Non-smooth interpolation of graph signals

Antoine Mazarguil, Laurent Oudre, Nicolas Vayatis

Non-smooth interpolation of graph signals

Antoine Mazarguil[a,*], Laurent Oudre[a], Nicolas Vayatis[a]

[a] *Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, F-91190, Gif-sur-Yvette, France.*

Abstract

We consider the problem of signal interpolation on graphs, i.e. recovering one or multiple graph signal values from incomplete measurements. We propose a review of the graph signal interpolation methods, which enlightens the restrictive underlying hypothesis of signal smoothness over the graph. We formulate a new interpolation framework based on a locality criterion designed to handle non-smooth signals, and provide corresponding specific solutions. We validate the proposed methods on synthetic and real-world interpolation problems, and analyse the stability of our methods regarding the interpolation problem main parameters.

Keywords: graph signal processing, interpolation, matrix completion, inpainting, signal recovery, matrix completion, structural equation model.

Contents

*Corresponding author

1. Introduction

In recent years, there has been a significant increase in the production and storage of data from multiple sources.The underlying structure of these sources can be modeled as a network that carries information about the organization and interaction of data sources throughout the controlled process.

The presence of missing values in network data streams is an important issue that may arise in most real-world scenarios. This loss of information can be due to many different factors, such as sensor default, loss in data transfer or system maintenance. Data interpolation methods are developed in this context, in order to replace the missing values with reliable estimates. In the common case of sensor network measurements, the information provided by the proximity or interaction of the sensors can be a decisive asset for the quality of the interpolation. Recent works have proposed to use the underlying structure of the data to guide the interpolation process. In particular, Graph Signal Processing (GSP) [1, 2] has been introduced as an intuitive framework to deal with data lying on a network structure, which applies to a wide class of classical use cases such as traffic [3, 4], meteorology [5], temperature [6], human or point cloud motion [7], radio [8], neurosciences [9, 10, 11] and molecule analysis [12]. Reconsidering the interpolation task in the context of graph processing tools appears as a natural approach to tackle the complex task of missing data estimation.

The main question that arises when dealing with sensor networks is how to improve a signal processing task with knowledge of the sensor topology. Several approaches for the interpolation of graph signals under known topology exist in the literature (see Section 3 of this manuscript).Most methods share the same assumption about the knowledge provided by the graph structure, namely that the nodes connected by the graph have similar values.This assumption, called *smoothness*, can lead to a deteriorated prediction if the signal under study has dissimilar values between the nodes that are connected with respect to the ground truth topology.

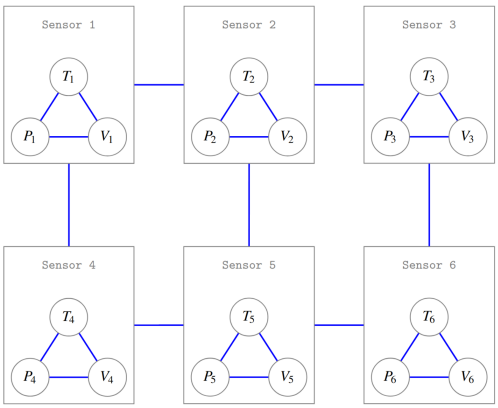
Especially, as will be illustrated in Section 2, in several practical use cases, the smoothness hypothesis is clearly not satisfied. For example, consider a process that involves two highly negatively correlated sensors. This information cannot be used with the smoothness constraint, that will assume that neighboring nodes must have similar values. Another example is the case of information senders and receivers connected by a network. Since their roles are inherently different, there is no a-priori reason for the connected agents to exchange similar amounts of messages. Nevertheless the number of interactions of an agent can be estimated from its neighbors. Finally, when considering multimodal graph signals (for instance weather data such as temperature/pressure/rainfall), it is straightforward that the quantities measured by a specific sensor are not similar.Yet these measurements carry information about each other, which must be taken into account by an appropriate interpolation method.

Recently, linear Structural Equation Model (lSEM) have been applied in the field of Graph Signal Processing for a joint inference of a new topology and the unknown part of the signals [13]. While the underlying linear model allows for anti-correlations in the signals, this method is not designed to take into consideration a previous topology of a sensor network, and a new topology is learned only from the signals.

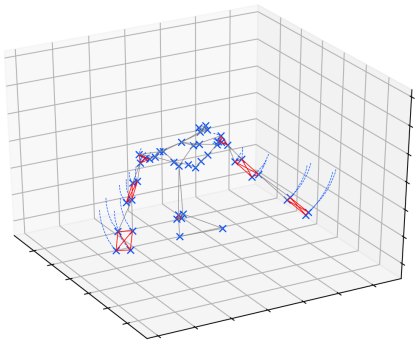
In this manuscript, the aim is to find a legitimate trade-off between taking the topology as is, and learning a new topology from scratch. To this end, our main hypothesis is that the proximity of two data sources in the sensor network indicates that they have a higher probability of sharing information, and thus are more likely to contribute to the reconstruction of the other. We propose a novel penalization for the learning of a lSEM that favors the contribution of local nodes to the reconstruction process.

Contributions. . The contributions of the paper are as follows:

- **A literature review on graph signal interpolation techniques** under a formalism that allows the comparison of most techniques.
- **A novel penalization for structural equation learning** that enforces proximity of the reconstruction weights.



(a) Multimodal graph of 6 sensors collecting temperature, pressure and rainfall



(b) Human upper body 3D motion during arm lowering. The red connections correspond to sensors that are located on the same rigid body part. Their relative positions are therefore constants.

Figure 1: Two motivating examples : a multimodal graph (left) and a 3D sensor network (right).

- **Four algorithms** for the search of the associated optimization problems, along with convergence results for some of them.
- **A benchmark of the presented methods** on synthetic and real-world datasets.
- **A stability analysis of the proposed methods** regarding the interpolation problem parameters.

2. Background

Before introducing the problem we want to solve and the notations of the article, we will start this section by introducing several motivating examples that will highlight the foundations of the proposed approach.

2.1. Motivating examples

Temperature / Pressure / Rainfall graph signals. In this example, we consider a network of sensors that measure three quantities at different positions on a territory: temperature, pressure and rainfall. An example of such a network of sensors is shown in figure 1a. The usual methodology to deal with the different modalities of the signals is to consider a cartesian product of the sensor graph and a complete graph connecting the different modalities, allowing the problem to be handled by classical one-dimensional GSP techniques. However, there are no a-priori leads for the signal on the newly constructed graph to exhibit smoothness properties. In this specific example, without normalization, there is no reason to expect temperature and pressure data to display similar values. Yet, an interpolation methods should take into account all the available measurements on a sensor, should a specific modality be missing. This raises the question of how to perform the interpolation, while preserving the proximity information between the elements that is inherited by both the ground sensor network and the modalities.

Communication network. Consider a communication network composed of actors playing different roles, like senders and receivers. The number of messages exchanged per actor is a signal that naturally adapts to the topology induced by the network, but no regularity property is a priori displayed by the signals (powerful receivers can have a high number of messages exchanged while being

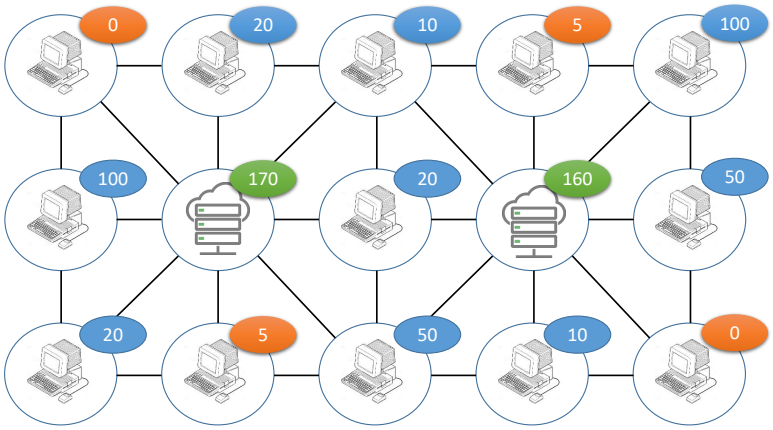


Figure 2: Communication network involving users and servers, enhanced with the count of exchanged messages over a period of time.

connected to small senders). Figure 2 displays an example of such a sensor network. Please note that on the figure, the signal is not smooth according to the structure. In the event of missing data, it seems relevant to estimate the unknown message count of a node from the message counts in its k-hop neighborhood. In other words, the relevant information for the interpolation is likely to be localized "near" the interpolated node. In this example, the structure of the network is a key object that needs to be taken into consideration for the interpolation of potentially missing data, and not using it could lead to deteriorated estimates. However, the structure itself is not sufficient, and relevant information is carried by the signals themselves, such as communication habits (i.e. correlations). An optimal solution should take advantage of both the signals and the network structure.

3D positions sensor network. 3D position sensors have been widely used in various application fields, from medicine to CGI. A natural graph can for instance be constructed by linking sensors belonging to the same body segments. In the case of missing or noisy data on a set of position signals, this underlying structure provides an indication on where the information can be found to perform the estimation. As an example, let us consider a subset of 4 sensors that are located on a rigid part of the body (e.g., a forearm), positioned such that they form a rectangle during the measurement process. Figure 1b displays an example of such a sensor network. In the figure, the red connections correspond to sensors attached to the same rigid body part. It is straightforward that the 3D positions over time of these 4 sensors, denoted by $p_1(t), p_2(t), p_3(t)$ and $p_4(t)$ follow a simple linear equation at each time step t, hereby revealing a candidate equation for the interpolation of $p_1(t)$: $\hat{p}_1(t) = p_2(t) + p_4(t) - p_3(t)$. A similar relation can be derived for any subset of sensors that are close enough to be considered on an nearly rigid sub-structure, or when a specific sensor can be approximated during the process as a weighted barycenter of a set of other sensors. In this particular example, the locality of the reconstruction weights with respect to the moving object is particularly relevant for large sensor networks, for it prevents the use of irrelevant linear relations that may appear with the high dimensionality.

Moreover, the analysis of 3D position data enhanced with a connectivity graph shows us that the signals cannot be approximated only by their low graph frequencies, invalidating the hypothesis of smooth signals. Figure 3 displays the time-average distribution of the signals energies in the graph

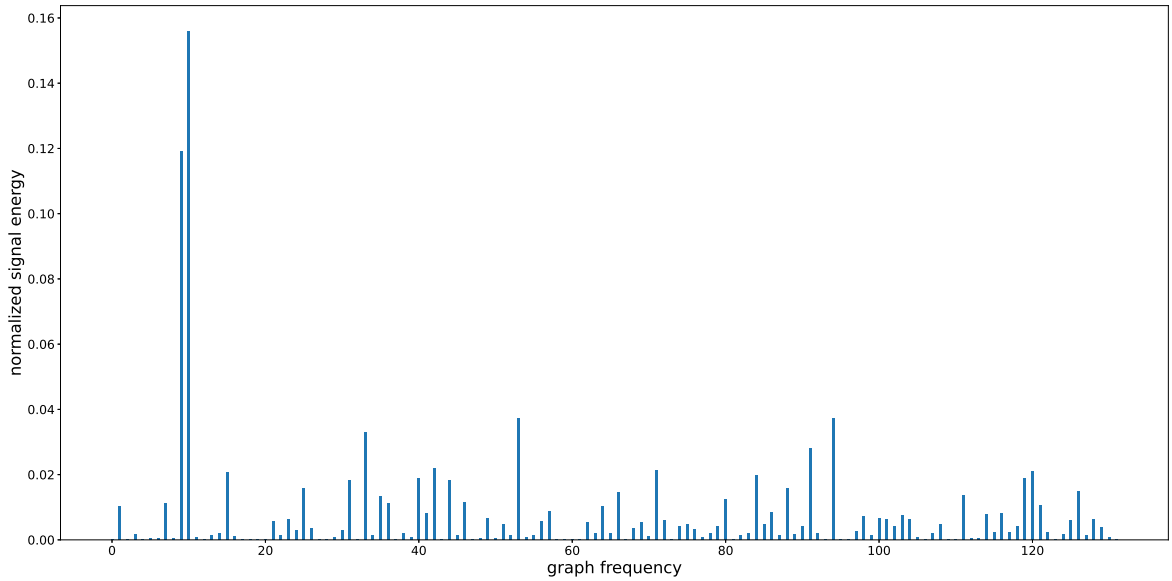


Figure 3: Time-mean energy of the **mocap** dataset in the Graph Fourier Transform space. The 3D data are centered at each timestep, inducing the inactivity of the 0 frequency. The values are normalized so that they sum to 1.

frequency domain for the **mocap** dataset (see Subsection 6.1 for details on the signals and the graph generation procedure).

2.2. Problem formulation

Let us consider a graph $\mathcal{G}$ containing N nodes, and L graph signals on $\mathcal{G}$ stored in a measurement matrix $\mathbf{T}$ of size $N \times L$. The matrix $\mathbf{T}$ contains missing values: the sets of known/unknown entries are respectively denoted as $\mathcal{K}/\mathcal{U}$. Provided the known measurements $\mathbf{T}_\mathcal{K}$, the aim of the interpolation task is to find a reconstruction matrix $\mathbf{X}$ that is coherent with $\mathbf{T}_\mathcal{K}$, but where all missing values are replaced by appropriate estimates.

In the article, two configurations will be investigated:

- The *noiseless case* where the known values in $\mathbf{T}$ are supposed to be noise free. In this configuration the known values in $\mathbf{X}$ and $\mathbf{T}$ are exactly equal, i.e.

$$\mathbf{X}_\mathcal{K} = \mathbf{T}_\mathcal{K}. \tag{1}$$

- The *denoising case* where the known values in $\mathbf{T}$ are supposed to be noisy and should therefore be denoised as well. In this configuration, the constraint for known values writes as

$$\mathbf{X}_\mathcal{K} = \mathbf{T}_\mathcal{K} + \mathbf{E}^{measure} \tag{2}$$

  where the measurement errors in $\mathbf{E}^{measure}$ are assumed to behave as Gaussian noise, i.e. $e_i^{measure} \sim \mathcal{N}(0, \sigma^2_{measure})$.

In order to synthesize the presentation of the different problem, we introduce a data-fitting function φ, such that $\varphi(\mathbf{X}_\mathcal{K} - \mathbf{T}_\mathcal{K})$ relates the noise scenario. The corresponding φ functions are defined as follows:

$$\varphi^{\text{noiseless}}(\mathbf{X} - \mathbf{T}) = \begin{cases} 0 & \text{if } \mathbf{X} - \mathbf{T} = \mathbb{0} \\ +\infty & \text{otherwise} \end{cases} \tag{3}$$

$$\varphi^{\text{denoising}}(\mathbf{X} - \mathbf{T}) = ||\mathbf{X} - \mathbf{T}||_F^2 \tag{4}$$

Symbol	Description	Dimension
$\mathbf{T}$	Measurement matrix	$\mathbb{R}^{N\times L}$
$\mathbf{X}$	Reconstruction matrix	$\mathbb{R}^{N\times L}$
$\mathbf{x}$	A graph signal	$\mathbb{R}^{N}$
x_i	Value of the signal $\mathbf{x}$ at node v_i	$\mathbb{R}$
$\mathbf{x}_n$	n^{th} row of matrix $\mathbf{X}$	$\mathbb{R}^{1\times L}$
$\mathbf{x}^l$	l^{th} column of matrix $\mathbf{X}$	$\mathbb{R}^{N\times 1}$
$\mathbf{X}_{\text{-}n}$	$\mathbf{X}$ deprived from the n^{th} row	$\mathbb{R}^{(N-1)\times L}$
$\mathbf{x}^l_{\text{-}n}$	$\mathbf{x}^l$ deprived from the n^{th} row	$\mathbb{R}^{(N-1)\times 1}$
$\mathcal{K}$ / $\mathcal{U}$	Set of known / unknown measurements	
$\mathbf{X}_\mathcal{K}$ / $\mathbf{X}_\mathcal{U}$	Vector of all known / unknown values	
$\mathbf{x}^l_\mathcal{K}$ / $\mathbf{x}^l_\mathcal{U}$	Known / unknown values of $\mathbf{x}^l$	
$\mathbf{W}$	Graph affinity matrix	$\mathbb{R}^{N\times N}$
$\mathbf{A}$	Matrix of the reconstruction weights	$\mathbb{R}^{N\times N}$
$\mathbf{a}^n$	Row n of the matrix $\mathbf{A}$	$\mathbb{R}^{1\times N}$
$\mathbf{a}^n_{\text{-}n'}$	Row n of the matrix $\mathbf{A}$ deprived of element n'	$\mathbb{R}^{1\times(N-1)}$
$a^{n'}_n$	Element at row n, column n' of the matrix $\mathbf{A}$	$\mathbb{R}$
$\mathbf{b}$	Vector of the reconstruction intercepts	$\mathbb{R}^{N\times 1}$
b_n	Reconstruction intercept at node n	$\mathbb{R}$

Figure 4: Notations

2.3. Notations and Graph Signal Processing tools

Let us consider a **weighted undirected connected** graph $\mathcal{G} = (V, E, \mathbf{W})$ composed of a finite collection of vertices $V = \{v_1, ..., v_N\}$, a set of edges $E \subset V \times V$ and a weight matrix $\mathbf{W} \in \mathbb{R}_+^{N\times N}$ for which the element $w_{i,j}$ quantifies the affinity between nodes v_i and v_j. The weight $w_{i,j}$ is equal to 0 for $(v_i, v_j) \notin E$, and since graph is **undirected**, we have $\mathbf{W} = \mathbf{W}^T$. The neighborhood $\mathcal{N}(v_i)$ of a node v_i is the set of nodes connected to v_i through an edge. The degree of a node is defined as the sum of all its connected weights, and the degree matrix D is the diagonal matrix $\mathbf{D} = \mathit{Diag}\,(\mathbf{W}\mathbb{1})$. Finally, we assume in this article that the graph $\mathcal{G}$ is **connected**, i.e. that a path can be found between every pair of nodes.

A **real signal on the graph** $\mathcal{G}$ is a function $V \longrightarrow \mathbb{R}$. We can represent the signal as a vector $\mathbf{y} \in \mathbb{R}^N$, in which $\mathbf{y}_i$ is the signal at the node v_i.

Given a graph $\mathcal{G} = (V, E, W)$, an **embedding** Φ of $\mathcal{G}$ in a D-dimensional space is a mapping $(V \to \mathbb{R}^D)$ such that a notion of proximity (i.e. similarity) is preserved from the graph domain to the Euclidean space. For the rest of the article, $\mathbf{\Phi}$ will denote the $\mathbb{R}^{N\times D}$ embedding matrix of the D-dimensional coordinates of the graph nodes representations.

3. Literature review

The problem of signal interpolation has been studied in a dense literature, with the specificity of being addressed by numerous mathematical fields (resulting in a wide variety of frameworks and denominations, such as interpolation, matrix completion, inpainting, missing value estimation or process inference). A clear distinction in this literature can be made on the use of a prior knowledge about the studied process, in the form of a dependence between signal modalities. A first class of algorithms performs interpolation using prior information about the sensor organization modeled by a graph structure. These methods find their foundations in the GSP literature, and assume that the observed process is related to the provided topology. The second category of methods does not assume prior knowledge of such a topology, and aims to learn the relations between the modalities from the known signal values. A discussion of the two categories of interpolation is presented in the next section.

Remark 1. *In the graph signal reconstruction literature, a large amount of methods assume that the selection of the set of missing values is performed along with the reconstruction task. These methods include graph signal sampling [14, 15, 16, 17] and are often used for graph signal compression [1, 18, 19]. In this scenario, the resulting interpolation method and the sampling task are deeply connected, and the interpolation method is irrelevant if the set of missing values is taken at random. Therefore, these methods, while being graph signal interpolation methods, will be omitted in our analysis.*

Remark 2. *Other works related to graph signal interpolation can be found in the GSP literature. For instance, the hypothesis of a dynamic process such as a diffusive field [20, 21, 22] can guide the estimation. Other works focus on estimating other statistics than the signals themselves, such as the second order statistics of the signals [23]. These approaches have been omitted in our analysis.*

3.1. Signal inference under known topology

Most GSP techniques for signal interpolation exploit the graph information through the use of a representation space. A representation space $\mathbf{\Phi}$ is defined by a family of vectors $(\Phi_1, ..., \Phi_D) \in \mathbb{R}^{N \times D}$, where each vector displays regularity properties regarding the graph structure. Representing a graph signal as a linear combination of elements from this family allows a comprehensive quantification of some signal properties, such as similarity between neighbors or signal variation regularity. Representing a graph signal as a linear combination of elements of this family allows a complete quantification of some properties of the signal, such as the similarity between neighbors or the regularity of the signal variation.

However, not all representation vectors in a family may have the same relevance to decompose the targeted signal. Some of them are less likely to appear in the decomposition, while others can be expected to be predominant. Rather than removing the less relevant elements from the representation family, this relevance can be included in the estimation process by reinforcing each vector of the family with a penalty. This penalty quantifies the degree of irrelevance of using each representation vector independently.

Given a penalized representation space and an interpolation problem, the main objective shifts from finding the values on the unknown nodes to finding the less penalized signal decomposition as linear combination of the representation vectors. This decomposition is obtained in such a way that the estimated signal fits the known values of the matrix $\mathbf{T}$. This way, the resulting interpolation respects the measurements while displaying the specificities induced by the representation space and its penalization. Once the signal decomposition $\mathbf{Z}^* \in \mathbb{R}^{D \times L}$ has been estimated, the interpolated signal can be written as a matrix product of the representation family matrix and the decomposition:

$$\mathbf{X}^* = \mathbf{\Phi}\mathbf{Z}^* \tag{5}$$

The interpolation task therefore rewrites as a representation learning problem, where the goal is to learn the decomposition $\mathbf{Z}^*$ that fits the set of known values $\mathbf{T}_{\mathcal{K}}$ and minimizes the penalization induced by the weights. This problem writes as

$$\mathbf{Z}^* = \underset{\mathbf{Z} \in \mathbb{R}^{D \times L}}{\operatorname{argmin}} \; \varphi\Big((\mathbf{T} - \mathbf{\Phi}\mathbf{Z})_{\mathcal{K}}\Big) \; + \; \gamma \sum_{d=1}^{D} h_d^2 ||\mathbf{z}_d|| \tag{6}$$

where

- $\mathbf{\Phi} \in \mathbb{R}^{N \times D}$ are the representation family
- $\mathbf{Z} \in \mathbb{R}^{D \times L}$ is the decomposition of the signal in the representation space and $\mathbf{z}_d$ the d-th row of $\mathbf{Z}$
- $\mathbf{h} \in \mathbb{R}^D$ is the vector of penalization weights and h_d the d-th element of $\mathbf{h}$
- $||.||$ is the penalization norm and γ the penalization hyper-parameter

Most graph signal interpolation algorithms can be expressed in this formalism and differ on two main points

- The representation space $\mathbf{\Phi}$ in which we have prior knowledge on the process
- The regularisation term parametrized by $\mathbf{h}$ and $||.||$, function of the signal decomposition in this representation space

An overview of the most common approaches is displayed on Table 1. First, let us describe the main parameters of this interpolation process.

3.1.1. Choice of the representation space

In the literature, the initial and preferred approach by most authors is the use of the Laplacian eigenmaps [24, 25, 26]. This representation space is obtained by taking the eigenvectors of a graph Laplacian, and is denoted as the spectral domain. The bandlimited version of the Laplacian eigenmaps is obtained by truncating the same family in order to keep only those eigenspaces whose eigenvalues are lower than a threshold, which allows to avoid the complete diagonalization. These approaches seek a solution as a combination of "low-frequency" components, that are relatively smooth on the graph. Other versions of the Laplacian can be used for the Laplacian eigenmaps embedding, such as the normalized Laplacian [27, 30] or the random walk Laplacian. These Laplacians normalize the strength of the weights by taking into account their importance in the local connections.

The Laplacian eigenmaps techniques are specific solutions to the problem of generating an Euclidean representation space for graph nodes, which is defined as graph embedding [34], and other graph embedding techniques can be used as representation space for the interpolation method. For instance, the diffusion map embedding can be used [33], therefore imparting regularity properties of this embedding to the reconstructed signal.

The graph-based frame theory [28, 29] is a special case of the previously described penalized framework, for the signal is sought as linear combination of elements from a family denoted as a frame. However, the notion of penalization is not introduced in this approach.

The conditions under which a frame is well suited for band-limited signals interpolation has been studied in [29]. Sufficient conditions for exact recovery have been established, and the stability in the case of noisy measurements has been quantified. These theoretical results allow the authors to design samplers suited for their approaches, minimizing the expected risk.

More generally, any family of vector that carries a graph-related specificity can be used for the interpolation. Another relevant option for the choice of the representation vectors is the graph wavelets [35, 36, 37], for they carry signal localization in both the node domain and the spectral domain at different scales. However, their applicability for signal interpolation remains to be studied.

3.1.2. Choice of the regularisation

Graph filters [38, 39] allow to rescale the signal decomposition in the spectral domain. These filters emphasize or reduce the energy of the signal in specific frequency ranges, and minimizing their response leads to the consideration of penalization weights introduced in the framework of Equation (6).

The expectation of a smooth signal corresponds to the minimization of the response of the signal to a simple graphical filter, provided by the Laplacian itself. It leads to the $\{\sqrt{\lambda_i}\}$ penalization [24, 27, 26, 32]. Bandlimitedness is a constraint that has been introduced in graph signal processing theory, that decomposes the signal in the Laplacian eigenspaces up to a cut-off eigenvalue. This approach corresponds to the D Laplacian eigenmaps with a specific value of D, and no penalization ($h = \mathbb{0}$) [24, 25]. The widely-used band-limited hypothesis of graph signals can be imposed to the reconstruction weights, by using a cut-off penalization, that associates respectively the low and high eigenspaces to a low and high penalization [30].

Graph ARMA filters [31] can be used as well in the interpolation process. These filters tends to approximate optimal graph signal responses through the use of rational polynomials of the Laplacian matrix, which allows a fast computation.

Table 1: Overview of GSP methods for graph signal interpolation

Paper	Algorithm	L	Representation space $\mathbf{\Phi}$	Data attachment function φ	Penalization norm $\|\|.\|\|$	Penalization weight $\mathbf{h}$
Narang 2013 [24]	Uniqueness set reconstruction	1	Bandlimited laplacian eigenmaps	$\varphi^{\text{noiseless}}$	L_2	$\mathbb{0}$
Narang 2013 [25]	Regularized intepolation	1	Bandlimited laplacian eigenmaps	$\varphi^{\text{denoising}}$	L_2	$\{ \exp(-1/\lambda_i) \lvert \lambda_i < w^* \}$
Kalofolias 2014 [26]	Graph Matrix Completion	> 1	Laplacian eigenmaps	$\varphi^{\text{denoising}}$	L_2	$\{ \sqrt{\lambda_i} \}$
Chen 2015 [27]	Graph Signal Inpainting	1	Normalized eigenmaps	$\varphi^{\text{noiseless}}$ $\varphi^{\text{denoising}}$	L_2	$\{ \sqrt{\lambda_i^{\mathcal{N}}} \}$
Chen 2015 [27]	Graph Matrix Completion	> 1	Normalized eigenmaps	$\varphi^{\text{noiseless}}$ $\varphi^{\text{denoising}}$	L_2	$\{ \sqrt{\lambda_i^{\mathcal{N}}} \}$
Chen 2015 [27]	Robust Graph Signal Inpainting	1	Normalized eigenmaps	$\varphi^{\text{noiseless}}$ $\varphi^{\text{denoising}}$	L_2	$\{ \sqrt{\lambda_i^{\mathcal{N}}} \}$
Chen 2015 [27]	Robust Graph Matrix Completion	> 1	Normalized eigenmaps	$\varphi^{\text{noiseless}}$ $\varphi^{\text{denoising}}$	L_2	$\{ \sqrt{\lambda_i^{\mathcal{N}}} \}$
Wang 2015 [28]	ILSR	1	ILSR Frame	$\varphi^{\text{noiseless}}$	L_2	$\mathbb{0}$
Wang 2015 [28]	IPR	1	IPR Frame	$\varphi^{\text{noiseless}}$	L_2	$\mathbb{0}$
Wang 2015 [28]	IWR	1	IWR Frame	$\varphi^{\text{noiseless}}$	L_2	$\mathbb{0}$
Tsitsvero 2016 [29]	Frame-based reconstruction	1	Sampling Frame	$\varphi^{\text{noiseless}}$	L_2	$\mathbb{0}$
Tsitsvero 2016 [29]	Frame-based reconstruction	1	Generalized Sampling Frame	$\varphi^{\text{noiseless}}$	L_2	$\mathbb{0}$
Romero 2016 [30]	Diffusion graph kernel	1	Normalized eigenmaps	$\varphi^{\text{denoising}}$	L_2	$\{ \exp(\sigma^2 \lambda_i^{\mathcal{N}}/2) \}$
Romero 2016 [30]	P-step random walk graph kernel	1	Normalized eigenmaps	$\varphi^{\text{denoising}}$	L_2	$\{ (a - \lambda_i^{\mathcal{N}})^{-p} \}$
Romero 2016 [30]	Regularized graph kernel	1	Normalized eigenmaps	$\varphi^{\text{denoising}}$	L_2	$\{ 1 + \sigma^2 \lambda_i^{\mathcal{N}} \}$
Romero 2016 [30]	Bandlimitation graph kernel	1	Normalized eigenmaps	$\varphi^{\text{denoising}}$	L_2	$\{ \beta \mathbb{1}(\{\lambda_i < w\}) + (1/\beta)\mathbb{1}(\{\lambda_i \geq w\}) \}$
Isufi 2016 [31]	Graph ARMA interpolation	1	Normalized eigenmaps	$\varphi^{\text{denoising}}$	L_2	$\{ P(\lambda_i)/Q(\lambda_i) \}$
Mashhadi 2017 [32]	Sparse Graph Interpolation	1	Laplacian eigenmaps	$\varphi^{\text{denoising}}$	L_1	$\{ \sqrt{\lambda_i} \}$
heimowitz 2019 [33]	Diffusion maps interpolation	1	Graph diffusion map	$\varphi^{\text{noiseless}}$ $\varphi^{\text{denoising}}$	L_1, L_0	$\{ (1 - \lambda_i^{\mathcal{N}})^{-1} \}$

Another important parameter of the regularization term is the norm used in the loss. Two classical regularisation terms can be expressed in this formalism. The Ridge regularizer corresponds to the penalization norm $||.||_2^2$ and has been widely used for closed form solution and convergence properties of iterative algorithms. More recently, the Lasso regularizer (penalization norm $||.||_1$) has been used to encourage sparsity in the representation space [32, 33].

3.1.3. Other approaches

Several additional terms can be added to the optimization problem (6). As usual matrix completion, graph matrix completion [26, 27] adds a nuclear norm term to the optimisation problem, that favors a low-rank reconstruction, which leads to a better representation of signal. Outliers can also be dealt with by the addition in the optimisation problem of a sparse outlier matrix [27]. In the rest of the article, the issue of outliers will not be assessed, however the proposed algorithms can be modified to handle it.

3.2. Joint signal and topology inference

In many practical use-cases, the graph associated with the studied process is unknown or inaccurate. A fundamental way to approach the task of interpolating missing signals without prior knowledge of the graph structure is to infer the dependency relationships between the variables from the known signal values.

The main hypothesis of most approaches is the Structural Equation Models (SEMs), which postulates that an equation is satisfied by the different dimensions of the signal. Although promising works have been done in order to address a broad class of equation natures [40, 41], most approaches focus on the linear SEM (lSEM), which states that for any signal $\mathbf{x}^{(l)}$, its value on node n can be expressed as:

$$x_n^{(l)} = \sum_{n \neq n'} a_n^{n'} x_{n'}^{(l)} + e_n^{(l)} \tag{7}$$

with $e_n^{(l)}$ a random noise. These equations can be put together in a matrix format, which leads to the matrix expression of lSEM:

$$\mathbf{X} = \mathbf{A}\mathbf{X} + \mathbf{E}^{model} \tag{8}$$

with $\mathrm{Diag}(\mathbf{A}) = \mathbb{0}$ and traditionally $(e^{model})_i^j \sim \mathcal{N}(0, \sigma^2_{\mathrm{model}})$. In the previous equation, $a_n^{n'}$ denotes the element of the matrix $\mathbf{A}$ at row n' and column n.

The model is described by two parameters :

- The reconstruction weights $\mathbf{A}$
- The noise parameter σ_{model}

In this framework, the elements of the matrix $\mathbf{A}$ correspond to the coefficients of the linear structural equation, and they can be assimilated to the weights of the structure carrying the process. In the methods presented in this subsection, the matrix $\mathbf{A}$ is learnt from the known signals, therefore inheriting the name of *Topology Inference Method*. These methods can be grouped into three main categories, presented in the following subsections.

3.2.1. Statistical approaches

In the statistics literature, several approaches have been introduced in order to perform the joint estimation of dependency relationships and unknown signals. The MICE algorithm (Multiple Imputation by Chained Equation) [42] has been introduced as a practical way to impute missing data. The approach consists in performing an initial simple imputation in the dataset, and to iteratively impute new values through linear regression assuming all the other nodes are known. A similar approach is produced by Principal Component Analysis (PCA) with multiple imputations [43, 44], in which all unknown values are estimated at each iteration, without node-by-node estimation. For a comprehensive survey of PCA imputation with missing values, see [45].

3.2.2. Matrix completion approaches

Matrix completion is a classical framework for signal interpolation and denoising, that relies on the key assumption that the complete matrix to be recovered satisfies low rank properties. This approach leads to rank minimization algorithms [46] that involve singular value decomposition and signal thresholding [47]. Comprehensive reviews of matrix completion methods have been conducted in the literature [48, 49, 50, 51, 52].

Remark 3. *It is relevant to note that under the low rank criterion, the nullspace of the modeled signal is not reduced to zero, which implies the lSEM relations. Therefore, learning the low-dimensional representation of the signal is strongly related to the estimation of a set of linear structural equations.*

3.2.3. Graph Topology Inference approaches

More recently, Joint Inference of Signal and Graph (JISG) [13] has been introduced as a novel approach to simultaneously estimate a graph and the unknown part of the signals. In this framework, the dependency structure of the nodes is modeled by a graph, which is estimated such that the signals are smooth on it. The smoothness criterion with regards to the graph defined by $\mathbf{A}$ is translated by signals self-similarity through the use of the adjacency operator (i.e. $\mathbf{X} \simeq \mathbf{AX}$), which implies the lSEM hypothesis displayed in Equation (8). Sufficient conditions for exact recovery of both the topology and the signals have been shown in the context of noise-free measurements.

Remark 4. *The issue of learning the relations that are satisfied by the signals, while taking into account the initial structure, remains mostly unstudied. A first approach to attend this problem is to perform smooth interpolation on a re-weighted version of the original graph [24], seeking a new structure on which the initially non-smooth signals respect smoothness criterion. However, this approach does not allow new connections to be added to the graph structure, therefore possibly leading to deteriorated predictions.*

4. Proposed method

As discussed in the literature review, most graph signal interpolation methods either assume that the signals are smooth with respect to a provided graph structure, or focus on learning a set of relations that is satisfied by the signals. In the present work, our proposal is to take into account a provided graph structure for the estimation of the model respected by the signals. Similarly to [13], we perform a paired estimation of a linear structural equation model and the missing data. However, this estimation is guided by a prior knowledge about the sensor dependency, i.e. a known graph structure.

Specifically, the assumption made by our model is that the proximity in the provided graph structure is an indicator of a high contribution in the Linear Structural Model Equations. This assumption is especially relevant when the studied process depends on a given graph structure, while the smoothness hypothesis is not respected on it (see Section 2 for motivating examples).

In our approach, two different structures are used to handle non-smooth graph signal interpolation:

- The structure of the data-sources, carried by a known graph, with weight matrix $\mathbf{W}$.
- The structural equations model that is respected by the signals, with coefficients $\mathbf{A}$ and $\mathbf{b}$.

The following section presents the methodology of our approach. First, we present the optimization problem along with the different terms composing it. Then, an emphasis is made of the graph locality penalisation and the possible graph distances that can be used.

4.1. Optimization problem description

Provided a graph $\mathcal{G}$, and a set of known measurements $\mathbf{T}_{\mathcal{K}}$, our aim is to estimate the full matrix $\mathbf{X}$ by solving the following optimization problem :

$$\begin{aligned}\widehat{\mathbf{A}}, \hat{\mathbf{b}}, \widehat{\mathbf{X}} \;=\; \underset{\mathbf{A},\mathbf{b},\mathbf{X}}{\mathbf{argmin}} \quad & \varphi(\; \mathbf{X}_{\mathcal{K}} - \mathbf{T}_{\mathcal{K}} \;) \\ & + \lambda \; ||\mathbf{X} - (\mathbf{A}\mathbf{X} + \mathbf{b}\mathbb{1}_L^T)||^2 \\ & + \mu \; \mathrm{Loc}(\mathbf{A}) \end{aligned} \tag{9}$$

$$\textbf{Subject to :} \;\; \mathrm{Diag}(\mathbf{A}) = \mathbb{0}.$$

In the previous equations, three kind of terms / constraints can be identified:

1. The **data** term takes the form of a function applied to the difference between the known values and their estimates.
2. The **model** term quantifies how well the affine model (lSEM) is respected by the estimated values.
3. The **graph** term enforces the localisation of the reconstruction weights on the graph structure.

The following section describes each term of the loss function and their impact on the optimization problem and the resulting interpolation.

4.2. Data term

As presented previously, two cases have to be considered :

- **The noiseless case.**

  The known measurements are assumed to be exact, which corresponds to the following function:

$$\varphi^{\text{noiseless}}(\mathbf{X}) = \begin{cases} 0 & \text{if } \mathbf{X} = \mathbb{0} \\ +\infty & \text{otherwise} \end{cases} \tag{10}$$

  In term of optimization resolution, this function corresponds to the addition of the constraint $\mathbf{X}_{\mathcal{K}} = \mathbf{T}_{\mathcal{K}}$ on the variables, which reduces the dimension of the reachable solution space.

- **The denoising case.**

  The known measurements are assumed to be noised, therefore the proximity between the known values and their estimation is desired, even though a difference is expected. The associated function is the Euclidean norm:

$$\varphi^{\text{denoising}}(\mathbf{X}) = ||\mathbf{X}||_F^2. \tag{11}$$

  Using this hypothesis allows for a compromise between fitting the model to the known measurements and finding a relevant model. The resulting interpolation incorporates a denoised version of the input known measurements, with regards to the model that has been estimated.

4.3. Model term

As mentioned in the beginning of this section, our method is based on the linear structural equation model, with the addition of an intercept to the linear equations. The reconstructed matrix $\mathbf{X}$ is assumed to follow an affine model written as :

As mentioned at the beginning of this section, our method is based on the linear structural equation model, with the addition of an intercept to the linear equations. The reconstructed matrix is assumed to respect an affine model written as follows:

$$\mathbf{X} = \mathbf{A}\mathbf{X} + \mathbf{b}(\mathbb{1}_L)^T + \mathbf{E}^{\text{model}}, \tag{12}$$

with $\mathrm{Diag}(\mathbf{A}) = \mathbb{0}$ and $(e^{model})_i^j \sim \mathcal{N}(0, \sigma^2_{\text{model}})$. The model is described by three parameters :

- The reconstruction weights $\mathbf{A}$
- The reconstruction intercepts $\mathbf{b}$
- The noise parameter σ_{model}

The underlying assumption is that the values of a signal on a node can be estimated from the values on other nodes, by an affine formula. Note that this model is general enough to include many graph signal models. For instance, any reconstruction method that obtains the missing values from a matrix product of the known values [24, 28, 30] naturally fits the lSEM hypothesis. In most interpolation methods, the use of this model is therefore implicit.

4.4. Graph term

In this manuscript, the graph is assumed to be an indicator of information shared between nodes, and our goal is to achieve a more robust reconstruction by encouraging the involvement of local nodes to the process. This criterion is induced by a penalization on the affine weights of the model parameters.

Given a distance obtained from the graph, $d_{\mathcal{G}} : V \times V \longrightarrow \mathbb{R}^+$, (such as the geodesic distance), the localization penalization is defined as :

$$\begin{aligned} \text{Loc}(\mathbf{A}) &= \sum_{i,j} d_{\mathcal{G}}(i,j)^2 (a^i_j)^2 \\ &= ||\mathbf{D}_{\mathcal{G}} \odot \mathbf{A}||^2 \end{aligned} \tag{13}$$

This penalty term quantifies how far the intensities of the weight vector $\mathbf{a}^i$ are from the node i in the sense of a distance induced by the graph. In other words, this penalty ensures that the contributions for signal reconstruction on node i is mostly carried on a set of nodes that are close to i. Note that the $\mathbf{a}^i$ values can be positive or negative, which allows to take into account negative contributions. Moreover, the notion of weight localization and smoothness of the reconstructed signal are not mutually exclusive, nor equivalent. For a model trained on smooth signals, the localization penalization term will not prevent the recovery of weights ensuring smoothness, therefore allowing a similar interpolation.

Figure 5 shows the intuition behind this penalty term. When the localization is small, the missing node is interpolated with only nearby nodes, forcing the reconstruction weights belonging to distant nodes to be close to zero.

Remark 5. *Notion of graph spread*

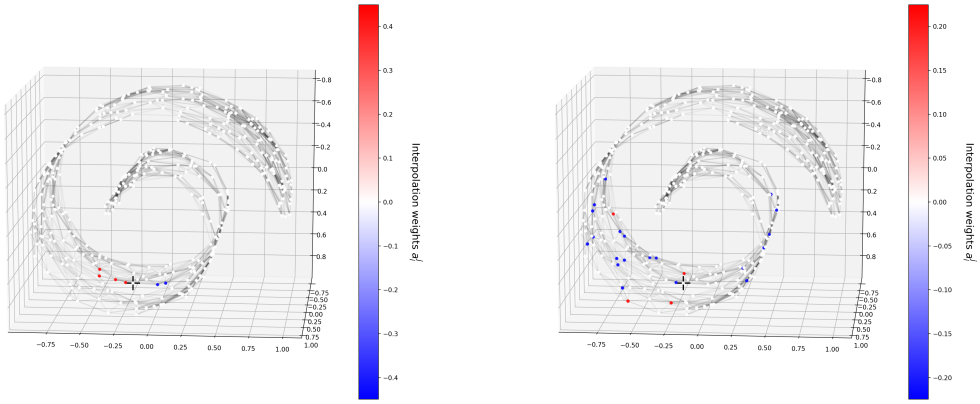
Our definition of the localization penalty is very similar to the notion introduced in graph uncertainty principle theory [53], in which the notion of signal spread on a graph is defined. Given a distance $d_{\mathcal{G}}$ obtained from the graph, the spread centered on the node i of a graph signal $\boldsymbol{z}$ is provided by

$$\Delta_i(\boldsymbol{z}) = \frac{1}{||\boldsymbol{z}||^2} \sum_{j=1}^{N} d_{\mathcal{G}}(i,j)^2 z_i^2. \tag{14}$$

This normalized function allows to quantify how far the intensity of the signal $\mathbf{z}$ is localized from a node i. In this article, the normalization factor is dropped in order to ensure the convexity of the sub-problems and the existence of closed form solutions.

4.5. Considered graph distances

As introduced at the beginning of this section, a main contribution of this paper is the graph localization penalty term described in Section 4.4. In this subsection, different graph distances are proposed, as well as their effect when incorporated to the optimization problem through the graph term (13).



(a) $\mathrm{Loc}(\mathbf{a}_i) = 1$ (b) $\mathrm{Loc}(\mathbf{a}_i) = 101.2$

Figure 5: Interpolation weights on a Swiss roll graph. The reconstructed node is marked with a cross. White nodes corresponds weights close to 0. Red (resp. Blue) nodes corresponds to a positive (resp. negative) reconstruction weight. The penalization is computed using the geodesic graph distance.

Geodesic distance. A direct measure of the distance between two nodes in a graph is the shortest path between the nodes. Formally, the geodesic distance between two nodes is the minimal sum of distances along any path between those nodes. The geodesic distance appears naturally in many applications and is well suited for graph structures provided as a set of pair-wise distances (rather than affinities). Although this distance allows the quantification of optimal travel distance, it fails to incorporate perhaps relevant graph information, like same cluster belonging or multiplicity of short paths between two nodes. In addition, the computation of the complete geodesic distance can be costly.

Laplacian eigenmaps distance. The issue of obtaining a graph distance in the context of affinity graphs is complex and unsolved [54]. To address this issue, one solution is to perform the D-laplacian eigenmaps of the graph[55], and compute the pairwise distance in the resulting embedding. Here, we describe the definition of this distance. For a given graph, the spectral decomposition of its Laplacian $\mathcal{L}$ is computed as follows :

$$\begin{aligned} \mathcal{L} &= \mathbf{U}\Lambda\mathbf{U}^T \\ \mathbf{U} &= (\mathbf{u}_1, ...\mathbf{u}_N) \in \mathcal{O}_N(\mathbb{R}) \\ \Lambda &= \mathrm{Diag}(\lambda_1, ..., \lambda_N),\ \lambda_1 \leq ... \leq \lambda_N \end{aligned} \tag{15}$$

The D-dimensional Laplacian eigenmap embedding Φ is then defined as $\Phi_D = (\mathbf{u}_1, ...\mathbf{u}_D) \ \in \mathbb{R}^{N\times D}$, and the according distance function is provided by the distance in the embedding space.

$$d^{\mathrm{eig}}_{\mathcal{G}}(i,j) = ||\Phi_D^T(\mathbf{e}_i - \mathbf{e}_j)||_F \tag{16}$$

This approach is computationally cheaper that computing the geodesic distance for a reasonable value of the dimension D, and the use of this distance as a notion of proximity between graph nodes has been largely justified by the spectral clustering method [56], which applies the K-means algorithm in the Laplacian eigenmaps space. More generally, a graph node distance can be derived from any embedding of the nodes, as the pair-wise distance in the embedding space. An interesting example is the weighted spectral embedding [57], for which the resulting distance can be interpreted as a mean commute time between the nodes. For an comprehensive review of node embedding techniques, see [58].

Graph affinity distance approximation. When dealing with large graphs, the computation of the distances presented above can become an issue. A proposition for incorporating an approximated notion

of distance with a low computational cost is to consider an expression directly obtained from the affinity matrix. For well chosen parameters $C > 0$ and $\alpha > 0$, we define the affinity distance approximation as follows :

$$d_{\mathcal{G}}^{\text{approx}}(i,j) = \begin{cases} \frac{1}{C+\alpha w_{i,j}} & \text{if } i \neq j \\ \frac{1}{C} & \text{otherwise} \end{cases} \tag{17}$$

This expression assigns a smaller distance for nodes connected by a strong affinity. However, this distance only characterizes the 1-hop neighborhood information, i.e. all non connected nodes have a constant distance of $1/C$. The main asset of this distance resides in its low computation complexity, while preserving a notion of graph proximity.

Null distance and uniform distance. In order to illustrate the interest of the addition of the graph distance penalty, our experiments in Section 7.5 include distances that do not carry any information on the graph structure. The null distance and the uniform distance are defined respectively as $d^{\text{null}}(i,j) = 0$ and $d^{\text{uniform}}(i,j) = \mathbb{1}(i \neq j)$. When the null distance (resp. uniform distance) is plugged in the problem (9), the estimation of the model parameters ($\mathbf{A}$, $\mathbf{b}$) consists of a set of classical linear (resp. Ridge) regressions. The optimization problem considered with the uniform distance is strongly related to the methodology proposed in [13]. The main difference is that the elastic-net penalization is used rather than the ridge penalization. Our experiments demonstrate how the addition of the graph penalization improves the quality of the interpolation in comparison to the algorithm proposed in [13] (c.f. section 7).

4.6. Properties of the proposed model

Remark 6. *Prior distribution on the model parameters*
Let us consider a Gaussian prior distribution on the model (12) parameters $\{a_{i,j}\}$, function of the pairwise nodes distances : $a_{i,j} \sim \mathcal{N}(0, \mu^{-1} d_G(i,j)^{-2})$. Then the a-posteriori maximum model likelihood leads to the optimization problem (9), with the same value for the parameter μ.

Property 1. *Bi-convexity*
The optimization problem (9) is bi-convex on the parameters $(\mathbf{X}, (\mathbf{A}, \mathbf{b}))$, i.e. the loss function is convex regarding $\mathbf{X}$ or $(\mathbf{A}, \mathbf{b})$ when the other parameters are fixed.

Convexity is a strong property that is often desired for optimization problems, as it yields interesting results such as the convergence of the parameters along the descent toward a global minimum and convergence rates of the resulting sequence [59, 60]. Block convex problems often appear when considering classical learning tasks, such as non-negative matrix factorisation [61, 62, 63] and dictionary learning [64, 65]. The property of bi-convexity encourages to consider alternated descents schemes. A rich literature exists on the optimization of block-convex functions [66, 67, 68], for which convergence properties can be studied. It is important to note that due to the non-convexity of the loss function, only a local minimum of the problem loss can be usually expected from the optimization procedure.

5. Resolution of the optimization problem

In this section, we describe the practical search of a local solution of the problem (9). As seen previously, this optimization problem is block-convex and will be solved by alternate minimization. The optimization scheme is therefore composed of two steps : the model estimation step (ME-step) which consists in updating $\mathbf{A}$ and $\mathbf{b}$ with fixed $\mathbf{X}$ and the signal imputation step (SI-step) which updates the matrix $\mathbf{X}$ with fixed $\mathbf{A}$ and $\mathbf{b}$. In this section, we will describe the resolution of the ME-step and four variants for the resolution of SI-step depending on the considered scenario (noiseless or noisy). Results and properties about their convergence are presented and discussed, as well as their complexities.

Since our problem is block-convex, it admits multiple local minima. Our proposal is to tackle this task with alternated proximal descent :

- **Alternated** : The search of a local minimum is performed by minimizing alternatively on the affine model parameters and on the estimated missing data.
- **Proximal** : A term of proximity to the previous iteration is added to the minimization problem to ensure algorithm convergence.
- **Descent** : The loss is strictly decreasing.

The use of proximal descent for this type of problems have been widely addressed in the literature [69, 70] and is the key in our context to ensure parameter convergence.

5.1. Model estimation : ME-step

Provided a complete data set $\mathbf{X} \in \mathbb{R}^{N \times L}$, one of the two main task to be performed is the estimation of the model parameters $(\mathbf{A}, \mathbf{b})$. The optimization problem is deduced by fixing the variable $\mathbf{X}$ in the bi-convex optimization problem. Whether the noiseless data-fitting function problem or its denoising version in problem (9) is preferred, the model estimation step leads to the same optimization problem:

$$\begin{aligned} \mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)} = \underset{\mathbf{A},\mathbf{b}}{\mathbf{argmin}} \; & \lambda \, ||\mathbf{X}^{(k)} - (\mathbf{A}\mathbf{X}^{(k)} + \mathbf{b}\mathbb{1}_L^T)||^2 \\ & + \mu \, ||\mathbf{D}_{\mathcal{G}} \odot \mathbf{A}||^2 \\ & + \nu \, ||\mathbf{A} - \mathbf{A}^{(k)}||^2 \\ \textbf{Subject to :} \; & \mathrm{Diag}(\mathbf{A}) = \mathbb{0} \end{aligned} \tag{18}$$

This optimization problem and can be solved for each distinct pair $(\mathbf{a}^n, b_n)$, for which it is reduced to a problem related to classical linear regression. These problems are convex, well studied, and admit a closed form solution given by the following expression. The proof can be found in appendix Appendix A.

$$\begin{aligned} \mathbf{P}_n &= \tilde{\mathbf{X}}_{\text{-}n} \tilde{\mathbf{X}}_{\text{-}n}^T + \frac{\mu}{\lambda} \mathbf{D}[n] + \frac{\nu}{\lambda} \mathbf{I}_{N-1} \\ \widehat{\mathbf{a}}_{\text{-}n}^n &= \mathbf{P}_n^{-1} \left(\tilde{\mathbf{X}}_{\text{-}n} \tilde{\mathbf{x}}_n^T + \frac{\nu}{\lambda} (\mathbf{a}_{\text{-}n}^n)^{(k)} \right) \\ \hat{a}_n^n &= 0 \\ \hat{b}_n &= \frac{1}{L} \left(\mathbf{x}_n \mathbb{1}_L - \mathbb{1}_L^T \mathbf{X}_{\text{-}n}^T \widehat{\mathbf{a}}_{\text{-}n}^n \right), \end{aligned} \tag{19}$$

where

- $\tilde{\mathbf{X}}$ is the centered matrix $\mathbf{X}$ such that all rows have zero mean

$$\tilde{\mathbf{X}} = \mathbf{X} - \frac{1}{L} \mathbf{X} \, \mathbb{1}_L \mathbb{1}_L^T$$

- $\mathbf{D}[n] \in \mathbb{R}^{N-1 \times N-1}$ is the diagonal matrix defined by

$$\mathbf{D}[n] = \mathrm{diag} \left((\mathbf{D}_{\mathcal{G}})_{-n}^{n} \right)$$

The proof can be found in Appendix A. The associated algorithm is presented in Algorithm (1).

5.2. Signal Imputation : SI-step

Provided the linear model parameters $(\mathbf{A}, \mathbf{b})$, the signal imputation task consists in estimating the data matrix $\mathbf{X}$ that minimizes the objective function (9). The reduced problem differs depending on whether one considers the noiseless or denoising problem. However, in both cases, a closed form solution exists for each graph signal $\mathbf{x}^l$ independently. For the sake of clarity, the l exponent will be

Algorithm 1 Model estimation algorithm

;

Inputs: $\mathbf{X} \in \mathbb{R}^{N\times L}$, $\mathbf{D} \in \mathbb{R}^{N\times N}$, λ, μ
Output: $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{b} \in \mathbb{R}^{N}$
$\mathbf{A} = \mathbb{0}_{N,N}$
$\mathbf{b} = \text{Mean}(\mathbf{X}, \text{axis} = 1)$
$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{b}\mathbb{1}_L^T$
$\mathbf{C} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$
for i = 1..N **do**
  $D_i = \text{Diag}(D[i, -i])$
  $A[i, -i] = (C[-i, -i] + \frac{\lambda}{\mu} D_i)^{-1} C[-i, i]$
end for

removed in the following with $\mathbf{x} = \mathbf{x}^l$. Provided a graph signal $\mathbf{x}$ and its associated measurement $\mathbf{t}$, it can be reordered such that the following structures hold :

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_\mathcal{K} \\ \mathbf{x}_\mathcal{U} \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_{\mathcal{K},\mathcal{K}} & \mathbf{A}_{\mathcal{K},\mathcal{U}} \\ \mathbf{A}_{\mathcal{K},\mathcal{U}} & \mathbf{A}_{\mathcal{U},\mathcal{U}} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_\mathcal{K} \\ \mathbf{b}_\mathcal{U} \end{bmatrix} \tag{20}$$

And the optimization problem for the signal $\mathbf{x}$ is reduced to the following form :

$$\begin{aligned} \mathbf{x}_\mathcal{U}^{(k+1)}, \mathbf{x}_\mathcal{K}^{(k+1)} = \underset{\mathbf{x}_\mathcal{U}, \mathbf{x}_\mathcal{K}}{\text{argmin}} \quad & \varphi(\mathbf{x}_\mathcal{K} - \mathbf{t}_\mathcal{K}) \\ & \lambda \, ||\mathbf{x} - \mathbf{A}\mathbf{x} - \mathbf{b}||^2 \\ & + \nu \, ||\mathbf{x} - \mathbf{x}^{(k)}||^2 \end{aligned} \tag{21}$$

Noiseless case. In the noiseless case, the closed form solution is given by the following expression. The proof can be found in Appendix B.

$$\begin{cases} \mathbf{x}_\mathcal{K}^{(k+1)} = \mathbf{t}_\mathcal{K} \\ \mathbf{x}_\mathcal{U}^{(k+1)} = \mathbf{C}^{-1}(\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3), \end{cases} \tag{22}$$

where

- $\mathbf{C} = \mathbf{A}_{\mathcal{K},\mathcal{U}}^T \mathbf{A}_{\mathcal{K},\mathcal{U}} + (\mathbf{I} - \mathbf{A}_{\mathcal{U},\mathcal{U}})^T (\mathbf{I} - \mathbf{A}_{\mathcal{U},\mathcal{U}}) + \frac{\nu}{\lambda} \mathbf{I}_{|\mathcal{U}|}$
- $\mathbf{v}_1 = \mathbf{A}_{\mathcal{K},\mathcal{U}}^T (\mathbf{b}_\mathcal{K} - (\mathbf{I} - \mathbf{A}_{\mathcal{K},\mathcal{K}}) \, \mathbf{t}_\mathcal{K})$
- $\mathbf{v}_2 = (\mathbf{I} - \mathbf{A}_{\mathcal{U},\mathcal{U}})^T (\mathbf{b}_\mathcal{U} - \mathbf{A}_{\mathcal{U},\mathcal{K}} \, \mathbf{t}_\mathcal{K})$
- $\mathbf{v}_3 = \frac{\nu}{\lambda} \mathbf{x}_\mathcal{U}^{(k)}$.

Denoising case. In the noisy case, the closed form solution is given by

$$\mathbf{x}^{(k+1)} = \mathbf{Q}^{-1} \left(\begin{bmatrix} \mathbf{t}_\mathcal{K} \\ \mathbb{0}_\mathcal{U} \end{bmatrix} - \lambda (\mathbf{I} - \mathbf{A})^T \mathbf{b} + \nu \, \mathbf{x}^{(k)} \right), \tag{23}$$

where

$$\mathbf{Q} = \text{Diag}(\begin{bmatrix} \mathbb{1}_\mathcal{K} \\ \mathbb{0}_\mathcal{U} \end{bmatrix}) + \lambda \, (\mathbf{I} - \mathbf{A})^T (\mathbf{I} - \mathbf{A}) + \nu \, \mathbf{I}_N. \tag{24}$$

As expected, the solutions for the noiseless case coincide with the measurements on known nodes,while the denoising version finds a trade-off between model fit and proximity to the known

values. We note that the missing values estimation for each signal requires the inversion of a $|\mathcal{U}| \times |\mathcal{U}|$ matrix in the noiseless case, and an inversion of a $N \times N$ matrix in the denoising case. However, the inverted matrix is a function of $\mathbf{A}$ and the distribution of missing values for the estimated signal. Therefore the inversion only needs to be performed once per missing profile among the L signals (if several signals have exactly the same set of missing values, only one inversion is required). Moreover, the independence of the interpolation on the different signals could allow to consider distributed solutions if the problem shape is too large. Nevertheless, we will see in the next section that it is possible to relax the SI-step by avoiding the inversion of the large matrices.

5.3. Relaxed approaches for the SI-step

As seen previously, solving the SI-step involves closed form solutions which are computationally cumbersome when working with large graphs. This motivates the introduction of less complex approximated solutions . In other non graph-related signal interpolation techniques, such as the MICE algorithm [71], the fitted model is used to estimate the missing values at step $(k+1)$ from the values at step (k). A similar relaxation can be performed here on the studied problems, lowering the complexity of the imputation step.

The previous consideration reshapes the optimization problem (9) step as follows :

$$\begin{aligned} \mathbf{X}^{(k+1)} = \underset{\mathbf{X}}{\operatorname{argmin}} \quad & \varphi(\mathbf{X}_{\mathcal{K}} - \mathbf{T}_{\mathcal{K}}) \\ & + \lambda \, ||\mathbf{X} - (\mathbf{A}\mathbf{X}^{(k)} + \mathbf{b}\mathbb{1}_L^T)||^2 \\ & + \nu \, ||\mathbf{X} - \mathbf{X}^{(k)}||^2 \end{aligned} \tag{25}$$

Note that the main change here is that $\mathbf{X}$ is replaced by $\mathbf{X}^{(k)}$ in the data and graph terms, which reduces complexity considerably. The following proofs of closed-form solutions can be found in Appendix C.

Noiseless case. In the noiseless case, the closed form solution for the relaxed algorithm is given by:

$$\begin{cases} \mathbf{X}_{\mathcal{K}}^{(k+1)} = \mathbf{T}_{\mathcal{K}} \\ \mathbf{X}_{\mathcal{U}}^{(k+1)} = \frac{1}{\lambda+\nu} \left(\lambda \, (\mathbf{A}\mathbf{X}^{(k)} + \mathbf{b}\mathbb{1}_L^T)_{\mathcal{U}} \; + \; \nu \, \mathbf{X}_{\mathcal{U}}^{(k)} \right) \end{cases} \tag{26}$$

Denoising case. In the noisy case, the closed form solution for the relaxed algorithm is given by:

$$\begin{cases} \mathbf{X}_{\mathcal{K}}^{(k+1)} = \frac{1}{1+\lambda+\nu} \left(\lambda (\mathbf{A}\mathbf{X}^{(k)} + \mathbf{b}\mathbb{1}_L^T)_{\mathcal{K}} + \; \nu \, \mathbf{X}_{\mathcal{K}}^{(k)} + \mathbf{T}_{\mathcal{K}} \right) \\ \mathbf{X}_{\mathcal{U}}^{(k+1)} = \quad \frac{1}{\lambda+\nu} \left(\lambda (\mathbf{A}\mathbf{X}^{(k)} + \mathbf{b}\mathbb{1}_L^T)_{\mathcal{U}} + \; \nu \, \mathbf{X}_{\mathcal{U}}^{(k)} \right) \end{cases} \tag{27}$$

It is interesting to note that the estimation of the unknown values is the same in both the noisy and noiseless cases. These estimations can be interpreted are barycenters between the measurements, the prediction at the previous iteration and the signals at the previous iteration. The weights of this barycenter are the algorithm parameters, which provides an intuitive interpretation of their behaviour. The use of the relaxation only requires the computation of the linear model prediction according to the data at the previous iteration, avoiding complex matrix inversions. This prediction can be computed efficiently as a matrix product. As we will see in Section 7, these relaxed solutions still produce quality estimates, with a notable reduction of the complexity.

5.4. Practical implementation

Our proposal for the search of a local minimum of problem (9) is to perform an alternating optimization descent of the corresponding loss function. Subsection 5.1 describes the optimization step on the pair $(\mathbf{A}, \mathbf{b})$, while the solutions of the different imputation problems (22), (23), (26) and (27) describe four propositions for the imputation step of $\mathbf{X}$.

Algorithm 2 AGPR : Alternated Graph Penalized Regression

Inputs: $\mathbf{T}$, $\mathcal{K}$, $\mathbf{D}$, λ, μ, ν
Output: X^*
$\mathbf{X}^{(0)} = \textbf{Initialize}(\mathbf{T}, \mathcal{K})$
$\mathbf{A}^{(0)} = \mathbb{0}_{N,N}$
$\mathbf{b}^{(0)} = \mathbb{0}_N$
$k = 0$
while not $\textbf{Stop}(k, \mathbf{X}^{(k)}, \mathbf{A}^{(k)}, \mathbf{b}^{(k)}, \mathbf{T}, \mathcal{K})$ **do**
  $\mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)} = \textbf{Model}(\mathbf{X}^{(k)}, \mathbf{A}^{(k)}, \mathbf{b}^{(k)}, \mathbf{D}, \lambda, \mu, \nu)$
  $\mathbf{X}^{(k+1)} = \textbf{Imputation}(\mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)}, \mathbf{X}^{(k)}, \mathbf{T}, \mathcal{K}, \lambda, \nu)$
  $k = k + 1$
end while
$\mathbf{X}^* = \mathbf{X}^{(k)}$

Method	AGPR-NI	AGPR-rNI	AGPR-DI	AGPR-rDI
Data term	$\begin{cases} 0 & \text{if } \mathbf{X}_{\mathcal{K}} = \mathbf{T}_{\mathcal{K}} \\ +\infty & \text{otherwise} \end{cases}$	$\begin{cases} 0 & \text{if } \mathbf{X}_{\mathcal{K}} = \mathbf{T}_{\mathcal{K}} \\ +\infty & \text{otherwise} \end{cases}$	$\|\mathbf{X}_{\mathcal{K}} - \mathbf{T}_{\mathcal{K}}\|^2$	$\|\mathbf{X}_{\mathcal{K}} - \mathbf{T}_{\mathcal{K}}\|^2$
Model term	$\|\mathbf{X} - \mathbf{A}^{(k+1)}\mathbf{X} - \mathbf{b}\|^2$	$\|\mathbf{X} - \mathbf{A}^{(k+1)}\mathbf{X}^{(k)} - \mathbf{b}\|^2$	$\|\mathbf{X} - \mathbf{A}^{(k+1)}\mathbf{X} - \mathbf{b}\|^2$	$\|\mathbf{X} - \mathbf{A}^{(k+1)}\mathbf{X}^{(k)} - \mathbf{b}\|^2$
Complexity of the ME-step	$\mathcal{O}(N^4 + LN^2)$	$\mathcal{O}(N^4 + LN^2)$	$\mathcal{O}(N^4 + LN^2)$	$\mathcal{O}(N^4 + LN^2)$
Complexity of the SI-step	$\mathcal{O}(LN^3)$	$\mathcal{O}(LN^2)$	$\mathcal{O}(LN^3)$	$\mathcal{O}(LN^2)$
Complexity of one complete update	$\mathcal{O}(N^4 + LN^3)$	$\mathcal{O}(N^4 + LN^2)$	$\mathcal{O}(N^4 + LN^3)$	$\mathcal{O}(N^4 + LN^2)$
Theorical convergence of the parameters	Yes	Yes for $\nu > \lambda$	Yes	Yes for $\nu > \lambda$

Table 2: Summary table of the presented algorithms

The general framework of the Alternated Graph Penalized Regression (AGPR) is presented in Algorithm 2. The details of the two variants (NI : noiseless imputation, DI : denoising imputation) and their relaxed counterparts (rNI and rDI) are presented in Table 2.

In the rest of the article, we initialize the missing values to the mean of the known signals on each node. Note that our algorithm assumes that no node and no signal is completely missing. The proximal descent parameter ν is set to 1. Note that a too large value of the ν parameter prevents the descent to converge to a local minimum of the loss function and reduces the effective convergence speed. On the other hand, a too low value of the parameter allows the descent variables to reach distant points, which is usually associated to overfitting behaviours.

5.5. Convergence properties

Property 2. *Convergence of the AGPR-NI and AGPR-DI algorithms*
Assuming $\nu > 0$, the sequence of parameters generated by algorithm 2 with the following imputation step converges.

- *Noiseless imputation (AGPR-NI)*
- *Denoising imputation (AGPR-DI)*

The proof of this property can be found in Appendix D.

6. Results

We now apply the proposed graph signal interpolation algorithms to a wide range of use cases. In order to do so, we consider different datasets, either generated according to classical graph signal

models or based on real data. For each simulation, a random set of values is selected and removed from the signal before the interpolation.

6.1. Benchmark datasets

In order to demonstrate the wide variety of possible interpolation use cases handled by our methods as well as the validity of each of them, we now compare our interpolation methods against state-of-the-art methods on different datasets:

- **Low band.** A set of L = 20 band-limited signal on a random binomial connected graph of N = 200 nodes. Only the 40 first frequencies of the Graph Fourier Transform are active, and their activity is drawn according to a normal centered distribution of unit variance.
- **Cluster low band.** A set of L = 20 band-limited signal on a graph presenting 4 clusters. The graph is generated from a set of N = 200 points in a plane, distributed with normal distribution over 4 clusters each center on a corner of the unit square. The graph is generated using the k-NN procedure with k = 5. Figure 6 displays the constructed graph. The 40 first frequencies of the Graph Fourier Transform are active, and their activity is drawn according to a normal centered distribution of unit variance.
- **High band.** A set of L = 20 signals with the 10 lowest frequencies and the 10 highest frequencies active on a binomial connected graph of N = 200 nodes. The frequency activities are drawn according to a normal centered distribution of unit variance.
- **Low band flipped.** A dataset $\mathbf{T}_{\text{smooth}} \in \mathbb{R}^{200\times 20}$ generated according to the same process as **Low band**. However, for each node, the signal is changed to its opposite with probability 0.5. Namely, the low band flipped dataset is generated by $\mathbf{T}_{\text{flip}} = \text{Diag}(r_1, ..., r_N)\mathbf{T}_{\text{smooth}}$, with $\mathbb{P}(r_i = 1) = \mathbb{P}(r_i = -1) = 0.5$. By construction, note that this signal is not low band.
- **Brittany.** [72] The temperature of N = 31 locations in Brittany, each hour for a month (L = 744). The associated graph is computed by K-nearest-neighbor (k = 5) with the sensors geographic positions. Weights are assigned to the edges through the use of a Gaussian kernel with the geographic distance ($w_{i,j} = exp(d(i,j)^2/\sigma^2)$). The parameter σ^2 was set to the variance of the geographic distances, 0.6.
- **Mocap.** [73] : A 3D times series dataset of N = 44 sensors fixed on a human body while performing tasks (L = 796). The associated graph is obtained by connecting neighboring sensors on the body through a knn procedure with k = 6. Weights are assigned to the edges through the use of a Gaussian kernel with the average 3D distance. The parameter σ^2 was set to the variance of the average 3D distances, 450. The multivariation of the signal is handled by considering a product of the sensor graph with a 3-cycle of weights 1. Figure 7 presents the sensor graph.
- **Walking dog.** This mesh of N = 251 nodes is uniformly distributed on a walking dog for L = 51 frames. The signal is multivariate, composed of the 3 dimensions. The associated graph is obtained by K-nearest-neighbor ($K = 5$) with the sensor average 3D distances. Weights are assigned to the edges through the use of a gaussian kernel with the average 3D distance. The parameter σ^2 was set to the variance of the average 3D distances, 181. The multivariation is handled as in the **Mocap** dataset. Figure 7 presents the sensor graph.

For all the datasets, we randomly select 25% of the nodes, and remove 20% or 50% of their values.

6.2. Methods and evaluation metrics

Several methods are tested and compared to our algorithm :

- **Mean interpolation.** This interpolation estimates each unknown signal by the mean of all known values on this node. It serves as a baseline for the estimation of the interpolation difficulty.

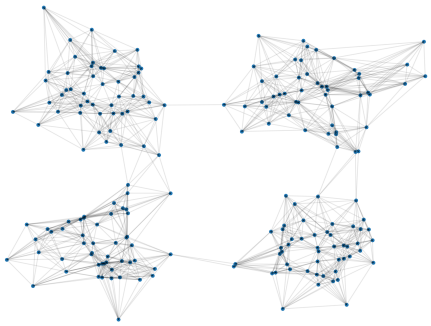


Figure 6: The **Cluster low band** graph.

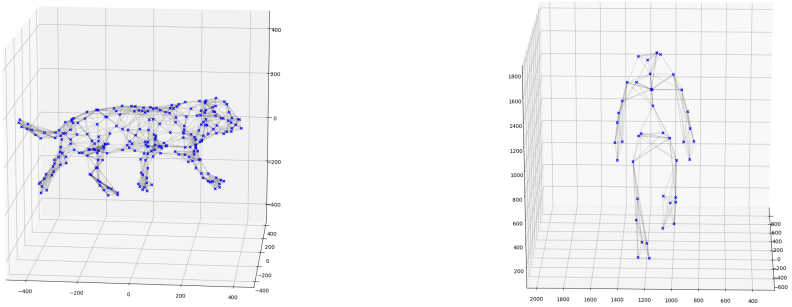


Figure 7: The **walking dog** graph (left) and the **mocap** graph (right)

- **Matrix completion techniques.** Two methods from classical (non graph-related) were implemented. A noiseless matrix completion [48] and a denoising matrix completion [49]. Both methods seek a complete matrix of low rank that fits the known signal values. All parameters have been chosen carefully through K-fold validation with K = 5.

- **Smooth GSP methods.** Two graph signal interpolation methods based on smoothness were implemented. Low band graph signal interpolation [24] and local set based frame reconstruction [28]. In order to give more credit for smooth interpolation methods, especially in the case of non-smooth graph signals, a second graph was learned according to the graph learning procedure described in [26]. The new graph is estimated from the known part of the signals, such that they verify smoothness on the structure. This approach allows to perform an GSP interpolation that takes into account all known signals, rather than interpolating each signal independently.

- **Graph matrix completion.** Two graph matrix completion were implemented. Graph Signal Matrix Completion via Total Variation Minimization [74] and Matrix Completion on Graph [26]. These methods perform the interpolation through rank minimization of the recovered matrix and smoothness of the interpolated signals. In the same approach as GSP methods, both the known Laplacian and a learned Laplacian have been used. All parameters have been chosen carefully through K-fold validation with K = 5.

- **Paired topology and signal inference.** Joint Inference of Signals and Graphs (JISG) [13]

has been implemented. This method learns both the Linear SEM structure and the estimated signals, with a elastic-net penalization of the model weights. All parameters have been chosen carefully through K-fold validation with K = 5.

The presented datasets are very different in term of variance and ease to interpolate. Therefore, in order to compare the quality of the algorithms on the different datasets, an error quantification metric taking these parameters into account must be introduced. In the rest of the manuscript, the quality of a reconstruction will be assessed with the normalized root squared error and its decibel version, defined as follows :

$$\text{nRSE} = \frac{||\mathbf{X}_{\mathcal{U}} - \hat{\mathbf{X}}_{\mathcal{U}}||_2}{||\mathbf{X}_{\mathcal{U}}||_2} \tag{28}$$

$$\text{nRSE-db} = -\log_{10}(\text{nRSE}) \tag{29}$$

6.3. Results

The results are presented in Table 3. For each method and test case, we computed the normalized root squared error (nRSE) of the reconstructions. First of all, we can see that our methods perform quite well on the presented datasets. In particular, AGPR shows the best performance on real world data, compared to the non-regularized and smooth methods. The relaxed methods (AGPR-rNI and AGPR-rDI) perform as well as their non-relaxed counterparts (AGPR-NI and AGPR-DI) but their execution time is significantly reduced.

Low band. As expected, the smooth signals are exactly reconstructed by the method of Narang et al. [24], which is designed for such signals. Yet, we see that our methods perform quite well even for the interpolation of smooth signals . Indeed, the smoothness criterion induces the existence of a good affine relationship with a weight vector localized on close hops, which is retrieved by our methods.

Cluster low band. As low band signals, the smooth signals are exactly reconstructed by the method of Narang et al. [24] as well. Again, our methods are the second best performers. The smoothness of the signal on this graph presenting cluster indicates high correlation of node signals values within each cluster. Here, the localization penalty operates a selection of the nodes, favoring inner-cluster contribution.

High band. For non-smooth signals, our approach obtains the best results. As expected, all smoothness-based approaches perform better with a learned version of the Laplacian, on which the signals are smoother. The regression model is able to address negative correlations between nodes, which naturally occurs in high band signals (activity in high frequencies induces high variations between neighbors). Therefore, the use of locally negative contributions is highly relevant for the reconstruction. These negative correlations cannot be handled by approaches relying on a smoothness assumption, and the selection of local contribution is not performed for JISG [13], leading to deteriorated predictions.

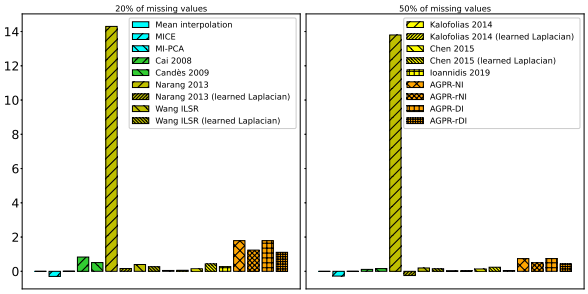
Low band flipped. The proposed methods yield the best results. For a substantial amount of nodes, the strong correlation between neighbors has been flipped into strong anti-correlation, thus rendering smoothness-based method inadequate while the estimation process for our methods remains strictly unchanged. This example demonstrates the wider class of graph processes that can be addressed by our approach.

Brittany. Here, the relaxed algorithms AGPR-rNI and AGPR-rDI obtain the best results. For the temperature graph, the observed signal is smooth in the 3D space since it corresponds to measurements of a continuous physical variable. However, the smoothness characteristic drops when the graph is obtained from an irregular or non-exhaustive sample of the original space. In this case, the constructed graph can be an approximation of the underlying space, while still presenting relevant information about the studied process. By learning the reconstruction weights from the joint information of the graph structure and the known signals, this lack of correlation can be identified, focusing the reconstruction weights on relevant nodes. In other words, the learned data model is less sensitive to the graph construction technique that generated $\mathcal{G}$. Since the loss function is expressed as the sum of

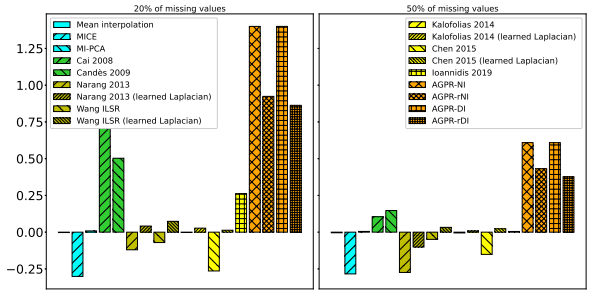
Table 3: nRSE-db table for the presented interpolation methods.

Method	Input laplacian	low band		cluster low band		high band		low band flipped		brittany		mocap		walking dog	
Missing ratio		0.2	0.5	0.2	0.5	0.2	0.5	0.2	0.5	0.2	0.5	0.2	0.5	0.2	0.5
Mean interpolation	x	-9.72e-04	-3.39e-03	-3.90e-03	-3.35e-03	-2.66e-03	-4.20e-03	-1.61e-03	-4.40e-03	1.99e+00	1.99e+00	1.05e-01	1.05e-01	6.78e-01	6.74e-01
MICE [71]		-2.97e-01 (*)	-2.79e-01 (*)	-3.03e-01 (*)	-2.82e-01 (*)	-3.01e-01 (*)	-2.84e-01 (*)	-2.97e-01 (*)	-2.81e-01 (*)	2.33e+00	2.06e+00	1.42e+00 (*)	1.32e+00 (*)	◊	◊
MI-PCA [44]		9.20e-03	2.11e-03	3.19e-03	-5.65e-04	8.69e-03	5.14e-03	9.00e-03	-2.75e-03	2.50e+00	2.47e+00	5.01e-01	4.96e-01	1.01e+00	1.00e+00
Cai 2008 [48]	x	8.34e-01	1.13e-01	8.64e-01	1.10e-01	7.81e-01	1.05e-01	8.24e-01	1.09e-01	1.50e-02	6.72e-03	4.14e-04	1.90e-04	9.19e-03	4.26e-03
Candès 2009 [49]	x	5.09e-01	1.58e-01	5.17e-01	1.54e-01	5.03e-01	1.47e-01	5.17e-01	1.54e-01	1.50e-02	6.72e-03	4.14e-04	1.90e-04	9.19e-03	4.26e-03
Narang 2013 [24]	Known	**1.43e+01**	**1.38e+01**	**1.40e+01**	**1.40e+01**	-1.20e-01	-2.74e-01	-6.87e-02	-2.18e-01	3.38e-16	2.41e-16	-8.49e-01	-4.35e+00	6.39e-02	5.46e-02
	Learned	1.64e-01	-2.47e-01	1.19e-01	-1.51e-01	4.16e-02	-1.01e-01	-5.83e-01	-3.86e-01	-4.82e-16	1.0e+00	7.58e-01	-3.43e+00	8.59e-01	8.54e-01
Yang ILSR [28]	Known	3.98e-01	1.96e-01	3.81e-01	1.85e-01	-7.07e-02	-4.90e-02	-2.30e-02	-1.11e-02	4.82e-17	1.45e-16	1.34e-02	1.15e-02	6.70e-02	4.72e-02
	Learned	2.70e-01	1.49e-01	1.76e-01	1.04e-01	7.37e-02	3.28e-02	1.30e-01	8.19e-02	1.93e-16	4.26e-02	3.06e-01	1.58e-01	5.19e-01	2.47e-01
Kalofolias 2014 [26]	Known	5.01e-02	2.82e-02	3.48e-02	1.79e-02	-2.20e-03	-5.25e-03	1.61e-02	4.03e-03	9.14e-01	9.10e-01	4.28e-02	3.72e-02	1.23e-01	8.91e-02
	Learned	6.02e-02	3.21e-02	4.85e-02	2.35e-02	2.73e-02	8.29e-03	5.87e-02	2.15e-02	8.60e-01	8.77e-01	1.06e-01	8.65e-02	7.11e-01	7.02e-01
Chen 2015 [27]	Known	1.50e-01	1.36e-01	2.23e-02	6.49e-02	-2.64e-01	-1.51e-01	-6.86e-02	-3.75e-02	8.27e-01	8.01e-01	-1.80e-01	-7.61e-02	1.93e-16	4.82e-17
	Learned	4.33e-01	2.39e-01	1.91e-02	1.09e-01	1.32e-02	2.27e-02	2.01e-01	1.06e-01	7.29e-01	7.36e-01	8.10e-01	7.30e-01	8.41e-01	7.84e-01
Ioannidis 2019 [13]	x	2.74e-01	4.46e-02	2.42e-01	2.06e-02	2.62e-01	4.83e-03	2.96e-01	5.48e-02	2.47e+00	2.45e+00	1.03e+00	9.54e-01	1.03e+00	9.59e-01
AGPR-NI	Known	1.79e+00	7.41e-01	2.25e+00	1.02e+00	**1.40e+00**	**6.09e-01**	**1.74e+00**	**7.98e-01**	2.62e+00	2.56e+00	**2.25e+00**	**1.40e+00**	1.67e+00	5.91e-01
AGPR-rNI	Known	1.24e+00	5.04e-01	1.97e+00	6.34e-01	9.24e-01	4.31e-01	1.19e+00	4.94e-01	**2.66e+00**	**2.59e+00**	2.00e+00	**1.41e+00**	**1.70e+00**	**1.42e+00**
AGPR-DI	Known	1.79e+00	7.41e-01	2.25e+00	1.02e+00	**1.40e+00**	**6.09e-01**	**1.74e+00**	**7.98e-01**	2.62e+00	2.56e+00	**2.25e+00**	**1.40e+00**	1.67e+00	5.91e-01
AGPR-rDI	Known	1.11e+00	4.31e-01	1.96e+00	6.28e-01	8.63e-01	3.77e-01	1.06e+00	4.22e-01	**2.66e+00**	**2.58e+00**	1.80e+00	1.12e+00	**1.72e+00**	**1.39e+00**

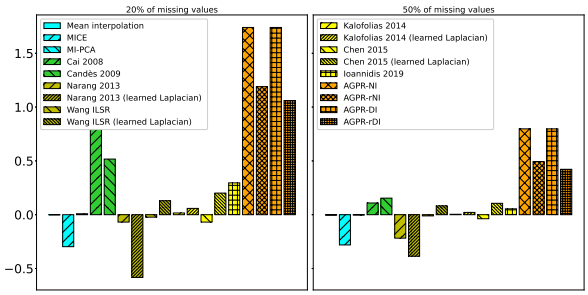
(*) : methods that did not interpolate all the missing values. The nRSE-db is calculated only from the interpolated values.
(◊) : methods that could not be computed in reasonable time (> 10h).



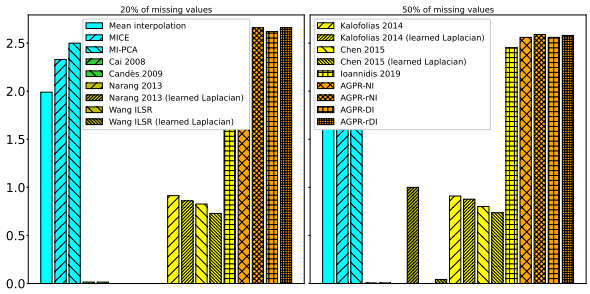
(a) nRSE-db for the **low band** dataset.



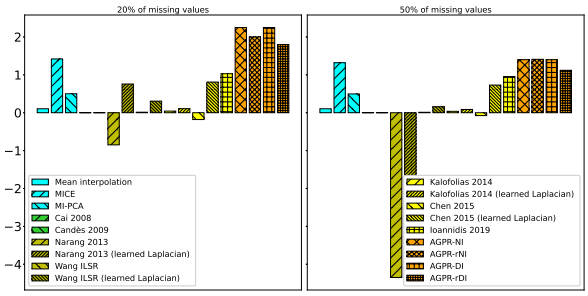
(b) nRSE-db for the **high band** dataset.



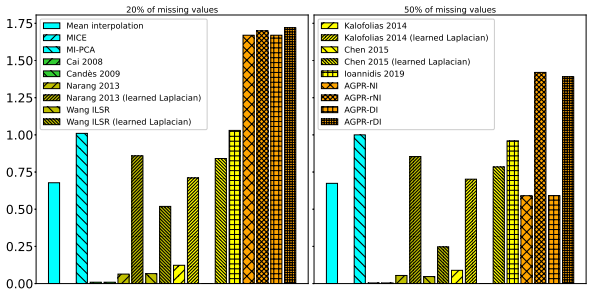
(c) nRSE-db for the **low band flipped** dataset.



(d) nRSE-db for the **brittany** dataset.



(e) nRSE-db for the **mocap** dataset.



(f) nRSE-db for the **walking dog** dataset.

Figure 8: Bar-plots of the nRSE-db for the presented interpolation methods.

the reconstruction error and localization, a natural trade-off occurs, allowing the model to search for information at more distant nodes if it means improving the quality of the reconstruction.
Mocap. Our approach obtains the best results. Thanks to the non-smooth framework, the multivariate setting is handled correctly by our method. The signals of each of the 3 dimensions (respectively X, Y and Z) are deeply correlated, therefore the values on the different components of each node carry information on other dimension even though these values may lie in different ranges.

7. Discussion

7.1. Parameters

In this section, we investigate the stability of the proposed algorithms with regards to the algorithm and problem parameters. Three different types of parameters can be identified :

Algorithm hyper-parameters.

- Model parameter λ : this parameter controls the adequacy of the estimated values with the learned affine model.
- Graph parameter μ : this parameter penalizes the proximity of the reconstruction weights on the graph structure. As this parameter increases, the interpolation weights of the linear model tends to be closer to the reconstructed nodes.

Problem parameters.

- The ratio of missing values r_m defined as

$$r_m = \frac{|\mathcal{U}|}{NL}.$$

  Note that this ratio alone does not represent the complexity induced by the profile of missing values.

- The noise ratio r_n defined as

$$r_n = \sigma_{\text{measure}}/\sigma_X,$$

  where σ_{measure} is the noise level defined in (2) and $\sigma_{\mathbf{X}}$ the standard deviations of $\mathbf{X}$. In this article, it is assumed that the known entries may differ from the values that fit the underlying linear model. The quality of this fit with respects to the overall signal is here quantified by the presented ratio.

- The ratio of problem dimensions r_d defined as

$$r_d = \frac{N}{L}.$$

  Indeed, the interpolation task strongly differs whether we consider the case of a few signals on a large network or a large set of signals on a small network.

7.2. Experimental setup

In order to study the stability of the proposed algorithm with respect to the presented parameters, we choose a specific dataset generation procedure that allows a consistent comparison and parameter investigation.

1. The graph $\mathcal{G}$ is composed of $N = 100$ points uniformly sampled in a 2D square, connected through 5-nearest neighbors.
2. A set of $L = 100$ signals are generated using the **Low Band Flipped** method ($r_d = 1$).

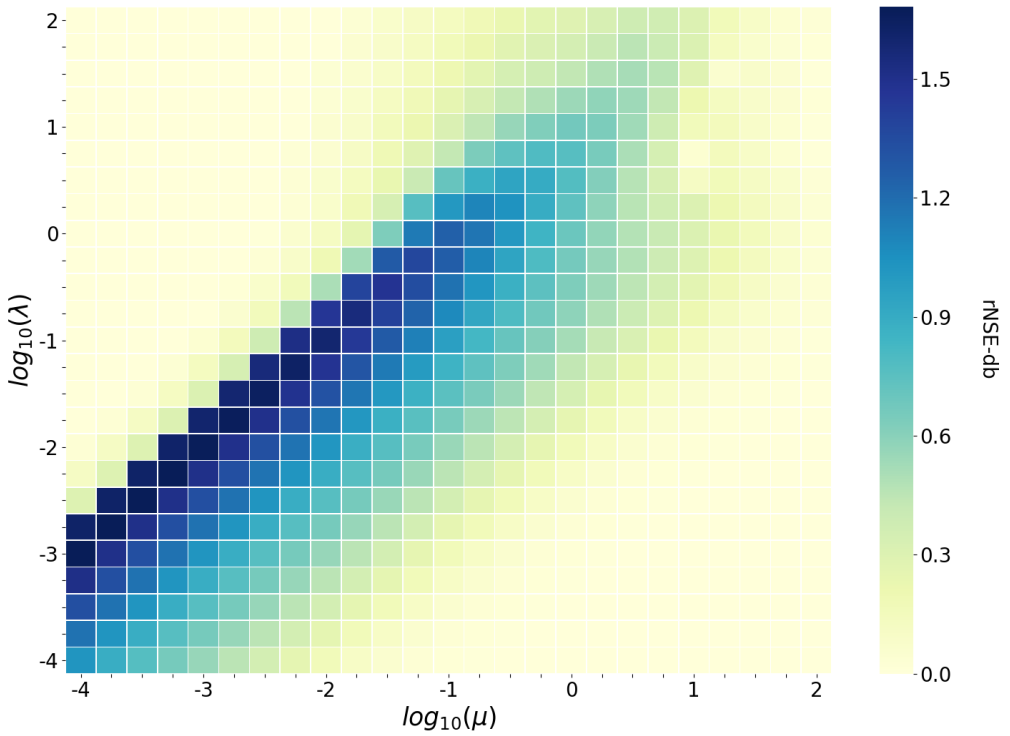


Figure 9: Heatmap of the average nRSE-db on the baseline dataset as a function of λ and μ

3. 20% of the values are missing ($r_m = 0.2$).
4. No noise is added to the signal ($r_n = 0$).
5. The optimization parameters are set to $\lambda = 10^{-1}$ and $\mu = 10^{-2}$.

Then, from this baseline configuration, we generate similar datasets by fixing all parameters but one, in order to investigate the influence of this parameter on the interpolation results.

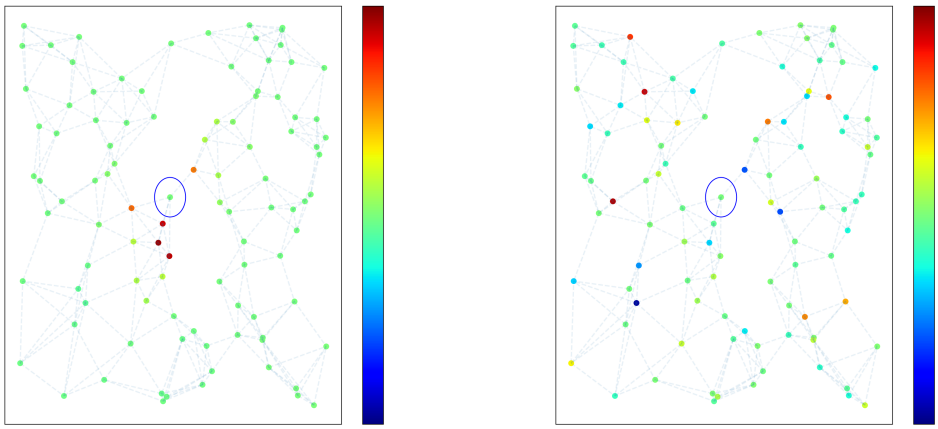
7.3. Influence of the algorithm hyper-parameters λ and μ

In order to demonstrate the influence of parameter hyper optimization, this section present a study of the interpolation quality function of λ and μ.

Figure 9 presents the heatmap of nRSE-db of the AGPR-NI algorithm interpolation for different values of the parameters λ and μ. The experiment shows two main behaviours.

First, the interpolation becomes inefficient when any of the parameter becomes either too large or too insignificant. A too large value of the hyper-parameter λ leads to an over-fitted linear model which is not reliable for the prediction of unknown values, while a too small value makes the model inefficient. Regarding the value of the hyper-parameter μ, its scale controls the strength of the graph-based penalization, and a non-relevant scale can either cancel the use of the graph in the interpolation, or drag the linear model parameters towards 0.

Second, we observe that the ratio of the parameters λ/μ plays an key role in the interpolation quality. Indeed, we expect the estimated values to be close to the known ones, i.e. the data fitting term in the optimization problem will be especially low. In this case, only the ratio of parameters λ/μ is decisive in the optimization problem. This ratio quantifies how much the proximity of the weights localization prevails over the respect of the linear model. This allows to reduce the complexity of the



(a) Optimal parameters $\lambda/\mu = \lambda^*/\mu^*$

(b) Sub-optimal parameters $\lambda/\mu = 100 \times \lambda^*/\mu^*$

Figure 10: Learned weights reconstructing the center node (circled) on the baseline dataset.

cross validation by only exploring different values of this ratio rather than performing a complete grid search of the optimal parameters, as long as the data fitting term prevails on the others.

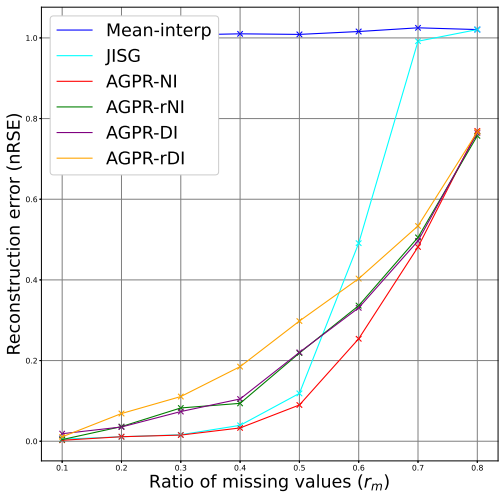
The effect of a too large value of the ratio λ/μ is displayed in figure 10. This figure presents the reconstruction weights $\mathbf{a}_i$ involved in the interpolation of the circled node for different values of the ratio λ/μ. Figures 10a and 10b display the reconstruction vector learned for respectively the optimal parameter ratio ($\lambda/\mu = \lambda^*/\mu^*$) and a sub-optimal parameter ratio ($\lambda/\mu = 100 \times \lambda^*/\mu^*$) . As the visual illustrates, if the ratio λ/μ becomes too large, the linear reconstruction term in (9) prevails on the localisation penalty (14). Thus, the optimization problem allows for large contributions from a wide set of distant nodes in order to achieve an improved linear reconstruction, which cancels the desired localization property.

7.4. Influence of the problem parameters r_m, r_n and r_d

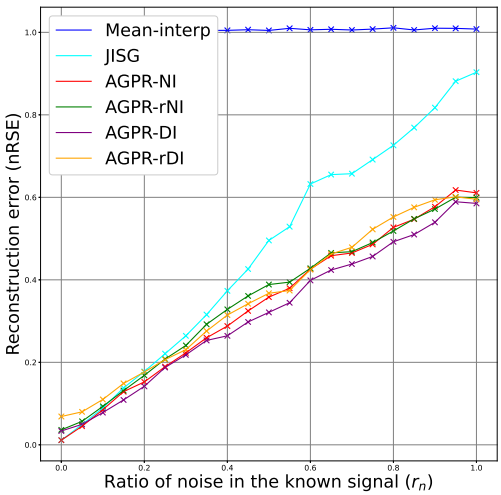
Figure 11 presents the nRSE of the four proposed algorithms (**NI**, **rNI**, **DI**, and **rDI**) as a function of the problem parameters r_m, r_n and r_d.

The ratio of missing values r_m is a key parameter of the interpolation problem. By studying the results of our methods when the count of missing values increases (Figure 11a), we see that the exact methods **NI** show better results than the other methods. Moreover, the noiseless methods are more accurate because the experiments are performed with a noise ratio $r_n = 0$. We remark that even for large values of missing ratio (from 40% to 80%), our methods present interesting performances, especially when compared to (JISG). When considering noiseless measurements, it seems natural that the denoising method will perform poorer than its noiseless counterpart. However, the strict distinction between the two variants is diffucult to understand. In fact, the noiseless problem can be assimilated to the limit of the denoising method when the parameters λ and μ tend to zero, with a constant ratio λ/μ.

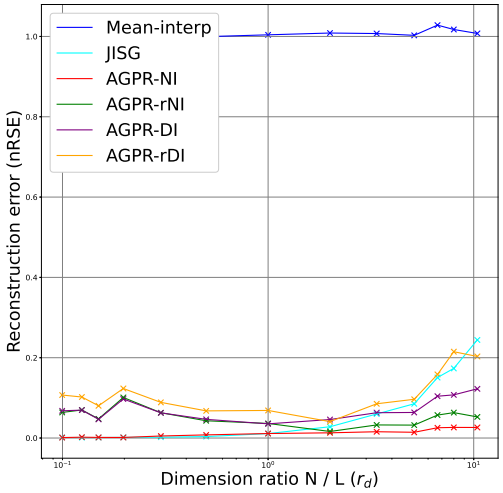
The second experiment in Figure 11b presents the evolution of the reconstruction errors as a function of the noise ratio r_n in the measurements. As expected, the noiseless method is the best performer when the noise ratio is low, especially below 10%. As the noise ratio increases above this threshold, the denoising algorithm **AGPR-DI** present better performances than its noiseless and relaxed counterpart. For significant noise in the measurements, the closed form solutions involved in the non-relaxed methods are more likely to find irrelevant local minimas of the objective function,



(a) nRSE of the AGPR methods for different missing ratio r_m on the same dataset



(b) nRSE of the AGPR methods for different noise ratio r_n on the same dataset



(c) nRSE of the AGPR methods for different dimension ratio r_d

Figure 11: Stability of the proposed methods with respect to the problem parameters r_m, r_n and r_d

resulting in an over-fitting state. As the figure illustrates, it appears that in the presence of high noise the relaxed methods that are updated using approximation rules are less robust than their non-relaxed counterparts.

The effect of varying the dimension ratio r_d is presented in Figure 11c. First, we see that for $N < L$ (i.e. more signal than nodes) and without noise in the measurements, the **AGPR-NI** and **JISG** methods are the best performers. In these conditions, the abundance of signals allow the methods to recover the true nodal correlations and therefore the signals topology. In fact, we were able to check that for the data points with $r_d < 0.5$, the condition for exact topology estimation presented in [13] was verified, justifying the results. The drop in estimation quality near the region $r_d = 10$ is due to the over-fitting behaviour that appears when the count of signals is too narrow for a good selection among the large set of nodes available for the interpolation. Our method therefore avoids the use of non-relevant correlations in the signals for the data model estimation. In this scenario, the condition for exact reconstruction presented in [13] becomes difficult to satisfy.

7.5. Influence of the graph distance

In order to illustrate the suitability of the proposed algorithms, we compare the performances of the **AGPR-NI** algorithm with five different graph distances. Figure 12 presents the interpolation quality with the different distances regarding the same problem parameters defined in Section 7.1. The computation of the different graph distances are detailed in Section 4.5:

- The null distance d_{null}
- The uniform distance d_{unif}
- The affinity approximation distance d_{aff}
- The affinity geodesic distance d_{geo}
- The laplacian eigenmaps distance d_{eig}

As a baseline comparison, the analysis is performed against Joint Inference of Signals and Graphs (**JISG**) [13].

While this difference in interpolation quality between the graph distances and the uniform distance is relatively low for small values of r_m, it becomes significant for more than 30%.

Figure 12a presents the performances of our algorithm with the different distances as a function of r_m. As we can see, the performances are poorer for the null distance and the affinity approximation distance. Our method performed with the uniform distance and **JISG** display similar results, with better results for **JISG**. This demonstrates that the elastic-net penalization is a more relevant regularizer than the classical ridge regularizer. As the ratio of missing values increases, the methods using the geodesic distance and the laplacian eigenmaps distance becomes the best performers, illustrating that for this specific graph process, the information provided by the graph is an important asset for the signal interpolation. For small amount of missing data, the linear fit is able to recover the localisation in the reconstruction weights directly from the high correlations / anti-correlations between the signals. This behaviour fades when too few known values are used to estimate these correlations, and the contribution of the graph localisation penalty becomes a key asset in recovering relevant weights. Moreover, both graph distances (geodesic and Laplacian) seem perform similarly, although the Laplacian eigenmaps distance shows slightly poorer results. This last difference is due to the fact that this distance is only an approximation of the exact distance along the graph structure.

Figure 12b shows the influence of parameter r_n: we notice that for most noise ratios, the geodesic distance and the Laplacian eigenmaps distance are the best performers (except for $r_n = 0$), illustrating the benefits of the graph knowledge in this interpolation task. The results shown by the eigenmaps distance are slightly inferior to the geodesic interpolation, which is due to the eigenmaps distance being assailable to an approximation of the graph distance, slightly misleading the signal estimation. As the noise level increases, all the performances become degraded. However, high **nRSE** are reached

sooner by the non graph-related methods. Classical linear and ridge regression allow all the nodes to contribute equally in the reconstruction. Therefore they are more likely to find random co-variations , while undermining the high local co-variations that are relevant in this specific graph process. Here, the graph penalty prevents such a behaviour, favouring among all possible regressions the ones respecting the locality criterion.

Figure 12c presents the stability of interpolation method with the different graph distances for different dimension ratio r_d. We notice that for problem dimensions such that $L >> N$, the information provided by the graph distance is not relevant for the interpolation. Indeed, the locality of the reconstruction weights, which is a relevant criterion for the model, results in strong correlations in the signal values. This correlation can be retrieved when we dispose of a sufficient set of signals. However, when the ration r_d increases, the problem meets over-fitting issues when considering a linear, ridge or elastic-net regression. Therefore, the information brought by the graph becomes highly relevant, improving the interpolation.

8. Conclusion and future works

In this paper, we began by presenting a literature review under an unifying formalism that allows for a relevant understanding and comparison of the key differences between classical graph signal interpolation techniques, most of which are based on the signal smoothness assumption. In order to use the information provided by a graph in an interpolation task without this restrictive assumption, we proposed a penalization that enforces locality of the reconstruction weights of a linear structural equation model.

We defined a new graph signal interpolation task as an optimization problem and provided a general solution by using the alternating minimization method. The effectiveness of the proposed method was demonstrated on synthetic data as well as on real network data, and compared to the graph signal interpolation methods presented in the review. The stability and reliability of the proposed methods were studied over a wide range of problem parameter, providing guidelines for the use of the method on real world applications.

Future works include the consideration of a L_1 graph localization penalization in the interpolation process (similarly to [13]), which will encourage sparsity in the reconstruction weights through a LASSO-like regression. An other relevant approach related to the presented algorithms is the use a generalized linear model to model the data dependency, introducing non-linearity in the relationships between the nodes.

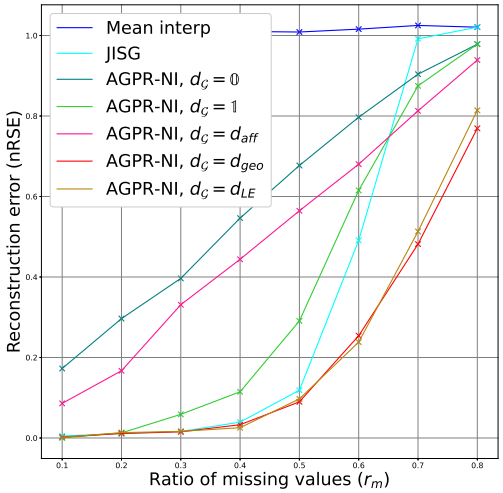
References

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

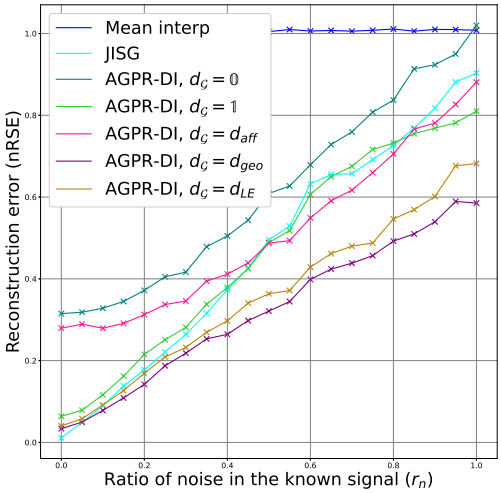
[2] A. Ortega, P. Frossard, J. Kovačević, J. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, pp. 808–828, 2018.

[3] J. A. Deri and J. M. Moura, "New york city taxi analysis with graph signal processing," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2016, pp. 1275–1279.

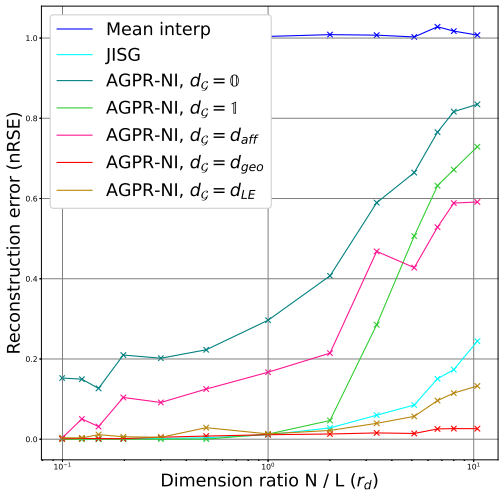
[4] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 2018.



(a) nRSE of the AGPR-NI method for different distances and missing ratio r_m



(b) nRSE of the AGPR-NI method for different distances and noise ratio r_n



(c) nRSE of the AGPR-NI method for different distances and dimension ratio r_d

Figure 12: Stability of the AGPR-NI with different graph distances

[5] B. Girault, "Stationary graph signals using an isometric graph translation," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 1516–1520.

[6] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Laplacian matrix learning for smooth graph signal representation," in *2015 IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 3736–3740.

[7] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 817–829, 2017.

[8] A. E. C. Redondi, "Radio map interpolation using graph signal processing," *IEEE Communications Letters*, vol. 22, no. 1, pp. 153–156, Jan 2018.

[9] W. Huang, T. A. Bolton, J. D. Medaglia, D. S. Bassett, A. Ribeiro, and D. Van De Ville, "A graph signal processing perspective on functional brain imaging," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 868–885, 2018.

[10] K. G. Oweiss, *Statistical signal processing for neuroscience and neurotechnology*. Academic Press, 2010.

[11] L. Goldsberry, W. Huang, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Brain signal analytics from graph signal processing perspective," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 851–855.

[12] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 772–10 781.

[13] V. N. Ioannidis, Y. Shen, and G. B. Giannakis, "Semi-blind inference of topologies and dynamical processes over dynamic graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2263–2274, 2019.

[14] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *Proceedings of the IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6510–6523, 2015.

[15] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Proceedings of the Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 446–475, 2018.

[16] S. Chen, A. Sandryhaila, and J. Kovačević, "Sampling theory for graph signals," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 3392–3396.

[17] D. E. Olivier Tzamarias, P. Akyazi, and P. Frossard, "A novel method for sampling bandlimited graph signals," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 126–130.

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

[19] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Reconstruction of graph signals through percolation from seeding nodes," *IEEE Transactions on Signal Processing*, vol. 64, no. 16, pp. 4363–4378, 2016.

[20] Y. M. Lu and M. Vetterli, "Distributed spatio-temporal sampling of diffusion fields from sparse instantaneous sources," in *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2009, pp. 205–208.

[21] J. Ranieri, A. Chebira, Y. M. Lu, and M. Vetterli, "Sampling and reconstructing diffusion fields with localized sources," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4016–4019.

[22] S. Reddy and S. P. Chepuri, "Sampling and reconstruction of diffusive fields on graphs," *arXiv preprint arXiv:1907.00892*, 2019.

[23] S. P. Chepuri and G. Leus, "Graph sampling for covariance estimation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 451–466, 2017.

[24] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 5445–5449.

[25] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 491–494.

[26] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs," in *Neural Information Processing Systems 2014, Workshop" Out of the Box: Robustness in High Dimension"*, 2014.

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

[28] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2432–2444, May 2015.

[29] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.

[30] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 764–778, Feb 2017.

[31] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, 2016.

[32] M. B. Mashhadi, M. Fallah, and F. Marvasti, "Interpolation of sparse graph signals by sequential adaptive thresholds," in *2017 International Conference on Sampling Theory and Applications (SampTA)*, July 2017, pp. 266–270.

[33] A. Heimowitz and Y. C. Eldar, "The nystrom extension for signals defined on a graph," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4199–4203.

[34] H. Cai, V. Zheng, and K. Chen-Chuan Chang, "A comprehensive survey of graph embedding: Problems, techniques and applications," *Computing Research Repository (CoRR)*, vol. abs/1709.07604, 2017.

[35] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53–94, 2006.

[36] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[37] M. Gavish, B. Nadler, and R. R. Coifman, "Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning." in *ICML*, 2010, pp. 367–374.

[38] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4117–4131, 2017.

[39] N. Tremblay, P. Gonçalves, and P. Borgnat, "Design of graph filters and filterbanks," in *Cooperative and Graph Signal Processing*. Elsevier, 2018, pp. 299–324.

[40] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 787–807, 2018.

[41] H.-T. Wai, A. Scaglione, B. Barzel, and A. Leshem, "Joint network topology and dynamics recovery from perturbed stationary points," *IEEE Transactions on Signal Processing*, vol. 67, no. 17, pp. 4582–4596, 2019.

[42] S. v. Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in r," *Journal of statistical software*, pp. 1–68, 2010.

[43] J. Josse, J. Pagès, and F. Husson, "Multiple imputation in principal component analysis," *Advances in data analysis and classification*, vol. 5, no. 3, pp. 231–246, 2011.

[44] J. Josse, F. Husson *et al.*, "missmda: a package for handling missing values in multivariate data analysis," *Journal of Statistical Software*, vol. 70, no. 1, pp. 1–31, 2016.

[45] S. Dray and J. Josse, "Principal component analysis with missing values: a comparative survey of methods," *Plant Ecology*, vol. 216, no. 5, pp. 657–667, 2015.

[46] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[47] R. Vidal, Y. Ma, and S. S. Sastry, "Principal component analysis," in *Generalized principal component analysis*. Springer, 2016, pp. 25–62.

[48] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[49] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[50] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[51] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems," *Big Data Mining and Analytics*, vol. 1, no. 4, pp. 308–323, 2018.

[52] L. T. Nguyen, J. Kim, and B. Shim, "Low-rank matrix completion: A contemporary survey," *IEEE Access*, vol. 7, pp. 94 215–94 237, 2019.

[53] A. Agaskar and Y. M. Lu, "A spectral graph uncertainty principle," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.

[54] B. Pasdeloup, R. Alami, V. Gripon, and M. Rabbat, "Toward an uncertainty principle for weighted graphs," in *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 1496–1500.

[55] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *Nips*, vol. 14, no. 14, 2001, pp. 585–591.

[56] ——, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in neural information processing systems*, 2002, pp. 585–591.

[57] T. Bonald, A. Hollocou, and M. Lelarge, "Weighted spectral embedding of graphs," in *Allerton Conference on Communication, Control, and Computing*, Oct 2018, pp. 494–501.

[58] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.

[59] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of operations research*, vol. 23, no. 4, pp. 769–805, 1998.

[60] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[61] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[62] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[63] ——, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[64] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[65] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, 2009, pp. 1033–1040.

[66] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Mathematical methods of operations research*, vol. 66, no. 3, pp. 373–407, 2007.

[67] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[68] M. Nikolova and P. Tan, "Alternating structure-adapted proximal gradient descent for nonconvex block-regularised problems," 2018.

[69] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[70] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, 2013.

[71] S. Buuren and C. Groothuis-Oudshoorn, "Mice: Multivariate imputation by chained equations in r," *Journal of Statistical Software*, vol. 45, 12 2011.

[72] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *Proceedings of the IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 481–496, Sep. 2018.

[73] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05," 2007, http://mocap.cs.cmu.edu/.

[74] S. Chen, A. Sandryhaila, J. Moura, and J. Kovacevic, "Signal recovery on graphs: Variation minimization," *Proceedings of the IEEE Transactions on Signal Processing*, vol. 63, 09 2015.

Appendix A. Closed form solutions for model estimation step

Provided $\mathbf{X} = \mathbf{X}^{(k)}$, $\mathbf{D}_\mathcal{G}$ and $\mathbf{A}^{(k)}$, the solution of (18) can be expressed row-wise as follows :

$$\begin{aligned}\{\hat{\mathbf{a}}^n\}, \{\hat{b}_n\} = \underset{\{\mathbf{a}^n\},\{b_n\}}{\mathbf{argmin}}\ & \lambda \sum_{n=1}^{N} ||\mathbf{x}_n - \mathbf{X}^T\mathbf{a}^n - b_n\mathbb{1}_L||^2 \\ & +\mu \sum_{n=1}^{N} ||(\mathbf{D}^\mathcal{G})^n \odot \mathbf{a}^n||^2 \\ & +\nu \sum_{n=1}^{N} ||\mathbf{a}^n - \mathbf{a}^{n,(k)}||^2 \\ \textbf{Subject to :}\ & \{\mathbf{a}_n^n = 0\}\end{aligned} \tag{A.1}$$

with the following notations introduced in subsection 5.1. In the presented sums, each term depends of only one of the rows a^n. Therefore, the optimization can be performed independently with regards to each row. The optimization problem for any row a^n becomes :

$$\begin{aligned}\hat{\mathbf{a}}^n, \hat{b}_n = \underset{\mathbf{a}^n, b_n}{\mathbf{argmin}}\ & \lambda\, ||\mathbf{x}_n - \mathbf{X}^T\mathbf{a}^n - b_n\mathbb{1}_L||^2 \\ & +\mu\, ||(\mathbf{D}^\mathcal{G})^n \odot \mathbf{a}^n||^2 \\ & +\nu\, ||\mathbf{a}^n - \mathbf{a}^{n,(k)}||^2 \\ \textbf{Subject to :}\ & a_n^n = 0\end{aligned} \tag{A.2}$$

The condition $a_n^n = 0$ can be dropped by considering only the unconstrained part of the row $\mathbf{a}^n$ (i.e $\mathbf{a}^n_{-n}$), and the penalty loss $||(\mathbf{D}^\mathcal{G})^n \odot \mathbf{a}^n||^2$ can be rewritten as a matrix product including the diagonal matrix $\mathbf{D}[n]$ introduced in the solution.

$$\begin{aligned}\hat{\mathbf{a}}^n_{-n}, \hat{b}_n = \underset{\mathbf{a}^{-n}_{-n}, b_n}{\mathbf{argmin}}\ & \lambda\, ||\mathbf{x}_n - (\mathbf{X}_{-n})^T\mathbf{a}^n_{-n} - b_n\mathbb{1}_L||^2 \\ & +\mu\, (\mathbf{a}^n_{-n})^T\, \mathbf{D}[n]\, \mathbf{a}^n_{-n} \\ & +\nu\, ||\mathbf{a}^n_{-n} - \mathbf{a}^{n,(k)}_{-n}||^2\end{aligned} \tag{A.3}$$

The minimized function is strictly convex with regard to $(\mathbf{a}^n_{-n}, b_n)$ (for $\eta > 0$). The optimal solution satisfies the following first-order partial derivative equations :

$$\begin{aligned}0 &= \lambda\, \mathbf{X}_{-n}(\mathbf{X}^T_{-n}\mathbf{a}^n_{-n} + b_n\mathbb{1}_L - \mathbf{x}_n)\ +\ \mu\, \mathbf{D}[n]\, \mathbf{a}^n_{-n}\ +\ \nu\, (\mathbf{a}^n_{-n} - \mathbf{a}^{n,(k)}_{-n}) \\ 0 &= \lambda\, \mathbb{1}^T_L(\mathbf{x}_n - \mathbf{X}^T_{-n}\mathbf{a}^n_{-n} - \hat{b}_n\mathbb{1}_L)\end{aligned}$$

Which lead to the closed form solution proposed in subsection 5.1.

Appendix B. Closed form expression of S.I. step

Using the notations introduced in (C.1), the known part of any signal is constrained to be the known values, therefore inducing $\mathbf{x}_{\mathcal{K}}^{(k+1)} = \mathbf{t}_{\mathcal{K}}$. Replacing $\mathbf{x}_{\mathcal{K}}$ by this value, the optimization problem for the noiseless imputation is performed on $\mathbf{x}_{\mathcal{U}}$ step can be rewritten as follows :

$$\begin{aligned}\mathbf{x}_{\mathcal{U}}^{(k+1)} = \underset{\mathbf{x}_{\mathcal{U}}}{\operatorname{argmin}} \quad & \lambda \, || \begin{bmatrix} \mathbf{t}_{\mathcal{K}} \\ \mathbf{x}_{\mathcal{U}} \end{bmatrix} - \mathbf{A} \begin{bmatrix} \mathbf{t}_{\mathcal{K}} \\ \mathbf{x}_{\mathcal{U}} \end{bmatrix} - \mathbf{b}||^2 \\ & + \nu \, || \begin{bmatrix} \mathbf{t}_{\mathcal{K}} \\ \mathbf{x}_{\mathcal{U}} \end{bmatrix} - \mathbf{x}^{(k)}||^2 \end{aligned} \tag{B.1}$$

The block matrix multiplication with $\mathbf{A}$ can be performed, and the squared L_2 norms can be separated.

$$\begin{aligned}\mathbf{x}_{\mathcal{U}}^{(k+1)} = \underset{\mathbf{x}_{\mathcal{U}}}{\operatorname{argmin}} \quad & \lambda \, ||\mathbf{t}_{\mathcal{K}} - \mathbf{A}_{\mathcal{K},\mathcal{K}}\mathbf{t}_{\mathcal{K}} - \mathbf{A}_{\mathcal{K},\mathcal{U}}\mathbf{x}_{\mathcal{U}} - \mathbf{b}_{\mathcal{K}}||^2 \\ & + \lambda \, ||\mathbf{x}_{\mathcal{U}} - \mathbf{A}_{\mathcal{U},\mathcal{K}}\mathbf{t}_{\mathcal{K}} - \mathbf{A}_{\mathcal{U},\mathcal{U}}\mathbf{x}_{\mathcal{U}} - \mathbf{b}_{\mathcal{U}}||^2 \\ & + \nu \, ||\mathbf{x}_{\mathcal{U}} - \mathbf{x}_{\mathcal{U}}^{(k)}||^2 \end{aligned} \tag{B.2}$$

This problem is strongly convex for $\nu > 0$, and the closed form solution presented in (22) is obtained by solving the first order differential condition.

Appendix C. Closed form expression of relaxed S.I. step

Using the notations introduced in (C.1), the optimization problem for the denoising imputation is performed step can be rewritten as follows :

$$\begin{aligned}\mathbf{x}^{(k+1)} = \underset{\mathbf{x}_{\mathcal{U}},\, \mathbf{x}_{\mathcal{K}}}{\operatorname{argmin}} \quad & ||\mathrm{Diag}(\begin{bmatrix} \mathbb{1}_{\mathcal{K}} \\ \mathbb{0}_{\mathcal{U}} \end{bmatrix})(\mathbf{x} - \mathbf{t})||^2 \\ & \lambda \, ||\mathbf{x} - \mathbf{A}\mathbf{x} - \mathbf{b}||^2 \\ & + \nu \, ||\mathbf{x} - \mathbf{x}^{(k)}||^2 \end{aligned} \tag{C.1}$$

This problem is strongly convex for $\nu > 0$, and the closed form solution presented in (23) is obtained by solving the first order differential condition.

Appendix D. Proof of global convergences of exact descent

Let us introduce the optimized loss

$$f(\, \mathbf{X}, \mathbf{A}, \mathbf{b}) = \varphi(\mathbf{X}_{\mathcal{K}} - \mathbf{T}_{\mathcal{K}}) + \lambda \, ||\mathbf{X} - \mathbf{A}\mathbf{X} - \mathbf{b}\mathbb{1}^T||^2 \, + \mu \, \mathrm{Loc}(\mathbf{A})$$

The sequence $(\mathbf{X}^{(k)}, \mathbf{A}^{(k)}, \mathbf{b}^{(k)})$ is constructed as follows :

$$\mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)} = \underset{\mathbf{A},\mathbf{b}}{\operatorname{argmin}} \; f(\, \mathbf{X}^{(k)}, \mathbf{A}, \mathbf{b}) \; + \; \nu ||\mathbf{A} - \mathbf{A}^{(k)}||^2 \tag{D.1}$$

$$\mathbf{X}^{(k+1)} = \underset{\mathbf{X}}{\operatorname{argmin}} \; f(\, \mathbf{X}, \mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)}) \; + \; \nu ||\mathbf{X} - \mathbf{X}^{(k)}||^2 \tag{D.2}$$

In equation (D.1), the set of feasible $\mathbf{A}$ and $\mathbf{b}$ includes the parameters at previous iteration $\mathbf{A}^{(k)}$ and $\mathbf{b}^{(k)}$. Therefore, the achieved minimum at $\mathbf{A}^{(k+1)}$ and $\mathbf{b}^{(k+1)}$ is necessarily lower than the values at $\mathbf{A}^{(k)}$ and $\mathbf{b}^{(k)}$. By plugging the values at iteration (k) and $(k+1)$, we get the following inequality :

$$f(\mathbf{X}^{(k)}, \mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)}) + \nu||\mathbf{A}^{(k+1)} - \mathbf{A}^{(k)}||^2 \leq f(\mathbf{X}^{(k)}, \mathbf{A}^{(k)}, \mathbf{b}^{(k)})$$

Using a similar rationale on (D.2) with $\mathbf{X}^{(k)}$ and $\mathbf{X}^{(k+1)}$, we obtain :

$$f(\mathbf{X}^{(k+1)}, \mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)}) + \nu||\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}||^2 \leq f(\mathbf{X}^{(k)}, \mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)})$$

By summation of the two previous inequalities, we have :

$$f(\mathbf{X}^{(k+1)}, \mathbf{A}^{(k+1)}, \mathbf{b}^{(k+1)}) + \nu||\mathbf{A}^{(k+1)} - \mathbf{A}^{(k)}||^2 + \nu||\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}||^2 \leq f(\mathbf{X}^{(k)}, \mathbf{A}^{(k)}, \mathbf{b}^{(k)})$$

By summation over k, we get, for any $p \in \mathbb{N}$:

$$\nu \left(\sum_{k=0}^{p} ||\mathbf{A}^{(k+1)} - \mathbf{A}^{(k)}||^2 + \sum_{k=0}^{p} ||\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}||^2 \right) \leq f(\mathbf{X}^{(0)}, \mathbf{A}^{(0)}, \mathbf{b}^{(0)})$$

From this, we can deduce that :

- The sequences $\{\mathbf{A}^{(k)}\}$ and $\{\mathbf{X}^{(k)}\}$ are bounded. Therefore, they each admit a converging subsequence.
- The sequences $\{||\mathbf{A}^{(k)} - \mathbf{A}^{(k+1)}||^2\}$ and $\{||\mathbf{X}^{(k)} - \mathbf{X}^{(k+1)}||^2\}$ tend to 0.

From this, we deduce that the sequences $\{\mathbf{A}^{(k)}\}$ and $\{\mathbf{X}^{(k)}\}$ converge. $\mathbf{b}^{(k+1)}$ can be expressed as a continuous function of $\mathbf{A}^{(k+1)}$ and $\mathbf{X}^{(k)}$, therefore the sequence $\{\mathbf{b}^{(k)}\}$ converges too.