.t. the original problem (4). We attribute the latter to the fact that the surrogate constraints in (6) provide similar guarantees on stochastic deviations as the variance constraint in (4) [cf. (7)]. For large probabilities p, the surrogate exhibits a lower expected performance but tighter standard deviation. This is because the surrogate is a stronger constraint, i.e., the surrogate is a strict bound of the original [cf. (7)].

Training sensitivity: We evaluate the effects of training parameters on the variance-constrained learning, i.e., the GRES(p, q) realizations N for empirical estimations [cf. (11)] in Fig. 3(a), the gradient steps Γ at the primal phase [cf. (10)] in Fig. 3(b) and the number of dropping edges M_d in Fig. 3(c).

Fig. 3(a) shows that the expected cost fluctuates with N and the fluctuation reduces as N becomes large, while the standard deviation decreases with N. This is because the empirical estimation with a larger N approximates better the variance and the corresponding model becomes more stable, which however takes more training time. The model with $N = 1$ has the largest variance despite better expected performance. We attribute the latter to that the estimated variance with $N = 1$ is more inaccurate, which may lead to a more relaxed constraint; hence, a lower effect during training and a better expected performance. However, the corresponding constraint is not as tight/strict as it is with a large N so the model has the largest variance.

Fig. 3(b) shows that the variance-constrained learning converges faster and to a lower value as Γ increases. This corroborates Theorem 3 since more gradient steps approach better the optimal solution at each primal phase, which reduces the error size ξ and accelerates the convergence. It is remarkable from Fig. 3(a)–(b) that small values of graph realizations, e.g., $N \geq 10$, and gradient steps, e.g., $\Gamma \geq 1$, achieve a satisfactory performance, indicating an efficient implementation.

Fig. 3(c) shows that the expected accuracy decreases and the standard deviation increases with M_d. This follows our finding in Theorem 1 that more unstable edges increase the graph stochasticity and the latter degrades the performance.

Finally, we corroborate the relation between the variance and the discrimination power analyzed in Section IV. Fig. 3(d) shows that the expected Lipschitz constant C_L of stochastic graph filters increases with the variance bound C_v. This corresponds to the theoretical finding in Theorem 1 that constraining the variance may lead to a less discriminative architecture, which contains filters with less variability in their frequency responses.

Hyper-parameter sensitivity: we perform the variance-constrained learning under different hyper-parameters; namely, the number of features F in Fig. 4(a), the number of layers L in Fig. 4(b), the graph size n in Fig. 4(c), and the second-order moment bound C_s in Fig. 4(d), to show their effects on the learning performance. The number of GRES(p, q) realizations is set to $N = 10$.

Fig. 4(a) shows that the expected performance and the variance increase with the number of features F. The former is because of the improved representational capacity, while the latter is because SGNNs with more features generate outputs with more randomness and may require a larger N to estimate the variance for constrained learning. Similar results and discussions apply on different numbers of layers L in Fig. 4(b). In Fig. 4(c), we see that the expected performance decreases with the graph size n because the problem becomes more difficult. The variance increases with n because SGNNs over larger graphs suffer more the randomness. In this case, the variance-constrained learning needs a larger N to have a better estimation of the variance. Fig. 4(d) illustrates that both the expected performance and the variance increase with the value of C_s. This is because a constraint with a larger C_s affects less the cost function but results in a looser constraint, indicating a trade-off between the expected performance and the variance as suggested by Theorem 1.

B. Recommender Systems

We now show an interesting application of the proposed approach in the diversity-enhancing recommender system

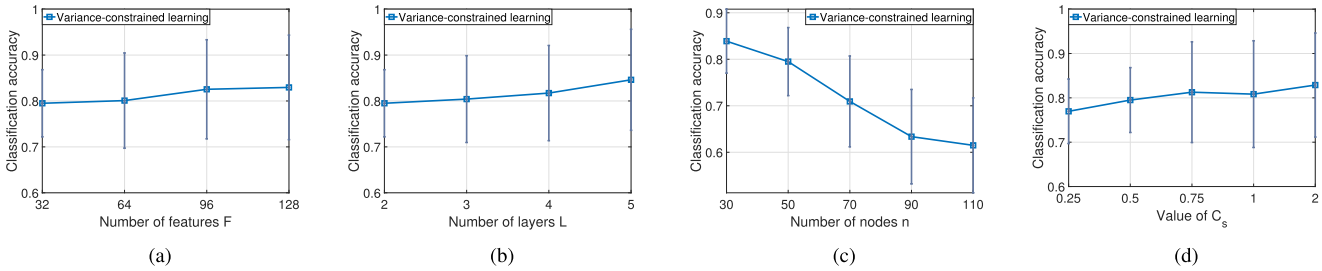


Fig. 4. Expected classification accuracy and standard deviation with different hype-parameters. (a) Different numbers of features F. (b) Different numbers of layers L. (c) Different graph sizes n. (d) Different values of C_s.

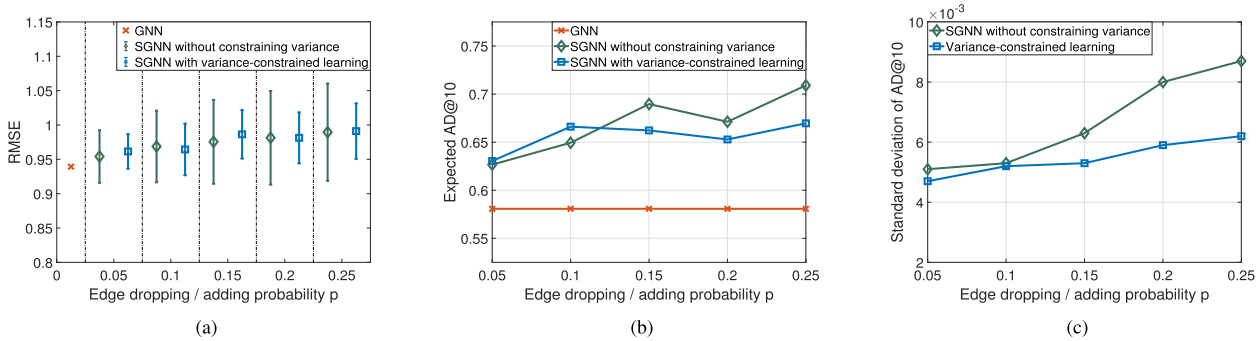


Fig. 5. (a) Expected RMSE and standard deviation of the GNN, the SGNN w/o the variance-constrained learning for movie recommendation. (b)-(c) Expected AD and standard deviation of the GNN, the SGNN w/o the variance-constrained learning for movie recommendation.

(RecSys). We consider the MovieLens 100K dataset, which comprises 943 users and 1682 movies [61]. Following the pre-processing steps in [26], we build the graph by considering nodes as movies and edges as similarities between them. We compute the movie similarity via the Pearson correlation and keep the 35 edges with the highest correlation. The graph signal is the ratings given by a user to the movies, where the signal value is zero if the movie is unrated.

In the RecSys, accuracy measures how well we predict the ratings a user has given to the movies. However, high accuracy is not necessarily linked to a better user satisfaction. Diversity also plays an important role, which measures the capability of the RecSys to include items of different categories in the recommendation list [63]. To measure accuracy we use the root mean squared error (RMSE), which is a standard criterion for the rating-based RecSys. To measure diversity we use the aggregated diversity for the recommendation list containing top ten items (AD@10), which is defined as the number of different items included in the list. A lower RMSE indicates a better accuracy and a higher AD implies a more diversified RecSys, i.e., the system does not overfit accuracy by recommending only niche items. The joint goal is to tweak the accuracy-diversity trade-off, i.e., predict accurate ratings and increase the recommendation diversity.

Parameterization: We consider the SGNN comprising a single layer with $F = 32$ filters of order $K = 4$ and the Leaky ReLU nonlinearity. The graph stochasticity throughout the architecture is leveraged as a training strategy to aid diversity because it will randomly remove some similarity edges between movies and connect different movies with each other [26]. We consider the first 35 edges with the highest correlation may be dropped and the next 20 edges may be added with a probability p, corresponding to the GRES(p, q) model with $p = q$ for simplicity. The constraint bounds are set as $C_f = 0$ and $C_s = 0.5$.

Performance: We compare the accuracy-diversity trade-off of the vanilla GNN, the SGNN with and without the variance-constrained learning. Fig. 5(a) shows the expected RMSE and the standard deviation under different edge dropping/adding probabilities $p \in [0.05, 0.25]$. For a lower $p \to 0$, the graph is stable and the SGNN exhibits comparable accuracies to the GNN; for a higher p, the graph varies more dramatically and the SGNN degrades gradually. The variance-constrained learning accounts for the variance during training, and thus maintains a lower standard deviation around the expected RMSE. Contrarily, the baseline method ignores this factor and has a higher standard deviation that increases with p.

Fig. 5(b)–(c) display the expected AD@10 and the standard deviation around it. The SGNN improves the diversity compared to the GNN, which can be explained by the involved graph stochasticity. While restricting the variance during training, the variance-constrained learning achieves a comparable (slightly lower) AD@10 to the baseline method. This result together with the well-controlled RMSE in Fig. 5(a) indicate that the variance-constrained learning exhibits a better accuracy-diversity trade-off.

VIII. Conclusion

We proposed a variance-constrained learning strategy for stochastic graph neural networks that achieves a trade-off between the expected performance and stochastic deviations.

This strategy adheres to solving a constrained stochastic optimization problem. We developed a primal-dual learning method to solve the problem in the dual domain, which alternates gradient updates between the SGNN parameters and the dual variable. The variance-constrained learning can be interpreted as a self-learning variance regularizer that provides explicit guarantees for stochastic deviations. A statistical analysis on the SGNN output is conducted to identify how the output variance is decreased and indicates the constrained variance comes at the expense of the discrimination power. We further analyzed the duality gap of the variance-constrained optimization problem and the convergence of the primal-dual learning method, which characterize the solution suboptimality and provide theoretical guarantees for the performance. Numerical results corroborate that the variance-constrained learning finds a favorable balance between the optimal performance and the deviation degradation.

REFERENCES

[1] Z. Gao, E. Isufi, and A. Ribeiro, "Variance-constrained learning for stochastic graph neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5245–5249.

[2] 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.

[3] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[4] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[5] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128–138, Nov. 2020.

[6] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2020.

[7] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 974–983.

[8] W. Fan et al., "A graph neural network framework for social recommendations," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2033–2047, May 2022.

[9] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Proc. Conf. Robot Learn.*, 2020, pp. 671–682.

[10] Z. Gao, F. Gama, and A. Ribeiro, "Wide and deep graph neural network with distributed online learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 3862–3877, 2022.

[11] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, 2020.

[12] Z. Gao, M. Eisen, and A. Ribeiro, "Resource allocation via graph neural networks in free space optical fronthaul networks," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.

[13] Z. Gao, Y. Shao, D. Gunduz, and A. Prorok, "Decentralized channel management in WLANs with graph neural networks," 2022, *arXiv:2210.16949*.

[14] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 5680–5695, 2020.

[15] H. Kenlay, D. Thanou, and X. Dong, "Interpretable stability bounds for spectral graph filters," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5388–5397.

[16] H. Kenlay, D. Thano, and X. Dong, "On the stability of graph convolutional neural networks under edge rewiring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 8513–8517.

[17] R. Levie, W. Huang, L. Bucci, M. Bronstein, and G. Kutyniok, "Transferability of spectral graph convolutional neural networks," *J. Mach. Learn. Res.*, vol. 22, no. 272, pp. 1–59, 2021.

[18] A. Parada-Mayorga and A. Ribeiro, "Algebraic neural networks: Stability to deformations," *IEEE Trans. Signal Process.*, vol. 69, pp. 3351–3366, 2021.

[19] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, "Graph filters for signal processing and machine learning on graphs," 2022, *arXiv:2211.08854*.

[20] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Filtering random graph processes over random time-varying graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4406–4421, Aug. 2017.

[21] A. Zou, K. Kumar, and Z. Hou, "Distributed consensus control for multi-agent systems using terminal sliding mode and Chebyshev neural networks," *Int. J. Robust Nonlinear Control*, vol. 23, no. 3, pp. 334–357, 2013.

[22] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp. 736–751, Dec. 2018.

[23] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3315–3326, Jul. 2008.

[24] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized time-varying formation control for multi-robot systems," *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 1029–1043, 2014.

[25] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, R. Yu, and Y. Liu, "Latent space model for road networks to predict time-varying traffic," in *Proc. 22nd ACM Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1525–1534.

[26] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3697–3707.

[27] R. Berg, T. Kipf, and M. Welling, "Graph convolutional matrix completion," in *Proc. 24th ACM Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1–7.

[28] E. Isufi, M. Pocchiari, and A. Hanjalic, "Accuracy-diversity trade-off in recommender systems via graph convolutions," *Inf. Process. Manage.*, vol. 58, no. 2, 2021, Art. no. 102459.

[29] L. Ben Saad, B. Beferull-Lozano, and E. Isufi, "Quantization analysis and robust design for distributed graph filters," *IEEE Trans. Signal Process.*, vol. 70, pp. 643–658, 2022.

[30] H. S. Nguyen, Y. He, and H. T. Wai, "On the stability of low pass graph filter with a large number of edge rewires," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 5568–5572.

[31] Z. Gao, E. Isufi, and A. Ribeiro, "Stability of graph convolutional neural networks to stochastic perturbations," *Signal Process.*, vol. 188, 2021, Art. no. 108216.

[32] Z. Gao, E. Isufi, and A. Ribeiro, "Stochastic graph neural networks," *IEEE Trans. Signal Process.*, vol. 69, pp. 4428–4443, 2021.

[33] W. Feng et al., "Graph random neural networks for semi-supervised learning on graphs," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 22092–22103.

[34] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–17.

[35] Z. Gao, S. Bhattacharya, L. Zhang, R. S. Blum, A. Ribeiro, and B. M. Sadler, "Training robust graph neural networks with topology adaptive edge dropping," 2021, *arXiv:2106.02892*.

[36] J. Chen, J. Zhu, and L. Song, "Stochastic training of graph convolutional networks with variance reduction," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 942–950.

[37] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–15.

[38] D. Zou, Z. Hu, Y. Wang, S. Jiang, Y. Sun, and Q. Gu, "Layer-dependent importance sampling for training deep and large graph convolutional networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11249–11259.

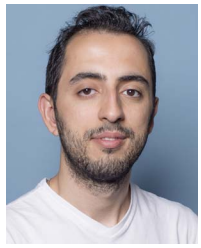
[39] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4563–4572.

[40] Z. Liu et al., "Bandit samplers for training graph neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 6878–6888.

[41] W. Cong, R. Forsati, M. Kandemir, and M. Mahdavi, "Minimal variance sampling with provable guarantees for fast training of graph neural networks," in *Proc. 26th ACM Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1393–1403.

[42] M. Fey, J. E. Lenssen, F. Weichert, and J. Leskovec, "GNNAutoScale: Scalable and expressive graph neural networks via historical embeddings," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 3294–3304.
[43] L. F. O. Chamon, Y. C. Eldar, and A. Ribeiro, "Functional nonlinear sparse models," *IEEE Trans. Signal Process.*, vol. 68, pp. 2449–2463, 2020.
[44] 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, May 2013.
[45] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.
[46] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Discrete signal processing on graphs: Frequency analysis," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, Apr. 2018.
[47] M. Coutino, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2320–2333, May 2019.
[48] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. 24th ACM Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2847–2856.
[49] K. Guo, J. Chen, and Y. Huang, "Outage analysis of cooperative communication network with hardware impairments," *Frequenz*, vol. 69, no. 9/10, pp. 443–449, 2015.
[50] S. Perugini, M. Goncalves, and E. Fox, "Recommender systems research: A connection-centric survey," *J. Intell. Inf. Syst.*, vol. 23, no. 2, pp. 107–143, 2004.
[51] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.
[52] J. Nocedal and S. Wright, *Numerical Optimization*. Berlin, Germany: Springer, 2006.
[53] R. L. Harrison, "Introduction to monte carlo simulation," in *Proc. AIP Conf.*, vol. 1204, no. 1, pp. 17–21, 2010.
[54] Z. Gao, A. Koppel, and A. Ribeiro, "Balancing rates and variance via adaptive batch-size for stochastic optimization problems," *IEEE Trans. Signal Process.*, vol. 70, pp. 3693–3708, 2022.
[55] S. Srivastava, *A Course on Borel Sets*, vol. 180. Berlin, Germany: Springer, 2008.
[56] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, 1991.
[57] B. Sriperumbudur, K. Fukumizu, and G. Lanckriet, "On the relation between universality, characteristic kernels and rkhs embedding of measures," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 773–780.
[58] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.
[59] S. Boyd, S. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[60] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, 2nd ed., Berlin, Germany: Springer, 2012, pp. 421–436.
[61] M. Harper and J. Konstan, "The movielens datasets: History and context," *ACM Trans. Interactive Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2015.
[62] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2010, pp. 1–15.
[63] C. Aggarwal et al., *Recommender Systems*, vol. 1. Berlin, Germany: Springer, 2016.

Zhan Gao (Graduate Student Member, IEEE) received the M.Sc. degree in systems engineering from the University of Pennsylvania, Pennsylvania, PA, USA. He is currently with the Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K. He has been a Research Intern with Intel Corporation, Beijing, China and a Visiting Researcher with the Department of Intelligence Science and Technology, Kyoto University, Kyoto, Japan. His research interests include graph signal processing, graph neural networks, and their applications on wireless networked systems.



Elvin Isufi (Member, IEEE) was born in Vlore, Albania, in 1989. He received the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in 2019, and the M.Sc. degree (*cum laude*) from the University of Perugia, Perugia, Italy, in 2014. He is currently an Assistant Professor with the Multimedia Computing Group, Delft University of Technology. Prior to that, he was a Postdoctoral Researcher with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. His research interests include intersection of signal processing, mathematical modeling, machine learning, and network theory.