LOCALLY STATIONARY TIME VERTEX PROCESS MODELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENIZ ASLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2026

Approval of the thesis:

LOCALLY STATIONARY TIME VERTEX PROCESS MODELS

submitted by **DENIZ ASLAN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Nevzat Güneri Gençer
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Elif Vural
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Sevinç Figen Öktem Seven
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Elif Vural
Electrical and Electronics Engineering, METU _____

Prof. Dr. Hakan Çevikalp
Electrical and Electronics Engineering, ESOGU _____

Date:13.01.2026

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Deniz Aslan

Signature :

ABSTRACT

LOCALLY STATIONARY TIME VERTEX PROCESS MODELS

Aslan, Deniz
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Elif Vural

January 2026, 76 pages

Graph signal processing provides a powerful framework for analyzing data defined over irregular network structures. In many real-world applications, time-vertex signals are observed only partially due to sensing failures, communication constraints, or limited data availability, making the learning of reliable signal models a challenging task. In particular, estimating parametric models from incomplete time-vertex observations requires capturing both temporal dynamics and graph-dependent statistical structures. Existing approaches based on joint wide-sense stationarity (JWSS) offer effective tools for modeling global statistical properties of time-vertex signals; however, their global stationarity assumptions often fail to represent local variations that naturally arise across both temporal and graph dimensions. In this thesis, we address the problem of estimating time-vertex signals with missing observations by learning parametric models that exhibit locally stationary behavior over time and graph. The proposed framework extends the concept of local stationarity to the time-vertex setting by modeling a structured covariance matrix derived from the sample covariance through a JWSS-inspired formulation. Model parameters are estimated via an alternating optimization algorithm that efficiently exploits the covariance structure for model fitting.

Experimental results on synthetic and real data demonstrate that the proposed approach achieves lower normalized mean error (NME) and mean absolute error (MAE) compared to reference time-vertex methods. Overall, this study introduces a covariance-driven parametric framework for learning locally stationary time-vertex processes, enabling more reliable modeling of complex graph-temporal signals.

Keywords: Graph Signal Processing, Locally Stationary Time-Vertex Processes, Local Stationarity, Moving Average Processes

ÖZ

YEREL DURAĞAN ZAMAN DÜĞÜM SÜREÇ MODELLERİ

Aslan, Deniz
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Elif Vural

Ocak 2026 , 76 sayfa

Çizgeler üzerinde sinyal işleme, düzensiz ağ yapıları üzerinde tanımlanan verilerin analizinde etkili çözümler sunmaktadır. Gerçek dünya uygulamalarında, zaman-düğüm sinyalleri çoğunlukla algılama hataları, haberleşme kısıtları veya sınırlı veri mevcudiyeti nedeniyle eksik gözlemlerle elde edilmektedir. Bu durum, güvenilir sinyal modellerinin öğrenilmesini önemli ölçüde zorlaştırmaktadır. Özellikle eksik zaman-düğüm gözlemlerinden parametrik modellerin öğrenilmesi, hem zamansal dinamiklerin hem de çizge yapısına bağlı istatistiksel özelliklerin birlikte hesaplanmasını gerektirmektedir. Ortak geniş anlamda durağanlık (Joint Wide-Sense Stationarity, JWSS) temelli mevcut yaklaşımlar, zaman-düğüm sinyallerinin genel istatistiksel özelliklerini modellemede etkili olsa da, bu yöntemlerde kullanılan genel anlamda durağanlık varsayımları zaman ve çizge boyutlarında ortaya çıkan yerel değişimleri temsil etmekte yetersiz kalmaktadır. Bu tezde, eksik gözlemler içeren zaman-düğüm sinyallerinin tahmin edilmesi problemi ele alınmakta ve hem zaman hem de çizge boyutlarında yerel olarak durağan davranış sergileyen parametrik modellerin öğrenilmesi amaçlanmaktadır. Önerilen yaklaşım, örnek kovaryans matrisinden türetilen yapılandırılmış bir kovaryans modeli aracılığıyla yerel durağanlık kavramını zaman-düğüm

bağlamına genişletmektedir. Model parametreleri, kovaryans yapısını etkin biçimde kullanan alternatif optimizasyon tabanlı bir algoritma ile tahmin edilmektedir. Sentetik ve gerçek veri kümeleri üzerinde gerçekleştirilen deneyler, önerilen yöntemin mevcut yöntemlere kıyasla daha düşük normalize ortalama hata (NME) ve ortalama mutlak hata (MAE) değerleri elde ettiğini göstermektedir. Sonuç olarak, bu çalışma karmaşık zaman-düğüm sinyallerinin daha güvenilir bir şekilde modellenmesine olanak sağlayan, kovaryans temelli parametrik bir öğrenme çerçevesi sunmaktadır.

Anahtar Kelimeler: Çizge Sinyal İşleme, Yerel Durağan Zaman-Düğüm Süreçleri, Yerel Durağanlık, Hareketli Ortalama Süreçleri

This thesis is dedicated to the memory of my dear aunt, Emel, rip.

ACKNOWLEDGMENTS

This thesis could not have been completed without the contributions of several individuals. Above all, I would like to sincerely thank my thesis supervisor, Assoc. Prof. Dr. Elif Vural, for her guidance and continuous support throughout this research. During the more challenging phases of the thesis, her patient and understanding approach were invaluable. This support made it possible for the work to progress and reach its final form.

I would also like to thank my family for their unwavering support and understanding throughout this demanding period. Their trust and encouragement provided a stable foundation that allowed me to focus on my academic goals. In addition, I gratefully acknowledge my company TUBITAK BILGEM ILTAREN for providing a supportive and flexible working environment, which made it possible to balance professional responsibilities with graduate research.

Finally, I would like to thank Ayça for her constant presence, love, patience, and emotional support during challenging times. Her understanding and motivation contributed significantly to maintaining my focus and resilience throughout this process.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xv

LIST OF ABBREVIATIONS xvii

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation and Problem Definition 1

1.2 Proposed Methods and Models 3

1.3 Contributions 5

1.4 The Outline of the Thesis 6

2 RELATED WORK 7

2.1 Notation 7

2.2 Fundamentals of Graph Signal Processing 8

2.3 Time-Vertex Signal Processing 12

2.4 Graph Signal Interpolation and Estimation of Missing Observations Approaches . . . 17

2.4.1 Non-Parametric Methods . . . 17

2.4.2 Parametric Methods . . . 18

2.4.3 Learning Based Methods . . . 22

2.4.4 Discussion . . . 24

3 LOCALLY STATIONARY TIME VERTEX PROCESS . . . 25

3.1 Stationarity in Graph Signal Processing . . . 25

3.2 Locally Stationary Time-Vertex Processes . . . 26

3.3 Components of the Proposed Process Model . . . 27

3.3.1 Joint Filter . . . 27

3.3.2 Time-Vertex Membership Functions . . . 28

3.4 Theoretical Covariance of the LSTVP . . . 30

3.5 Empirical Covariance Estimation . . . 33

3.6 Learning Framework for the LSTVP Model . . . 34

3.7 Optimization Details and Complexity Analysis . . . 40

4 EXPERIMENTAL RESULTS . . . 45

4.1 Performance Analysis of LSTVP . . . 47

4.1.1 Effect of the Number of Realizations . . . 53

4.1.2 Effect of Noise Level . . . 56

4.1.3 Effect of Model Complexity . . . 58

4.1.4 Effect of Regularization Parameters . . . 60

4.2 Comparative Experiments . . . 61

5 CONCLUSIONS 65

REFERENCES 69

LIST OF TABLES

TABLES

Table 4.1 Variation of the NME μ_2 and μ_3 where $\mu_1 = 0$ 61

Table 4.2 Variation of the NME with different μ_1 values 61

LIST OF FIGURES

FIGURES

Figure 2.1 Sample Graph 9

Figure 2.2 Sample Graph Signal 11

Figure 2.3 Examples of time–vertex signals at different time indices τ. Each subfigure illustrates the graph signal snapshot at a specific time instant, highlighting the temporal evolution across the graph domain. . . 13

Figure 2.4 Joint Kernel 16

Figure 3.1 Overall pipeline of the proposed algorithm 44

Figure 4.1 Visualization of the first two membership functions (g_1 and g_2) used in the synthetic data generation. For each membership, the left column shows the spatial activation over the graph domain, while the right column illustrates its variation over time for $T = 6$. 49

Figure 4.2 Visualization of the remaining membership functions (g_3 and g_4) used in the synthetic data generation. The left column shows the spatial activation over the graph domain, while the right column illustrates the temporal variation for $T = 6$. 50

Figure 4.3 Visualization of the joint kernels associated with the $K = 4$ membership functions used in the synthetic data generation. Each joint kernel corresponds to the membership functions shown in Fig. 4.1 and Fig. 4.2, and illustrates distinct local spectral characteristics in the joint time-vertex domain. 51

Figure 4.4 Estimated joint power spectral density (JPSD) of synthetic time–vertex signals generated using the proposed LSTVP model. The distribution of spectral energy across graph and temporal frequencies reflects the combined effect of the locally varying membership functions and their associated low-pass joint kernels. 52

Figure 4.5 Effect of the number of realizations on reconstruction performance. The figure reports the variation of the normalized mean error (NME) and mean absolute error (MAE) as a function of the number of available realizations for two different noise values. The ratio of missing samples is fixed to 25%. 55

Figure 4.6 Effect of noise level on reconstruction performance. The figure illustrates the variation of the normalized mean error (NME) and mean absolute error (MAE) as a function of the signal-to-noise ratio (SNR) for two different number of realizations ($R = 70$ and $R = 1000$), missing samples fixed to 25%. 57

Figure 4.7 Effect of model complexity on covariance estimation performance. The covariance discrepancy (CD) is reported as a function of different model complexity parameters, illustrating the impact of varying K, Q, and M on the accuracy of the learned LSTVP covariance structure. 59

Figure 4.8 NME (left) and MAE (right) results for the Molène dataset. . . . 62

Figure 4.9 NME (left) and MAE (right) results for the NOAA dataset. . . . 63

LIST OF ABBREVIATIONS

ABRREVIATONS

ARMA	Auto Regressive Moving Average
CD	Covariance Discrepancy
DFT	Discrete Fourier Transform
GFT	Graph Fourier Transform
GSP	Graph Signal Processing
IDFT	Inverse Discrete Fourier Transform
IGFT	Inverse Graph Fourier Transform
IJFT	Inverse Joint Fourier Transform
JFT	Joint Fourier Transform
JPSD	Joint Power Spectral Density
JWSS	Joint Wide Sense Stationary
k-NN	k nearest neighbors
k-SVD	k-Singular Value Decomposition
LMMSE	Linear Minimum Mean Square Error
LSGP	Locally Stationary Graph Processes
LSTVP	Locally Stationary Time-Vertex Processes
MA	Moving Average
MAE	Mean Absolute Error
MSE	Mean Square Error
NME	Normalized Mean Error
SDP	Semidefinite Programming
SNR	Signal to Noise Ratio

SOCP	Second Order Cone Programming
WSS	Wide Sense Stationary

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

In recent years, rapid advancements in technology have led to a substantial increase in both the number and complexity of data-generating systems. In this context, graph-based modeling has emerged as a powerful and versatile framework for representing and analyzing signals collected from systems characterized by irregular or non-Euclidean topologies, such as sensor networks. This modeling paradigm forms the basis of Graph Signal Processing (GSP), an emerging field that generalizes classical signal processing tools to signals defined on the vertices of a graph. Beyond the spatial variability induced by the graph structure, the signals of interest often exhibit temporal dynamics as well. Accordingly, modeling such data as time-vertex signals, which jointly capture variations over both the temporal and topological domains, provides a comprehensive and effective approach to signal representation and analysis.

Graph Signal Processing (GSP) provides a principled framework for analyzing data supported on irregular structures such as graphs, extending classical signal processing tools—including the Fourier transform, sampling, and filtering—to non-Euclidean domains [1, 2]. Over the past decade, GSP techniques have found successful applications in a wide range of domains, such as sensor networks [2], social networks [31], traffic flow analysis [5], and brain connectivity studies [24]. The ability of GSP to capture both local and global dependencies enables efficient modeling and interpretation of high-dimensional structured data.

However, many real-world, graph-supported signals are not only spatially structured but also evolve over time. In such cases, traditional graph-based or temporal mod-

els alone are insufficient, as they fail to capture the joint dependencies across time and graph domains. This limitation has motivated the development of Time-Vertex Graph Signal Processing (TV-GSP) frameworks, which extend GSP to handle temporal dynamics and allow joint filtering, spectral analysis, and modeling across both dimensions [6, 15].

Stationarity is a central concept in both classical and graph-based signal processing. In traditional time-series analysis, a wide-sense stationary (WSS) process is characterized by a constant mean and an autocovariance function that depends only on the time lag [33]. In the graph setting, the notion of WSS is adapted by requiring the covariance to be jointly diagonalizable with the graph Laplacian, which ensures that statistical properties remain invariant under graph shifts [6]. When extended to the time-vertex domain, stationarity implies invariance not only across time shifts but also across graph translation operators, leading to models in which the joint covariance structure is globally homogeneous [8].

Despite these advances, most existing methods assume that the underlying data are fully observed and globally wide-sense stationary (WSS) [6, 8]. In practice, these assumptions rarely hold. Signals collected from networked systems often suffer from incomplete or missing observations due to sensor malfunctions, packet loss, or intentional subsampling. Such missing data severely degrades the performance of conventional estimators and leads to biased or undefined sample covariances.

Furthermore, the global stationarity assumption rarely holds in practice. In many real-world settings, the second-order statistics of time-vertex signals vary significantly across different regions of the graph or at different times. For example, in meteorological networks, temperature and humidity patterns evolve smoothly over geographical regions but exhibit distinct local behaviors due to micro-climates. In transportation or traffic networks, congestion levels differ across urban zones and vary throughout the day, yielding nonstationary time-vertex dynamics. Similarly, in social networks, information diffusion or user activity patterns fluctuate across communities, making the assumption of uniform global stationarity unrealistic. These scenarios demonstrate that the covariance structure often changes locally, necessitating models that can capture position-dependent and time-varying statistical characteristics.

The global WSS assumption imposes strict homogeneity that cannot accommodate the local temporal or spatial variations observed in realistic systems. Consequently, there is a strong need for models that can (i) incorporate both graph structure and temporal evolution, (ii) allow their parameters to be estimated reliably even when only partial observations are available, and (iii) flexibly capture local stationarity in the time-vertex domain.

Motivated by these challenges, this thesis focuses on modeling time-vertex graph signals using a parametric framework that characterizes local stationarity jointly in both time and vertex dimensions. The proposed model, referred to as the Locally Stationary Time Vertex Process (LSTVP), aims to learn low-order parametric filters that efficiently capture the underlying dynamics while maintaining computational tractability. Once such a model is learned, it can be used to estimate missing observations and reconstruct the full signal with improved statistical reliability.

The motivation behind this study stems from the growing demand for accurate and statistically valid estimation methods in systems where both temporal dynamics and graph topology play critical roles—especially under incomplete or noisy measurements. By bridging the gap between temporal modeling and graph-based signal representation, the proposed framework seeks to provide a robust foundation for analyzing, reconstructing, and predicting complex time-vertex signals.

1.2 Proposed Methods and Models

To address the challenges discussed in the previous section, this thesis proposes a parametric modeling framework for time-vertex signals with missing observations. While existing approaches, such as non-parametric graph filtering, dictionary learning, and deep unrolling-based inference methods, have achieved notable success in signal recovery tasks, they often rely on large amounts of training data or neglect the underlying stochastic structure of the process. In contrast, our focus is on developing a compact, interpretable, and statistically grounded model that explicitly captures the joint variability of signals over both time and graph domains.

The main goal of this thesis is to learn a parametric time-vertex process model that in-

corporates local stationarity across both time and graph domains. The learned model provides a flexible foundation for several downstream tasks such as interpolation, and recovering missing time-vertex observations. Unlike traditional methods that either disregard the underlying graph topology or rely on global stationarity assumptions, our approach is based on learning locally stationary models that allow for smooth statistical variations across time and vertex dimensions. By formulating the signal generation process through a moving average (MA) representation, the proposed framework exploits the joint time-vertex covariance structure for reliable estimation. This formulation provides a unified perspective that bridges previous efforts on ARMA-based modeling of time-vertex graph signals [21] and locally stationary graph processes [19], enabling accurate recovery of incomplete time-vertex data.

More specifically, the proposed model extends the concept of locally stationary processes, which are well established in classical time-series analysis [22], to the joint time-vertex domain. The model represents the process as a time-varying combination of MA components, where each component describes the second-order statistics associated with a local region in the time-vertex space. This parametric representation yields a compact description of the covariance structure, which is particularly advantageous when only partial observations are available.

A key component of the framework is the introduction of *membership functions*, which capture local variability in both the time and vertex domains. In the temporal domain, the memberships describe how the contributions of different components evolve gradually over time, ensuring continuity in local statistics. In the vertex domain, they act as node-dependent weights that determine the spatial influence of each component on the overall signal. By enforcing smoothness constraints on these memberships, the model prevents abrupt transitions, leading to a more realistic and interpretable description of the underlying process.

We formulate the learning of locally stationary time-vertex processes as a covariance fitting problem. Given partially observed realizations of the process, the joint covariance matrix is estimated and subsequently fitted to the parametric form implied by the model. While the overall problem is not jointly convex, both the membership-learning and MA-parameter estimation subproblems can be relaxed to convex formulations

when treated separately. This structure enables an alternating optimization strategy that preserves tractability and yields stable parameter estimates. In this framework, alternating optimization and suitable relaxation techniques allow the parameters of both the covariance model and the membership functions to be learned directly from incomplete data without requiring fully observed realizations.

After learning the proposed parametric model, we demonstrate its effectiveness through time-vertex interpolation experiments on both real-world datasets and synthetically generated signals. In this setting, the learned covariance structure enables the recovery of missing signal entries by exploiting the model's locally stationary second-order statistics. Specifically, we employ a linear minimum mean-square error (LMMSE) estimator that uses the joint time-vertex covariance implied by the learned model to predict the unobserved samples. This allows the reconstructed signal to remain consistent with both the temporal dynamics and the graph topology, illustrating the practical value and generalizability of the proposed approach.

In summary, the proposed framework integrates parametric stochastic modeling with smoothly varying memberships to generalize local stationarity to the time-vertex domain. By unifying concepts from parametric stochastic process models, it provides a principled and computationally tractable approach for the estimation and recovery of incomplete time-vertex signals.

1.3 Contributions

The main contributions of this thesis can be summarized as follows:

- We extend the concept of locally stationary processes from graph-only signals [19] to time-vertex graph signals, providing a novel and comprehensive framework that captures both temporal and structural variations simultaneously. In doing so, we enhance parametric time-vertex estimation models by incorporating local stationarity, rather than relying solely on global modeling assumptions [21].
- We demonstrate the effectiveness of the proposed framework in recovering

missing time-vertex observations. Unlike classical approaches that require complete data or impose strong global assumptions, our method leverages locally stationary parametric modeling to estimate missing entries using only partial observations.

1.4 The Outline of the Thesis

The remainder of this thesis is organized as follows:

In Chapter 2, we review the related literature and provide the necessary background on graph signal processing, stationary and locally stationary processes, and existing approaches for handling missing data.

In Chapter 3, we introduce the proposed time-vertex locally stationary parametric model. This chapter presents both the theoretical formulation of the model and the associated parameter learning framework, including the derivation of the learning objectives and optimization steps.

In Chapter 4, we present experimental results obtained from both synthetic and real-world datasets, and provide a comparative performance evaluation against existing methods.

Finally, Chapter 5 concludes the thesis by summarizing the main findings and discussing possible directions for future research.

CHAPTER 2

RELATED WORK

This section reviews the existing literature that forms the foundation of our study, covering key developments in graph signal processing, time–vertex models, wide-sense and locally stationary graph processes, and parameter estimation methods used in the field of graph signal processing.

2.1 Notation

To ensure clarity and consistency, this section summarizes the notation used throughout the thesis. Scalars are represented by italic letters (e.g., $a \in \mathbb{R}$), while vectors and matrices are denoted by boldface letters—lowercase for vectors (e.g., $\mathbf{x} \in \mathbb{R}^N$) and uppercase for matrices (e.g., $\mathbf{A} \in \mathbb{R}^{N \times M}$). Calligraphic symbols denote sets (e.g., $\mathcal{S}$), and the cardinality of a set $\mathcal{S}$ is written as $|\mathcal{S}|$. The notation $\mathbf{A}(i,j)$ refers to the (i,j)-th element of the matrix $\mathbf{A}$. The Euclidean norm of a vector is denoted by $\|\cdot\|_2$, while $\|\cdot\|_F$ stands for the Frobenius norm of a matrix. Transpose and Hermitian transpose operations are written as $(\cdot)^T$ and $(\cdot)^H$, respectively. The elementwise (Hadamard) product is represented by $\circ$, and the Kronecker product by $\otimes$. We use $\mathbf{I}_N$ to denote the N-dimensional identity matrix, and $\mathbf{0}_{N\times M}$ and $\mathbf{1}_{N\times M}$ for all-zero and all-one matrices of size $N \times M$, respectively. Absolute value operations applied to a matrix refer to elementwise application, i.e., $|\mathbf{A}|(i,j) = |\mathbf{A}(i,j)|$. Elementwise inequality between matrices is indicated by $\preceq$, meaning that $\mathbf{A} \preceq \mathbf{B}$ implies $\mathbf{A}(i,j) \leq \mathbf{B}(i,j)$ for all (i,j). The covariance matrix of a random vector $\mathbf{x}$ quantifies the second–order dependencies between its entries and is computed by centering the vector around its

mean. It is given by

$$\mathbf{C}_x = \mathbb{E}\big[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^H\big] \tag{2.1}$$

which captures how the fluctuations of $\mathbf{x}$ around their expected value are correlated across different components.

2.2 Fundamentals of Graph Signal Processing

Graph Signal Processing (GSP) provides a theoretical framework for extending classical signal processing concepts to data supported on irregular structures, such as networks, meshes, or manifolds. Unlike traditional signals defined over regular and uniformly sampled domains, graphs offer a flexible representation for modeling pairwise relationships through their vertices and weighted edges [12, 13, 51]. An example of such a graph structure is illustrated in Figure 2.1. This framework enables the analysis, filtering, and modeling of complex data supported on networks by leveraging spectral methods grounded in graph theory [52]. A weighted and undirected graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denotes the set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges, and $\mathbf{W} \in \mathbb{R}^{N \times N}$ is the weight matrix whose entries w_{ij} quantify the similarity or interaction strength between nodes v_i and v_j. Although various graph models exist in the literature—such as directed, undirected, weighted, unweighted, or bipartite graphs—this thesis focuses on an undirected and weighted graph $\mathcal{G}$, which is well suited for spectral graph analysis. In an undirected graph $\mathcal{G}$, $\mathbf{W}$ is symmetric and nonnegative, i.e., $w_{ij} = w_{ji} \geq 0$.

In practice, graphs are often formed by linking each vertex to a subset of nearby points in the data domain. A common choice is the k-nearest neighbor (k-NN) construction, where edges are created between a node and its k most proximate neighbors. The strength of these connections is then encoded through a similarity function, frequently modeled by a Gaussian kernel to reflect smoothly decaying affinities,

$$w_{ij} = \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_j\|^2}{2\sigma^2}\right) \tag{2.2}$$

with $\mathbf{r}_i$ and $\mathbf{r}_j$ denoting the positions of the corresponding nodes, and σ controlling the bandwidth of the kernel.

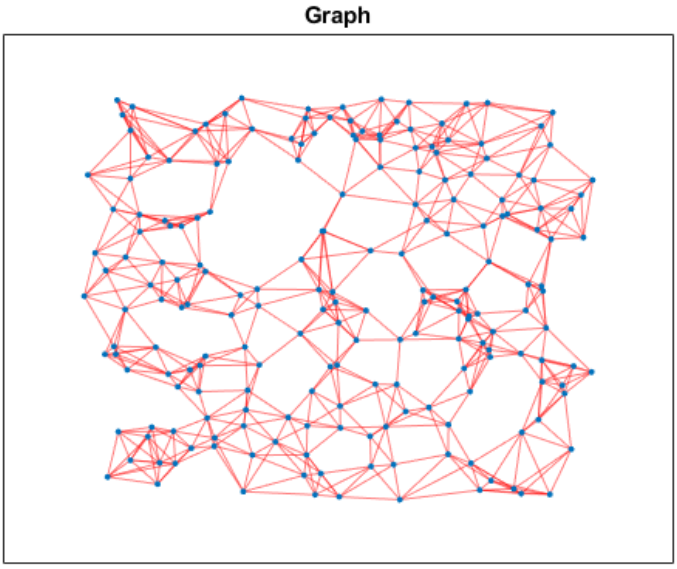


Figure 2.1: Sample Graph

The degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a diagonal matrix with elements $d_{ii} = \sum_{j=1}^{N} w_{ij}$ representing the total connection strength of node i. A central operator in graph signal processing is the graph Laplacian, which plays a role analogous to the second order discrete differential operator in classical signal processing [2], [23], [56]. It measures how a signal varies across the graph by comparing the value at each node with those of its neighbors, thereby encoding notions of smoothness, variation, and flow. This property makes the Laplacian the cornerstone of spectral analysis on graphs, as its eigen-structure provides a natural definition of graph frequencies and forms the foundation for filtering, interpolation, and statistical modeling of graph-supported data. The combinatorial graph Laplacian is defined as in (2.3)

$$\mathbf{L}_G = \mathbf{D} - \mathbf{W}, \tag{2.3}$$

while the normalized Laplacian is given by in (2.4)

$$\mathbf{L}_G = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}. \tag{2.4}$$

In many applications, the normalized Laplacian is preferred over the combinatorial Laplacian because it mitigates the influence of degree variability across the graph. By incorporating degree normalization, this operator provides a scale-invariant measure

of node interactions, thereby allowing signals defined on graphs with heterogeneous connectivity patterns to be compared more consistently. Moreover, the normalized Laplacian has a well-bounded spectrum within the interval [0,2], which offers mathematical convenience in spectral analysis and improves numerical stability in filtering and optimization procedures. Both forms are real, symmetric, and positive semidefinite, thus admitting an eigen-decomposition

$$\mathbf{L}_G = \mathbf{U}_G \mathbf{\Lambda}_G \mathbf{U}_G^\top, \tag{2.5}$$

where $\mathbf{U}_G = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ contains the orthonormal eigenvectors, and

$$\mathbf{\Lambda}_G = \operatorname{diag}(\lambda_1, \ldots, \lambda_N) = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{bmatrix}, \tag{2.6}$$

holds the nonnegative eigenvalues. The eigenvalues λ_n play a role analogous to frequencies in classical Fourier analysis, where small λ_n correspond to slowly varying components of the signal across the graph, while large λ_n capture rapid variations [1][2][3].

A graph signal is defined as a mapping $x : \mathcal{V} \to \mathbb{R}$ that assigns a scalar value to each node. In vector form, it is represented as in (2.7),

$$\mathbf{x} = [x(v_1), x(v_2), \ldots, x(v_N)]^\top \in \mathbb{R}^N. \tag{2.7}$$

The value $x(v_i)$ can represent, for example, a temperature measurement collected by a sensor located at node v_i. A synthetically generated graph signal is illustrated in Figure 2.2.

The eigenvectors of the Laplacian form a complete orthonormal basis for $\mathbb{R}^N$, referred to as the graph Fourier basis. Using this basis, the Graph Fourier Transform (GFT) of a graph signal $\mathbf{x}$ is defined as in (2.8)

$$\hat{\mathbf{x}} = \mathbf{U}_G^\top \mathbf{x} \tag{2.8}$$

and the Inverse Graph Fourier Transform (IGFT) is given by in (2.9)

$$\mathbf{x} = \mathbf{U}_G \hat{\mathbf{x}}. \tag{2.9}$$

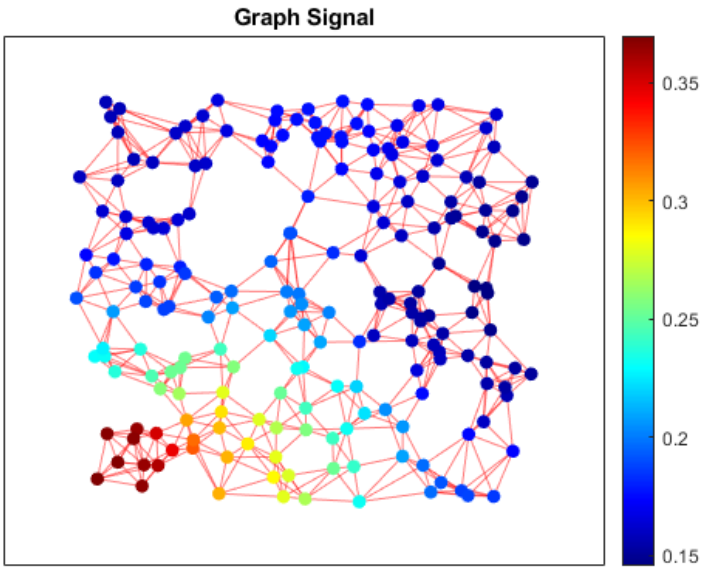


Figure 2.2: Sample Graph Signal

Each spectral coefficient $\hat{x}(\lambda_n) = \mathbf{u}_n^\top \mathbf{x}$ represents the contribution of the n-th graph frequency to the overall graph signal. Similar to the classical Fourier transform, smooth signals on the graph have most of their spectral energy concentrated at low graph frequencies, which correspond to eigenvectors that exhibit slow variation across the topology. Conversely, high graph frequencies represent rapidly oscillating components that change sharply between adjacent nodes, capturing discontinuities or localized fluctuations in the signal. Beyond such global spectral representations, localized analysis on graphs has been investigated through vertex-frequency representations, which jointly characterize signals in the vertex and spectral domains [10]. Representative examples include the windowed graph Fourier transform (WGFT) and related multiscale graph wavelet representations, which extend classical time-frequency and multiresolution analysis concepts to graph domains [46], [54], [57].

Filtering operations on graphs are defined in the spectral domain by modifying these frequency components. Let $g(\lambda)$ denote a graph filter kernel that determines how each graph frequency is scaled. The filtered signal $\mathbf{y}$ is obtained as

$$\mathbf{y} = g(\mathbf{L}_G)\mathbf{x} = \mathbf{U}_G g(\mathbf{\Lambda}_G)\mathbf{U}_G^\top \mathbf{x} \tag{2.10}$$

where $g(\mathbf{\Lambda}_G) = \mathrm{diag}(g(\lambda_1), \ldots, g(\lambda_N))$. In practice, the filter $g(\mathbf{L}_G)$ is often approx-

imated as a polynomial function of the Laplacian:

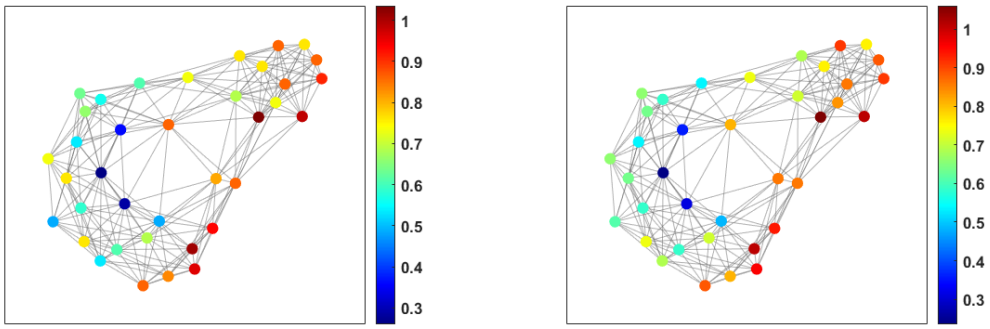
$$g(\mathbf{L}_G) = \sum_{k=0}^{K-1} a_k \mathbf{L}_G^k \tag{2.11}$$

where $\{a_k\}$ are polynomial coefficients. This form allows localized filtering operations, since $\mathbf{L}_G^k$ captures k-hop neighborhood interactions. Polynomial graph filters are widely used in practical implementations due to their computational efficiency and scalability for large-scale graphs [1, 2].

Graph signal processing thus generalizes fundamental operations such as convolution, frequency analysis, and spectral filtering to irregular graph domains. In this analogy, the graph Laplacian serves as a discrete differential operator, and its eigenvalues and eigenvectors provide a notion of graph frequencies analogous to temporal frequencies in classical signal processing. In the following section, these concepts are extended to the joint analysis of time and vertex domains, forming the foundation of time-vertex graph signal processing.

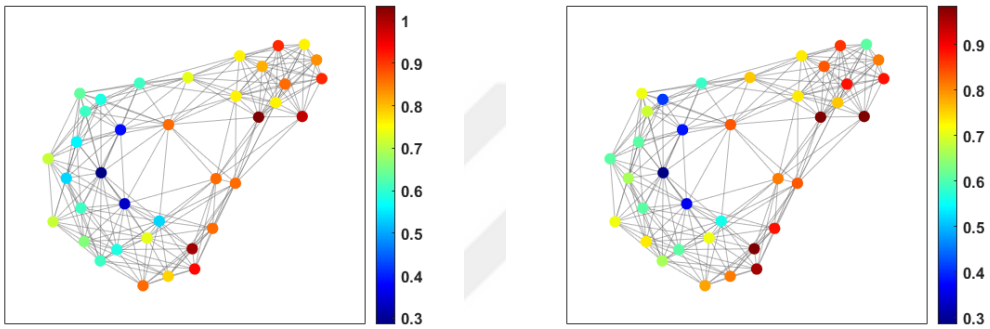
2.3 Time-Vertex Signal Processing

Graph Signal Processing provides tools for analyzing data supported on irregular structures; however, many real-world graph-based datasets evolve not only across the vertices of a graph but also over time. Examples include temperature readings in environmental sensor networks, traffic measurements across road junctions, and EEG signals recorded from multiple cortical regions. Such signals exhibit dependencies in both the spatial and temporal domains, motivating the development of *time-vertex signal processing (TVSP)*, which jointly models dynamics across graph and time dimensions [7, 8]. A time-vertex signal is represented as a matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$, where N denotes the number of graph vertices and T denotes the number of temporal samples. Each column $\mathbf{x}_t$ captures the graph signal at time index t, while each row describes the temporal evolution of a particular node. Representative examples of time–vertex signals at different time instants are illustrated in Fig. 2.3.



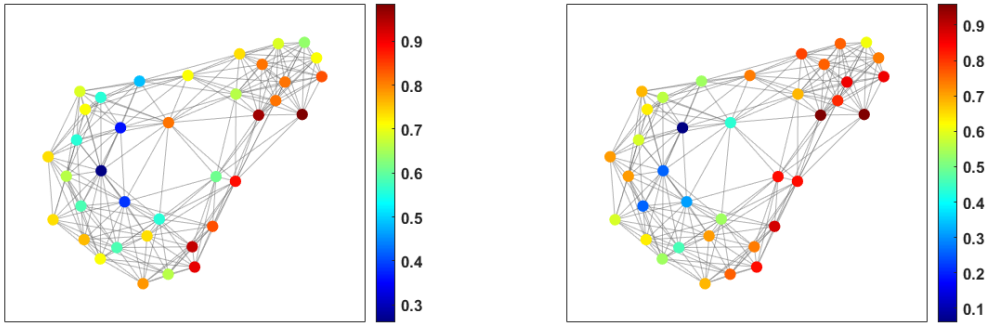
(a) $\tau = 1$

(b) $\tau = 2$



(c) $\tau = 3$

(d) $\tau = 4$



(e) $\tau = 5$

(f) $\tau = 6$

Figure 2.3: Examples of time–vertex signals at different time indices τ. Each subfigure illustrates the graph signal snapshot at a specific time instant, highlighting the temporal evolution across the graph domain.

For convenience in spectral and statistical analysis, the time-vertex signal is vectorized as

$$\mathbf{x} = \mathrm{vec}(\mathbf{X}) \in \mathbb{R}^{NT}, \tag{2.12}$$

where $\mathrm{vec}(\cdot)$ denotes the vectorization operator that stacks the columns of a matrix into a single column vector, thereby enabling the use of joint operators acting simultaneously in both the graph and time domains.

To capture variability across space and time, TVSP employs two Laplacian operators: the graph Laplacian $\mathbf{L}_G$, defined in Section 2.2, and a temporal Laplacian $\mathbf{L}_T$ that embeds dependencies along the time axis. The temporal Laplacian is typically constructed either from a path graph, reflecting first-order temporal relationships, or from a circulant structure when periodicity is assumed. Their eigendecompositions are given by in (2.13a) and (2.13b)

$$\mathbf{L}_G = \mathbf{U}_G \mathbf{\Lambda}_G \mathbf{U}_G^\top \tag{2.13a}$$

$$\mathbf{L}_T = \mathbf{U}_T \mathbf{\Lambda}_T \mathbf{U}_T^\top \tag{2.13b}$$

where $\mathbf{U}_G$ and $\mathbf{U}_T$ contain the spatial and temporal eigenvectors, respectively. Let $\mathbf{U}_T \in \mathbb{C}^{T\times T}$ denote the normalized discrete Fourier transform (DFT) basis matrix. Its (t,τ)-th entry is defined as

$$[\mathbf{U}_T]_{t,\tau} = \frac{1}{\sqrt{T}}\, e^{j\omega_\tau t}, \qquad t,\tau = 1,\ldots,T, \tag{2.14}$$

where the discrete angular frequencies are given by

$$\omega_\tau = \frac{2\pi(\tau-1)}{T}. \tag{2.15}$$

Accordingly, the diagonal matrix $\mathbf{\Lambda}_T$ is defined as

$$\mathbf{\Lambda}_T = \mathrm{diag}(\omega_1,\ldots,\omega_T), \tag{2.16}$$

where $\mathrm{diag}(\cdot)$ denotes the operator that forms a diagonal matrix from the entries of its argument. When $\mathbf{L}_T$ is circulant, its eigenvectors correspond to the discrete Fourier transform (DFT) basis [8, 6]. By combining the spatial and temporal domains, TVSP defines a joint Laplacian operator as [8]

$$\mathbf{L}_J = \mathbf{L}_T \otimes \mathbf{I}_N \;+\; \mathbf{I}_T \otimes \mathbf{L}_G \tag{2.17}$$

where $\otimes$ denotes the Kronecker product. The associated joint eigenbasis is

$$\mathbf{U}_J = \mathbf{U}_T \otimes \mathbf{U}_G \tag{2.18}$$

which enables a spectral representation simultaneously across graph and time frequencies. Using this joint basis, the *Joint Fourier Transform (JFT)* of a time-vertex signal is defined as in (2.19)

$$\hat{\mathbf{X}} = \mathbf{U}_G^\top \mathbf{X} \mathbf{U}_T \tag{2.19}$$

with the inverse transform given by

$$\mathbf{X} = \mathbf{U}_G \hat{\mathbf{X}} \mathbf{U}_T^\top. \tag{2.20}$$

Furthermore, for notational convenience and computational simplicity, the joint Fourier transform (JFT) and its inverse (IJFT) are commonly defined through the joint Fourier basis matrix $\mathbf{U}_J$ as [48, 50]

$$\hat{\mathbf{x}} = \mathbf{U}_J^H \mathbf{x}, \tag{2.21}$$

and

$$\mathbf{x} = \mathbf{U}_J \, \hat{\mathbf{x}}, \tag{2.22}$$

respectively, where $(\cdot)^H$ denotes the Hermitian transpose. Throughout the remainder of the thesis, all formulations are expressed using this vectorized representation for notational consistency. The JFT provides a two-dimensional frequency-domain representation, enabling the design of filtering operations that account for both temporal and graph-frequency content of the signal.

In time-vertex signal processing, linear filtering operations can be characterized in the joint spectral domain by a two-dimensional kernel defined over graph and temporal frequencies. Specifically, the joint spectral kernel $h(\mathbf{\Lambda}_G, \mathbf{\Omega}_T) \in \mathbb{C}^{N \times T}$ collects the filter responses over all joint frequency pairs, where $\mathbf{\Lambda}_G = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of graph Laplacian eigenvalues and $\mathbf{\Omega}_T = \mathrm{diag}(\omega_1, \ldots, \omega_T)$ is the diagonal matrix of discrete temporal frequencies. The (n, τ)th entry of $h(\mathbf{\Lambda}_G, \mathbf{\Omega}_T)$ corresponds to the scalar response $h(\lambda_n, \omega_\tau)$, which determines how the joint frequency component indexed by (λ_n, ω_τ) is amplified or attenuated by the filter. Figure 2.4 illustrates a typical joint spectral kernel whose response is concentrated at low graph and temporal frequencies, indicating a joint low-pass filtering behavior that promotes smoothness across both the graph and time domains.

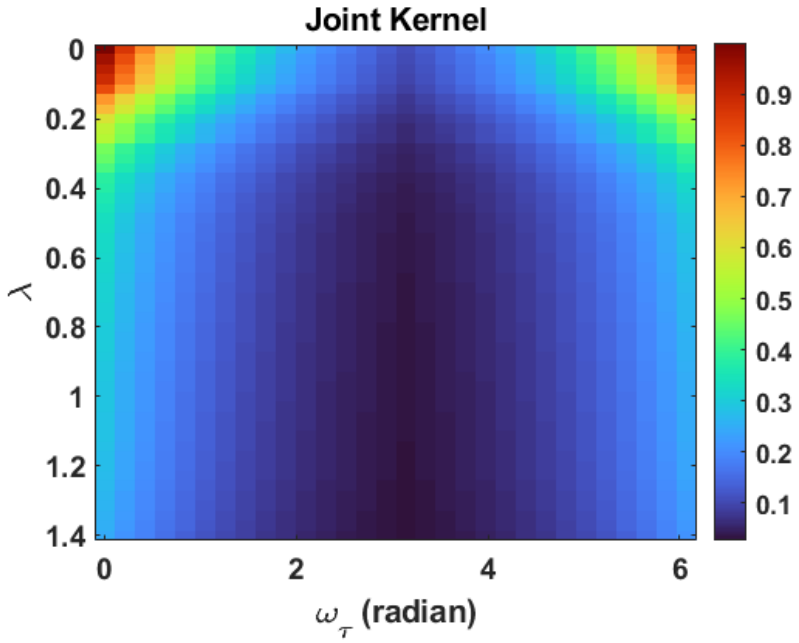


Figure 2.4: Joint Kernel

A joint spectral filter with kernel $h(\mathbf{\Lambda}_G, \mathbf{\Omega}_T)$ can be represented in vectorized form as

$$\mathbf{f} = \mathrm{vec}(h(\mathbf{\Lambda}_G, \mathbf{\Omega}_T)) \in \mathbb{C}^{NT}. \tag{2.23}$$

The corresponding diagonal matrix representation is then given by

$$\mathbf{F} = \mathrm{diag}(\mathbf{f}) \in \mathbb{C}^{NT \times NT}, \tag{2.24}$$

where $\mathbf{F}$ denotes the matrix counterpart of the joint spectral kernel $h(\mathbf{\Lambda}_G, \mathbf{\Omega}_T)$ in the vectorized joint spectral domain. Using this representation, the joint spectral filtering operation can be expressed as

$$\mathbf{y} = \mathbf{U}_J \, \mathbf{F} \, \mathbf{U}_J^H \, \mathbf{x} \; \in \; \mathbb{R}^{NT}, \tag{2.25}$$

where $\mathbf{U}_J$ denotes the joint Fourier basis and $(\cdot)^H$ represents the Hermitian transpose. The joint spectral framework also enables the definition of second-order statistical quantities for time-vertex random processes, most notably the joint covariance structure and the joint power spectral density (JPSD). For zero-mean processes, the JPSD provides a spectral description of how signal energy is distributed across joint graph and temporal frequency components.

In this representation, the covariance information of the process admits a compact characterization in the joint spectral domain, where each joint frequency component

contributes independently to the overall signal variance. The JPSD thus serves as a fundamental tool for analyzing and modeling the second-order behavior of time-vertex signals.

Overall, the time–vertex framework provides a unified structure for analyzing, filtering, and modeling signals that evolve jointly across graph and time domains. While these concepts establish the mathematical foundation for joint processing, practical reconstruction and estimation tasks often require additional modeling assumptions or problem-specific formulations. In the next section, we review existing approaches for interpolating time–vertex signals and estimating missing observations, situating them within the broader landscape of reconstruction methods.

2.4 Graph Signal Interpolation and Estimation of Missing Observations Approaches

The recovery of graph-supported signals from incomplete or missing measurements has emerged as a fundamental challenge in modern data analysis, particularly when the observations exhibit both spatial structure and temporal evolution. The interplay between graph topology and time dynamics has led to the development of a broad spectrum of reconstruction techniques, each relying on different assumptions about smoothness, statistical behavior, or underlying generative models. In the context of time–vertex signal processing, these methods can be broadly categorized into non-parametric, parametric, and learning-based approaches. This section reviews the principal contributions within each category, emphasizing their modeling philosophies, strengths, and inherent limitations

2.4.1 Non-Parametric Methods

Non-parametric interpolation methods rely on the structural properties of the underlying graph without assuming a specific parametric model for the signal. Many of these techniques build upon the notion that real-world graph signals tend to vary smoothly over the topology, an idea formalized in early works on graph filters and graph-based harmonic analysis, as well as in semi-supervised learning with local and global con-

sistency and graph learning from smooth signals [1, 2, 44, 47]. In this setting, recovering missing vertex values typically involves solving optimization problems that penalize large variations across graph edges, often through Laplacian smoothness or total-variation regularizers. These smoothness-driven reconstructions have been applied in diverse tasks ranging from environmental sensing to brain imaging [24].

A second major branch of non-parametric approaches arises from graph sampling theory. Foundational studies demonstrate that if a graph signal is bandlimited with respect to the Graph Fourier Transform, then perfect reconstruction is possible from a sufficiently informative subset of nodes [15, 16]. These results have inspired numerous sampling and recovery strategies that view interpolation as a graph-aware extension of classical Shannon theory, where the role of the frequency domain is played by the Laplacian eigenvectors [5]. Approaches based on successive local aggregations [15] further generalize this perspective by enabling distributed implementations in large-scale networks. Related adaptive formulations have also been proposed for graph signal reconstruction, where both the signal estimates and the sampling strategy are updated online to track time-varying conditions [45], including robust adaptive algorithms designed for graph-supported data [60].

In time–vertex settings, non-parametric methods are typically extended by imposing joint smoothness priors across both domains. The work of Grassi et al. [14] introduced scalable joint operators capable of promoting regularity in space and time simultaneously, enabling the reconstruction of spatio-temporal data without explicitly presuming a statistical generative model. Related non-parametric formulations have also considered signal estimation over dynamically evolving graph topologies, where the underlying neighborhood structure changes over time [61]. While non-parametric approaches offer flexibility and do not require parameter learning, they may struggle in scenarios where the signal exhibits non-stationary or locally structured behaviors that smoothness priors cannot capture effectively.

2.4.2 Parametric Methods

Parametric reconstruction methods assume that the signal is generated by an underlying stochastic process, typically characterized through spectral representations [58].

In the graph domain, the concept of wide-sense stationarity was formalized in [4, 6], establishing that a stationary graph process has a covariance matrix that is diagonalizable by the Laplacian eigenvectors. This framework enables parametric estimation of the graph power spectral density (PSD) and supports reconstruction strategies that exploit second-order statistics. Extensions of this spectral perspective to time-varying settings have also been considered under additional structural assumptions. In particular, Fukuhara *et al.* [63] propose a parametric statistical framework for time-varying graph signals based on period estimation, which is subsequently exploited for graph Wiener filtering.

Time–vertex extensions of stationarity were later established by Loukas and Perraudin [8], who introduced the joint power spectral density (JPSD) to characterize dependencies across both space and time. Under this model, interpolation reduces to estimating or filtering in the joint spectral domain, where the eigenstructure of the joint Laplacian provides a natural basis for designing reconstructions. However, classical joint stationarity assumes globally consistent second-order behavior, which may be restrictive in real-world scenarios where dynamics change across time or spatial subregions.

Several parametric modeling approaches have been proposed to describe the temporal and spatial dynamics of signals evolving over graphs. In particular, models based on autoregressive moving-average (ARMA) [49, 53] and vector autoregressive moving-average (VARMA) processes extend classical time-series formulations by incorporating graph-structured dependencies. Isufi et al. [9] introduced graph VARMA recursions for forecasting time series over networks, demonstrating their effectiveness in spatio-temporal prediction tasks. Subsequent works have further explored parametric process models to address time variability and non-stationary behavior in graph-supported signals. For instance, Qiu et al. [34] studied time-varying graph signal reconstruction under changing graph topologies, while Güneyi et al. [21] proposed parametric time-vertex process models that enable learning from incomplete spatio-temporal observations.

Following the general formulation introduced in [21], a parametric time-vertex ARMA

process can be expressed in the generic form

$$\mathbf{x}_t = \sum_{p=1}^{P} \mathbf{A}_p(\mathbf{L}_G)\, \mathbf{x}_{t-p} + \sum_{q=0}^{Q} \mathbf{B}_q(\mathbf{L}_G)\, \mathbf{w}_{t-q}, \tag{2.26}$$

where $\{\mathbf{w}_t\}$ denotes zero-mean white noise, and $\mathbf{A}_p(\cdot)$ and $\mathbf{B}_q(\cdot)$ represent graph filters parameterized as functions of the graph Laplacian $\mathbf{L}_G$. When the graph filters $\mathbf{A}_p(\mathbf{L}_G)$ and $\mathbf{B}_q(\mathbf{L}_G)$ are restricted to polynomial forms of the graph Laplacian, the ARMA model admits a finite-dimensional parametric representation. In particular, assuming

$$\mathbf{A}_p(\mathbf{L}_G) = \sum_{k=0}^{K} a_{pk}\mathbf{L}_G^k, \qquad \mathbf{B}_q(\mathbf{L}_G) = \sum_{m=0}^{M} b_{qm}\mathbf{L}_G^m, \tag{2.27}$$

the time-vertex ARMA process in (2.26) can be expressed as

$$\mathbf{x}_t = \sum_{p=1}^{P}\sum_{k=0}^{K} a_{pk}\, \mathbf{L}_G^k\, \mathbf{x}_{t-p} + \sum_{q=0}^{Q}\sum_{m=0}^{M} b_{qm}\, \mathbf{L}_G^m\, \mathbf{w}_{t-q}, \tag{2.28}$$

where $\{a_{pk}\}$ and $\{b_{qm}\}$ denote the polynomial graph filter coefficients. This formulation highlights the parametric structure of the ARMA model, in which both temporal and graph dependencies are captured through a finite set of coefficients.

While the ARMA formulation provides a flexible and expressive modeling framework, this thesis does not pursue autoregressive recursions. Instead, we focus on moving-average (MA) type parametric representations, which lead to structured second-order models that are particularly convenient for learning and interpolation from partially observed data.

Accordingly, we consider Moving-Average (MA) type time-vertex processes of the form

$$\mathbf{x}_t = \sum_{q=0}^{Q-1} \mathbf{B}_q(\mathbf{L}_G)\, \mathbf{w}_{t-q}, \tag{2.29}$$

which describe the signal as the output of a finite-order graph filtering operation applied to the innovation sequence.

$$\mathbf{x}_t = \sum_{q=0}^{Q-1}\sum_{m=0}^{M-1} b_{qm}\, \mathbf{L}_G^m\, \mathbf{w}_{t-q}. \tag{2.30}$$

This formulation can be interpreted as a special case of the ARMA model in (2.28), obtained by omitting the autoregressive terms. When we take the JFT of both sides

of (2.30), we have the spectral representation of the joint filter as [9, 21]

$$h_{\mathrm{MA}}(\lambda_n, \omega_\tau) = \sum_{q=0}^{Q-1} \sum_{m=0}^{M-1} b_{qm}\, \lambda_n^m\, e^{-j\omega_\tau q} \tag{2.31}$$

where $n = 1, \ldots, N$ and $\tau = 1, \ldots, T$. The MA perspective admits a natural extension to the joint time-vertex spectral domain, where the filtering operation can be characterized through a parametric joint spectral kernel. This viewpoint directly motivates the joint MA modeling framework developed in the subsequent chapters of this thesis. It is worth noting that the parametric ARMA and MA formulations discussed above are typically developed under global stationarity assumptions in the joint time-vertex domain. Under this setting, the statistical properties of the process are assumed to remain invariant across time and graph domains, enabling compact spectral representations and tractable parametric models.

While global stationarity provides a convenient modeling framework, it may be restrictive in practical scenarios where the signal characteristics vary over time or across different regions of the graph. Such limitations motivate the development of more flexible models that can capture local variations in time-vertex processes, which constitutes the main focus of the proposed framework in this thesis.

Motivated by the limitations of globally stationary models, an alternative line of work has investigated *local stationarity* concepts in the graph domain, where second-order statistics are allowed to vary across the graph while remaining approximately stationary within local neighborhoods. A representative example is the locally stationary graph process framework proposed by Canbolat et al. [19], which provides a parametric description of spatially varying covariance structure through graph-based filtering operators. In this framework, the observed graph signal is modeled as a superposition of multiple locally weighted stationary components driven by a white innovation process. In the following, we briefly review the core formulation of this graph-only locally stationary model, as it serves as the main conceptual building block for the proposed time-vertex extension developed in Chapter 3.

Specifically, a locally stationary graph process can be written as [19]:

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{G}_k\, \mathbf{U}_G\, h_k(\mathbf{\Lambda}_G)\, \mathbf{U}_G^\top\, \mathbf{w}, \tag{2.32}$$

where $\mathbf{w} \in \mathbb{R}^N$ denotes a zero-mean white noise process with unit variance, and $\mathbf{U}_G$ and $\mathbf{\Lambda}_G$ correspond to the eigenvector and eigenvalue matrices of the graph Laplacian, respectively. The function $h_k(\mathbf{\Lambda}_G)$ represents a graph spectral filter. The matrices $\mathbf{G}_k \in \mathbb{R}^{N \times N}$ are diagonal membership matrices of the form

$$\mathbf{G}_k = \mathrm{diag}(\mathbf{g}_k), \tag{2.33}$$

where $\mathbf{g}_k \in \mathbb{R}^N$ is a nonnegative membership vector associated with the k-th component process. The n-th entry of $\mathbf{g}_k$ specifies the relative contribution of the k-th stationary component at node n. Through these membership vectors, the model captures spatially varying behavior by allowing different stationary components to dominate in different regions of the graph.

It is important to note that the locally stationary graph process model proposed by Canbolat et al. [19] is formulated solely in the graph domain and does not account for temporal dynamics. In this framework, local stationarity is defined with respect to the graph structure, allowing the statistical behavior of the process to vary across different regions of the graph while remaining time-invariant.

While this graph-only formulation provides a powerful tool for modeling spatially heterogeneous processes, it is not sufficient for signals whose characteristics also evolve over time. In many practical applications, both temporal variability and graph-dependent behavior are inherently coupled. This observation motivates the extension of local stationarity concepts to the joint time–vertex domain, which constitutes the main focus of the proposed framework developed in this thesis.

Overall, parametric methods offer interpretability and strong statistical foundations but often require accurate estimation of spectral parameters or neighborhood structures. Their performance can degrade when global stationarity assumptions are violated or when the underlying graph topology is not well aligned with the observed data.

2.4.3 Learning Based Methods

More recently, deep learning has played a pivotal role in advancing time–vertex reconstruction. Neural architectures such as unrolling networks [30, 36] emulate it-

erative optimization procedures and have demonstrated strong reconstruction performance, particularly in scenarios with structured noise or nonlinear relationships. Graph neural networks (GNNs) have also been adapted to spatio-temporal data, including space–time GNNs [35] and time-varying GNN-based recovery models [37, 38, 64]. Related works have further incorporated multiscale temporal representations within GNN architectures to efficiently capture time-varying dynamics on graphs [68]. Beyond neural network–based architectures, alternative learning paradigms rooted in dynamical systems have also been explored for modeling time-varying graph signals.

More recently, diffusion-based learning frameworks have emerged as a powerful alternative for modeling complex dependencies in graph-structured data. By leveraging stochastic diffusion processes, these methods have been applied to graph representation learning, generative modeling, and spatio-temporal graph forecasting, enabling flexible modeling of uncertainty and multi-scale dynamics on graphs [65, 66, 67].

In particular, Koopman operator-based methods learn a linear representation of nonlinear temporal dynamics in a lifted space, enabling data-driven modeling of graph signal evolution [62]. While effective for capturing complex time-varying behaviors, these approaches are primarily geared toward modeling system evolution and prediction, rather than learning structured covariance models for statistical inference.

Despite their empirical success, recent surveys have pointed out that purely data-driven learning approaches may suffer from limited interpretability and reduced robustness when domain structure is not explicitly enforced [31]. This observation has motivated growing interest in structure-aware and model-driven learning paradigms, which incorporate graph signal processing principles to balance flexibility with statistical interpretability.

Before the recent surge of deep learning models, several structure-aware learning paradigms laid the foundations for incorporating graph geometry into data-driven inference. Graph-based semi-supervised learning approaches exploit the intrinsic geometry of data by enforcing smoothness over a graph or manifold, an idea that has strongly influenced subsequent developments in graph signal modeling and inference [40]. Related semi-supervised learning formulations model the data as Gaussian fields defined over a graph and recover unknown signal values via harmonic func-

tions, thereby exploiting global smoothness with respect to the graph structure [41]. From a signal representation perspective, learning the underlying graph structure directly from data has been shown to be crucial for capturing meaningful dependencies in graph-supported signals [39].

2.4.4 Discussion

While learning-based methods often demonstrate strong empirical performance compared to classical techniques in practice, they typically require large training datasets, carefully designed architectures, and significant computational resources. Moreover, their interpretability is limited compared to parametric or model-driven approaches, and their generalization to new graphs or unseen temporal patterns is not guaranteed.

Consequently, existing interpolation approaches for time-vertex signals span a broad methodological spectrum. Non-parametric methods provide flexibility and ease of implementation, but rely heavily on smoothness assumptions that may not hold in heterogeneous or rapidly evolving settings. Parametric methods offer interpretability and a strong statistical grounding, yet often impose global stationarity assumptions or fixed model structures that limit their ability to capture local variability.

Notably, none of these families fully address the joint challenges posed by missing data, local variability across both space and time, and the need for structured parametric models that explicitly capture second-order statistical dependencies. These limitations motivate the development of more expressive frameworks, such as the locally stationary time-vertex parametric models introduced in Chapter 3, which aim to combine the interpretability of parametric approaches with the flexibility necessary to handle non-stationary graph-supported signals.

CHAPTER 3

LOCALLY STATIONARY TIME VERTEX PROCESS

In this chapter, we introduce a parametric modeling framework for *locally stationary time-vertex processes* (LSTVPs) that extends existing stationary and graph-only locally stationary models to jointly account for temporal variability and graph-dependent structure. Motivated by the limitations of globally stationary assumptions and the inability of graph-only local models to capture temporal evolution, the proposed framework allows the second-order statistics of time-vertex signals to vary smoothly across both time and graph domains.

Building upon the concepts reviewed in Chapter 2, we first present the formal definition and structural components of the proposed LSTVP model. We then characterize its second-order statistical properties and develop a learning framework for estimating the model parameters from partially observed data.

3.1 Stationarity in Graph Signal Processing

Classical graph signal models are commonly built upon the assumption of global stationarity. In the graph domain, this assumption leads to the notion of graph wide-sense stationary (GWSS) processes, where second-order statistics are invariant under graph shifts [6].

Definition 1 (Graph Wide-Sense Stationary (GWSS) Process)**.** *A random graph signal* $\mathbf{x} \in \mathbb{R}^N$ *is said to be* graph wide-sense stationary (GWSS) *if its first- and second-order statistics are invariant under graph shifts. Specifically,*

1. *the mean is constant over the vertices,*

$$\mathbb{E}[\mathbf{x}] = \mu\, \mathbf{1}_N, \tag{3.1}$$

2. *the covariance matrix is diagonalizable by the graph Fourier basis* $\mathbf{U}_G$*,*

$$\mathbf{C}_x = \mathbb{E}[\mathbf{x}\mathbf{x}^H] = \mathbf{U}_G\, h(\mathbf{\Lambda}_G)\, \mathbf{U}_G^H, \tag{3.2}$$

where $h(\mathbf{\Lambda}_G)$ *denotes the graph power spectral density (GPSD).*

This framework naturally extends to time-vertex signals through the concept of joint wide-sense stationarity (JWSS), which enforces invariance under both temporal and graph shifts [8].

Definition 2 (Joint Wide-Sense Stationary (JWSS) Process)**.** *A time-vertex signal* $\mathbf{x} \in \mathbb{R}^{NT}$ *is said to be* jointly wide-sense stationary (JWSS) *if its first- and second-order statistics are invariant under joint temporal and graph shifts [55]. Formally,*

1. *the mean is constant over time and vertices,*

$$\mathbb{E}[\mathbf{x}] = \mu\, \mathbf{1}_{NT}, \tag{3.3}$$

2. *the covariance matrix admits the joint diagonalization*

$$\mathbf{C}_x = \mathbb{E}[\mathbf{x}\mathbf{x}^H] = \mathbf{U}_J\, h(\mathbf{\Lambda}_G, \mathbf{\Omega}_T)\, \mathbf{U}_J^H, \tag{3.4}$$

where $\mathbf{U}_J \in \mathbb{C}^{NT\times NT}$ *and* $h(\mathbf{\Lambda}_G, \mathbf{\Omega}_T) \in \mathbb{C}^{NT\times NT}$ *denote the eigenvectors of the joint Laplacian and joint power spectral density (JPSD), respectively.*

3.2 Locally Stationary Time-Vertex Processes

While joint wide-sense stationary (JWSS) models provide a powerful representation for globally stationary time-vertex processes, they are inherently limited in capturing signals whose statistical properties evolve over time or vary across the graph. In many real-world applications, such nonstationarities arise due to localized temporal events, spatial heterogeneity, or slowly varying dynamics, rendering the global stationarity assumption overly restrictive. To address this limitation, we introduce a parametric modeling framework based on *local stationarity* in the joint time-vertex domain.

Definition 3 (Locally Stationary Time–Vertex Process)**.** *A stochastic time-vertex signal* $\mathbf{x} \in \mathbb{R}^{NT}$ *is said to be a* locally stationary time-vertex process (LSTVP) *if it admits the decomposition*

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{G}_k \mathbf{U}_J \mathbf{A}_k \mathbf{U}_J^{\mathrm{H}} \mathbf{w}, \tag{3.5}$$

where $\mathbf{w} \in \mathbb{R}^{NT}$ *denotes a zero-mean white noise vector with unit variance,* $\mathbf{U}_J$ *is the joint Fourier basis,* $\mathbf{G}_k = \mathrm{diag}(\mathbf{g}_k) \in \mathbb{R}^{NT \times NT}$ *is a diagonal membership matrix encoding the time-vertex localization of the* k*-th component, and* $\mathbf{A}_k \in \mathbb{C}^{NT \times NT}$ *is a diagonal matrix representing the joint spectral kernel associated with the* k*-th local stationary component.*

In this formulation, each component represents a locally stationary contribution defined over a specific region of the time-vertex domain. The diagonal operators $\mathbf{G}_k = \mathrm{diag}(\mathrm{vec}(g_k)) \in \mathbb{R}^{NT \times NT}$ encode *time-vertex membership functions* that modulate the spatial and temporal support of the kth component, thereby localizing its influence. The matrices $\mathbf{A}_k \in \mathbb{C}^{NT \times NT}$ are diagonal in the joint spectral domain and specify the joint filtering behavior associated with each local component.

Overall, the proposed LSTVP model represents a global time-vertex process as a superposition of K locally stationary components, each characterized by a distinct joint spectral filter and a corresponding membership function. This structured representation induces a parametric covariance model that is particularly well suited for learning from incomplete or partially observed time-vertex data, as developed in the subsequent sections.

3.3 Components of the Proposed Process Model

3.3.1 Joint Filter

To explicitly relate the joint spectral kernel to its matrix representation, we define the joint spectral filter $\mathbf{A}_k$ as a diagonal operator whose entries are given by the vectorized joint frequency response.

$$\mathbf{A}_k = \mathrm{diag}(\mathrm{vec}(h_k(\mathbf{\Lambda}_G, \mathbf{\Omega}_T))) \tag{3.6}$$

Here, $h_k(\mathbf{\Lambda}_G, \mathbf{\Omega}_T) \in \mathbb{C}^{N\times T}$ denotes the joint spectral kernel evaluated over all graph and temporal frequency pairs. The corresponding joint filtering operator is constructed by vectorizing this kernel and embedding it into a diagonal matrix $\mathbf{A}_k$.

To obtain a learnable representation, each locally stationary component is modeled using a joint moving–average (MA) parametrization in the spectral domain. This parametrization corresponds to a finite-order process model defined in the time-vertex domain. Specifically, by decomposing the global process in (3.5) into its K locally stationary components as $\mathbf{x} = \sum_{k=1}^{K} \mathbf{x}^{(k)}$, the value of the k-th local component at time t, denoted by $\mathbf{x}_t^{(k)}$, is modeled in this work using the following time–vertex domain representation:

$$\mathbf{x}_t^{(k)} = \sum_{q=0}^{Q-1} \sum_{m=0}^{M-1} b_{qmk}\, \mathbf{L}_G^m \mathbf{w}_{t-q}, \tag{3.7}$$

where $\mathbf{L}_G$ denotes the graph Laplacian, $\mathbf{w}_t$ is a zero–mean temporal white noise process, b_{qmk} are the joint MA filter coefficients, M controls the polynomial order in the graph spectral variable, and Q determines the temporal order of the filter.

Taking the joint Fourier transform of (3.7) yields a separable polynomial structure in the joint spectral domain. As a result, the joint spectral kernel $h_k(\lambda_n, \omega_\tau)$ evaluated at the joint frequency pair (λ_n, ω_τ) is given by [9, 21]

$$h_k(\lambda_n, \omega_\tau) = \sum_{q=0}^{Q-1} \sum_{m=0}^{M-1} b_{qmk}\, \lambda_n^m e^{-j\omega_\tau q}. \tag{3.8}$$

3.3.2 Time-Vertex Membership Functions

The localization of each component in the proposed LSTVP model is governed by the time-vertex membership matrices $\mathbf{G}_k$. Each membership matrix is defined as a diagonal operator of the form

$$\mathbf{G}_k = \mathrm{diag}(\mathbf{g}_k) \in \mathbb{R}^{NT\times NT}, \tag{3.9}$$

where $\mathbf{g}_k \in \mathbb{R}^{NT}$ is a membership function defined over the joint time-vertex domain. The ith entry of $\mathbf{g}_k$ quantifies the contribution of the kth locally stationary component at the corresponding time-vertex location.

Beyond indicating the degree of activation of each local component over the time-vertex domain, the membership matrix $\mathbf{G}_k$ also acts as a weighting mechanism that modulates the amplitude of the observed signal. Specifically, each diagonal entry of $\mathbf{G}_k$ scales the signal value at the corresponding time-vertex location, thereby controlling the relative influence of the k-th locally stationary component on the overall signal magnitude.

While this representation is expressive, directly learning the diagonal entries of $\{\mathbf{G}_k\}$ would require estimating NT parameters per component, which is prohibitive in high-dimensional or incomplete observation settings. To obtain a compact and learnable formulation in high–dimensional or incomplete observation settings, we introduce a structured parametrization of the membership functions. Rather than directly estimating the diagonal entries of each membership matrix, which would require learning NT parameters per component, we adopt a dictionary-based representation inspired by prior work on learning locally stationary graph processes [20].

In particular, following the data-driven modeling approach proposed by Canbolat et al. [20], we leverage the observation that graph signal realizations often exhibit dominant structural patterns that are closely related to the underlying regions of local stationarity. From this perspective, both the observed signals and the associated membership functions are shaped by the same latent time-vertex structure, suggesting that dictionary atoms learned directly from the signal realizations can serve as effective prototypes for time-vertex localization patterns. This insight motivates the use of a signal-adapted dictionary to parametrize the membership functions in a compact and interpretable manner. Specifically, each membership matrix is defined as

$$\mathbf{G}_k = \mathrm{diag}(\mathbf{g}_k) = \mathrm{diag}(\mathbf{D}\mathbf{v}_k), \qquad \mathbf{D} \in \mathbb{R}^{NT\times P},\ \mathbf{v}_k \in \mathbb{R}^P, \tag{3.10}$$

where $\mathbf{D}$ denotes a dictionary matrix whose columns represent prototypical time-vertex localization patterns. The dictionary $\mathbf{D}$ is learned offline from the available (possibly incomplete) observations using the K-SVD algorithm [17, 18], while $\mathbf{v}_k$ contains the expansion coefficients associated with the k-th locally stationary component.

Unlike hard segmentation or clustering approaches, the proposed formulation allows for *soft memberships*, whereby multiple components may contribute to the signal

at a given time-vertex point with different weights. This flexibility is essential for modeling smoothly varying nonstationary behavior across time and graph vertices, as commonly encountered in real-world time-vertex signals.

To promote local stationarity, the membership functions are encouraged to vary smoothly over the joint time-vertex domain. This smoothness is quantified through the quadratic form induced by the joint Laplacian, which measures the rate of variation of each membership function across time and graph dimensions. Specifically, we consider the inequality

$$g_k^\top \mathbf{L}_J g_k \leq C, \tag{3.11}$$

where $\mathbf{L}_J$ denotes the joint Laplacian operator and C represents an upper bound on the allowable joint variation of the k-th membership function. Rather than imposing a rigid pointwise constraint, this bound acts as a regularization budget that limits the overall rate of change of the membership functions, thereby favoring slowly varying memberships and approximately stationary behavior within each local region. In contrast to globally stationary models, the LSTVP framework allows the second-order statistics of the signal to adapt smoothly across both time and graph vertices, thereby providing a more expressive representation for complex real-world time-vertex processes.

3.4 Theoretical Covariance of the LSTVP

The primary objective of this thesis is to learn a parametric model that captures the second-order statistical structure of nonstationary time-vertex signals. We focus on learning a compact set of model parameters that govern the underlying covariance structure of the process. This perspective is particularly well suited for scenarios involving incomplete or partially observed data, where reliable estimation of second-order statistics enables principled signal reconstruction.

Within the proposed LSTVP framework, the signal is modeled as a superposition of locally stationary components characterized by joint spectral filters and time-vertex membership functions. As a result, the model induces a structured and parameterized covariance matrix whose form is fully determined by the unknown filter and

membership parameters. Learning the model therefore amounts to estimating these parameters such that the theoretical covariance implied by the LSTVP model matches the empirical covariance computed from the observed data.

In this section, we derive the theoretical covariance matrix of the proposed LSTVP model starting from the generative signal representation. This derivation forms the foundation of the subsequent learning formulation, where model parameters are estimated by minimizing a covariance mismatch criterion. The effectiveness of the learned parametric model is ultimately evaluated through the task of estimating missing observations, as demonstrated in the experimental results in Chapter 4.

We begin by rewriting the proposed LSTVP signal model in a more compact and convenient form to facilitate the derivation of its second-order statistics. Recall that each locally stationary component is filtered in the joint spectral domain through the operator $\mathbf{U}_J\mathbf{A}_k\mathbf{U}_J^H$. For notational simplicity, we define the corresponding joint filtering operator as

$$\mathbf{H}_k \triangleq \mathbf{U}_J\mathbf{A}_k\mathbf{U}_J^H. \tag{3.12}$$

Using this definition, the LSTVP signal model in (3.5) can be equivalently expressed as

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{G}_k\mathbf{H}_k\mathbf{w}, \tag{3.13}$$

where $\mathbf{w} \in \mathbb{R}^{NT}$ denotes a zero-mean white noise process with unit variance. To further simplify the representation, we collect the contributions of all locally stationary components into a single linear operator. Specifically, we define

$$\mathbf{\Gamma} \triangleq \sum_{k=1}^{K} \mathbf{G}_k\mathbf{H}_k, \tag{3.14}$$

which aggregates the effects of the joint spectral filters and the corresponding time-vertex membership functions. With this definition, the generative signal model reduces to the compact form

$$\mathbf{x} = \mathbf{\Gamma}\mathbf{w}. \tag{3.15}$$

This representation makes explicit the linear transformation that maps the driving white noise process to the observed time-vertex signal. As a result, the second-order statistics of $\mathbf{x}$ are fully characterized by the operator $\mathbf{\Gamma}$. In the following, we leverage this structure to derive the theoretical covariance matrix of the proposed LSTVP

model. Based on the compact signal representation in (3.15), the theoretical covariance matrix of the time-vertex signal $\mathbf{x}$ is given by

$$\mathbf{C}_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right] = \mathbb{E}\left[\mathbf{\Gamma}\mathbf{w}\mathbf{w}^H\mathbf{\Gamma}^H\right]. \tag{3.16}$$

In this work, the driving process $\mathbf{w}$ is assumed to be a zero-mean white Gaussian noise vector with unit variance. Under this assumption, the noise covariance satisfies

$$\mathbb{E}\left[\mathbf{w}\mathbf{w}^H\right] = \mathbf{I}_{NT}, \tag{3.17}$$

where $\mathbf{I}_{NT}$ denotes the $NT \times NT$ identity matrix [32]. Substituting (3.17) into (3.16), the theoretical covariance matrix of the LSTVP signal simplifies to

$$\mathbf{C}_x = \mathbf{\Gamma}\mathbf{\Gamma}^H. \tag{3.18}$$

To make the dependence of the theoretical covariance on the model parameters explicit, we expand $\mathbf{\Gamma}\mathbf{\Gamma}^H$ using the definition in (3.14). Recall that $\mathbf{\Gamma} = \sum_{k=1}^{K} \mathbf{G}_k\mathbf{H}_k$; therefore,

$$\mathbf{C}_x = \mathbf{\Gamma}\mathbf{\Gamma}^H = \left(\sum_{k=1}^{K} \mathbf{G}_k\mathbf{H}_k\right)\left(\sum_{\ell=1}^{K} \mathbf{G}_\ell\mathbf{H}_\ell\right)^H \tag{3.19}$$

$$= \left(\sum_{k=1}^{K} \mathbf{G}_k\mathbf{H}_k\right)\left(\sum_{\ell=1}^{K} \mathbf{H}_\ell^H\mathbf{G}_\ell^H\right) \tag{3.20}$$

$$= \sum_{k=1}^{K}\sum_{\ell=1}^{K} \mathbf{G}_k\mathbf{H}_k\mathbf{H}_\ell^H\mathbf{G}_\ell^H. \tag{3.21}$$

Since each membership matrix is diagonal and real-valued, we have $\mathbf{G}_\ell^H = \mathbf{G}_\ell \in \mathbb{R}^{NT\times NT}$. Moreover, using the definition $\mathbf{H}_k = \mathbf{U}_J\mathbf{A}_k\mathbf{U}_J^H$ from (3.12), the cross-term $\mathbf{H}_k\mathbf{H}_\ell^H$ becomes

$$\mathbf{H}_k\mathbf{H}_\ell^H = \left(\mathbf{U}_J\mathbf{A}_k\mathbf{U}_J^H\right)\left(\mathbf{U}_J\mathbf{A}_\ell\mathbf{U}_J^H\right)^H \tag{3.22}$$

$$= \mathbf{U}_J\mathbf{A}_k\mathbf{U}_J^H\mathbf{U}_J\mathbf{A}_\ell^H\mathbf{U}_J^H = \mathbf{U}_J\mathbf{A}_k\mathbf{A}_\ell^H\mathbf{U}_J^H, \tag{3.23}$$

where we used the unitarity of the joint Fourier basis, $\mathbf{U}_J^H\mathbf{U}_J = \mathbf{I}_{NT\times NT}$. Substituting (3.23) into (3.21), we obtain the parameterized covariance structure

$$\mathbf{C}_x = \mathbf{\Gamma}\mathbf{\Gamma}^H = \sum_{k=1}^{K}\sum_{\ell=1}^{K} \mathbf{G}_k\mathbf{U}_J\left(\mathbf{A}_k\mathbf{A}_\ell^H\right)\mathbf{U}_J^H\mathbf{G}_\ell. \tag{3.24}$$

We emphasize that the covariance structure in (3.24) is parametric in nature, as neither the joint spectral matrices $\mathbf{A}_k$ nor the membership matrices $\mathbf{G}_k$ are free variables. The matrices $\mathbf{A}_k$ are fully determined by the finite set of moving-average coefficients $\{b_{qmk}\}$ defining the joint MA kernels introduced in (3.8), while the matrices $\mathbf{G}_k = \mathrm{diag}(\mathbf{g}_k)$ are parameterized through the corresponding membership functions $\mathbf{g}_k$ subject to smoothness priors over the joint domain.

3.5 Empirical Covariance Estimation

The theoretical covariance matrix derived in the previous section provides a parametric description of the proposed LSTVP model. In practice, however, the true covariance of the underlying process is unknown and must be estimated from a finite set of observed signal realizations. This empirical estimate serves as the reference against which the model-based covariance is matched during the learning stage.

Let $\{\mathbf{x}^{(r)}\}_{r=1}^{R}$ denote R independent realizations of a time-vertex signal generated by the same underlying process, where each realization is vectorized as $\mathbf{x}^{(r)} \in \mathbb{R}^{NT}$. Assuming zero-mean observations, the empirical covariance matrix is computed as

$$\widehat{\mathbf{C}}_x = \frac{1}{R} \sum_{r=1}^{R} \mathbf{x}^{(r)} \mathbf{x}^{(r)H}. \tag{3.25}$$

As the number of realizations R increases, the empirical covariance $\widehat{\mathbf{C}}_x$ converges to the true covariance $\mathbf{C}_x$ under standard ergodicity assumptions. In the finite-sample regime, however, the estimate may be noisy or incomplete, particularly in scenarios involving missing observations. The goal of the proposed learning framework is therefore to estimate the model parameters such that the structured theoretical covariance induced by the LSTVP model closely matches the empirical covariance computed from the available data.

In practice, rather than directly using the raw sample covariance matrix $\widehat{\mathbf{C}}_x$ as input to the learning algorithm, we exploit prior structural assumptions to obtain a more robust covariance estimate. Specifically, we compute a covariance matrix that is constrained to satisfy the joint wide-sense stationarity (JWSS) property, following the framework introduced in [8]. This amounts to projecting the empirical covariance onto the set

of matrices that are diagonalizable by the joint Fourier basis $\mathbf{U}_J \in \mathbb{C}^{NT\times NT}$. More precisely, the set of JWSS covariance matrices can be written as [8]

$$\mathbf{C}_{\mathrm{JWSS}} = \left\{\mathbf{C} \in \mathbb{C}^{NT\times NT} \;:\; \mathbf{C} = \mathbf{U}_J \,\mathrm{diag}(\mathbf{p})\mathbf{U}_J^{\mathrm{H}},\ \mathbf{p} \in \mathbb{C}^{NT}\right\}. \tag{3.26}$$

Given the empirical covariance $\widehat{\mathbf{C}}_x$, its projection onto $\mathcal{C}_{\mathrm{JWSS}}$ is obtained by retaining only the diagonal of the joint spectral representation:

$$\widehat{\mathbf{C}}_{\mathrm{JWSS}} = \mathbf{U}_J \;\mathrm{diag}_{\mathrm{M}}\Big(\;\mathrm{diag}_{\mathrm{V}}\left(\mathbf{U}_J^{\mathrm{H}}\widehat{\mathbf{C}}_x\mathbf{U}_J\right)\Big)\,\mathbf{U}_J^{\mathrm{H}} \tag{3.27}$$

where the $diag_V(\cdot)$ operation denotes the extraction of the diagonal components of a matrix, and the $diag_M(\cdot)$ operation refers to a diagonal matrix that contains the given vector on its diagonal. In the proposed learning framework, this structured covariance matrix is used as the input to the covariance $\widehat{\mathbf{C}}_x$ matching objective, leading to improved numerical stability and robustness compared to directly fitting the unstructured sample covariance.

It is worth emphasizing that, although the proposed LSTVP framework aims to learn a locally stationary time-vertex model, the covariance matrix used as input to the learning algorithm is globally constrained to satisfy the JWSS property. From a modeling perspective, this choice can be interpreted as a form of implicit regularization. By projecting the empirical covariance onto the JWSS subspace, the learning process is guided by a globally stationary second-order structure, preventing the model from overfitting to excessive local variations in the data.

As a result, the learned membership functions and joint filter kernels are encouraged to capture localized deviations from a global stationary baseline, rather than arbitrarily adapting to noise or spurious fluctuations. In this sense, the proposed approach can be viewed as learning a local model that remains anchored to a globally stationary covariance structure, striking a balance between model flexibility and robustness.

3.6 Learning Framework for the LSTVP Model

Having derived the model-induced theoretical covariance $\mathbf{C}_x$ in Section 3.4 and the empirical covariance estimate $\widehat{\mathbf{C}}_x$ from the available realizations in Section 3.5, we now formulate a learning framework to estimate the unknown LSTVP parameters.

The key idea is to exploit the structured covariance representation implied by the proposed generative model: the joint spectral filters and the time-vertex membership functions jointly determine $\mathbf{C}_x$ through a finite set of parameters. We therefore learn the model by minimizing a covariance mismatch criterion that encourages the theoretical covariance to match the empirical covariance, while incorporating regularization terms and constraints that promote smooth memberships and physically meaningful parameter estimates. This section presents the resulting optimization problem and the algorithmic strategy used to solve it.

The core learning objective is to fit the model-induced covariance $\mathbf{C}_x = \mathbf{\Gamma}\mathbf{\Gamma}^{\mathrm{H}}$ to the empirical covariance estimate $\widehat{\mathbf{C}}_x$, where $\mathbf{\Gamma}\mathbf{\Gamma}^{\mathrm{H}}$ depends on the collection of unknown LSTVP parameters, namely the joint moving-average coefficients $\{b_{qmk}\}$ and $\{v_k\}$ defining the membership functions $\{\mathbf{g}_k\}$. A natural data-fitting criterion is therefore given by the covariance mismatch

$$\arg\min \left\| \widehat{\mathbf{C}}_x - \mathbf{\Gamma}\mathbf{\Gamma}^{\mathrm{H}} \right\|_F^2, \tag{3.28}$$

Recall from (3.18) that the theoretical covariance of the proposed LSTVP model is given by

$$\mathbf{C}_x = \mathbf{\Gamma}\mathbf{\Gamma}^{H}, \tag{3.29}$$

with $\mathbf{\Gamma} = \sum_{k=1}^{K} \mathbf{G}_k \mathbf{H}_k$ and $\mathbf{H}_k = \mathbf{U}_J \mathbf{A}_k \mathbf{U}_J^H$.

The following lemma will be useful in our derivations [69]

Lemma 1. *For any matrices* $\mathbf{Y}, \mathbf{B} \in \mathbb{R}^{N\times N}$ *and diagonal matrices* $\mathbf{D}, \mathbf{E} \in \mathbb{R}^{N\times N}$*, the following identity holds*

$$\mathbf{D}(\mathbf{Y} \circ \mathbf{B})\mathbf{E} = \mathbf{Y} \circ (\mathbf{DBE}), \tag{3.30}$$

where $\circ$ *denotes the Hadamard (element-wise) product.*

We next use Lemma 1 to rewrite the theoretical covariance in a form that explicitly separates the joint spectral filtering structure from the time-vertex membership weighting. Recall from (3.24) that the covariance admits

$$\mathbf{C}_x = \sum_{k=1}^{K} \sum_{\ell=1}^{K} \mathbf{G}_k \, \mathbf{U}_J \left(\mathbf{A}_k \mathbf{A}_\ell^H \right) \mathbf{U}_J^H \mathbf{G}_\ell. \tag{3.31}$$

Since $\mathbf{G}_k$ and $\mathbf{G}_\ell$ are diagonal membership matrices, the (i,j)th entry of the term $\mathbf{G}_k\mathbf{M}\mathbf{G}_\ell$ can be written as

$$[\mathbf{G}_k\mathbf{M}\mathbf{G}_\ell]_{i,j} = [\mathbf{G}_k]_{i,i}\ [\mathbf{M}]_{i,j}\ [\mathbf{G}_\ell]_{j,j} = [\mathbf{M}]_{i,j}\ [\mathbf{G}_k\mathbf{1}_{NT\times NT}\mathbf{G}_\ell]_{i,j}\,, \tag{3.32}$$

where $\mathbf{1}_{NT\times NT}$ denotes the all-ones matrix. The identity in (3.32) is precisely the element-wise modulation captured by Lemma 1. Therefore, by setting

$$\mathbf{Y} = \mathbf{U}_J(\mathbf{A}_k\mathbf{A}_\ell^H)\mathbf{U}_J^H, \quad \mathbf{B} = \mathbf{1}_{NT\times NT}, \quad \mathbf{D} = \mathbf{G}_k, \quad \mathbf{E} = \mathbf{G}_\ell,$$

Lemma 1 yields the equivalent representation

$$\mathbf{G}_k\,\mathbf{U}_J(\mathbf{A}_k\mathbf{A}_\ell^H)\mathbf{U}_J^H\mathbf{G}_\ell = \big(\mathbf{U}_J(\mathbf{A}_k\mathbf{A}_\ell^H)\mathbf{U}_J^H\big) \circ \big(\mathbf{G}_k\mathbf{1}_{NT\times NT}\mathbf{G}_\ell\big). \tag{3.33}$$

Substituting (3.33) into (3.31), we obtain

$$\mathbf{C}_x = \sum_{k=1}^{K}\sum_{\ell=1}^{K} \big(\mathbf{U}_J(\mathbf{A}_k\mathbf{A}_\ell^H)\mathbf{U}_J^H\big) \circ \big(\mathbf{G}_k\mathbf{1}_{NT\times NT}\mathbf{G}_\ell\big), \tag{3.34}$$

which makes explicit that each component pair (k,ℓ) contributes through an element-wise weighting of a joint spectral covariance kernel by time-vertex membership interactions.

The decomposition in (3.34) highlights that the effect of local stationarity is entirely captured by the time-vertex membership interactions $\mathbf{G}_k\mathbf{1}_{NT\times NT}\mathbf{G}_\ell$. We now express the diagonal joint spectral operators $\{\mathbf{A}_k\}$ in terms of the joint MA parameters. For each joint frequency pair (λ_n,ω_τ), we define the joint basis vector as [21]

$$\mathbf{u}_{n,\tau} \triangleq \big[\lambda_n^0 e^{-j\omega_\tau 0},\ \ldots,\ \lambda_n^{M-1}e^{-j\omega_\tau(Q-1)}\big]^\top \in \mathbb{C}^{QM}, \tag{3.35}$$

and the stacked coefficient vector

$$\mathbf{b}_k \triangleq \big[b_{0,0,k},\ \ldots,\ b_{Q-1,M-1,k}\big]^\top \in \mathbb{R}^{QM}. \tag{3.36}$$

With these definitions, the joint MA kernel in (2.31) can be written compactly as the scalar inner product

$$h_k(\lambda_n,\omega_\tau) = \mathbf{u}_{n,\tau}^\top\mathbf{b}_k. \tag{3.37}$$

Recall that each diagonal matrix $\mathbf{A}_k$ is formed from the vectorized kernel values over all joint frequency pairs. Under the column-wise indexing $i = (\tau-1)N + n$, the

(i,i)th entry of $\mathbf{A}_k\mathbf{A}_\ell^H$ satisfies

$$\left[\mathbf{A}_k\mathbf{A}_\ell^H\right]_{i,i} = h_k(\lambda_n,\omega_\tau)\, h_\ell(\lambda_n,\omega_\tau)^H \tag{3.38}$$

$$= \left(\mathbf{u}_{n,\tau}^\top \mathbf{b}_k\right)\left(\mathbf{u}_{n,\tau}^\top \mathbf{b}_\ell\right)^H \tag{3.39}$$

$$= \mathbf{u}_{n,\tau}^\top \mathbf{b}_k \mathbf{b}_\ell^H \mathbf{u}_{n,\tau}^*, \tag{3.40}$$

where $(\cdot)^*$ denotes the complex conjugate. By combining the membership parametrization $\mathbf{G}_k = \mathrm{diag}(\mathbf{D}\mathbf{v}_k)$ which is defined in (3.10) with the joint MA filter parametrization $h_k(\lambda_n,\omega_\tau) = \mathbf{u}_{n,\tau}^\top \mathbf{b}_k$, the theoretical covariance $\mathbf{C}_x = \mathbf{\Gamma}\mathbf{\Gamma}^H$ in (3.18) becomes a fourth-order function of the optimization variables. Consequently, the covariance fitting objective in (3.28) is a quartic (and thus non-convex) optimization problem with respect to the unknown membership coefficients $\{\mathbf{v}_k\}$ and filter coefficients $\{\mathbf{b}_k\}$.

To obtain a tractable learning formulation, we adopt a convex relaxation via lifting by stacking the component-wise parameters into global vectors. Similar convex relaxation strategies based on lifting have been explored in prior work for learning structured graph process models; see, e.g., [19, 21]. Building upon these ideas, we adopt a lifting-based convex relaxation that is specifically tailored to the proposed LSTVP model, where both the joint MA coefficients and the time-vertex membership parameters are embedded into global lifted variables. Let

$$\mathbf{v} = \left[\mathbf{v}_1^\top\ \mathbf{v}_2^\top\ \cdots\ \mathbf{v}_K^\top\right]^\top \in \mathbb{R}^{PK}, \qquad \mathbf{b} = \left[\mathbf{b}_1^\top\ \mathbf{b}_2^\top\ \cdots\ \mathbf{b}_K^\top\right]^\top \in \mathbb{R}^{QMK}. \tag{3.41}$$

We then define the lifted matrix variables

$$\mathbf{V} \triangleq \mathbf{v}\mathbf{v}^\top \in \mathbb{R}^{PK\times PK}, \qquad \mathbf{B} \triangleq \mathbf{b}\mathbf{b}^\top \in \mathbb{R}^{QMK\times QMK}. \tag{3.42}$$

Recall from (3.34) that the theoretical covariance can be expressed through a Hadamard product, where the second factor involves the membership interaction term $\mathbf{G}_k\mathbf{1}_{NT\times NT}\mathbf{G}_\ell$. Since $\mathbf{G}_k$ and $\mathbf{G}_\ell$ are diagonal, we have

$$\mathbf{G}_k\mathbf{1}_{NT\times NT}\mathbf{G}_\ell = \mathbf{g}_k\mathbf{g}_\ell^\top. \tag{3.43}$$

Using the dictionary parametrization $\mathbf{g}_k = \mathbf{D}\mathbf{v}_k$ in (3.10), it follows that

$$\mathbf{g}_k\mathbf{g}_\ell^\top = \mathbf{D}\mathbf{v}_k\mathbf{v}_\ell^\top\mathbf{D}^\top. \tag{3.44}$$

Let $\mathbf{Z}_k \in \mathbb{R}^{P\times PK}$ denote the block-selection operator that extracts the kth P-dimensional block of $\mathbf{v}$, i.e.,

$$\mathbf{v}_k = \mathbf{Z}_k\mathbf{v}, \tag{3.45}$$

where $\mathbf{Z}_k = [\mathbf{0} \ \cdots \ \mathbf{I}_P \ \cdots \ \mathbf{0}]$ with $\mathbf{I}_P$ placed at the kth block position. Then,

$$\mathbf{v}_k \mathbf{v}_\ell^\top = (\mathbf{Z}_k \mathbf{v})(\mathbf{Z}_\ell \mathbf{v})^\top = \mathbf{Z}_k \mathbf{V} \mathbf{Z}_\ell^\top. \tag{3.46}$$

Substituting (3.46) into (3.44), the membership interaction term becomes

$$\mathbf{G}_k \mathbf{1}_{NT \times NT} \mathbf{G}_\ell = \mathbf{g}_k \mathbf{g}_\ell^\top = \mathbf{D} \mathbf{Z}_k \mathbf{V} \mathbf{Z}_\ell^\top \mathbf{D}^\top. \tag{3.47}$$

In an analogous manner, we express the joint spectral interaction term $\mathbf{A}_k \mathbf{A}_\ell^H$ in terms of a lifted matrix variable $\mathbf{B}$. Let $\mathbf{T}_k \in \mathbb{R}^{QM \times QMK}$ denote the block-selection operator whose kth block is the identity, i.e., $\mathbf{T}_k = [\mathbf{0} \ \cdots \ \mathbf{I}_{QM} \ \cdots \ \mathbf{0}]$, so that $\mathbf{b}_k = \mathbf{T}_k \mathbf{b}$. Then, for any (k, ℓ),

$$\mathbf{b}_k \mathbf{b}_\ell^\top = (\mathbf{T}_k \mathbf{b})(\mathbf{T}_\ell \mathbf{b})^\top = \mathbf{T}_k \mathbf{B} \mathbf{T}_\ell^\top. \tag{3.48}$$

Next, define the matrix $\mathbf{W} \in \mathbb{C}^{QM \times NT}$ by stacking the joint basis vectors $\mathbf{u}_{n,\tau}$ as columns,

$$\mathbf{W} \triangleq \big[\mathbf{u}_{1,1} \ \mathbf{u}_{2,1} \ \cdots \ \mathbf{u}_{N,1} \ \mathbf{u}_{1,2} \ \cdots \ \mathbf{u}_{N,T}\big], \tag{3.49}$$

where the column ordering follows the vectorization index $i = (\tau - 1)N + n$. Using (3.37) and (3.48), the diagonal entries of $\mathbf{A}_k \mathbf{A}_\ell^H$ can be written as

$$\big[\mathbf{A}_k \mathbf{A}_\ell^H\big]_{i,i} = h_k(\lambda_n, \omega_\tau)\, h_\ell(\lambda_n, \omega_\tau)^* = \mathbf{u}_{n,\tau}^\top \big(\mathbf{T}_k \mathbf{B} \mathbf{T}_\ell^\top\big) \mathbf{u}_{n,\tau}^*. \tag{3.50}$$

Equivalently, collecting all diagonal elements yields

$$\mathbf{A}_k \mathbf{A}_\ell^H = \mathbf{I}_{NT} \circ \Big(\mathbf{W}^\top \big(\mathbf{T}_k \mathbf{B} \mathbf{T}_\ell^\top\big) \mathbf{W}^*\Big). \tag{3.51}$$

Substituting the membership interaction term in (3.47) and the joint filter interaction term in (3.51) into the Hadamard covariance representation in (3.34), we obtain the theoretical covariance explicitly in terms of the lifted variables $\mathbf{V}$ and $\mathbf{B}$ as

$$\mathbf{\Gamma}\mathbf{\Gamma}^H = \sum_{k=1}^{K} \sum_{\ell=1}^{K} \Big(\mathbf{U}_J \big(\mathbf{I}_{NT} \circ \big(\mathbf{W}^\top (\mathbf{T}_k \mathbf{B} \mathbf{T}_\ell^\top) \mathbf{W}^*\big)\big) \mathbf{U}_J^H\Big) \circ \Big(\mathbf{D} \mathbf{Z}_k \mathbf{V} \mathbf{Z}_\ell^\top \mathbf{D}^\top\Big). \tag{3.52}$$

The first factor inside the summation encodes the joint spectral covariance contribution induced by the MA filter parameters through $\mathbf{B}$, whereas the second factor captures the time-vertex localization imposed by the membership parameters through $\mathbf{V}$.

By substituting the lifted covariance model $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^H$ derived in (3.52) into the covariance fitting criterion in (3.28), we obtain an optimization problem over the lifted variables $\mathbf{V}$ and $\mathbf{B}$. Due to the bilinear coupling between these variables in $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^H$, the resulting objective is not jointly convex in $(\mathbf{V}, \mathbf{B})$. Nevertheless, the problem admits a biconvex structure: for a fixed $\mathbf{V}$, the covariance model becomes an affine (hence the mismatch becomes quadratic) function of $\mathbf{B}$, and the objective is convex in $\mathbf{B}$; similarly, for a fixed $\mathbf{B}$, the objective is convex in $\mathbf{V}$. This property motivates the use of an alternating optimization (block coordinate descent) strategy, where we iteratively update $\mathbf{B}$ while holding $\mathbf{V}$ fixed, and vice versa, until convergence (e.g., [28]). Accordingly, we define the following data-fitting objective:

$$\begin{aligned} f_1(\mathbf{V}, \mathbf{B}) = \min_{\mathbf{V}, \mathbf{B}} \quad & \left\| \widehat{\mathbf{C}}_x - \boldsymbol{\Gamma}\boldsymbol{\Gamma}^H \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{V} \succeq \mathbf{0}, \qquad \mathbf{B} \succeq \mathbf{0}. \end{aligned} \tag{3.53}$$

While the data-fitting objective $f_1(\mathbf{V}, \mathbf{B})$ enforces consistency between the model-induced covariance and the empirical estimate, it does not, by itself, impose any structural regularity on the learned membership functions. In particular, without additional constraints, the estimated memberships may exhibit abrupt variations across the time-vertex domain, which contradicts the assumption of local stationarity. To encode the prior knowledge that membership functions should vary smoothly over time and graph vertices, we introduce an additional regularization term that promotes smoothness of the membership coefficients.

Specifically, we define the following smoothness-promoting objective:

$$f_2(\mathbf{V}) = \sum_{k=1}^{K} \mathbf{v}_k^\top \mathbf{D}^\top \mathbf{L}_J \mathbf{D}\, \mathbf{v}_k, \tag{3.54}$$

which penalizes rapid variations of the corresponding membership functions $\mathbf{g}_k = \mathbf{D}\mathbf{v}_k$ over the joint time-vertex domain. Using the lifted variable $\mathbf{V} = \mathbf{v}\mathbf{v}^\top$, the objective in (3.54) can be equivalently expressed as

$$f_2(\mathbf{V}) = \sum_{k=1}^{K} \operatorname{tr}\left(\mathbf{Z}_k^\top \mathbf{D}^\top \mathbf{L}_J \mathbf{D}\, \mathbf{Z}_k \mathbf{V}\right), \tag{3.55}$$

which is linear in $\mathbf{V}$, as defined in (3.42).

In addition to data fidelity and smoothness regularization, the proposed learning framework seeks low-dimensional representations for both the membership and joint filter parameters. Specifically, since the lifted variables $\mathbf{V} = \mathbf{v}\mathbf{v}^\top$ and $\mathbf{B} = \mathbf{b}\mathbf{b}^\top$ are constructed from underlying low-dimensional coefficient vectors, the ideal solutions are expected to be low-rank. Promoting low-rank structure not only reduces the effective number of degrees of freedom, but also improves robustness and generalization, particularly in settings with limited or incomplete observations.

Motivated by these considerations, an ideal formulation of the learning problem can be expressed as

$$\begin{aligned} \min_{\mathbf{V},\mathbf{B}} \quad & f_1(\mathbf{V},\mathbf{B}) + \mu_1 f_2(\mathbf{V}) \\ \text{s.t.} \quad & \mathbf{V} \succeq \mathbf{0}, \qquad \mathbf{B} \succeq \mathbf{0}, \qquad \operatorname{rank}(\mathbf{V}) = 1, \qquad \operatorname{rank}(\mathbf{B}) = 1. \end{aligned} \tag{3.56}$$

The rank-one constraints in (3.56) enforce the lifted variables to admit factorizations of the form $\mathbf{V} = \mathbf{v}\mathbf{v}^\top$ and $\mathbf{B} = \mathbf{b}\mathbf{b}^\top$, which is consistent with the underlying parametric model. However, these rank constraints are non-convex and render the optimization problem computationally intractable in general. As a result, directly enforcing rank-one solutions is not feasible for large-scale problems.

To obtain a tractable formulation, we adopt a convex relaxation strategy in which the rank-one constraints are replaced by trace regularization. Under the positive semidefinite constraints, the trace function serves as a convex surrogate that promotes low-rank solutions. This leads to the following relaxed optimization problem:

$$\begin{aligned} \min_{\mathbf{V},\mathbf{B}} \quad & f_1(\mathbf{V},\mathbf{B}) + \mu_1 f_2(\mathbf{V}) + \mu_2 \operatorname{tr}(\mathbf{B}) + \mu_3 \operatorname{tr}(\mathbf{V}) \\ \text{s.t.} \quad & \mathbf{V} \succeq \mathbf{0}, \qquad \mathbf{B} \succeq \mathbf{0}, \end{aligned} \tag{3.57}$$

where μ_1, μ_2, and μ_3 are positive regularization parameters that control the trade-off between data fitting, membership smoothness, and model complexity.

3.7 Optimization Details and Complexity Analysis

The alternating optimization problem summarized in Algorithm 1 is implemented by solving a sequence of convex subproblems derived from the data-fitting formulation

in (3.53). Although the overall problem is not jointly convex in $(\mathbf{V}, \mathbf{B})$ in (3.57), fixing one variable renders the objective convex with respect to the other, resulting in semidefinite programs (SDPs) at each iteration.

The alternating optimization procedure requires an initial feasible point for the lifted variables $\mathbf{V}$ and $\mathbf{B}$. In practice, we initialize the algorithm from the membership-related variable $\mathbf{V}$, while setting $\mathbf{B}$ to a positive semidefinite unknown matrix. Specifically, $\mathbf{V}^{(0)}$ is initialized as a random positive semidefinite matrix scaled by the square root of the Frobenius norm of the structured covariance estimate $\widehat{\mathbf{C}}_x$. This data–adaptive scaling was found to yield stable convergence behavior in our experiments.

In practice, the convex subproblems arising in the proposed learning framework are solved using the CVX modeling environment, which provides a high-level interface for specifying disciplined convex optimization problems [25]. As underlying solvers, SDPT3 [29] or MOSEK [26] can be employed to handle the resulting semidefinite programs (SDPs) [11]. This solver–modeling combination enables the proposed approach to efficiently process time-vertex covariance matrices while enforcing the positive semidefinite constraints imposed on the lifted variables $\mathbf{V}$ and $\mathbf{B}$.

In this work, the MOSEK solver is primarily adopted due to its favorable numerical robustness and computational efficiency when dealing with large-scale SDPs [27]. Empirical observations indicate that MOSEK achieves faster convergence and improved numerical stability compared to alternative solvers, making it particularly well suited for the high-dimensional lifted covariance learning problems considered in this thesis.

The overall computational complexity of the proposed algorithm is dominated by the alternating optimization (AO) procedure. In particular, the updates of the matrices $\mathbf{V}$ and $\mathbf{B}$, with dimensions $PK \times PK$ and $QMK \times QMK$, respectively, constitute the most significant steps in terms of computation. These subproblems are addressed using interior-point based SDP solvers, whose computational cost depends polynomially on both the number of decision variables and the number of affine constraints.

Following standard complexity results based on the HKM algorithm [29], the worst-case complexity of interior-point SDP solvers scales as $\mathcal{O}(mn^3 + m^2n^2)$ with respect

to the number of affine constraints m and the number of decision variables n. In the proposed alternating optimization scheme, two dominant SDP subproblems are solved at each iteration.

The SDP associated with $\mathbf{V} \in \mathbb{R}^{(PK)\times(PK)}$ admits a worst-case computational complexity on the order of

$$\mathcal{O}\big(\mathrm{poly}(PK)\big), \tag{3.58}$$

whereas the SDP involving the joint filter covariance variable $\mathbf{B} \in \mathbb{R}^{(QMK)\times(QMK)}$ exhibits a worst-case complexity scaling as

$$\mathcal{O}\big(\mathrm{poly}(QMK)\big), \tag{3.59}$$

where $\mathrm{poly}(\cdot)$ denotes a polynomial of degree at least three. Although the worst-case complexity increases polynomially with the number of local components K, the filter orders Q and M, and the number of dictionary atoms P, it is important to note that the resulting complexity expressions in (3.58) and (3.59) are independent of the signal dimensions N and T. In practice, the parameters P, K, Q, and M are chosen such that $P, K, Q, M \ll N, T$, reflecting the use of compact local models and low-order joint filters. As a result, despite the polynomial dependence on these parameters, the overall computational burden remains manageable for moderate problem sizes.

Meanwhile, in addition to the solver-related complexity, a non-negligible portion of the computational cost arises from the construction and manipulation of symbolic optimization expressions during the alternating optimization procedure. In particular, the parametric covariance matrix (3.52) is formed using symbolic decision variables within the CVX framework, and the associated matrix operations, such as multiplications and linear mappings required to express the objective and constraints in solver-compatible form, introduce a certain computational overhead.

This modeling-related cost becomes more pronounced as the dimension of the time-vertex signal, given by $N \times T$, increases. Empirical observations indicate that the time required to construct and process these symbolic expressions grows with the signal dimension, even before invoking the underlying SDP solver. Consequently, the overall runtime of the proposed algorithm is influenced not only by the theoretical complexity of the SDP solver, but also by the cost of transforming high-dimensional parametric covariance models into a solver-ready representation.

In this chapter, we introduced a parametric learning framework for locally stationary time-vertex processes. Starting from a structured signal model, we derived the corresponding theoretical covariance representation and formulated a covariance-based learning objective to estimate the model parameters from data. By exploiting suitable parameterizations and convex relaxations, the resulting non-convex problem was cast into a biconvex form and solved using an alternating optimization strategy. The proposed approach provides a flexible yet tractable mechanism for capturing localized spatio-temporal statistical variations in time-vertex signals.

Figure 3.1 illustrates the overall pipeline of the proposed algorithm, highlighting the main processing stages from partially observed time-vertex signal realizations to the final signal reconstruction. Specifically, the figure summarizes the empirical JWSS covariance estimation, dictionary learning for membership parametrization, LSTVP model learning via alternating optimization, and the subsequent LMMSE-based signal reconstruction in a unified framework.

In the next chapter, we evaluate the proposed learning framework through a series of experiments on synthetic and real-world datasets. The experimental results assess the ability of the learned models to accurately represent time-vertex statistics and to enable effective estimation of missing observations under different operating conditions.

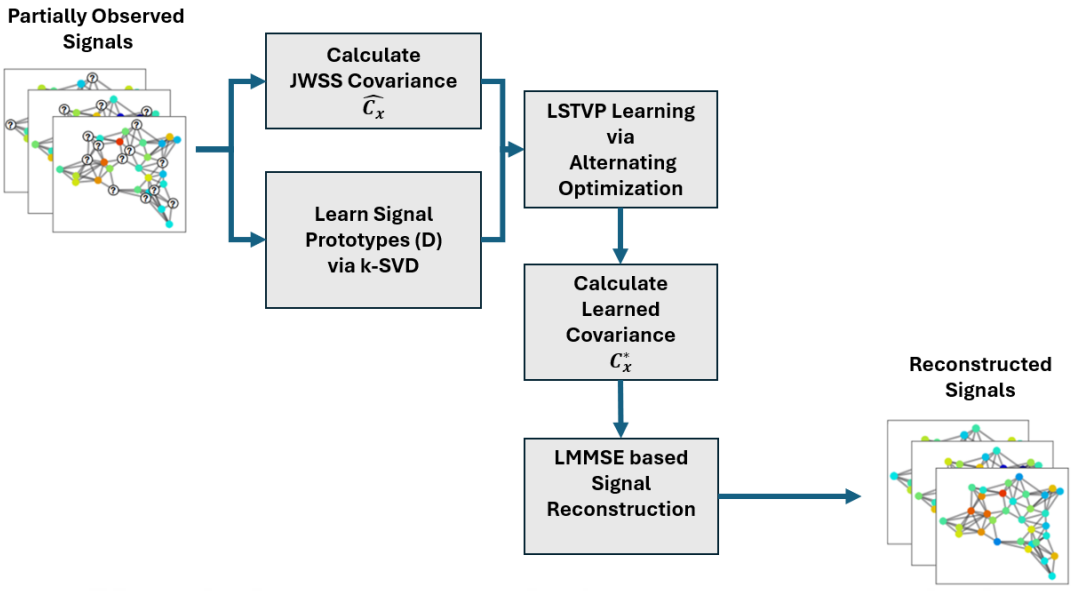


Figure 3.1: Overall pipeline of the proposed algorithm

Algorithm 1 Learning LSTVP Model Parameters via Alternating Optimization

Require: Structured covariance $\widehat{\mathbf{C}}_x$, dictionary matrix $\mathbf{D}$ and number of dictionary atoms P, number of components K, joint filter orders Q and M, regularization parameters μ_1, μ_2, μ_3, maximum iterations $I_{\max}$.

Ensure: Membership functions $\{g_k\}_{k=1}^K$, joint spectral filters $\{h_k\}_{k=1}^K$

Initialize $\mathbf{V}^{(0)} \succeq \mathbf{0}$

Set iteration counter $i \leftarrow 0$

repeat

  $i \leftarrow i + 1$

  Update B:

  $\mathbf{B}^{(i)} = \arg\min_{\mathbf{B} \succeq \mathbf{0}} \; f_1(\mathbf{V}^{(i-1)}, \mathbf{B}) + \mu_2 \operatorname{tr}(\mathbf{B})$

  Update V:

  $\mathbf{V}^{(i)} = \arg\min_{\mathbf{V} \succeq \mathbf{0}} \; f_1(\mathbf{V}, \mathbf{B}^{(i)}) + \mu_1 f_2(\mathbf{V}) + \mu_3 \operatorname{tr}(\mathbf{V})$

until $i \geq I_{\max}$

Extract $\{\mathbf{v}_k\}$ from rank-one decomposition of $\mathbf{V}^{(i)}$

Extract $\{\mathbf{b}_k\}$ from rank-one decomposition of $\mathbf{B}^{(i)}$

Compute $\mathbf{g_k} = \mathbf{D}\mathbf{v}_k, \quad k = 1, \ldots, K$

Construct $h_k(\lambda_n, \omega_\tau)$ from $\mathbf{b}_k$

return $\{\mathbf{g_k}, \mathbf{h_k}\}_{k=1}^K$

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, we evaluate the performance of the proposed framework through a series of experiments on both synthetic and real-world datasets. Experiments on synthetic data are first conducted to isolate and analyze the effects of key modeling and optimization parameters under controlled conditions. These experiments provide insights into the robustness, stability, and identifiability of the learned model. Subsequently, the proposed method is applied to a real dataset to demonstrate its practical effectiveness and to compare its performance against existing approaches.

Before presenting the experimental results, we first describe the evaluation procedure used throughout this chapter. Since the proposed learning framework is covariance-driven, both the empirical and model-based covariance matrices are normalized prior to evaluation. This normalization is motivated by the fact that LMMSE estimation depends on relative correlation structures rather than absolute signal scale. Consequently, scale information does not affect estimation performance and can be safely removed to enable a fair comparison between different covariance models.

To assess the quality of the learned covariance structure independently of the reconstruction task, we consider the following covariance discrepancy measure:

$$\mathrm{CD} = \frac{\|\mathbf{C}_x - \mathbf{C}_x^\star\|_F}{\|\mathbf{C}_x\|_F}, \tag{4.1}$$

where $\mathbf{C}_x^\star$ denotes the normalized covariance matrix learned by the proposed LSTVP model and $\mathbf{C}_x$ denotes the normalized ground-truth covariance matrix. This metric captures deviations in both the overall correlation pattern and the energy distribution of the covariance structure.

The primary application considered in this chapter is the recovery of missing obser-

vations in time–vertex signals. To this end, we assume that a subset of the signal samples is observed, while the remaining entries are missing. Given the learned covariance matrix, missing values are estimated using a linear minimum mean square error (LMMSE) estimator.

Let $\mathbf{x} \in \mathbb{R}^{NT}$ denote the complete time-vertex signal and $\mathbf{y} = \mathbf{S}\mathbf{x}$ represent the observed signal, where $\mathbf{S}$ is a binary sampling operator selecting the available observations. In all experiments, the sampling operator $\mathbf{S}$ selects the observed entries *uniformly at random* over the joint time-vertex grid. Let $\mathbf{z}$ denote the vector of missing entries of $\mathbf{x}$. Given the observations $\mathbf{y} = \mathbf{S}\mathbf{x}$, the missing entries are estimated via the LMMSE estimator as

$$\widehat{\mathbf{z}} = \mathbf{C}_{zy}\mathbf{C}_{yy}^{-1}\mathbf{y}, \tag{4.2}$$

where $\mathbf{C}_{zy}$ and $\mathbf{C}_{yy}$ are submatrices of the learned covariance matrix corresponding to the cross-covariance and auto-covariance of the missing and observed components, respectively.

To quantitatively evaluate the reconstruction accuracy, we consider several standard error metrics computed over all missing entries and averaged across R independent signal realizations. Specifically, the normalized mean error (NME) and mean absolute error (MAE) are defined as

$$\mathrm{NME} = \frac{1}{R}\sum_{r=1}^{R}\frac{\left\|\mathbf{z}^{(r)} - \widehat{\mathbf{z}}^{(r)}\right\|_2}{\left\|\mathbf{z}^{(r)}\right\|_2}, \tag{4.3a}$$

$$\mathrm{MAE} = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{R_z}\left\|\mathbf{z}^{(r)} - \widehat{\mathbf{z}}^{(r)}\right\|_1, \tag{4.3b}$$

where R denotes the number of independent realizations, R_z denotes the number of missing samples in each realization, $\mathbf{z}^{(r)}$ and $\hat{\mathbf{z}}^{(r)}$ denote the true and estimated vectors of missing entries at the r-th realization, respectively, and $\|\cdot\|_2$ denotes the Euclidean norm.

In addition to defining the evaluation metrics, we briefly summarize below the baseline reconstruction methods used in all experiments.

Specifically, we compare the proposed LSTVP framework against several reference methods that have been previously introduced and discussed in Chapter 2 and Chapter 3. As a first baseline, we consider the JWSS model [8], which is a nonparametric

approach assuming global joint wide-sense stationarity across both time and graph domains. Under this assumption, the second-order statistics of the data are represented via a single global joint non-parametric filter. This model serves as a natural benchmark to assess the limitations of global stationarity assumptions, which are explicitly relaxed in the proposed LSTVP framework by allowing statistical properties to vary locally over time and graph.

As a second baseline, we include a global moving-average (MA) model, introduced in Chapter 2 with (2.29) and (2.30), which relies on similar global stationarity assumptions as JWSS but adopts a parametric formulation. By modeling the joint filtering behavior using a finite set of parameters, this approach provides a more structured and computationally efficient alternative to nonparametric JWSS, while still enforcing globally consistent behavior across time and graph domains.

Finally, we consider a global joint ARMA-based approach [9, 21], defined in Chapter 2 with (2.26) and (2.28), which also assumes a globally stationary time-vertex process but employs a richer parametric model by incorporating autoregressive dependencies in addition to moving-average components. While this model offers increased representational capacity compared to the global MA formulation, it does not account for local variations in either the time or graph domains.

In contrast to these baselines, the proposed LSTVP model provides a parametric and locally adaptive representation by expressing the time-vertex process as a linear combination of multiple joint filters, each of which can capture distinct statistical structures associated with different graph regions and time intervals. This increased modeling flexibility enables LSTVP to represent a broader class of nonstationary time-vertex processes, at the cost of increased model complexity, and forms the central motivation for the comparative evaluation presented in this section.

4.1 Performance Analysis of LSTVP

This section evaluates the proposed LSTVP framework on synthetic time-vertex signals, for which the ground-truth model parameters and observation process are fully controlled. In all experiments, we consider graphs with $N = 30$ vertices and time

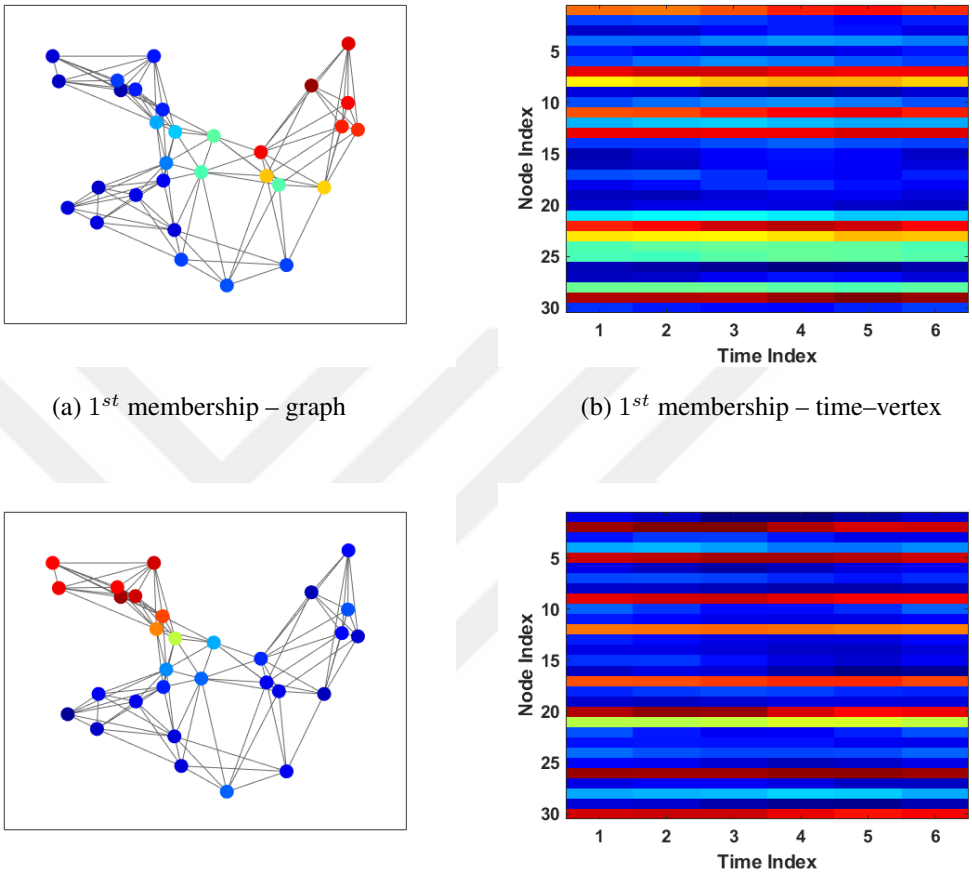
series of length $T = 6$. The synthetic signals are generated according to the locally stationary time-vertex process model introduced as (3.5), ensuring full consistency between the data generation mechanism and the assumed reconstruction model.

The use of synthetic data enables a systematic and isolated analysis of key factors affecting estimation performance, including model order parameters, sampling ratios, and noise levels, without interference from real-world measurement artifacts. Unless otherwise stated, the membership functions are constructed to induce locally varying statistical behavior across both the graph and time domains, reflecting heterogeneous regions with distinct joint spectral characteristics.

In the synthetic data generation process, $K = 4$ membership functions are constructed to induce smoothly varying local behavior across both the graph and time domains. Each membership function assigns spatially and temporally localized weights to the underlying joint filters, thereby controlling the regions over which specific second-order statistical structures are active. The memberships are designed to vary smoothly over vertices and time indices, ensuring gradual transitions between different local regimes rather than abrupt changes. Figure 4.1 and 4.2 illustrates the resulting membership functions, highlighting their smooth and localized structure across the joint time–vertex domain.

To evaluate robustness to noise, additive noise is incorporated into the observation model, and performance is reported as a function of the signal-to-noise ratio (SNR), defined as the ratio between the signal power of the noiseless time-vertex process and the power of the added noise component.

While the membership functions define where different statistical behaviors are active over the joint time-vertex domain, the corresponding joint kernels specify the local second-order structures associated with each region. In the proposed LSTVP model, each membership modulates a distinct joint filter, resulting in locally varying spectral characteristics across time and graph domains. The synthetic process is therefore generated as a superposition of multiple locally active components, each governed by its own joint kernel. To further illustrate this mechanism, we visualize the joint kernels associated with each membership function in the following figures.



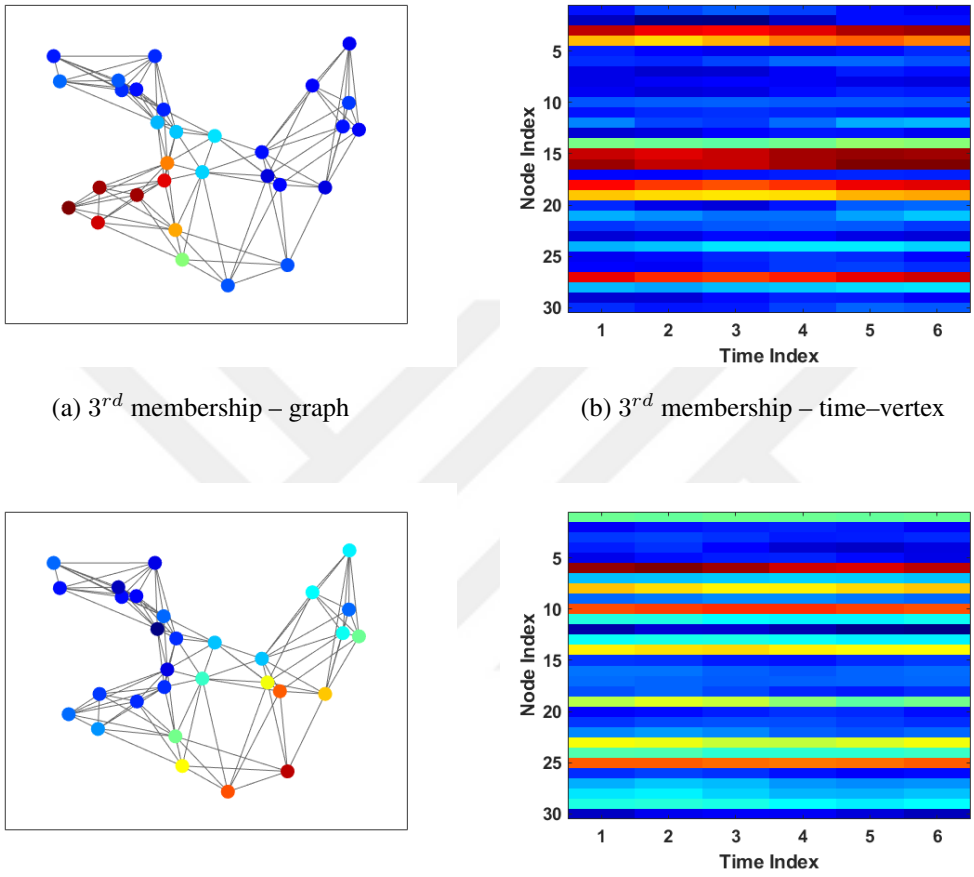
(a) 1^{st} membership – graph

(b) 1^{st} membership – time–vertex

(c) 2^{nd} membership – graph

(d) 2^{nd} membership – time–vertex

Figure 4.1: Visualization of the first two membership functions (g_1 and g_2) used in the synthetic data generation. For each membership, the left column shows the spatial activation over the graph domain, while the right column illustrates its variation over time for $T = 6$.



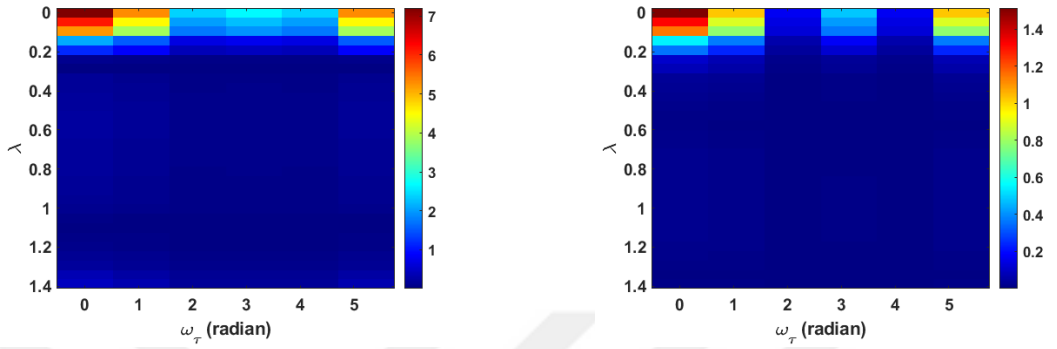
(a) 3^{rd} membership – graph

(b) 3^{rd} membership – time–vertex

(c) 4^{th} membership – graph

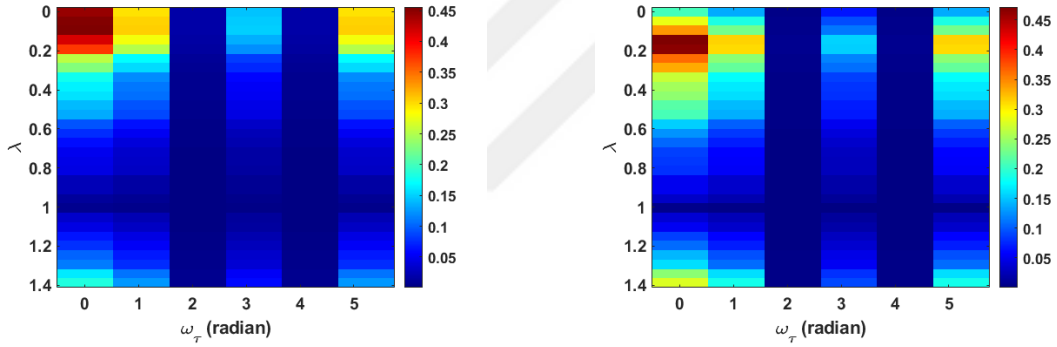
(d) 4^{th} membership – time–vertex

Figure 4.2: Visualization of the remaining membership functions (g_3 and g_4) used in the synthetic data generation. The left column shows the spatial activation over the graph domain, while the right column illustrates the temporal variation for $T = 6$.



(a) 1^{st} joint kernel

(b) 2^{nd} joint kernel



(c) 3^{rd} joint kernel

(d) 4^{th} joint kernel

Figure 4.3: Visualization of the joint kernels associated with the $K = 4$ membership functions used in the synthetic data generation. Each joint kernel corresponds to the membership functions shown in Fig. 4.1 and Fig. 4.2, and illustrates distinct local spectral characteristics in the joint time-vertex domain.

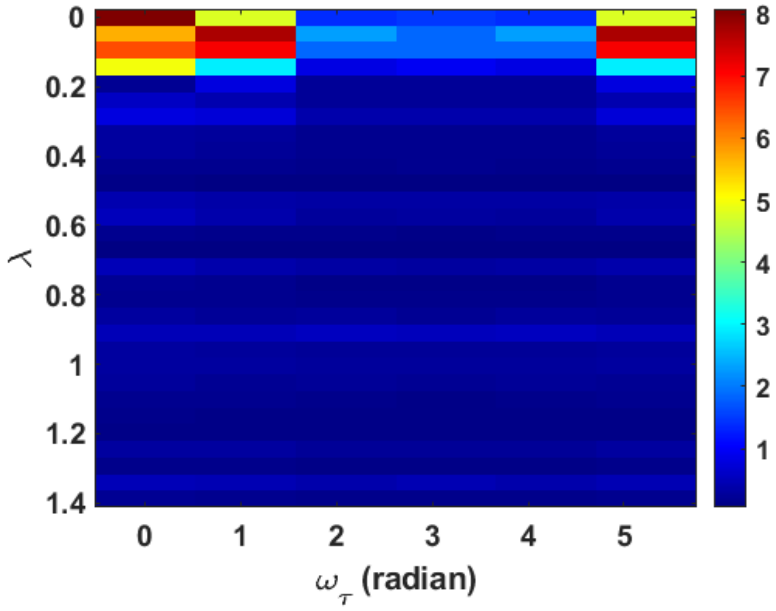


Figure 4.4: Estimated joint power spectral density (JPSD) of synthetic time–vertex signals generated using the proposed LSTVP model. The distribution of spectral energy across graph and temporal frequencies reflects the combined effect of the locally varying membership functions and their associated low-pass joint kernels.

In the synthetic data generation process, the joint kernels are selected to exhibit low-pass characteristics while covering different regions of the joint spectral domain. Specifically, each kernel emphasizes a distinct portion of the spectrum, thereby inducing varying degrees of smoothness across the graph and time dimensions. This design ensures spectral diversity among locally active components while remaining consistent with the parametric LSTVP model. The resulting joint kernels are illustrated in Fig. 4.3.

Combining the membership functions and joint kernels within the proposed LSTVP generative model yields synthetic time-vertex signal realizations exhibiting locally varying second-order statistics. To provide a global spectral interpretation of the resulting process, we examine the joint power spectral density (JPSD) of the generated signals. The JPSD captures how energy is distributed across graph and temporal frequencies and reflects the interaction between locally active memberships and their associated joint kernels. The estimated JPSD of representative signal realizations is illustrated in Fig. 4.4.

4.1.1 Effect of the Number of Realizations

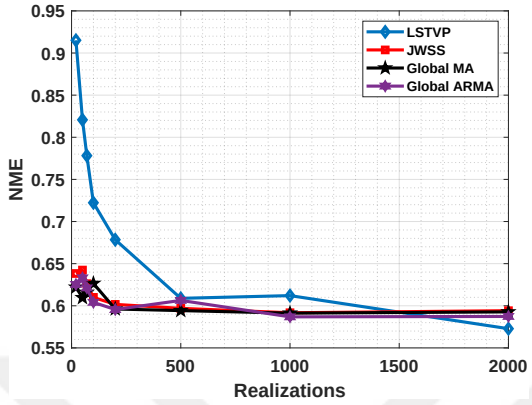
The number of available signal realizations plays a key role in the reliability of covariance estimation. This experiment evaluates its effect on the accuracy and stability of the learned model. Figure 4.5 illustrates the effect of the number of available realizations on the reconstruction performance of the proposed LSTVP-based framework under different noise conditions. The top row corresponds to a low signal-to-noise ratio scenario ($\mathrm{SNR} = 0$ dB), while the bottom row shows the same performance metrics under a higher SNR regime ($\mathrm{SNR} = 10$ dB).

Under low-SNR conditions, increasing the number of realizations leads to a consistent reduction in both the mean absolute error (MAE) and the normalized mean error (NME). This behavior indicates that, even in noisy environments, the availability of a larger number of realizations significantly improves the reliability of covariance estimation, which in turn enhances the accuracy of the LMMSE-based reconstruction. However, in this regime, the proposed LSTVP model may exhibit limited performance gains compared to simpler baseline methods due to its higher representational

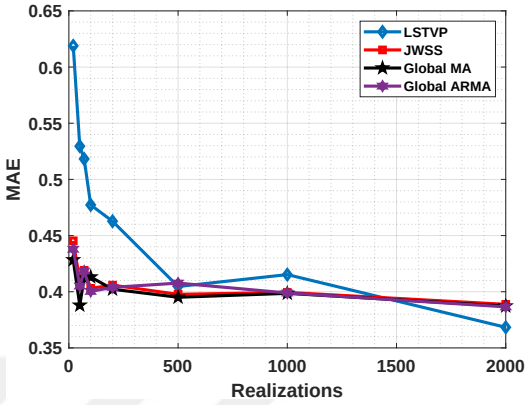
capacity, which can partially fit noise-dominated components when the SNR is very low.

As the noise level decreases, the beneficial impact of increasing the number of realizations becomes more pronounced. In particular, for $\mathrm{SNR} = 10$ dB, both MAE and NME exhibit a steeper decline as the number of realizations grows, demonstrating that improved noise conditions allow the proposed method to more effectively exploit the additional statistical information provided by multiple realizations. In this regime, the locally adaptive and parametric structure of the LSTVP framework becomes advantageous, enabling it to outperform methods relying on global stationarity assumptions.

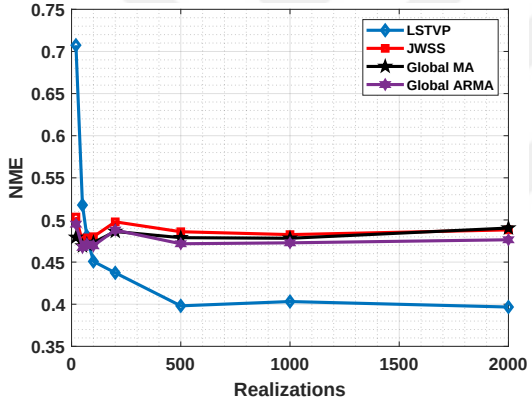
It is also observed that, for both SNR levels, the reconstruction error does not converge to zero as the number of realizations increases. At low SNR, this behavior is expected due to the presence of irreducible noise, which results in a nonzero residual error even with infinitely many realizations. At higher SNR levels, the remaining error can be attributed to finite model orders, regularization effects, as well as the fact that $\mathrm{SNR} = 10$ dB still corresponds to a non-negligible noise regime.



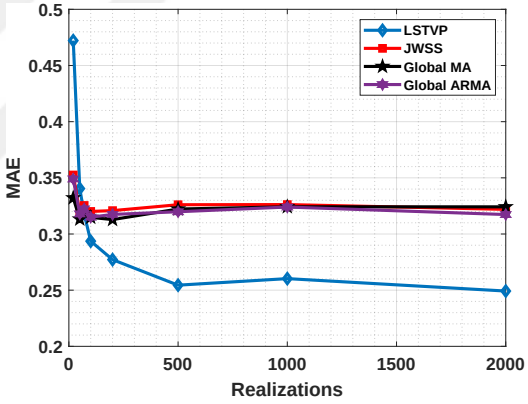
(a) Realizations vs. NME (SNR $= 0$ dB)



(b) Realizations vs. MAE (SNR $= 0$ dB)



(c) Realizations vs. NME (SNR $= 10$ dB)



(d) Realizations vs. MAE (SNR $= 10$ dB)

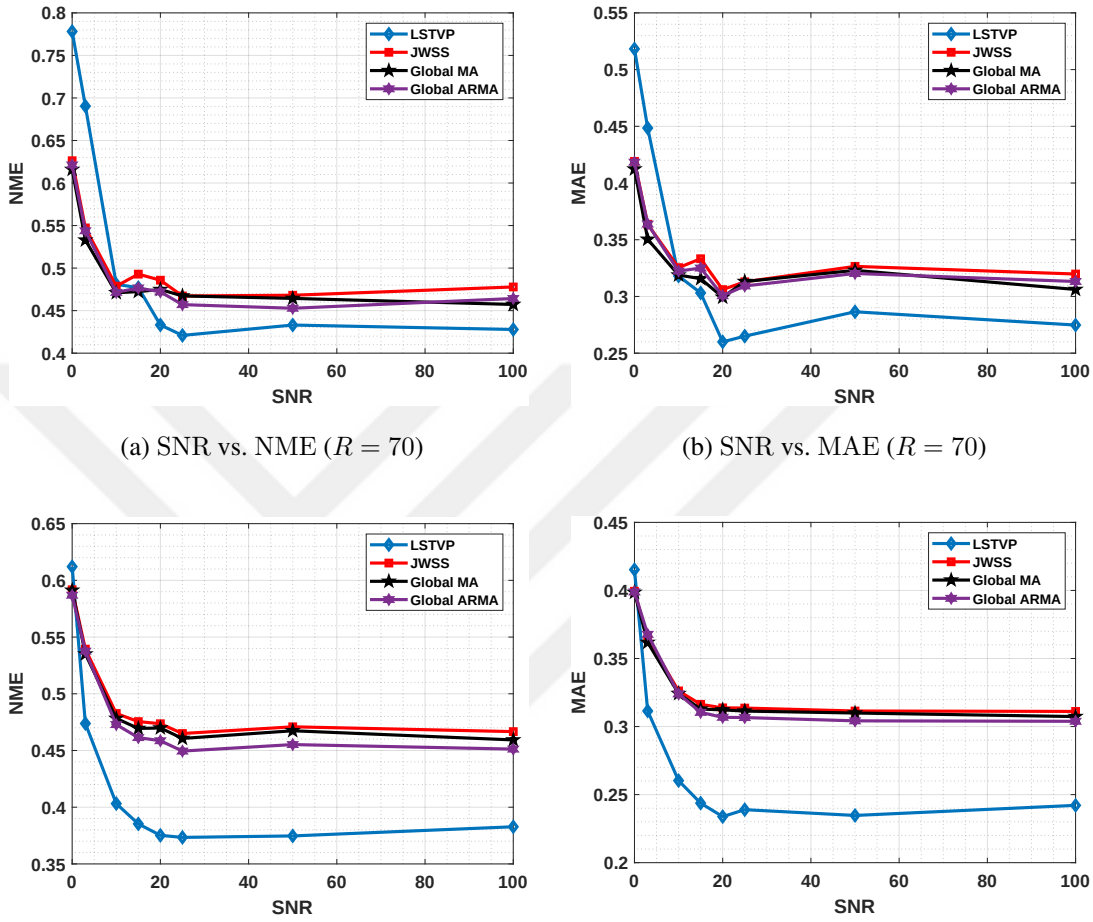
Figure 4.5: Effect of the number of realizations on reconstruction performance. The figure reports the variation of the normalized mean error (NME) and mean absolute error (MAE) as a function of the number of available realizations for two different noise values. The ratio of missing samples is fixed to 25%.

4.1.2 Effect of Noise Level

In this experiment, we analyze the robustness of the proposed method under increasing noise levels. Synthetic observations are corrupted with additive white Gaussian noise at different signal-to-noise ratio (SNR) values, and the resulting estimation performance is evaluated in terms of reconstruction accuracy.

Figure 4.6 illustrates the effect of the noise level on the reconstruction performance of the proposed method. As the signal-to-noise ratio (SNR) increases, both the normalized mean error (NME) and mean absolute error (MAE) tend to decrease, indicating that improved observation quality enhances estimation accuracy.

For a relatively small number of realizations ($R = 70$), the performance gain of the proposed LSTVP model becomes noticeable only at higher SNR values, suggesting that limited sample availability restricts the ability of the algorithm to reliably learn the underlying covariance structure under noisy conditions. In contrast, when the number of realizations is increased to $R = 1000$, significant improvements in both MAE and NME are observed already at lower SNR levels. These results demonstrate that increasing the number of realizations substantially improves the robustness of the proposed framework to noise.



(a) SNR vs. NME ($R = 70$)

(b) SNR vs. MAE ($R = 70$)

(c) SNR vs. NME ($R = 1000$)

(d) SNR vs. MAE ($R = 1000$)

Figure 4.6: Effect of noise level on reconstruction performance. The figure illustrates the variation of the normalized mean error (NME) and mean absolute error (MAE) as a function of the signal-to-noise ratio (SNR) for two different number of realizations ($R = 70$ and $R = 1000$), missing samples fixed to 25%.

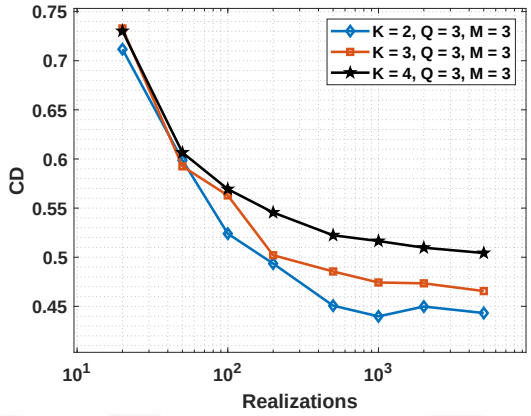
4.1.3 Effect of Model Complexity

In this experiment, we examine how varying the model complexity affects estimation performance. Figure 4.7 illustrates the impact of different model complexity parameters on the covariance estimation accuracy, as measured by the covariance discrepancy (CD).

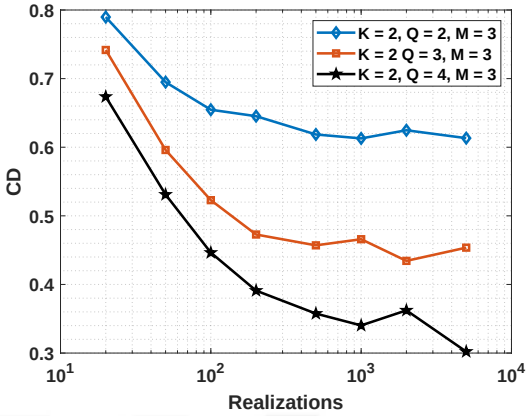
First, the results corresponding to the parameter K, which controls the number of locally stationary components and jointly determines the number of membership functions and joint filters, indicate that increasing K leads to higher model complexity. As the dimensionality of the parameter space grows with K, the learning problem becomes more challenging, manifesting as an increase in the covariance discrepancy. This behavior highlights the trade-off between model expressiveness and estimation reliability when selecting the number of local components.

In contrast, the parameter Q, which determines the temporal order of the joint moving-average filter, exhibits a clear and consistent positive effect on estimation performance in the considered synthetic setting. Increasing Q allows the model to exploit richer temporal correlation structures encoded in the structured covariance matrix provided as input to the learning algorithm. As a result, for the adopted data generation model and spectral characteristics, higher values of Q lead to a significant reduction in CD, highlighting the importance of temporal modeling capacity under these experimental conditions.

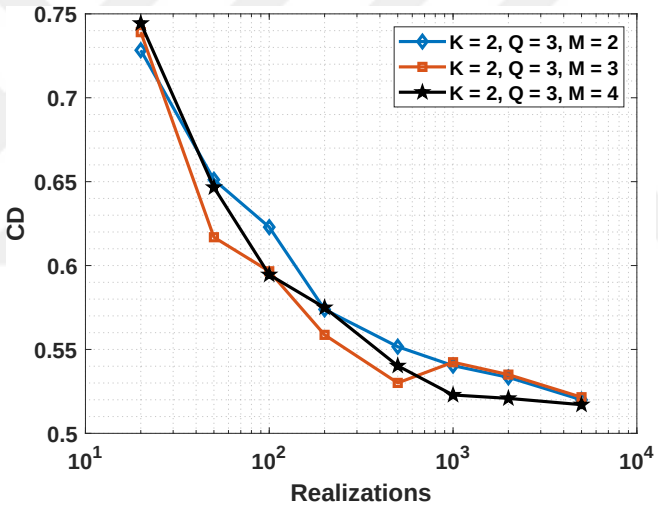
Finally, the effect of the graph spectral order parameter M, which controls the maximum power of the graph Laplacian in the joint filter representation, appears to be less pronounced in the considered experiments. The observed variations in CD across different values of M do not exhibit a clear monotonic trend for the employed synthetic signals, suggesting that increasing the graph spectral order beyond a certain point does not necessarily yield additional performance gains in this setting. This observation indicates that, for the assumed spatial correlation structure, the proposed framework may already capture the dominant graph-dependent patterns with relatively low graph spectral orders, while further increases in M primarily increase model complexity without substantially improving estimation accuracy.



(a) Effect of model complexity parameter K



(b) Effect of model complexity parameter Q



(c) Effect of model complexity parameter M

Figure 4.7: Effect of model complexity on covariance estimation performance. The covariance discrepancy (CD) is reported as a function of different model complexity parameters, illustrating the impact of varying K, Q, and M on the accuracy of the learned LSTVP covariance structure.

4.1.4 Effect of Regularization Parameters

The regularization parameters play a crucial role in stabilizing the learning process and preventing ill-posed solutions in the proposed framework. In this experiment, we investigate the effect of the regularization weights on the estimation performance by evaluating the normalized mean error (NME) over a wide range of parameter values. Unlike the previous experiments, this analysis is conducted on a real-world dataset, namely the Molène dataset, in order to assess the practical impact of regularization under realistic observation conditions.

For the Molène dataset [3], the underlying graph consists of $N = 37$ nodes with $T = 12$ temporal samples per realization, where 72 realizations are used for training and 660 realizations are reserved for testing. During training, 25% of the signal samples are randomly treated as missing observations and used for validation when tuning the regularization parameters.

Table 4.1 reports the variation of the NME with respect to the regularization parameters μ_2 and μ_3, while fixing $\mu_1 = 0$. The results clearly demonstrate that setting the regularization parameters to zero leads to inferior performance, indicating that purely data-driven covariance fitting is insufficient to yield reliable model estimates. Introducing moderate regularization significantly improves the reconstruction accuracy, confirming the necessity of imposing additional structural constraints on the learned parameters.

In particular, small but nonzero values of μ_2 and μ_3 consistently lead to lower NME values, whereas excessively large regularization weights result in a degradation of performance due to over-penalization. This behavior highlights the importance of balancing data fidelity and regularization in the learning objective.

The influence of the parameter μ_1 is further examined in Table 4.2, where the NME is reported for different values of μ_1 while keeping the remaining regularization parameters fixed. The results indicate that introducing moderate smoothness regularization improves estimation accuracy, whereas both very small and very large values of μ_1 yield suboptimal performance.

Overall, these results confirm that regularization is an essential component of the proposed learning framework. Appropriate selection of the regularization parameters enables robust covariance estimation and significantly enhances reconstruction performance, while setting these parameters to zero leads to noticeable performance degradation.

Table 4.1: Variation of the NME μ_2 and μ_3 where $\mu_1 = 0$

μ_2 \ μ_3	0	10^{-6}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10
0	0.3697	0.3732	0.3716	0.3720	0.3702	0.3481	0.3272	1.2932
10^{-6}	0.3707	0.3727	0.3689	0.3713	0.3716	0.3320	0.3519	1.1317
10^{-4}	0.3719	0.3725	0.3700	0.3698	0.3700	**0.3108**	**0.3000**	1.3483
10^{-3}	0.3701	0.3710	0.3711	0.3709	0.3670	0.3549	0.3352	1.3447
10^{-2}	0.3701	0.3720	0.3742	0.3720	0.3715	0.3551	0.3488	1.3514
10^{-1}	0.3728	0.3737	**0.3218**	0.3570	0.3807	0.3664	0.5757	1.4105
1	0.3478	0.3485	0.3218	0.3570	0.3671	0.3601	0.5596	1.3462
10	0.4089	0.4062	0.4051	0.4060	0.4015	0.3927	1.3467	1.3498

Table 4.2: Variation of the NME with different μ_1 values

μ_1	0	10^{-6}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10
NME	0.3734	0.3740	0.3744	0.3698	**0.3557**	0.3679	0.3611	1.4186

4.2 Comparative Experiments

To assess the effectiveness of the proposed LSTVP model, we compare its performance with several representative baseline methods introduced at the beginning of Chapter 4, each reflecting different modeling assumptions in time-vertex signal processing. The real-world experiments in this section are conducted one two datasets. The first dataset is the Molène weather dataset, which consists of hourly temperature measurements collected from $N = 37$ meteorological stations located in the Brittany region of France during January 2014 [3]. Each measurement station is modeled

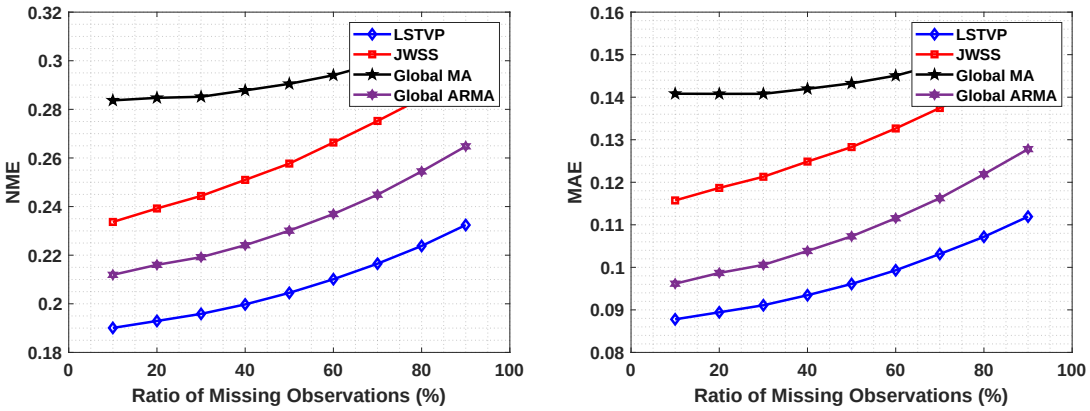


Figure 4.8: NME (left) and MAE (right) results for the Molène dataset.

as a vertex of a graph, while the spatial relationships between stations are captured through a 5-nearest neighbor (5-NN) graph constructed based on geographical proximity. The edge weights are defined using a Gaussian kernel applied to the latitude and longitude coordinates of the stations, resulting in a weighted undirected graph that reflects spatial correlations among nodes.

For the Molène dataset, the underlying graph consists of $N = 37$ nodes with $T = 12$ temporal samples per realization. A total of 72 realizations are used for training, while 660 realizations are reserved for testing. During the training phase, 25% of the signal samples are randomly treated as missing observations and used for validation when tuning the regularization parameters. Reconstruction performance is then evaluated on the held-out test realizations under varying missing observation ratios.

As observed in Figure 4.8, the proposed LSTVP model consistently achieves the lowest NME across all missing observation ratios, demonstrating superior robustness to data incompleteness compared to the competing methods. This behavior can be attributed to the ability of the LSTVP model to capture localized time-vertex variations through jointly learned membership functions and local joint spectral filters.

Beyond the performance of the proposed approach, clear trends can also be observed among the baseline methods. In particular, the global ARMA model generally outperforms the global MA model, which can be explained by its richer parametric structure that allows it to better fit the underlying process. Similarly, both parametric models (global MA and ARMA) achieve lower errors than the non-parametric JWSS model,

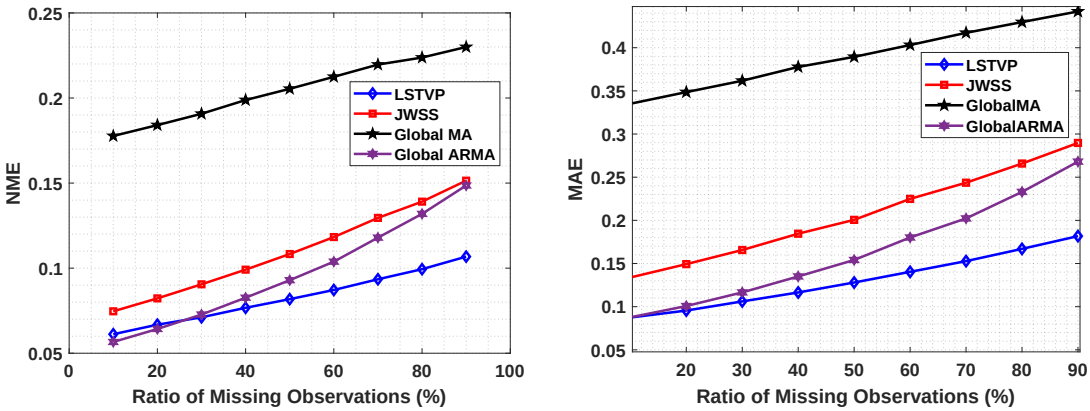


Figure 4.9: NME (left) and MAE (right) results for the NOAA dataset.

especially at higher missing ratios. This behavior is likely due to the fact that, in finite-sample regimes, parametric formulations enable more reliable estimation of second-order statistics compared to fully non-parametric approaches.

In contrast, the JWSS model relies on estimating a single global joint spectral representation without an explicit parametric structure, which limits its ability to adapt under severe observation loss. Overall, while the baseline methods exhibit different levels of robustness depending on their modeling assumptions, the proposed LSTVP framework consistently provides the most accurate reconstructions by jointly leveraging locality and parametric modeling in the time-vertex domain.

To further evaluate the robustness of the proposed framework under more heterogeneous spatial and temporal dynamics, we additionally consider a real-world climate dataset obtained from the National Oceanic and Atmospheric Administration (NOAA) [70]. The experiments are conducted using temperature measurements collected from $N = 65$ spatial locations over $T = 12$ temporal samples per realization. In this setting, 83 signal realizations are used for training, while 870 realizations are reserved for testing, following the same evaluation method adopted for the Molène dataset.

Figure 4.9 presents the reconstruction performance on the NOAA dataset in terms of the normalized mean error (NME) and mean absolute error (MAE) as a function of the missing observation ratios. Consistent with the trends observed on the Molène dataset, the proposed LSTVP framework achieves the lowest reconstruction error

across all missing ratios, demonstrating superior robustness compared to globally stationary baseline methods. As the proportion of missing observations increases, the performance gap between the proposed approach and the baseline methods becomes more pronounced.

Taken together, the results obtained on the Molène and NOAA datasets demonstrate that explicitly modeling local stationarity in the time-vertex domain leads to consistent performance improvements across distinct real-world scenarios.

CHAPTER 5

CONCLUSIONS

In this thesis, we addressed the problem of modeling, learning, and reconstructing time-vertex signals with missing observations by developing a parametric framework based on *Locally Stationary Time-Vertex Processes* (LSTVPs). Motivated by the limitations of globally stationary models in capturing localized temporal and spatial variations, we proposed a structured approach that extends joint wide-sense stationarity to a locally adaptive setting over the joint time-vertex domain.

The main contribution of this work is the formulation of a parametric covariance model that explicitly captures local stationarity through smoothly varying time-vertex membership functions. By modeling the second-order statistics as a superposition of locally weighted joint moving-average (MA) components, the proposed framework provides a flexible yet interpretable representation of time-vertex dependencies. This structure allows different stationary behaviors to dominate in different regions of the joint domain, thereby overcoming the restrictive assumptions of global stationarity.

To estimate the model parameters from partially observed data, we formulated the learning task as a covariance fitting problem. The resulting optimization problem is inherently non-convex due to the coupled dependence on the membership functions and joint filter coefficients. By exploiting appropriate parameterizations and convex relaxations, we showed that the problem admits a biconvex structure, which can be efficiently addressed via an alternating optimization scheme. The proposed learning framework incorporates positive semidefinite constraints, smoothness regularization, and low-rank structure, enabling stable estimation of the model parameters.

Beyond model learning, this thesis demonstrated the practical utility of the learned

LSTVP model for time-vertex signal interpolation. By leveraging the learned covariance structure, missing signal samples were reconstructed using a linear minimum mean square error (LMMSE) estimator. This application highlights that the proposed framework is not only descriptive in nature but also directly applicable to inference tasks involving incomplete time-vertex data.

Extensive experimental results on both synthetic and real-world datasets validated the effectiveness of the proposed approach. Synthetic experiments illustrated the impact of key modeling and optimization parameters, including the number of realizations, noise level, model complexity, and regularization strengths. Experiments on real-world datasets further demonstrated that the proposed LSTVP framework consistently outperforms state-of-the-art baselines, such as globally stationary and global MA/ARMA models, particularly in scenarios characterized by local non-stationarities. These results confirm the advantage of incorporating localized statistical modeling into time-vertex signal processing.

Despite its effectiveness, the proposed framework has certain limitations. The current formulation assumes a predefined graph structure and enforces smooth variations in the membership functions, which may not fully capture abrupt regime changes or highly dynamic graph topologies. In addition, covariance-based learning can become computationally demanding for very large-scale graphs or long temporal horizons, motivating further research into more scalable implementations.

Several directions for future work naturally emerge from this study. Potential extensions include learning or adapting the underlying graph structure, incorporating higher-order or non-Gaussian statistical models, and developing online or distributed learning algorithms suitable for large-scale settings. Another promising direction is the integration of the proposed parametric framework with learning-based approaches, such as deep unfolding or hybrid model-driven neural architectures, to combine interpretability with enhanced modeling flexibility.

In summary, this thesis provides a principled and effective framework for modeling, learning, and reconstructing time-vertex signals under local stationarity assumptions. By bridging theoretical modeling, optimization-based learning, and practical reconstruction tasks, the proposed LSTVP approach contributes to the broader field

of graph signal processing and offers new perspectives for analyzing complex time-vertex data.

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, pp. 83–98, Oct. 2012.

[2] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6163–6166, May 2013.

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

[4] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, 2017.

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

[6] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462–3477, July 2017.

[7] N. Perraudin, A. Loukas, F. Grassi, and P. Vandergheynst, "Towards stationary time-vertex signal processing," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 3914–3918.

[8] A. Loukas and N. Perraudin, "Stationary time-vertex signal processing,"

EURASIP Journal on Advances in Signal Processing, vol. 2019, no. 1, p. 36, 2019.

[9] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with VARMA recursions on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4870–4885, 2019.

[10] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.

[11] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[12] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, pp. 808–828, May 2018.

[13] A. Ortega, *Introduction to Graph Signal Processing*, Cambridge University Press, 2022.

[14] 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, pp. 817–829, Feb 2018.

[15] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Transactions on Signal Processing*, vol. 64, pp. 1832–1843, Apr 2016.

[16] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Transactions on Signal Processing*, vol. 63, pp. 6510–6523, Dec 2015.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[18] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2651–2664, Nov 2013.

[19] A. Canbolat and E. Vural, "Locally stationary graph processes," *IEEE Transactions on Signal Processing*, vol. 72, pp. 2323–2332, 2024.

[20] A. Canbolat and E. Vural, "An efficient algorithm for learning locally stationary graph processes," in *Proc. 32nd European Signal Processing Conference (EUSIPCO)*, 2024, pp. 2312–2316, doi: 10.23919/EUSIPCO63174.2024.10714929.

[21] E. T. Güneyi, B. Yaldız, A. Canbolat, and E. Vural, "Learning graph ARMA processes from time-vertex spectra," *IEEE Transactions on Signal Processing*, vol. 72, pp. 47–56, 2024.

[22] R. A. Silverman, "Locally stationary random processes," *IRE Transactions on Information Theory*, vol. 3, pp. 182–187, 1957.

[23] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

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

[25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," Available: http://cvxr.com/cvx, Mar. 2014.

[26] MOSEK ApS, "The MOSEK optimization toolbox for MATLAB," Version 10.0, Copenhagen, Denmark, 2023.

[27] MOSEK ApS, "Semidefinite optimization with MOSEK," ISMP Tutorial Slides, 2012.

[28] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[29] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "On the implementation and usage of SDPT3—a MATLAB software package for semidefinite-quadratic-linear programming, version 4.0," in *Handbook on Semidefinite, Conic and Polynomial Optimization*, M. Anjos and J. Lasserre, Eds. Springer, 2012, pp. 715–754.

[30] M. Nagahama, K. Yamada, Y. Tanaka, S. H. Chan, and Y. C. Eldar, "Graph signal restoration using nested deep algorithm unrolling," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3296–3311, 2022.

[31] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 128–146, 2020.

[32] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. McGraw–Hill, 2002.

[33] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, 1996.

[34] K. Qiu, X. Mao, X. Shen, X. Wang, T. Li, and Y. Gu, "Time-Varying Graph Signal Reconstruction," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3161–3176, 2020.

[35] S. Hadou, C. I. Kanatsoulis and A. Ribeiro, "Space-Time Graph Neural Networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2022. arXiv preprint arXiv:2110.02880, 2021.

[36] S. Chen and Y. C. Eldar, "Time-Varying Graph Signal Inpainting via Unrolling Networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, June 6–11, 2021, pp. 8092–8097.

[37] J. A. Castro-Correa, J. H. Giraldo, A. Mondal, M. Badiey, T. Bouwmans, and F. D. Malliaros, "Time-Varying Signals Recovery via Graph Neural Networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, June 2023, pp. 1–5.

[38] J. A. Castro-Correa, J. H. Giraldo, M. Badiey, and F. D. Malliaros, "Gegenbauer graph neural networks for time-varying signal reconstruction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11734–11745, 2024.

[39] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning Graphs From Data: A Signal Representation Perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, May 2019.

[40] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Machine Learning*, vol. 56, no. 1–3, pp. 209–239, 2004.

[41] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pp. 912–919, AAAI Press, 2003.

[42] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines*, B. Schölkopf and M. K. Warmuth, Eds., pp. 144–158, Springer, Berlin, Heidelberg, 2003.

[43] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, Jan. 2004.

[44] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16 (NIPS 2003)*, 2003.

[45] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive Graph Signal Processing: Algorithms and Optimal Sampling Strategies," *IEEE Transactions on Signal Processing*, vol. 66, no. 13, pp. 3584–3598, Jul. 2018.

[46] 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, Mar. 2011.

[47] V. Kalofolias, "How to learn a graph from smooth signals," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

[48] A. Loukas and P. Frossard, "Frequency analysis of temporal graph signals," in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016.

[49] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Separable autoregressive moving average graph-temporal filters," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 200–204, 2016.

[50] F. Grassi, N. Perraudin, and R. Ricard, "Tracking time-vertex propagation using dynamic graph wavelets," in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016.

[51] C. Zhang, D. Florêncio, and P. A. Chou, "Graph Signal Processing—A Probabilistic Framework," Microsoft Research Technical Report, 2015.

[52] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sep. 2014.

[53] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1931–1935, Nov. 2015.

[54] P. Valdivia, F. Díaz, F. Petronetto, C. T. Silva, and L. G. Nonato, "Wavelet-based visualization of time-varying data on graphs," in *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 1–8, Oct. 2015.

[55] A. Loukas, E. Isufi, and N. Perraudin, "Predicting the evolution of stationary graph signals," in *Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 60–64, 2017.

[56] R. Merris, "Laplacian matrices of graphs: A survey," *Linear Algebra and its Applications*, vol. 197–198, pp. 143–176, 1994.

[57] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "A windowed graph Fourier transform," in *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP)*, pp. 133–136, 2012.

[58] J. H. Giraldo, A. Mahmood, B. Garcia-Garcia, D. Thanou, and T. Bouwmans, "Reconstruction of time-varying graph signals via Sobolev smoothness," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 201–213, 2022.

[59] G. Lewenfus, W. A. Martins, S. Chatzinotas, and B. Ottersten, "Joint forecasting and interpolation of time-varying graph signals using deep learning," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 761–773, 2020.

[60] C. Peng, Y. Yan, and E. E. Kuruoglu, "Adaptive Graph Normalized Sign Algorithm," arXiv preprint arXiv:2405.04107, May 2024.

[61] Y. Yan, F. Zhao, and E. E. Kuruoglu, "Time-varying graph signal estimation via dynamic multi-hop topologies," arXiv preprint arXiv:2410.17625, Nov. 2024.

[62] S. Krishnan, J. Choi, and J. Park, "Learning time-varying graph signals via Koopman," arXiv preprint arXiv:2511.06493, Nov. 2025.

[63] T. Fukuhara, J. Hara, H. Higashi, and Y. Tanaka, "Period estimation for time-varying graph signals and its application to graph Wiener filter," in *Proc. Asia Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA ASC)*, 2025.

[64] J. A. Castro-Correa, J. H. Giraldo, A. Mondal, M. Badiey, T. Bouwmans, and F. D. Malliaros, "Time-varying signals recovery via graph neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2023,

[65] B. P. Chamberlain, J. Rowbottom, D. Eynard, F. Di Giovanni, X. Dong, and M. M. Bronstein, "Graph neural diffusion," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021, pp. 1356–1367.

[66] X. Luo, Z. Yin, J. Li, and S. Wang, "Graffe: Graph representation learning via diffusion probabilistic models," in *International Conference on Learning Representations (ICLR)*, 2022.

[67] H. Wen, Y. Lin, Y. Xia, H. Wan, Q. Wen, R. Zimmermann, and Y. Liang, "Diff-STG: Probabilistic spatio-temporal graph forecasting with denoising diffusion models," in *Proc. of the 31st ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, 2023.

[68] D. T. Nguyen, M. D. T. Nguyen, T. S. Hy, and R. Kondor, "Fast Temporal Wavelet Graph Neural Networks," arXiv preprint arXiv:2302.08643, Oct. 2023.

[69] G. P. H. Styan, “Hadamard products and multivariate statistical analysis,” *Linear Algebra and its Applications*, vol. 6, pp. 217–240, 1973.

[70] A. Arguez, I. Durre, S. Applequist, R. S. Vose, M. F. Squires, X. Yin, R. R. Heim, and T. W. Owen, “NOAA’s 1981–2010 U.S. climate normals: An overview,” *Bulletin of the American Meteorological Society*, vol. 93, no. 11, pp. 1687–1697, 2012.