MULTIMODAL GRAPH SIGNAL DENOISING VIA TWOFOLD GRAPH SMOOTHNESS REGULARIZATION WITH DEEP ALGORITHM UNROLLING

Masatoshi Nagahama[1] *and Yuichi Tanaka*[1,2]

[1]Tokyo University of Agriculture and Technology, Tokyo, Japan
[2]PRESTO, Japan Science and Technology Agency, Saitama, Japan

ABSTRACT

We propose a denoising method of multimodal graph signals with twofold smoothness regularization. Graph signal processing assumes that a signal has an underlying structure that is represented by a graph. In each node of the graph, we often have multimodal data or features that are correlated across modalities. Since these multimodal data are measured by various sensors, the observed data will be noisy. In this paper, we assume that a multimodal signal is smooth on *two underlying graphs*: One is a spatial graph (i.e., relationship among nodes) and the other is a modality graph (i.e., relationship among modalities). We formulate a regularized minimization problem based on smoothness on the twofold graphs. The problem is solved with an alternating minimization scheme. To avoid a hand-crafted parameter tuning that is usually costly and converges to local minima, we utilize *deep algorithm unrolling* (DAU) to train the parameters in the algorithm. To validate the proposed method, we conduct experiments on synthetic data and demonstrate that our method outperforms various existing graph signal denoising methods.

***Index Terms*—** Graph signal processing, deep algorithm unrolling, signal denoising.

1. INTRODUCTION

Signal restoration refers to the process of restoring an original signal from observation(s) deteriorated by noise, missing, and blur. Many restoration problems for regular signals, e.g., audio, image, and video signals, have been studied for decades [1, 2, 3]. Signal restoration is also an essential task for various machine learning problems as data cleansing.

Many signals have their own interconnectivity among samples. It can often be exploited for signal processing like signals on sensor, transportation, or brain networks, power grids, 3D meshes, and point clouds. These connectivities can be represented as graphs. Recently, graph signal processing (GSP) has attracted many signal processing and machine learning researchers [4, 5, 6]. Examples of GSP include sampling and restoration [7, 8, 9], denoising [10], and compression [11], to name a few.

In this paper, we consider graph signal denoising. It aims to seek the best possible signal from the observation containing noise. For example in sensor networks, some sensors may not work properly or have different specifications that result in noisy samples [12]. Many methods for graph signal restoration, including denoising, have been proposed so far based on regularized optimization [13], graph filters and filter banks [14, 15], deep learning on graphs [16], and algorithm unrolling [17, 10].

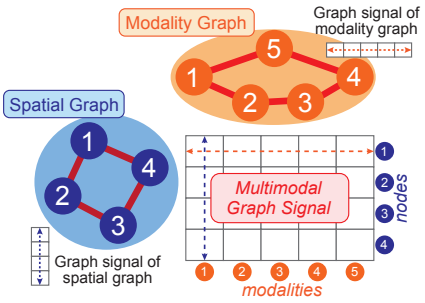


Fig. 1: An example of twofold graph assumption where a multimodal graph signal has its own spatial and modality graphs ($N = 4$ and $K = 5$).

A sensor can often capture multimodal features, and the feature values on a node may be correlated with each other. This can be easily understood by considering environmental sensors (temperature and air pressure) and 3D point clouds (coordinates). In other words, we can assume *nodes* have an underlying spatial structure (spatial graph), and *features* also have an underlying inter-feature structure (modality graph). We call this *twofold graph assumption* (TGA) in this paper. This assumption is illustrated in Fig. 1. The graph signal restoration performance may be improved if we consider TGA.

TGA has been utilized for a few GSP and machine learning tasks. For example, robust principal component analysis on graphs [18] considers two types of graphs based on data similarity and feature similarity. In matrix completion (i.e., interpolation), dual graph Laplacian regularization (DGLR) has been studied [19], similar to TGA. DGLR considers the data matrix has two underlying graphs, e.g., user graph and movie graph for recommendation systems. The matrix completion with DGLR is designed based on a convex optimization and its formulation composes of a data fidelity term and two graph smoothness terms. The smoothness terms regularize the Laplacian quadratic form, i.e., the second-order differences with neighboring signals.

These model-based approaches based on TGA need to select optimal parameter(s) for the regularization terms. The parameter selection directly (and strongly) affects the performance and the speed of convergence. However, many works only select the parameters in an ad-hoc manner and they are fixed throughout performing an iterative optimization.

In this paper, we propose a multimodal graph signal denoising method with twofold graph smoothness regularization (TGSR).

This work is supported in part by JST PRESTO (JPMJPR1935) and JSPS KAKENHI (20H02145).

Based on TGA, we formulate a denoising problem including the first-order smoothness terms of the twofold graphs. Its solver is implemented by an iterative algorithm. In contrast to many model-based methods, our approach automatically tunes the parameters from training data by employing deep algorithm unrolling (DAU) [20]. Thanks to DAU, the parameters in each iteration of the proposed method are trained to minimize the loss function via techniques from deep learning, e.g., back-propagation. Given the cascade of P-layer networks which corresponds to P iterations of the algorithm, the proposed method can suppress noise from the noisy input.

We conduct experiments on synthetic multimodal graph signals to verify the denoising performance of the proposed method. Our proposed method outperforms various existing methods including model-based and graph neural network-based approaches in terms of root mean squared error (RMSE).

Notation: Throughout this paper, vectors and matrices are written in bold and sets are written as calligraphic letters. Furthermore, $\|\cdot\|_1$ and $\|\cdot\|_F$ denote the ℓ_1 vector norm and the Frobenius norm, respectively.

We denote an undirected graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$. The numbers of vertices and edges are represented as $|\mathcal{V}| = N$ and $|\mathcal{E}|$, respectively. The edge weight between v_i and v_j is denoted as $w_{i,j} \in \mathbb{R}_{\geq 0}$. A weighted adjacency matrix of $\mathcal{G}$ is defined as an $N \times N$ matrix with $[\mathbf{W}]_{ij} = w_{i,j}$; $[\mathbf{W}]_{ij} = 0$ represents unconnected vertices. In this paper, we consider a graph that does not have self-loops, i.e., $[\mathbf{W}]_{ii} = 0$ for all i. The degree matrix of $\mathcal{G}$ is defined as a diagonal matrix $[\mathbf{D}]_{ii} = \sum_j w_{i,j}$. The symmetric normalized graph Laplacian matrix of $\mathcal{G}$ is given by $\boldsymbol{\mathcal{L}} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}$. Since $\boldsymbol{\mathcal{L}}$ is a real symmetric matrix, $\boldsymbol{\mathcal{L}}$ always has an eigendecomposition. Let the eigendecomposition of $\boldsymbol{\mathcal{L}}$ be $\boldsymbol{\mathcal{L}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$, where $\mathbf{U}$ is an eigenvector matrix and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$. The weighted graph incidence matrix is denoted as $\mathbf{M}$. For notation simplicity, we assign an integer to an edge as $\mathcal{E} = \{e_s\}_{s=1}^{|\mathcal{E}|}$. Then, the sth row and tth column of $\mathbf{M}$ corresponding to e_s and v_t as follows:

$$[\mathbf{M}]_{s,t} = \begin{cases} \sqrt{w_{i,j}} & e_s = (v_i, v_j) \text{ and } t = i, \\ -\sqrt{w_{i,j}} & e_s = (v_i, v_j) \text{ and } t = j, \\ 0 & \text{otherwise.} \end{cases}$$

The incidence matrix satisfies $\boldsymbol{\mathcal{L}} = \mathbf{M}^\top\mathbf{M}$.

A graph signal $\mathbf{x} \in \mathbb{R}^N$ is defined as a mapping from the vertex set to the set of real numbers, i.e., $x_n : \mathcal{V} \to \mathbb{R}$. Further, a multimodal graph signal having K modalities is represented by a matrix $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K)}] \in \mathbb{R}^{N \times K}$. Sometimes, $\mathbf{X}$ is also denoted as concatenated row vectors $\bar{\mathbf{x}}^{(k)} \in \mathbb{R}^{1 \times K}$, i.e., $\mathbf{X} = [(\bar{\mathbf{x}}^{(1)})^\top, \ldots, (\bar{\mathbf{x}}^{(N)})^\top]^\top$.

2. TWOFOLD GRAPH SMOOTHNESS REGULARIZATION AND DEEP ALGORITHM UNROLLING

In this section, we describe the details of the proposed method. The overview of the denoising algorithm is shown in Fig. 2.

2.1. Problem Setting

Suppose that an observed noisy multimodal graph signal $\mathbf{Y} \in \mathbb{R}^{N \times K}$ is modeled as

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{N \times K}$ and $\mathbf{N} \in \mathbb{R}^{N \times K}$ are the unknown ground truth multimodal graph signal and the matrix containing i.i.d. additive white Gaussian noise (AWGN), respectively. The main purpose of this paper is to restore $\mathbf{X}$ from $\mathbf{Y}$.

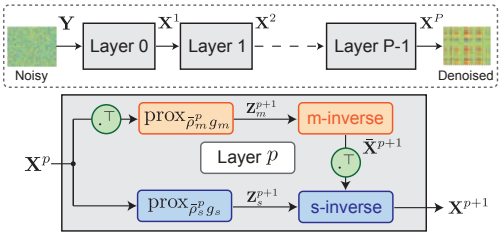


Fig. 2: Overview of the proposed architecture. The proposed architecture consists of P layers and is trainable using deep algorithm unrolling.

In this paper, we assume $\mathbf{x}^{(k)}$ in $\mathbf{X}$ is smooth on the spatial graph, and $\bar{\mathbf{x}}^{(n)}$ in it is also smooth on the modality graph. Hereafter, the spatial and modality graphs are denoted as $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s, \mathbf{W}_s)$ and $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m, \mathbf{W}_m)$, respectively. Later in this section, we also introduce the regularization terms using spatial and modality graph incidence matrices. They are represented as $\mathbf{M}_s$ and $\mathbf{M}_m$, respectively.

2.2. Twofold Graph Smoothness Regularization

A smooth signal on a graph $\mathcal{G}_q$ ($q \in \{s, m\}$) is represented as follows.

$$g_q(\mathbf{Z}_q) \leq \epsilon, \tag{2}$$

where g_q is some convex function that promotes sparsity, $\mathbf{Z}_s = \mathbf{M}_s\mathbf{X}$, and $\mathbf{Z}_m = \mathbf{M}_m\mathbf{X}^\top$. The incidence matrices applied to the signal act as first-order high-pass filters, and thus, the small ϵ indicates the signal is smooth on $\mathcal{G}_q$.

For denoising by promoting smoothness based on (2), we formulate the following minimization problem:

$$\begin{aligned} \min_{\mathbf{X},\mathbf{Z}_s,\mathbf{Z}_m} \ & \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\|_F^2 + \rho_s g_s(\mathbf{Z}_s) + \rho_m g_m(\mathbf{Z}_m), \\ & \text{subject to } \mathbf{Z}_s = \mathbf{M}_s\mathbf{X},\ \mathbf{Z}_m = \mathbf{M}_m\mathbf{X}^\top, \end{aligned} \tag{3}$$

where the first term is a data fidelity term and the second and third ones are for TGSR. Additionally, ρ_s and ρ_m are nonnegative parameters which control the strengths of the regularization. These parameters are later trained by DAU. Note that TGSR utilizes the first-order differences while DGLR considers the second-order ones [19].

In this paper, we consider to solve (3) with an iterative algorithm. We adopt the half-quadratic splitting (HQS) algorithm [21, 22] as its baseline, with the idea of introducing additional variables to solve different convex optimization problems and alternatingly updating each variable.

First, we introduce the following cost function:

$$\begin{aligned} \mathcal{L}_{\mu_s,\mu_m}(\mathbf{X}, \mathbf{Z}_s, \mathbf{Z}_m) = & \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\|_F^2 \\ & + \rho_s g_s(\mathbf{Z}_s) + \frac{\mu_s}{2}\|\mathbf{Z}_s - \mathbf{M}_s\mathbf{X}\|_F^2 \\ & + \rho_m g_m(\mathbf{Z}_m) + \frac{\mu_m}{2}\|\mathbf{Z}_m - \mathbf{M}_m\mathbf{X}^\top\|_F^2, \end{aligned} \tag{4}$$

where μ_s and μ_m are additional penalty parameters that are also made trainable later. Based on the HQS algorithm, the iterative steps

to solve (4) are derived from the partial derivatives of its terms with respect to $\mathbf{X}$, $\mathbf{Z}_s$, and $\mathbf{Z}_m$. The algorithm is represented as follows:

$$\mathbf{X}^{p+1} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y}-\mathbf{X}\|_F^2 + \frac{\mu_s}{2}\|\mathbf{Z}_s - \mathbf{M}_s\mathbf{X}\|_F^2 + \frac{\mu_m}{2}\|\mathbf{Z}_m - \mathbf{M}_m\mathbf{X}^\top\|_F^2, \tag{5a}$$

$$\mathbf{Z}_s^{p+1} = \arg\min_{\mathbf{Z}_s} g_s(\mathbf{Z}_s) + \frac{\mu_s}{2\rho_s}\|\mathbf{Z}_s - \mathbf{M}_s\mathbf{X}\|_F^2, \tag{5b}$$

$$\mathbf{Z}_m^{p+1} = \arg\min_{\mathbf{Z}_m} g_m(\mathbf{Z}_m) + \frac{\mu_m}{2\rho_m}\|\mathbf{Z}_m - \mathbf{M}_m\mathbf{X}^\top\|_F^2. \tag{5c}$$

All the subproblems above are convex.

Since the problem of interest (5a) contains $\mathbf{X}^\top$ as well as $\mathbf{X}$, it is difficult to obtain its closed-form solution directly. Hence, we further split (5a). With $\bar{\mathbf{X}} := \mathbf{X}^\top$, (5a) is reformulated as follows.

$$\bar{\mathbf{X}}^{p+1} = \arg\min_{\bar{\mathbf{X}}} \frac{1}{2}\|\mathbf{Y}^\top - \bar{\mathbf{X}}\|_F^2 + \frac{\mu_m}{2}\|\mathbf{Z}_m - \mathbf{M}_m\bar{\mathbf{X}}\|_F^2, \tag{6a}$$

$$\mathbf{X}^{p+1} = \arg\min_{\mathbf{X}} \frac{1}{2}\|(\bar{\mathbf{X}}^{p+1})^\top - \mathbf{X}\|_F^2 + \frac{\mu_s}{2}\|\mathbf{Z}_s - \mathbf{M}_s\mathbf{X}\|_F^2. \tag{6b}$$

Since generic alternating minimization algorithms can reorder the steps because of its modular property, we first update $\mathbf{Z}_s$ and $\mathbf{Z}_m$, then $\mathbf{X}$ is updated.

As a result, the iterative algorithm works as follows. First, the initial variable is set to $\mathbf{X}^0 = \mathbf{Y}$. Then, the algorithm proceeds in the following order:

$$\mathbf{Z}_m^{p+1} = \mathrm{prox}_{\bar{\rho}_m g_m}(\mathbf{M}_m(\mathbf{X}^p)^\top), \tag{7a}$$

$$\mathbf{Z}_s^{p+1} = \mathrm{prox}_{\bar{\rho}_s g_s}(\mathbf{M}_s\mathbf{X}^p), \tag{7b}$$

$$\bar{\mathbf{X}}^{p+1} = (\mathbf{I}+\mu_m\boldsymbol{\mathcal{L}}_m)^{-1}(\mathbf{Y}^\top + \mu_m\mathbf{M}_m^\top\mathbf{Z}_m^{p+1}), \tag{7c}$$

$$\mathbf{X}^{p+1} = (\mathbf{I}+\mu_s\boldsymbol{\mathcal{L}}_s)^{-1}((\bar{\mathbf{X}}^{p+1})^\top + \mu_s\mathbf{M}_s^\top\mathbf{Z}_s^{p+1}), \tag{7d}$$

where $\bar{\rho}_s = \rho_s/\mu_s$, $\bar{\rho}_m = \rho_m/\mu_m$, and prox$(\cdot)$ is a proximal operator.[1] The inversions in (7c) and (7d) are referred to as *m-inverse* and *s-inverse*, respectively.

In this paper, we use $g_q(\cdot) = \|\cdot\|_1$, which is widely used for promoting sparsity. The regularization term is also represented as follows:

$$\|\mathbf{M}\mathbf{x}\|_1 = \textstyle\sum_{j\in\mathcal{N}_i}\sqrt{w_{ij}}\,|x_i - x_j|, \tag{8}$$

where $\mathcal{N}_i$ is a set of vertices connecting with v_i. This regularization is also called as the graph total variation (GTV) [23, 10]. The proximal operators of ℓ_1 norm in (7a) and (7b) are known as the soft-thresholding function [24].

2.3. Deep Algorithm Unrolling

In our iterative algorithm (7a)–(7d), we need to select four hyperparameters $\{\bar{\rho}_q, \mu_q\}$ in each iteration appropriately. These parameters are usually fixed in all iterations. They are often selected in an ad-hoc manner, however, it is costly and often leads to that the algorithm converges to local minima.

To avoid this parameter selection problem, we install trainable parameters in each iteration and they are trained from available data using DAU [20]. Instead of fixed parameters in every iteration, we make all the parameters $\bar{\rho}_q^p, \mu_{q,1}^p$ and $\mu_{q,2}^p$ trainable. In other words, the proposed method can contain different parameters for all iterations and they are trained from the available training data. This approach results in faster convergence and lower reconstruction error (demonstrated in the next section).

[1] Proximal operator is defined as $\mathrm{prox}_{\gamma f}(\mathbf{x}) = \arg\min_{\mathbf{u}\in\mathbb{R}^N}\{f(\mathbf{u}) + \frac{1}{2\gamma}\|\mathbf{u}-\mathbf{x}\|^2\}$.

2.4. Acceleration and Spectral Interpretation

Since (7c) and (7d) contain matrix inversion and it will take $\mathcal{O}(N^3)$ complexity, the algorithm becomes computationally expensive. To avoid this, the *s-* and *m-inverses* can be accelerated using the polynomial approximation. The *s-* and *m-inverses* can be rewritten as $(\mathbf{I}+\mu_q\boldsymbol{\mathcal{L}}_q)^{-1} = \mathbf{U}f(\boldsymbol{\Lambda}_q;\mu_q)\mathbf{U}^\top$, where

$$f(\boldsymbol{\Lambda}_q;\mu_q) = \mathrm{diag}((1+\mu_q\lambda_{q,1})^{-1},\ldots,(1+\mu_q\lambda_{q,N})^{-1}). \tag{9}$$

The inversion of a diagonal matrix takes $\mathcal{O}(N)$. From this aspect, we can accelerate the algorithm to precompute the eigendecomposition of graph Laplacians before running the algorithm and repeatedly use the eigenvector matrix $\mathbf{U}$.

Specifically, the function (9) is regarded as a graph low-pass filter based on the frequency response $f(\lambda;\mu) = 1/(1+\mu\lambda)$, i.e., high frequency coefficients are suppressed to zero. Therefore, (7c) and (7d) can be seen as iterative graph low-pass filters along modality and spatial graphs.

The proposed method is referred to as *twofold graph smoothness regularization with deep algorithm unrolling* (TGSR-DAU). Its detailed algorithm is shown in **Algorithm 1**.

Algorithm 1 TGSR-DAU: Graph signal denoising via twofold graph smoothness regularization with deep algorithm unrolling

Input: $\mathbf{Y}$, $\mathbf{X}^0 = \mathbf{Y}$, P, and $\{\bar{\rho}_q^p, \mu_{q,1}^p, \mu_{q,2}^p\}_{p=0}^{P-1}(q\in\{s,m\})$.

for $p = 0,\ldots,P-1$ **do**
- $\mathbf{Z}_m^{p+1} \leftarrow \mathrm{prox}_{\bar{\rho}_m^p g_m}(\mathbf{M}_m(\mathbf{X}^p)^\top)$
- $\mathbf{Z}_s^{p+1} \leftarrow \mathrm{prox}_{\bar{\rho}_s^p g_s}(\mathbf{M}_s\mathbf{X}^p)$
- $\bar{\mathbf{X}}^{p+1} \leftarrow \mathbf{U}_m f(\boldsymbol{\Lambda}_m;\mu_{m,1}^p)\mathbf{U}_m^\top(\mathbf{Y}^\top + \mu_{m,2}^p\mathbf{M}_m^\top\mathbf{Z}_m^{p+1})$
- $\mathbf{X}^{p+1} \leftarrow \mathbf{U}_s f(\boldsymbol{\Lambda}_s;\mu_{s,1}^p)\mathbf{U}_s^\top((\bar{\mathbf{X}}^{p+1})^\top + \mu_{s,2}^p\mathbf{M}_s^\top\mathbf{Z}_s^{p+1})$

end for

Output: $\mathbf{X}^P$.

3. EXPERIMENTS

In this section, we perform denoising experiments of multimodal graph signals and compare the proposed method with existing methods in terms of RMSE.

3.1. Setup

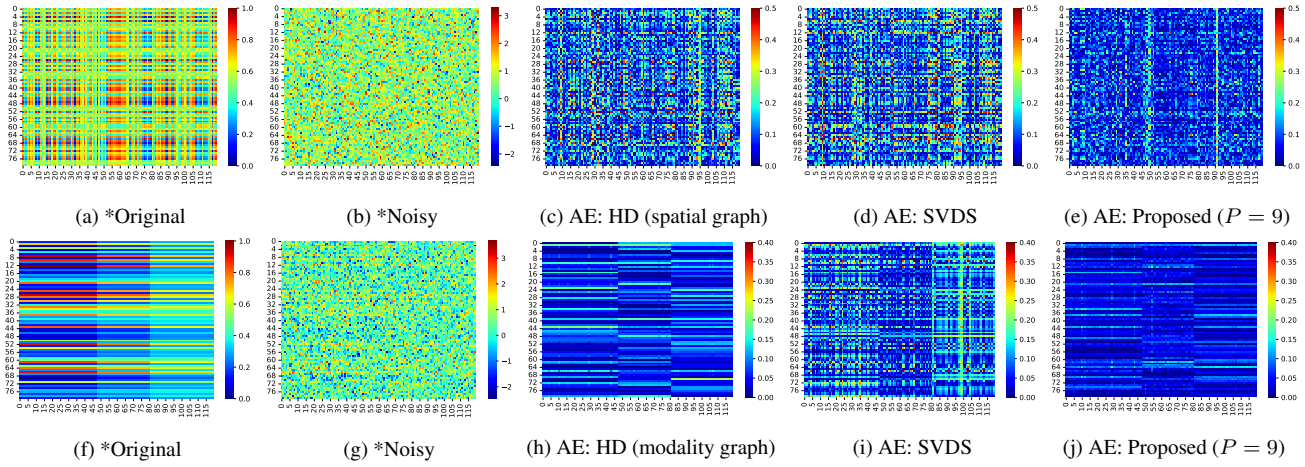
For the experiments, we create synthetic multimodal graph signals $\mathbf{X}$ assumed to be smooth on both spatial and modality graphs. In this paper, we used the following twofold graphs:

1. Spatial sensor graph & modality sensor graph;
2. Spatial sensor graph & modality community graph.

The numbers of vertices are set to $N = 80$ and $K = 120$. These graphs mimic sensor networks with environmental sensors.

A smooth graph signal is generated by a linear combination of the eigenvectors of graph Laplacian.

For the twofold graph 1, we generate a smooth graph signal matrix as $\mathbf{X} = [\mathbf{U}_{s,8}\mathbf{d}_s][\mathbf{U}_{m,8}\mathbf{d}_m]^\top$, where $\mathbf{U}_{s,8} \in \mathbb{R}^{N\times 8}$ and $\mathbf{U}_{m,8} \in \mathbb{R}^{K\times 8}$ are the first eight eigenvectors in $\mathbf{U}_s$ and $\mathbf{U}_m$, respectively. Additionally, $\mathbf{d}_s, \mathbf{d}_m, \in \mathbb{R}^8$ are expansion coefficients



(a) *Original (b) *Noisy (c) AE: HD (spatial graph) (d) AE: SVDS (e) AE: Proposed ($P = 9$)

(f) *Original (g) *Noisy (h) AE: HD (modality graph) (i) AE: SVDS (j) AE: Proposed ($P = 9$)

Fig. 3: Denoising results ($\sigma = .75$). Top: Twofold graph 1. Bottom: Twofold graph 2. Original scale is indicated with *. The others show the absolute errors (AEs) from the original signal.

whose element is randomly generated by uniform distribution between $[0, 1]$.

For the twofold graph 2, we generate piecewise constant signals composed of three communities for the feature-wise direction. That is, a graph signal matrix is generated as $\mathbf{X} = [\mathbf{U}_{s,8}\mathbf{d}_s]\mathbf{c}^\top$, where $\mathbf{c}$ is a community assignment vector and $\mathbf{c} = [\mathbf{c}_1^\top \ \mathbf{c}_2^\top \ \mathbf{c}_3^\top]^\top \in \mathbb{R}^K$ where $\mathbf{c}_i \in \mathbb{R}^{K_i}$ with $i = \{1, 2, 3\}$. Let $\mathbf{1}_{K_i}$ be the vector having K_i entries equal to 1. The vector of the ith community is represented as $\mathbf{c}_i = \beta_i \mathbf{1}_{K_i}$ with a random integer β_i selected from $[1, 5]$.

We first generate 300 samples of multimodal signals with each of the above twofold graphs. The min-max normalization is applied to each of them to be mapped into $[0, 1]$. Finally, the synthetic noisy signal matrices are generated by adding AWGN ($\sigma = \{.25, .50, .75\}$) matrices to $\mathbf{X}$. The generated datasets are separated into the following three subsets: training (60%), validation (20%), and test (20%).

The denoising performance is compared with the existing single graph-based methods listed as follows:

- Diffusion with heat kernel (HD) [25];
- Graph low-pass filter $h(\lambda) = (1 + \tau\lambda)^{-1}$ (GLF);
- Singular value decomposition with hard shrinkage (SVDS) [26];
- Graph convolution networks (GCN) [27]; and
- Graph unrolling trend filter (GUTF) [17].

These methods are applied to a noisy matrix with the spatial or modality graph. The hyperparameters of HD, GLF, and SVGS are chosen to minimize the average RMSE of the validation set, and we report the RMSE of the test set. For GCN, GUTF, and the proposed method, their parameters are trained with the training set.

The configuration of the proposed method is determined by our preliminary experiments. The layer size is set to $P = \{1, 5, 9\}$ to show the influence of the number of layers. The proposed method is trained for 30 epochs with Adam optimizer whose learning rate is set to 0.01 [28].

3.2. Results

Experimental results are summarized in Table 1. The proposed method outperforms the existing methods in terms of the average RMSE for test data. Specifically, the proposed method with $P = 9$ is superior to the existing methods for all cases. The layer size clearly contributes to the RMSE improvement. As observed, deep-learning-based methods (i.e., GCN and GUTF) do not work properly in this experimental setting.

Table 1: Denoising results (average RMSE for test data).

Graphs	Sensor & Sensor			Sensor & Community		
Methods / σ	.25	.50	.75	.25	.50	.75
Noisy	.250	.500	.749	.250	.500	.751
GLF (spatial)	.094	.124	.144	.119	.170	.205
GLF (modality)	.095	.126	.147	.045	.070	.093
HD (spatial)	.089	.126	.148	.107	.157	.198
HD (modality)	.090	.126	.152	.039	.068	.092
SVDS	.069	.123	.149	.075	.105	.133
GCN (spatial)	.131	.126	.132	.202	.200	.205
GCN (modality)	.130	.124	.129	.132	.132	.136
GUTF	.387	.382	.400	.363	.364	.363
Ours ($P = 1$)	.067	.091	.106	.039	.061	.079
Ours ($P = 5$)	**.051**	.080	.099	.030	.046	**.060**
Ours ($P = 9$)	**.051**	**.078**	**.098**	**.029**	**.045**	**.060**

Within the comparison of the proposed methods, it was found that the RMSE for the twofold graph 2 is better than that for the twofold graph 1. This suggests that the assumed signal model is more effective for piecewise constant signals.

The visualizations of the signals and errors are shown in Fig. 3 for $\sigma = .75$. As seen in the figure, the proposed method has successfully suppress noise in both directions, while the other single graph-based methods often smooth only one direction.

4. CONCLUSIONS

In this paper, we proposed a multimodal graph signal denoising method via twofold graph smoothness regularization with deep algorithm unrolling. The iterative algorithm for solving the minimization problem is derived from the half-quadratic splitting method. We unrolled it to learn its internal parameters from the available dataset. The experimental results showed that the proposed method outperforms several existing methods.

5. REFERENCES

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE ICCV*, 1998, pp. 839–846.

[2] T. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 231–241, 2001.

[3] D. Garcia, "Robust smoothing of gridded data in one and higher dimensions with missing values," *Computational Statistics & Data Analysis*, vol. 54, no. 4, pp. 1167–1178, 2010.

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

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

[6] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[7] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling signals on graphs: From theory to applications," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 14–30, 2020.

[8] J. Hara, Y. Tanaka, and Y. C. Eldar, "Generalized graph spectral sampling with stochastic priors," in *Proc. ICASSP*, 2020, pp. 5680–5684.

[9] Y. Tanaka and Y. C. Eldar, "Generalized sampling on graphs with subspace and smoothness priors," *IEEE Trans. Signal Process.*, vol. 68, pp. 2272–2286, 2020.

[10] M. Nagahama, K. Yamada, Y. Tanaka, S. H. Chan, and Y. C. Eldar, "Graph signal restoration using nested deep algorithm unrolling," *arXiv:2106.15910 [cs, eess]*, 2021.

[11] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *Proc. ICASSP*, 2012, pp. 3921–3924.

[12] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. ICASSP*, 2013, pp. 5445–5449.

[13] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, 2015.

[14] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal and Inf. Process. over Networks*, vol. 2, no. 2, pp. 137–148, 2016.

[15] Y. Tanaka and A. Sakiyama, "M-channel oversampled graph filter banks," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3578–3590, 2014.

[16] T. H. Do, D. Minh Nguyen, and N. Deligiannis, "Graph autoencoder for graph signal denoising," in *Proc. ICASSP*, 2020, pp. 3322–3326.

[17] S. Chen, Y. C. Eldar, and L. Zhao, "Graph unrolling networks: Interpretable neural networks for graph signal denoising," *IEEE Trans. Signal Process.*, vol. 69, pp. 3699–3713, 2021.

[18] N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, "Fast robust PCA on graphs," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 4, pp. 740–756, 2016.

[19] F. Wang, Y. Wang, G. Cheung, and C. Yang, "Graph sampling for matrix completion using recurrent gershgorin disc shift," *IEEE Trans. Signal Process.*, vol. 68, pp. 2814–2829, 2020.

[20] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, 2021.

[21] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imaging Sci.*, vol. 1, no. 3, pp. 248–272, 2008.

[22] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image deblurring via algorithm unrolling," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 666–681, 2020.

[23] S. Ono, I. Yamada, and I. Kumazawa, "Total generalized variation for graph signals," in *Proc. IEEE Conf. CVPR*, 2015, pp. 5456–5460.

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

[25] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recognition*, vol. 41, no. 11, pp. 3328–3342, 2008.

[26] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt 3$," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5040–5053, 2014.

[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, p. 14.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.