©SHUTTERSTOCK.COM/PAVEL_KLIMENKO

Alexandre Hippert-Ferrer, Aude Sportisse, Amirhossein Javaheri, Mohammed Nabil El Korso, and Daniel P. Palomar

Missing Data in Signal Processing and Machine Learning

Models, methods, and modern approaches

Introduction

Missing data are a pervasive challenge across scientific disciplines, occurring whenever observations are partially unavailable due to sensor failures, measurement limitations, human error, or natural phenomena. In signal processing (SP) and machine learning (ML), incomplete data can severely compromise analysis, distort parameter estimates, and undermine predictive performance.

From sensor networks and radar systems to medical imaging and financial time series, practitioners across domains must routinely grapple with the question: *How should we handle missing data?*

- In biomedical SP, equipment malfunctions frequently cause missing recordings. For instance, electroencephalograms (EEGs) often contain faulty measurements due to electrode disconnections or motion artifacts, prompting analysts to discard affected segments and creating data gaps.

Dr. Çağatay Candan acted as the associate editor for this article and recommended it for publication.

Digital Object Identifier 10.1109/MSP.2026.3682182
Date of current version: 7 August 2026

- In image processing, images must often be restored from partial observations caused by distortions such as scratches, noise, or occlusions. In remote sensing, for example, optical sensors are limited by cloud cover and prone to hardware failures, such as the well-known malfunction of the Landsat ETM+ scan-line corrector, which produced images with systematic gaps.
- In sensor array processing, multiple antennas record signals from a small number of sources, and measurements may be lost during transmission or due to sensor failures. As the number of underlying sources is typically much smaller than the number of sensors, the resulting data matrix can be modeled as the sum of a strictly low-rank matrix, representing the true signals plus a noise matrix.
- In financial data analysis, traded assets often exhibit missing values due to nontrading periods, lack of liquidity, or data handling errors. Accurate imputation is critical for applications such as portfolio optimization, backtesting, and risk modeling.
- In recommender systems, the Netflix Prize dataset (released in 2006) comprised ratings from approximately 480,000 users over 18,000 movies, yet only about 1% of entries were observed; the competition challenged participants to predict missing ratings from this extremely sparse matrix.
- In graph-structured data, signals defined on networks, such as temperature measurements in sensor networks, user interactions on social networks, or traffic patterns in transportation systems, frequently contain missing entries due to device malfunctions or privacy constraints.

This challenge of extracting information from incomplete observations is far from new. Early statistical work by Wilks in 1932 addressed parameter estimation for bivariate normal distributions with missing values, motivated by incomplete questionnaires in social sciences and government statistics. Lord extended this framework to multivariate settings in 1955 [1]. However, since the early 1970s, the field truly flourished following Rubin's seminal work, which formalized the theoretical foundations of missing-data mechanisms, i.e., the probabilistic relationship between missingness and data values [2]. This framework established that different causes of missingness require fundamentally different analytical approaches. Combined with advances in computational capacity, Rubin's theory catalyzed major methodological developments spanning statistical inference, data analysis, and ML. Over the past three decades in particular, the SP and ML communities have witnessed tremendous progress in adapting classical techniques to handle incomplete signals and in developing entirely new methodologies in core domains, including compressive sensing, parameter estimation, graph SP, subspace tracking (ST), filtering, regression, classification, and time series analysis.

Practitioners typically face three fundamental tasks when confronting missing data: *imputing* missing entries to obtain complete datasets, *estimating* parameters or signals from incomplete observations, and *learning* predictive models despite missing values. Each of these analytical tasks presents distinct challenges and has inspired specialized methodologies, yet these areas are deeply interconnected.

Scope and contributions of this tutorial. This article aims to provide practitioners with comprehensive and accessible tools for dealing with missing data in modern SP/ML applications. While not aiming for exhaustive coverage, we focus on both classical and recent techniques tailored to the needs of the SP/ML community. Our tutorial makes three primary contributions.

First, we provide a clear and unified framework for organizing missing-data strategies around three fundamental tasks: *imputation* (replacing missing entries with predicted values), *estimation* (inferring unknown parameters or signals from incomplete data), and *learning* (predicting target values from incomplete observations or using generative (deep) learning approaches to impute missing values). While these categories are permeable and often intertwined, this organization clarifies the landscape and helps practitioners identify appropriate methods for their applications.

Second, we highlight major methodological advances through detailed case studies grounded in real-world scenarios, demonstrating how contemporary techniques address imputation, estimation, and learning tasks in challenging settings involving irregular and high-dimensional structured data and complex missingness mechanisms.

Third, we reinterpret key statistical concepts through a consistent formalism familiar to the SP community, making foundational theory accessible while connecting it to modern algorithmic implementations, including deep learning approaches.

We begin by introducing the notations used throughout the tutorial, followed by formal definitions of key concepts to model missing data. The tutorial then proceeds through three main sections corresponding to the fundamental tasks of imputation, estimation, and learning, each ending with a brief takeaway. We conclude with a general summary and discuss open challenges and future research directions. Table 1 provides a guideline for readers seeking a general overview of the material covered in this tutorial.

Basic concepts of missing data

This section defines fundamental concepts for handling missing data: the missing-data pattern, the missing-data mechanism, and the notion of ignorability. We represent the data by a $p \times n$ matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_n]$, where variables (or features) are arranged along the rows, and i.i.d. observations (or samples) $\boldsymbol{x}_j$, for $j = 1, \dots, n$, are arranged along the columns. Table 2 summarizes the notation adopted throughout this tutorial.

Missing-data pattern

To handle missing data systematically, it is convenient to define the missing-data *pattern* [3], which characterizes the structure and shape of missingness. We introduce an indicator matrix $\boldsymbol{M} \in \{0, 1\}^{p \times n}$ such that, for the ith element of the jth sample of the data matrix $\boldsymbol{X}$:

$$M_{i,j} = \begin{cases} 0 & \text{if } X_{i,j} \text{ is missing,} \\ 1 & \text{if } X_{i,j} \text{ is observed.} \end{cases} \tag{1}$$

Figure 1 illustrates this notation, showing how the complete data matrix $\boldsymbol{X}$ can be partitioned into its observed ($\boldsymbol{X}_{\mathrm{o}}$) and missing ($\boldsymbol{X}_{\mathrm{m}}$) components. The actually observed data are noted

Table 1. A section-by-section overview of the tutorial, highlighting unified formulations, key covered methods, and the corresponding references in the text.

Framework	Formulation	Algorithms/approaches	In-text reference
Imputation	Imputation via sampling $\hat{\boldsymbol{X}} = (\mathbf{1}_{p\times n} - \boldsymbol{M}) \odot \boldsymbol{Z} + \boldsymbol{M} \odot \boldsymbol{X}$	Mean, model-based, MIs	Equation (3)
	Imputation as an optimization problem $\hat{\boldsymbol{X}} = \underset{\boldsymbol{X}}{\operatorname{argmin}}\, d(\boldsymbol{X}, \boldsymbol{M}) + \alpha\, g(\boldsymbol{X}; \boldsymbol{\Theta}) + \lambda\, h(\boldsymbol{X})$	Graph signal recovery, matrix completion	Equation (4)
Estimation	Ignorable missing-data mechanism $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, f(\boldsymbol{X}_{\text{o}}; \boldsymbol{\theta})$	EM and its variants	Table 5
	Nonignorable missing-data mechanism $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, f(\boldsymbol{X}_{\text{o}}, \boldsymbol{M}; \boldsymbol{\theta})$	SEM and its variants	Table 5
	Prior information $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, d(\boldsymbol{\theta}, \boldsymbol{X}_{\text{o}}) + \lambda R(\boldsymbol{\theta})$	A priori: MAP-EM	Equation (16)
		Reparameterization based	Equation (18)
Learning	Generative approaches for imputation $\hat{\boldsymbol{X}}_{\text{m}} = I_{\hat{\zeta}}(\boldsymbol{X}_{\text{o}}, \boldsymbol{M}), \hat{\boldsymbol{\zeta}} = \underset{\zeta}{\operatorname{argmin}}\, \mathbb{E}\{\mathcal{D}_{\zeta}(\boldsymbol{X}_{\text{o}}, \boldsymbol{M})\}$	MVAE, MIWAE: KL divergence	Equation (23)
		GAIN: adversarial loss	Equation (30)
		OT: Wasserstein distance	Equation (31)
	Label prediction with missing values $\boldsymbol{X}_{\text{train}}$ incomplete: estimate $f(\boldsymbol{y}\,|\,\boldsymbol{X})$ $\boldsymbol{X}_{\text{train}}$ and $\boldsymbol{x}_{\text{new}}$	Naive imputation then debiased algorithms	Equation (33)
	incomplete: estimate $f(\boldsymbol{y}\,|\,\boldsymbol{X}_{\text{o}})$	MIA method, Impute-then-regress strategies	Figure 3

OT: optimal transport; KL: Kullback-Leibler.

Table 2. The main notations used in the tutorial.

p	Number of variables or features	n	Number of observations or samples
$\boldsymbol{X}/X_{i,j}$	$p \times n$ data matrix/its (i, j)-th entry	$[\boldsymbol{X}]_{i,:}$	The ith row of $\boldsymbol{X}$
$\boldsymbol{M}$	Missing-data pattern	$\boldsymbol{x}_j/\boldsymbol{m}_j$	The jth column of $\boldsymbol{X}/\boldsymbol{M}$
$\boldsymbol{X}_{\text{m}}/\boldsymbol{X}_{\text{o}}$	Missing/observed part of the data	$[\boldsymbol{x}]_i$	The ith element of $\boldsymbol{x}$
$\boldsymbol{x}_{\text{m},j}/\boldsymbol{x}_{\text{o},j}$	Missing/observed part of sample $\boldsymbol{x}_j$	$\boldsymbol{\Sigma}$	Covariance or shape matrix
$\lVert\boldsymbol{X}\rVert_{1,1}/\lVert\boldsymbol{X}\rVert_{1,\text{off}}$	L_1 norm of $\boldsymbol{X}$/off-diagonal entries of $\boldsymbol{X}$	$\boldsymbol{\mu}$	Mean or location vector
$\boldsymbol{\theta}$	Vector of unknown parameters	$\odot$	Element-wise (Hadamard) product
$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)})$	Surrogate function at step k of EM	$f(\cdot)$	pdf or pmf
$\mathbb{E}_{\boldsymbol{x}\sim f(\cdot)}[g(\boldsymbol{x})]$	Expectation of $g(\boldsymbol{x})$ with $\boldsymbol{x} \sim f(\cdot)$	$\operatorname{tr}(\cdot)/(\cdot)^{\top}$	Trace operator/transpose operator
$\det(\cdot)/\det^{*}(\cdot)$	Determinant/pseudo-determinant operator	$f(\cdot; \boldsymbol{\theta})$	Identical to $f(\cdot\,|\,\boldsymbol{\theta})$ when $\boldsymbol{\theta}$ is deterministic
$\overset{\text{EVD}}{=}/\overset{\text{SVD}_{\lambda}}{=}$	Eigenvalue/soft thresholded SVD	$\succeq 0$	Is positive semidefinite

pdf: probability density function; pmf: probability mass function.

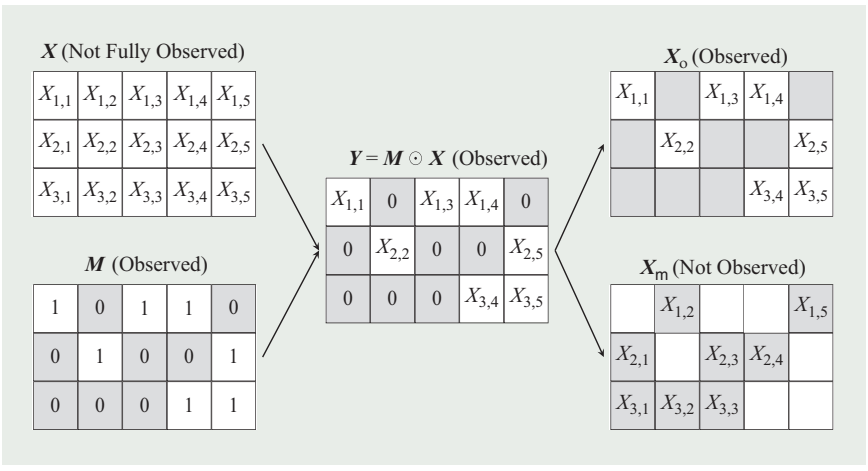

FIGURE 1. An illustration of the missing-data pattern. Blue cells indicate observed entries, and gray cells indicate missing entries. The observed matrix is $\boldsymbol{Y} = \boldsymbol{M} \odot \boldsymbol{X}$, where zeros correspond to missing entries. The complete data matrix $\boldsymbol{X}$ is decomposed into observed $(\boldsymbol{X}_{\text{o}})$ and missing $(\boldsymbol{X}_{\text{m}})$ components using the indicator matrix $\boldsymbol{M}$.

$\boldsymbol{Y} = \boldsymbol{M} \odot \boldsymbol{X}$. Missing-data patterns can be classified into five nonmutually exclusive categories, as illustrated in Figure 2:

1) *Univariate* missing data occur when a single variable is missing for some observations. For example, in environmental or industrial monitoring, a single sensor (e.g., humidity) may fail to report readings at certain time instances.
2) *Multivariate* missing data arise when multiple variables are missing simultaneously. For example, wearable health devices sensing multiple physiological signals (e.g., heart rate, skin temperature, and blood pressure) may fail together due to transmission loss.

3) The *monotone* pattern occurs when the incomplete data can be sorted such that the ith variable is at least as observed as the $(i+1)$-th variable for $i = 1, \ldots, p-1$. A common example arises in sensor arrays, where progressive sensor failures systematically lead to increasing missingness.
4) The *file-matching* pattern occurs in multimodal contexts where at least two sets of variables are never jointly observed. This is often observed when two sensing systems observe the same phenomenon but operate at different sampling frequencies.
5) The *general* pattern indicates that multiple blocks of data are missing across multiple variables in a nonstructured way. This is typical in imaging systems where pixels or regions are missing due to occlusions across multiple images.

Missing-data mechanism

The *cause* of missingness, not just its pattern, profoundly influences the choice of appropriate statistical inference techniques. Before Rubin's seminal work [2], the missing-data pattern $\boldsymbol{M}$ was mostly treated as a deterministic quantity and largely ignored in statistical analysis. Treating $\boldsymbol{M}$ as a random variable fundamentally changed the theory and practice of working with incomplete data, providing a principled framework for dealing with uncertainty in the missingness itself.

The missing-data pattern describes *which* values are missing but not the *relationship* between missingness and data values. This relationship is characterized by the missing-data *mechanism*, formally defined using the conditional distribution of $\boldsymbol{M}$ given $\boldsymbol{X}$, denoted $f(\boldsymbol{M} \mid \boldsymbol{X}; \boldsymbol{\phi})$, parameterized by $\boldsymbol{\phi} \in \Omega_{\boldsymbol{\phi}}$.

Definition 1 (Missing-data mechanism)

The missing-data mechanism is said:

- *Missing completely at random (MCAR):* The probability of a value being missing is independent of the data values, i.e., $f(\boldsymbol{M} \mid \boldsymbol{X}; \boldsymbol{\phi}) = f(\boldsymbol{M}; \boldsymbol{\phi})$.
- *Missing at random (MAR):* The probability of missingness depends only on the observed values, i.e., $f(\boldsymbol{M} \mid \boldsymbol{X}; \boldsymbol{\phi}) = f(\boldsymbol{M} \mid \boldsymbol{X}_{\mathrm{o}}; \boldsymbol{\phi})$.
- *Missing not at random (MNAR):* The probability of missingness depends on the values of the missing variables themselves, and possibly on the observed ones.

The definitions of missing-data mechanisms are frequently subject to discussion [4]. In this tutorial, we assume that the previous statements apply to all possible data patterns and values, corresponding to the everywhere mechanisms described by [4]. Some concrete examples of these mechanisms are presented in "Application 1: Modeling Missing Measurements in Wearable Health Signals."

Ignorability and likelihood inference

The distinction among MCAR, MAR, and MNAR is critical because it determines which methods yield valid inference, as explained later. Under MCAR or MAR, many standard techniques remain applicable; under MNAR, failing to model the missingness mechanism can produce severely biased estimates.

In most applications, the parameter $\boldsymbol{\phi}$ governing the missing-data mechanism is not of primary interest. Statistical inference typically targets the parameter $\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}$ of the data distribution. This leads to the notion of *ignorability*, a key concept that determines whether the missing-data mechanism must be explicitly modeled for valid inference.

Definition 2 (Ignorable missing-data mechanism)

A missing-data mechanism is ignorable for likelihood-based inference if the maximum likelihood estimate of $\boldsymbol{\theta}$ obtained from the observed-data likelihood $f(\boldsymbol{X}_{\mathrm{o}}; \boldsymbol{\theta})$ coincides with that obtained from the joint likelihood $f(\boldsymbol{X}_{\mathrm{o}}, \boldsymbol{M}; \boldsymbol{\theta}, \boldsymbol{\phi})$.

Under MCAR or MAR, the joint maximum likelihood estimation (MLE)

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}, \hat{\boldsymbol{\phi}}_{\mathrm{MLE}} = \underset{(\boldsymbol{\theta}, \boldsymbol{\phi}) \in \Omega_{\boldsymbol{\theta}, \boldsymbol{\phi}}}{\operatorname{argmax}} \log f(\boldsymbol{X}_{\mathrm{o}}, \boldsymbol{M}; \boldsymbol{\theta}, \boldsymbol{\phi})$$

factorizes such that the MLE for $\boldsymbol{\theta}$ can be obtained independently

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \underset{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}}{\operatorname{argmax}} \log f(\boldsymbol{X}_{\mathrm{o}}; \boldsymbol{\theta}),$$

since

$$\begin{aligned} \log f(\boldsymbol{X}_{\mathrm{o}}, \boldsymbol{M}; \boldsymbol{\theta}, \boldsymbol{\phi}) &= \log \int_{\mathcal{X}_{\mathrm{m}}} f(\boldsymbol{X}; \boldsymbol{\theta})\, f(\boldsymbol{M} \mid \boldsymbol{X}; \boldsymbol{\phi})\, d\boldsymbol{X}_{\mathrm{m}} \\ &\overset{\mathrm{MCAR/MAR}}{=} \log \left[\int_{\mathcal{X}_{\mathrm{m}}} f(\boldsymbol{X}; \boldsymbol{\theta})\, d\boldsymbol{X}_{\mathrm{m}} \cdot f(\boldsymbol{M} \mid \boldsymbol{X}_{\mathrm{o}}; \boldsymbol{\phi}) \right] \\ &\overset{(\boldsymbol{\theta}, \boldsymbol{\phi}) \text{ uncoupled}}{=} \log f(\boldsymbol{X}_{\mathrm{o}}, \boldsymbol{\theta}) + \log f(\boldsymbol{M} \mid \boldsymbol{X}_{\mathrm{o}}; \boldsymbol{\phi}). \end{aligned} \tag{2}$$

Here, $f(\boldsymbol{X}_{\mathrm{o}}; \boldsymbol{\theta}) = \int_{\mathcal{X}_{\mathrm{m}}} f(\boldsymbol{X}; \boldsymbol{\theta})\, d\boldsymbol{X}_{\mathrm{m}}$ denotes the *observed-data likelihood*. The conditions in (2) are referred to as ignorability conditions for likelihood inference, as summarized in Lemma 1.

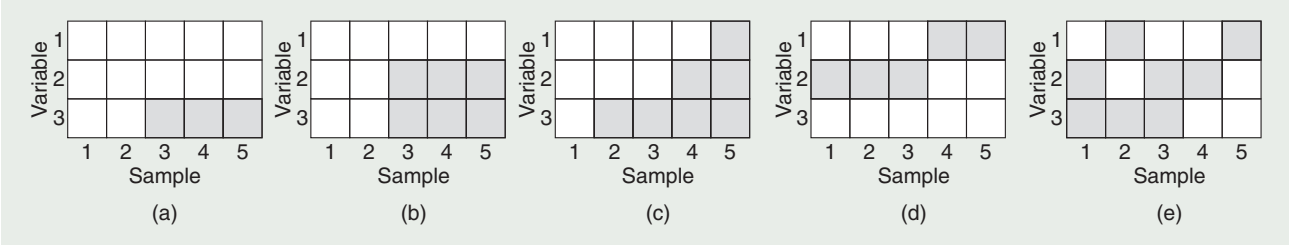

FIGURE 2. Different types of missing-data patterns for multivariate data with $p = 3$ variables and $n = 5$ samples. White cells indicate $\boldsymbol{X}_{\mathrm{o}}$; gray cells indicate $\boldsymbol{X}_{\mathrm{m}}$. (a) Univariate. (b) Multivariate. (c) Monotone. (d) File matching. (e) General.

Application 1: Modeling Missing Measurements in Wearable Health Signals

To illustrate the three missing-data mechanisms, consider a health monitoring system that records two variables over time: heart rate (measured via a wearable device) and blood pressure (measured via a cuff-based monitor). Missing data can arise for different reasons, each corresponding to a distinct mechanism.

1. *MCAR scenario:* The wearable device occasionally experiences technical failures, such as battery depletion or signal dropout. These failures occur independently of the actual measurements, so the remaining data are representative of the complete data (see Figure S1). Simple approaches such as mean imputation or ignoring missing entries may suffice (see the "Missing Data Imputation" section).
2. *MAR scenario:* Older patients may be less likely to complete blood pressure measurements regularly due to age-related forgetfulness or unfamiliarity with the technology. Here, missingness depends on an observed variable (age), making the mechanism ignorable (see Figure S2). More sophisticated algorithms that exploit the relationship between observed variables can effectively handle this case.
3. *MNAR scenario:* Patients may avoid checking their blood pressure when they suspect it is high, perhaps due to anxiety or denial. Missingness now depends on the unobserved value itself, causing the remaining data to systematically underrepresent high blood pressure readings (see Figure S3). Ignoring this mechanism leads to biased estimates; here, a valid inference requires explicitly modeling the relationship between missingness and the missing values.

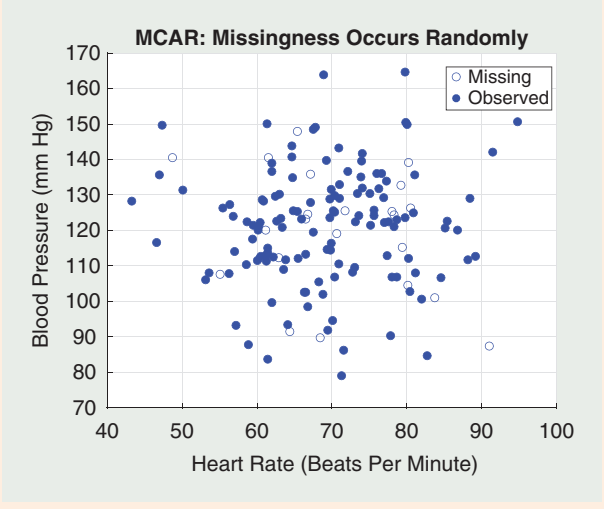

FIGURE S1. The MCAR mechanism. The missing data are independent of the data values.

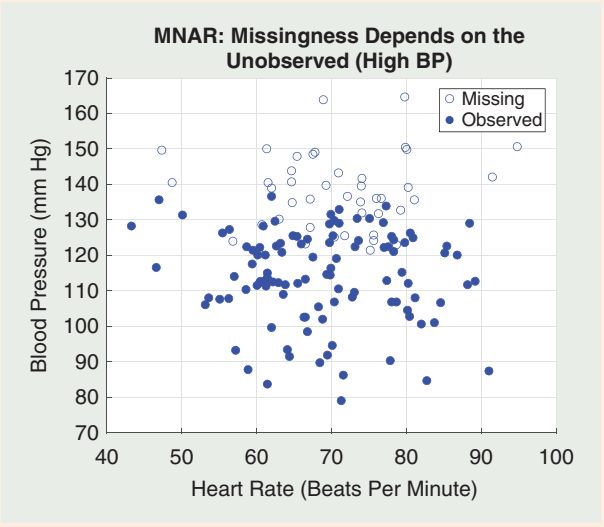

FIGURE S3. The MNAR mechanism. The missing data depend on unobserved data. BP: blood pressure.

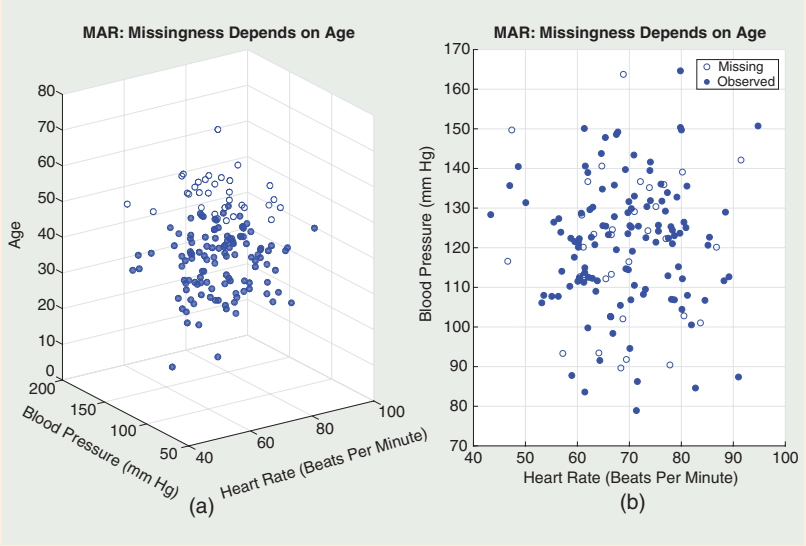

FIGURE S2. (a) and (b) MAR mechanism. The missing data depend on observed data; when adding the age variable (a), patients older than 50 years may be less likely to wear the monitoring device consistently.

Lemma 1 (Ignorability for likelihood inference [3])

The missing-data mechanism is ignorable for likelihood inference if

1) The mechanism is MCAR or MAR.
2) The parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$ are uncoupled in the sense that $\Omega_{\theta,\phi} = \Omega_{\theta} \times \Omega_{\phi}$.

When ignorability holds, likelihood-based inference can proceed using only the observed-data likelihood $f(\boldsymbol{X}_o; \boldsymbol{\theta})$, without modeling the missingness mechanism, which is a simplification in practice.

Missing-data imputation

Equipped with the basic tools to deal with missing values, we now turn to the task of missing-data imputation. This section discusses both classical and recent imputation approaches for various types of signals, including graphs and low-rank matrices. "Application 2: Multiple Imputation of Financial Time Series" further illustrates an application to time series data.

Imputation yields a complete dataset that can be analyzed using standard techniques that require fully observed data, such as parameter estimation, classification, or regression. Additionally, imputation may serve as an initialization step for more sophisticated methods. At this stage, one may ask: Why perform imputation at all? Why not simply discard samples containing missing values? This approach, known as *list-wise deletion* or *complete-case analysis*, is justifiable only when the remaining data are representative of the original distribution and when the proportion of missing values is small [5]. In many applications, deletion is problematic; for example, discarding all segments of an EEG containing transient artifacts may eliminate critical event-related information, fundamentally altering the interpretation of neural activity.

In what follows, the imputed dataset, denoted by $\hat{\boldsymbol{X}}$, is defined as

$$\hat{\boldsymbol{X}} = \underbrace{(\mathbf{1}_{p\times n} - \boldsymbol{M}) \odot \boldsymbol{Z}}_{\text{imputed missing values}} + \underbrace{\boldsymbol{M} \odot \boldsymbol{X}}_{\text{unchanged observed values}} \tag{3}$$

where $\mathbf{1}_{p\times n}$ is a $p \times n$ matrix of ones, and $\boldsymbol{Z}$ contains the values replacing the missing entries of $\boldsymbol{X}$.

Classical imputation methods

The simplest approach is probably *mean imputation*, in which each missing entry is replaced by the mean of the observed values for the corresponding variable, that is $[Z]_{i,:} = (\Sigma_j M_{i,j} X_{i,j})/\Sigma_j M_{i,j}, \forall i \in \{1, \ldots, p\}$. Although straightforward to implement, this method ignores relationships between variables and underestimates the variance. A modest improvement consists of imputing missing values using the conditional mean given observed values within the same sample, plus a random residual to preserve empirical variance. A classic and efficient alternative is to predict missing values from similar samples in the dataset [6]. For each sample with missing data, one identifies its k-nearest neighbors using a distance metric on complete variables and imputes by averaging their values. This method preserves local structure in the data but faces practical challenges as follows:

1) It requires defining an appropriate distance metric over incomplete variables.
2) The distance computation becomes unreliable when many variables are missing.
3) Selecting k is nontrivial and data dependent.

In general, simple methods suffer clear limitations; they fail to capture complex dependence structures and tend to perform poorly when the proportion of missing data is substantial. Nevertheless, they serve two important purposes; they provide interpretable baselines for benchmarking more advanced techniques, and they offer practical solutions when missingness is sparse (e.g., less than 5% for mean imputation) and occurs completely at random (MCAR). In the following, we present more sophisticated imputation methods designed for MCAR or MAR settings, that is, when the missing-data mechanism is assumed to be ignorable (see (2)).

Model-based imputation methods

In *model-based* imputation methods, the entries of $\boldsymbol{Z}$ are sampled from the conditional distribution of the missing data given the observed data, i.e., $\boldsymbol{Z} \sim f(\boldsymbol{X}_m \mid \boldsymbol{X}_o)$.[1] These approaches fall into two broad categories [7, Sects. 4.4–4.6]: *iterative conditional* methods, which directly learn $f(\boldsymbol{X}_m \mid \boldsymbol{X}_o)$, and *full generative* methods, which instead learn the joint distribution $f(\boldsymbol{X}) = f(\boldsymbol{X}_o, \boldsymbol{X}_m)$.

Iterative conditional methods

This first strategy consists of directly estimating the conditional distribution $f(\boldsymbol{X}_m \mid \boldsymbol{X}_o)$ without specifying the full joint distribution $f(\boldsymbol{X})$. The algorithm iterates the following steps until convergence:

1) For each variable i, fit a predictive model using complete samples of $[\boldsymbol{X}]_{i,:}$ (i.e., samples containing no missing values for that variable), with $[\boldsymbol{X}]_{i,:}$ as the target and $[\boldsymbol{X}]_{-i,:}$ (other variables) as predictors.
2) Leverage the fitted model to impute missing values in $[\boldsymbol{X}]_{i,:}$.
3) Repeat for all variables.

The predictor can take various forms, such as a random forest [8], linear regression, or other flexible models. A popular approach is multivariate imputation by chained equations (MICE) [7]. Although these techniques accommodate different data types effectively, they are computationally intensive; each variable requires training a separate model, involving the estimation of p different conditional distributions. Additionally, their convergence and theoretical properties rely on regularity conditions that may not always hold in practice.

Full generative methods

Alternatively, one can specify a parametric distribution for the complete data $\boldsymbol{X}$ and derive samples from the posterior conditional distribution $f(\boldsymbol{X}_m \mid \boldsymbol{X}_o)$. For example, assuming that each observation $\boldsymbol{x}_j \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for $j = 1, \ldots, n$, the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be estimated via the expectation–maximization (EM) algorithm [9] (detailed in the "Estimation With Missing Data Under a Parametric Model Framework" section). This parametric formulation also underlies several imputation strategies discussed later, including matrix completion (see the next section) and generative approaches described in the "Learning With Missing Data" section.

[1] There is a slight abuse of notation here. Rigorously, the conditional distribution is denoted $\boldsymbol{Z} \sim f(\cdot \mid \boldsymbol{X}_o)$.

Application 2: Multiple Imputation of Financial Time Series

Financial data of traded assets frequently contain missing values due to nontrading periods, lack of liquidity, data collection gaps, or handling errors. Traditional interpolation methods include *last observation carried forward* (LOCF), which repeats the most recent observed price, and *linear interpolation* of prices (equivalent to mean interpolation of returns). While simple, these approaches fail to capture the stochastic nature of financial data.

More principled imputation methods exploit structural knowledge about the data. In recommender systems, one leverages low-rank structure; in image inpainting, local geometric similarity. For financial time series, the key structure is *temporal dependence*. One approach is to apply the MICE method [7] with reengineered lagged variables as inputs. Alternatively, one can directly specify a time series model, such as a random walk or autoregressive (AR) process, fit it to the observed data, and use Bayesian inference to impute missing values from the posterior distribution [13]. Figure S4 compares naive interpolation methods with MICE and the model-based ImputeFin approach, demonstrating the superior realism of statistical imputation. When multiple assets are modeled jointly, one can further exploit cross-sectional dependencies.

In applications such as portfolio optimization, backtesting, and risk modeling, it is often desirable to generate multiple plausible imputations rather than a single point estimate, an approach known as *MI*. A probabilistic model for the missing values naturally enables this; one simply draws from the estimated posterior distribution as many times as needed. Figure S5 shows, in random order, the ground-truth S&P 500 index and three representative MIs obtained using the model-based approach of [13], implemented in the R package `imputeFin`.[S1]

As the figures illustrate, model-based statistical imputation performs remarkably well. The underlying method works as follows. To capture both temporal structure and the heavy-tailed nature of financial returns, an AR model is assumed

$$x_t = \mu + a x_{t-1} + \epsilon_t \tag{S1}$$

[S1] https://CRAN.R-project.org/package=imputeFin

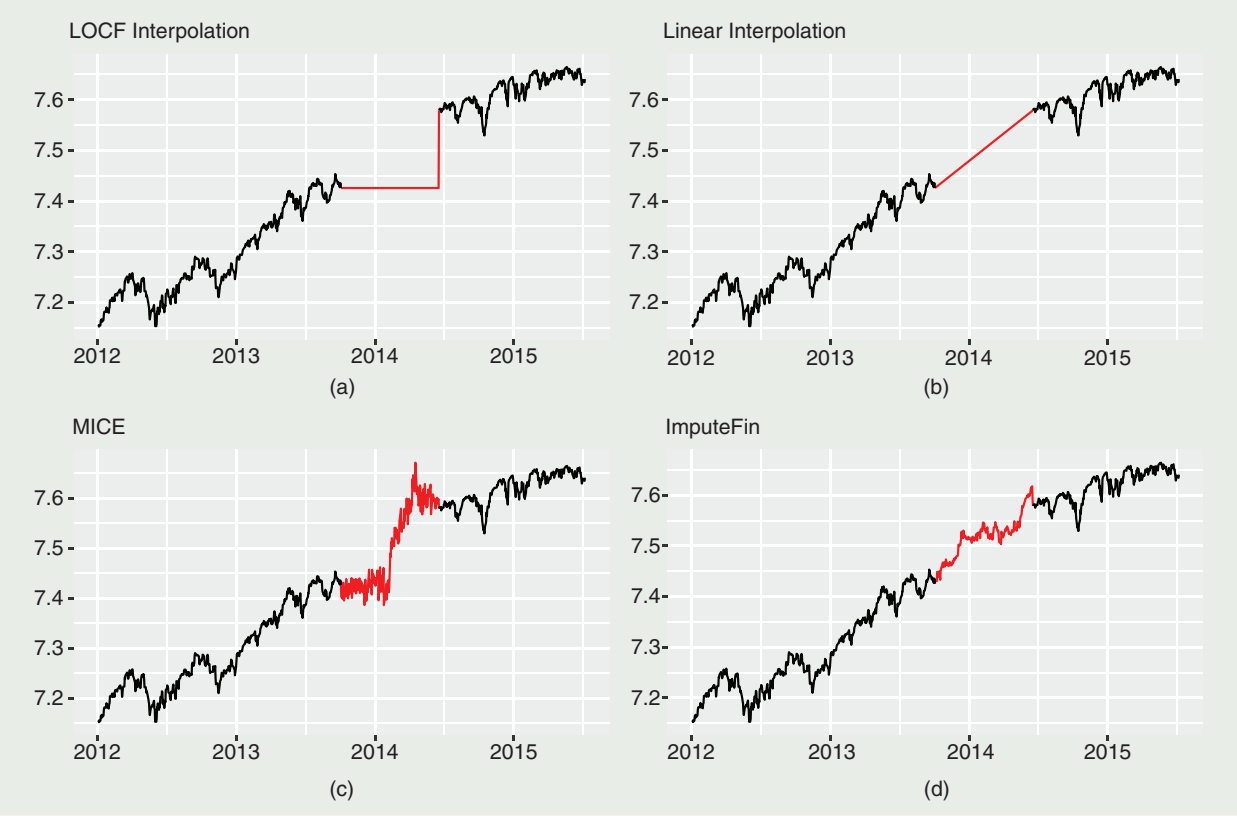

FIGURE S4. (a)–(d) A comparison of naive interpolation, MICE with reengineered lagged variables, and model-based imputation (ImputeFin) for financial time series.

(*Continued*)

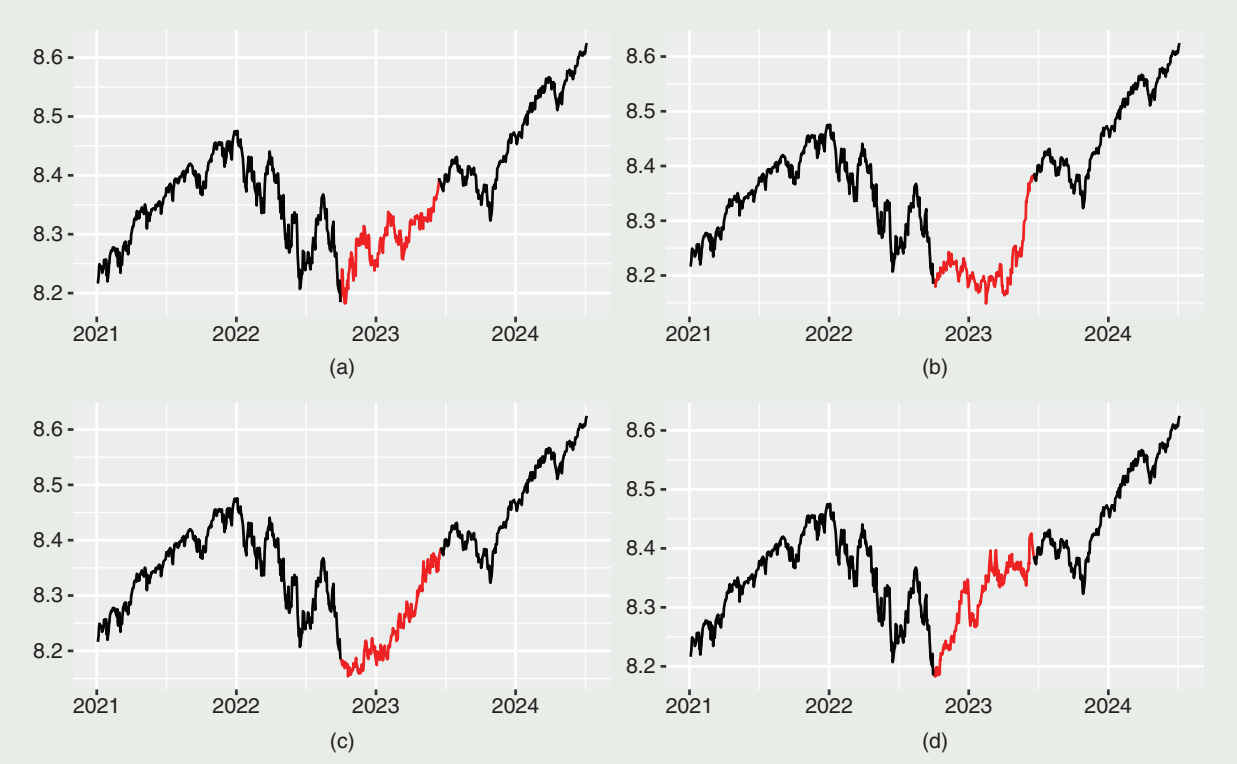

FIGURE S5. (a)–(d) MIs of a financial time series. Can you distinguish the ground truth from the imputations? Interested readers may find the answer in the "Conclusion" section.

where x_t is the time series value at time t, μ is the drift, a is the AR coefficient, and the innovations ϵ_t follow a Student's t distribution, $\epsilon_t \sim t(0, \sigma^2, \nu)$, with scale σ and degrees of freedom $\nu > 0$ controlling tail thickness. The parameter vector is $\boldsymbol{\theta} = (a, \mu, \sigma, \nu)$.

The key challenge is estimating $\boldsymbol{\theta}$ in the presence of missing values. With complete data and Gaussian innovations, this reduces to least squares; with Student's t innovations, the problem becomes more involved but remains tractable. However, with missing values, MLE requires solving

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \log f(\boldsymbol{x}_{\text{o}}; \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \log \int \prod_{t=2}^{n} f(x_t | x_{t-1}; \boldsymbol{\theta}) d\boldsymbol{x}_{\text{m}} \tag{S2}$$

where $\boldsymbol{x}_{\text{o}}$ and $\boldsymbol{x}_{\text{m}}$ denote the observed and missing components of the time series $\boldsymbol{x}$. This integral lacks a closed-form solution due to the Student's t likelihood. The method of [13] addresses this using the stochastic approximation EM (SAEM) algorithm with multiple Monte Carlo samples, which will be discussed in the "Estimation With Missing Data Under a Parametric Model Framework" section (see Table 5).

Multiple imputation methods

In single imputation methods, the entries of $\boldsymbol{Z}$ in (3) are sampled only once. In contrast, *multiple imputation* (MI) generates $M > 1$ plausible imputations for each missing value, resulting in M complete datasets. Each dataset is analyzed independently, and the results are pooled using standard procedures [2], [3] to account for both within-imputation and between-imputation variability. This approach provides a more robust and reliable analysis compared to single imputation. Its advantages are demonstrated through a detailed example involving financial time series data, presented in "Application 2: Multiple Imputation of Financial Time Series."

Imputation as a constrained optimization problem

In this section, we examine imputation approaches focusing on two case studies: signals defined on graphs (*graph signal recovery*) and signals with a low-rank structure (*matrix completion*). Assume that we have observations with missing entries as $\boldsymbol{Y} = \boldsymbol{M} \odot \boldsymbol{X}$ (see Figure 1). The imputed data matrix $\hat{\boldsymbol{X}}$ is obtained by solving a constrained optimization problem of the form[2]

$$\hat{\boldsymbol{X}} = \operatorname*{argmin}_{\boldsymbol{X}} d(\boldsymbol{X}, \boldsymbol{M}) + \alpha g(\boldsymbol{X}; \boldsymbol{\theta}) + \lambda h(\boldsymbol{X}) \tag{4}$$

[2]Note that (3) still applies as a general imputation framework for $\hat{\boldsymbol{X}}$. Due to the specific nature of the signals presented in this section (graphs and low-rank matrices), imputation largely benefits from the tools of optimization.

where $\alpha, \lambda \geq 0$ are regularization parameters. The three terms in this objective serve distinct purposes.

- *Fidelity term* $d(\boldsymbol{X},\boldsymbol{M})$*:* This measures the discrepancy between the reconstructed data and the available observations. A common choice is the squared Frobenius norm of the residual on observed entries.

$$d(\boldsymbol{X},\boldsymbol{M}) = \left\| \boldsymbol{M} \odot (\boldsymbol{Y} - \boldsymbol{X}) \right\|_F^2. \tag{5}$$

  To handle outliers, one can also use the Huber-loss function for the fidelity term, defined as

$$d(\boldsymbol{X},\boldsymbol{M}) = \sum_{i,j} H_\delta(M_{i,j}(Y_{i,j} - X_{i,j})) \quad \text{with}$$

$$H_\delta(x) = \frac{1}{2}x^2 \mathbb{1}_{|x| \leq \delta} + \left(|x| \delta - \frac{\delta^2}{2} \right) \mathbb{1}_{|x| > \delta}$$

  where δ is a constant (slope) parameter that approximately corresponds to the standard deviation of the inlier errors.
- *Smoothness term* $g(\boldsymbol{X};\boldsymbol{\theta})$*:* This quantifies how smoothly the data conform to the model structure encoded by the known parameters $\boldsymbol{\theta}$. For instance, in graph SP, this term enforces consistency with an underlying graph model, where $\boldsymbol{\theta}$ represents the graph Laplacian or adjacency matrix.
- *Regularization term* $h(\boldsymbol{X})$: This promotes the desired properties of the solution, such as bounded energy (via the Frobenius norm, $h(\boldsymbol{X}) = \left\| \boldsymbol{X} \right\|_F^2$) or low-rank structure (via the nuclear norm, $h(\boldsymbol{X}) = \|\boldsymbol{X}\|_*$).

Case Study I—Graph signal recovery

Many applications involve signals defined on graphs, such as sensor networks, social networks, or transportation systems. When such signals contain missing entries, imputation can leverage the underlying graph structure. A graph is typically represented by a (weighted) adjacency matrix $\boldsymbol{W}$, where the entry $W_{i,j}$ denotes the weight of the edge from node i to node j.

Using the framework in (4), with the fidelity term from (5) and $\boldsymbol{\theta} = \boldsymbol{W}$, the remaining terms of (4) can be distinguished as follows:

Smoothness: For an undirected graph with positively weighted edges, where $\boldsymbol{W} = \boldsymbol{W}^T \geq \boldsymbol{0}$, the following definition of smoothness is commonly utilized [10]:

$$g(\boldsymbol{X};\boldsymbol{W}) = \frac{1}{2}\sum_{i,j} W_{i,j} \left\| [\boldsymbol{X}]_{i,:} - [\boldsymbol{X}]_{j,:} \right\|_p^p. \tag{6}$$

Here, $\|\cdot\|_p$ denotes the L_p vector norm, typically chosen as the L_1 or L_2 norm. For $p = 2$, the term $g(\boldsymbol{X};\boldsymbol{W})$ is quadratic in $\boldsymbol{X}$ and is also referred to as *graph Tikhonov regularization*. Several works also employ a $p = 1$ variant of the graph-based smoothness, known as *graph total variation*, which is suitable for edge-preserving denoising. Based on this measure, a graph signal is smooth when the signal values at connected nodes are similar, with the degree of similarity proportional to the weight of the connection $W_{i,j}$. For graph signals with a temporal dimension (e.g., time series on a network), one can use *spatiotemporal smoothness* by replacing $\boldsymbol{X}$ in (6) with the first-order temporal difference matrix $\Delta(\boldsymbol{X}) = [\boldsymbol{x}_1, \boldsymbol{x}_2 - \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n - \boldsymbol{x}_{n-1}]$. The resulting measure simultaneously enforces spatial smoothness (via graph-domain similarities) and temporal smoothness (via finite differences of temporal samples). For directed graphs, where $\boldsymbol{W}$ is nonsymmetric and may have negative edge weights, a slightly different notion of smoothness can be defined by measuring signal variations across the edges as

$$g(\boldsymbol{X};\boldsymbol{W}) = \sum_{j=1}^{n} \left\| \boldsymbol{x}_j - \frac{1}{|\lambda|_{\max}(\boldsymbol{W})} \boldsymbol{W}\boldsymbol{x}_j \right\|_p^p \tag{7}$$

where $|\lambda|_{\max}(\boldsymbol{W})$ denotes the eigenvalue of $\boldsymbol{W}$ with the largest magnitude in absolute value [11]. This formulation, with $p = 1$ and $p = 2$, gives alternative definitions for graph total variation and Tikhonov regularization for directed graphs, respectively.

Regularization: In graph signal imputation, the regularization term is frequently selected as the Frobenius norm $h(\boldsymbol{X}) = \|\boldsymbol{X}\|_F^2$ or simply omitted by setting $h(\boldsymbol{X}) = 0$.

Using the smoothness term $g(\boldsymbol{X};\boldsymbol{W})$ with $p \geq 1$ and a convex function $h(\boldsymbol{X})$, the graph signal recovery problem becomes convex and can be efficiently solved using standard convex optimization methods, such as alternating direction method of multipliers (ADMM), proximal gradient, or majorization-minimization.

Case Study II—Low-rank matrix completion

Low-rank matrix completion addresses the recovery of a matrix $\boldsymbol{X}$ from partial observations $\boldsymbol{Y} = \boldsymbol{M} \odot \boldsymbol{X}$ under the assumption that $\boldsymbol{X}$ has a low-rank structure. The ideal problem is

$$\min_{\boldsymbol{X}} \quad \text{rank}(\boldsymbol{X}) \quad \text{subject to} \quad \boldsymbol{M} \odot (\boldsymbol{X} - \boldsymbol{Y}) = \boldsymbol{0}. \tag{8}$$

The inherent nonconvexity introduced by the rank constraint makes the problem NP-hard and therefore intractable in general. A standard convex relaxation replaces the rank constraint with the nuclear norm, fitting the generic framework (4) with the fidelity term given in (5) and

$$\textbf{Smoothness:} \quad g(\boldsymbol{X};\boldsymbol{\theta}) = 0 \tag{9}$$

$$\textbf{Regularization:} \quad h(\boldsymbol{X}) = \left\| \boldsymbol{X} \right\|_*, \tag{10}$$

where $\left\| \boldsymbol{X} \right\|_*$ denotes the nuclear norm (sum of singular values). The absence of an explicit smoothness penalty reflects the assumption that the data matrix itself, through its low-rank structure, embodies sufficient intrinsic regularity. This formulation is widely employed in recommender systems, sensor array processing, and related applications in which the observations are assumed to lie approximately in a low-dimensional subspace.

By noticing that $\left\| \boldsymbol{X} \right\|_* = \min_{\boldsymbol{U},\boldsymbol{V}:\boldsymbol{X}=\boldsymbol{U}\boldsymbol{V}^\top} 1/2 \left\| \boldsymbol{U} \right\|_F^2 + 1/2 \left\| \boldsymbol{V} \right\|_F^2$, where $\boldsymbol{U} \in \mathbb{R}^{p \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times r}$ such that $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$, and r is the rank of $\boldsymbol{X}$, the optimization problem in (4) can be rewritten as follows:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \left\| \boldsymbol{M} \odot (\boldsymbol{U}\boldsymbol{V}^\top - \boldsymbol{Y}) \right\|_F^2 + \lambda \left(\left\| \boldsymbol{U} \right\|_F^2 + \left\| \boldsymbol{V} \right\|_F^2 \right). \tag{11}$$

This new formulation is very close to the problem of retrieving an underlying subspace $\boldsymbol{U}$ from the data $\boldsymbol{Y}$ in the *batch* setting, i.e., when $\boldsymbol{Y}$ has a fixed number of observations (columns). It is also closely connected to online subspace estimation with missing data—commonly referred to as *subspace tracking (ST)*—which arises when the data matrix is constructed sequentially

from a stream of observations (see the "Estimation With Missing Data Under a Parametric Model Framework" section).

To solve the optimization problem (11), classical methods rely on proximal iterative algorithms based on singular value decomposition (SVD). In practice, the solution is usually computed by performing a grid search over the regularization parameter λ. More specifically, a widely used algorithm, soft-Impute [12], proceeds iteratively from an initial imputed matrix $\hat{\boldsymbol{X}}^{(0)}$ (e.g., obtained via mean imputation) and a fixed value of λ, alternating between the following two steps:

1) Compute the soft thresholded SVD of $\hat{\boldsymbol{X}}^{(k)}$, i.e., compute the SVD and retain the singular values higher than a threshold λ, $\hat{\boldsymbol{X}}^{(k)} \overset{\text{SVD}_\lambda}{=} \boldsymbol{U}^{(k)}\boldsymbol{S}_\lambda^{(k)}\boldsymbol{V}^{(k)\top}$. Here, $\boldsymbol{S}_\lambda^{(k)}$ is a diagonal matrix such that $[\boldsymbol{S}_\lambda^{(k)}]_{ii} = \max((\sigma_i - \lambda), 0)$, where σ_i are the singular values of $\hat{\boldsymbol{X}}^{(k)}$, for $i \in \{1, \ldots, p\}$.
2) Update the imputed data matrix (only on the missing values) with the quantity computed in step 1, as: $\hat{X}_{i,j}^{(k+1)} = (1 - M_{i,j})[(\boldsymbol{U}^{(k)}\boldsymbol{S}_\lambda^{(k)}\boldsymbol{V}^{(k)T})]_{i,j} + M_{i,j}X_{i,j}$.

Takeaway

For imputation, traditional schemes such as mean imputation provide computationally efficient baselines but underestimate variability and may introduce bias, particularly for MAR or MNAR data and under complex dependency structures. Robust alternatives include model-based techniques such as MI (e.g., MICE), which replace missing values with samples from the conditional distribution $f(\boldsymbol{X}_\text{m} \mid \boldsymbol{X}_\text{o})$. For data with specific structures, such as graph-structured signals and low-rank data, imputation can be formulated as a constrained optimization problem—see Table 3 for a detailed comparison of these two approaches. As a rule of thumb, simple methods suffice when missingness is sparse (less than 5 to 10%) and MCAR; otherwise, model-based approaches are strongly preferred.

Estimation with missing data under a parametric model framework

In this section, we address estimation and inference in the presence of missing data under a parametric model intended to capture the underlying data-generating process. The primary methodological framework is based on the construction of a likelihood-based objective function derived from MLE, under both ignorable and nonignorable missing-data mechanisms. This framework can be further regularized to incorporate structural assumptions or prior information available to the practitioner, such as low-rank constraints, parameter restrictions, or other domain-specific knowledge.

Statistical estimation under an ignorable mechanism

Assume that the parametric model associated with a data-generating process and the missingness mechanism for the jth sample reads $f(\boldsymbol{x}_j, \boldsymbol{m}_j; \boldsymbol{\theta}, \boldsymbol{\phi})$, where $\boldsymbol{\theta}$ denotes the deterministic unknown parameter of interest (e.g., the mean vector $\boldsymbol{\mu}$ in $\mathbb{R}^p$ or $\mathbb{C}^p$, or the covariance matrix $\boldsymbol{\Sigma}$, in the cone of positive semidefinite matrices), and $\boldsymbol{\phi}$ is the set of parameters governing the missingness mechanism. Finally, $\boldsymbol{x}_j$ and $\boldsymbol{m}_j$ denote the jth sample and missingness pattern (see Table 2). Under an *ignorable* missing-data mechanism with multiple independent samples, the MLE is given by:

$$\hat{\boldsymbol{\theta}}_\text{MLE} = \underset{\boldsymbol{\theta} \in \Omega_\theta}{\operatorname{argmax}} \sum_j \log \int f(\boldsymbol{x}_{\text{o},j}, \boldsymbol{x}_{\text{m},j}; \boldsymbol{\theta})\, d\boldsymbol{x}_{\text{m},j}.$$

In general, this problem does not admit a closed-form solution. An exception is the MLE of the mean and/or covariance matrix of a multivariate normal distribution from incomplete data with a monotone pattern under MCAR or MAR mechanisms [3], [14]. In most practical settings, MLE with incomplete data is carried out using the EM algorithm [15] or one of its variants. The EM algorithm is particularly well suited to missing-data problems as it treats the unobserved entries as latent variables. At iteration k, the E-step computes the conditional expectation of the complete-data log-likelihood given the observed data and the previous parameter estimate, while the M-step updates the parameters by maximizing this conditional expectation. The procedure is initialized at $k = 0$ with an initial estimate $\boldsymbol{\theta}^{(0)}$, typically obtained via a simple imputation scheme or one of the imputation methods described previously. The EM algorithm then alternates between the following two steps until convergence:

- *E-step (expectation)*: This step consists of computing a surrogate function, denoted by $Q(\cdot)$, which is the conditional expectation of the complete-data log-likelihood conditioned on the observed data and the previous estimate of the parameter $\boldsymbol{\theta}^{(k)}$. The complete data comprise both the observed data and latent data; in our setting, $\boldsymbol{X} = (\boldsymbol{X}_\text{o}, \boldsymbol{X}_\text{m})$, which yields the following:

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)}) = \mathbb{E}_{\boldsymbol{X}_\text{m} \sim f(\boldsymbol{X}_\text{m} \mid \boldsymbol{X}_\text{o}; \boldsymbol{\theta}^{(k)})}[\log f(\boldsymbol{X}_\text{o}, \boldsymbol{X}_\text{m}; \boldsymbol{\theta})]. \tag{12}$$

- *M-step (maximization)*: This step updates the unknown parameter by maximizing the surrogate function

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta} \in \Omega_\theta}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)}). \tag{13}$$

Table 3. A comparison between graph signal recovery and low-rank matrix completion.

Aspect	Graph signal recovery	Low-rank matrix completion
Structural prior	Graph smoothness	Low-rank structure
Optimization problem	Often convex with L_1 (Total Variation), L_2 (Tikhonov) smoothness term	Originally nonconvex Convex with nuclear norm
Vectors versus matrices	Handles both vectors and matrices	Primarily handles matrices
Graph dependence	Requires a known or learned graph	No graph in classical form
Typical algorithms	Proximal/gradient methods, ADMM	Soft-thresholded SVD, ADMM
Complexity driver	Size of the graph, vertex domain operations (if used)	Matrix size and rank, repeated SVDs
Main applications	Sensor networks	Image inpainting

The EM algorithm offers several advantages, including ease of implementation, a clear statistical interpretation without ad hoc imputation, and general convergence guarantees. However, when the proportion of missing information is high, convergence can be slow. In the following, we briefly outline strategies for implementing EM under different data distributions and present the main EM variants.

Case Study I—The Gaussian distribution

Assume that the samples $\boldsymbol{x}_j$, $j=1,\ldots,n$, are independent and that $\boldsymbol{x}_j \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Under a MAR or MCAR mechanism, the EM surrogate function reads

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) \propto n \log\det(\boldsymbol{\Sigma}) - \sum_j \mathbb{E}_{\boldsymbol{x}_{\mathrm{m},j}\sim f(\boldsymbol{x}_{\mathrm{m},j}|\boldsymbol{x}_{\mathrm{o},j};\boldsymbol{\theta}^{(k)})}[(\boldsymbol{x}_j-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu})].$$

First, since $\boldsymbol{x}_j$ contains the observed values $\boldsymbol{x}_{\mathrm{o},j}$ and the missing ones $\boldsymbol{x}_{\mathrm{m},j}$, the expectation is taken only with respect to the missing values. Second, the term under the expectation contains the sufficient statistics $\boldsymbol{x}_{\mathrm{m},j}$ and $\boldsymbol{x}_{\mathrm{m},j}\boldsymbol{x}_{\mathrm{m},j}^\top$. Consequently, being able to compute $\mathbb{E}_{\boldsymbol{x}_{\mathrm{m},j}\sim f(\boldsymbol{x}_{\mathrm{m},j}|\boldsymbol{x}_{\mathrm{o},j};\boldsymbol{\theta}^{(k)})}[\boldsymbol{x}_{\mathrm{m},j}]$ and $\mathbb{E}_{\boldsymbol{x}_{\mathrm{m},j}\sim f(\boldsymbol{x}_{\mathrm{m},j}|\boldsymbol{x}_{\mathrm{o},j};\boldsymbol{\theta}^{(k)})}[\boldsymbol{x}_{\mathrm{m},j}\boldsymbol{x}_{\mathrm{m},j}^\top]$ yields a closed-form expression for $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)})$. In the Gaussian setting, these expectations follow directly from the conditional distribution $\boldsymbol{x}_{\mathrm{m},j}\,|\,\boldsymbol{x}_{\mathrm{o},j} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathrm{m}|\mathrm{o}}, \boldsymbol{\Sigma}_{\mathrm{m}|\mathrm{o}})$, where

$$\boldsymbol{\mu}_{\mathrm{m}|\mathrm{o}} = \boldsymbol{\mu}_{\mathrm{m}} + \boldsymbol{\Sigma}_{\mathrm{m,o}}\,\boldsymbol{\Sigma}_{\mathrm{o,o}}^{-1}(\boldsymbol{x}_{\mathrm{o},j}-\boldsymbol{\mu}_{\mathrm{o}})$$

and

$$\boldsymbol{\Sigma}_{\mathrm{m}|\mathrm{o}} = \boldsymbol{\Sigma}_{\mathrm{m,m}} - \boldsymbol{\Sigma}_{\mathrm{m,o}}\,\boldsymbol{\Sigma}_{\mathrm{o,o}}^{-1}\boldsymbol{\Sigma}_{\mathrm{o,m}}$$

Here, $\boldsymbol{\mu}_{\mathrm{m}}$ denotes the mean of the missing components, $\boldsymbol{\Sigma}_{\mathrm{m,m}}$ their covariance matrix, and $\boldsymbol{\Sigma}_{\mathrm{m,o}}$ the cross-covariance between missing and observed components [3].

Case Study II—The scaled Gaussian distribution

A scaled Gaussian distribution—also called *compound Gaussian*—is beneficial as it adjusts variance to better fit the data, providing greater flexibility for capturing different scales [16]. This adaptability mitigates the influence of outliers, preventing them from disproportionately influencing the model. Namely, a vector $\boldsymbol{x}$ following a scaled Gaussian distribution admits the stochastic representation $\boldsymbol{x} \overset{d}{=} \sqrt{\tau}^{-1}\boldsymbol{n}$, where $\overset{d}{=}$ denotes equality in distribution, $\tau > 0$ is a random *texture* variable with distribution $f(\cdot;\boldsymbol{\psi})$ (unknown parameter $\boldsymbol{\psi}$), and $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian *speckle* component.[3]

[3]In this setting, $\boldsymbol{\theta}$ includes $\boldsymbol{\psi}$; assuming $\boldsymbol{\psi}$ known may lead to model mismatch, as illustrated in [17].

In the presence of missing values, a common trick is to apply the chain rule formula to reduce the problem to the standard Gaussian setting described previously. In this case, the complete data consist of $\{\boldsymbol{x}_{\mathrm{o},j}, \boldsymbol{x}_{\mathrm{m},j}, \tau_j\}_j$, and the EM surrogate function is

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) = \sum_j \mathbb{E}_{\tau_j\sim f(\tau_j|\boldsymbol{x}_{\mathrm{o},j};\boldsymbol{\theta}^{(k)})}[\mathbb{E}_{\boldsymbol{x}_{\mathrm{m},j}\sim f(\boldsymbol{x}_{\mathrm{m},j}|\tau_j,\boldsymbol{x}_{\mathrm{o},j};\boldsymbol{\theta}^{(k)})}[\log f(\boldsymbol{x}_{\mathrm{m},j}, \boldsymbol{x}_{\mathrm{o},j}|\tau_j;\boldsymbol{\theta})]].$$

The inner expectation with respect to $\boldsymbol{x}_{\mathrm{m},j}$ can be derived using the same methodology as in the Gaussian case. The outer expectation reduces to evaluating $\mathbb{E}_{\tau_j\sim f(\tau_j|\boldsymbol{x}_{\mathrm{o},j};\boldsymbol{\theta}^{(k)})}[\tau_j]$ and $\mathbb{E}_{\tau_j\sim f(\tau_j|\boldsymbol{x}_{\mathrm{o},j};\boldsymbol{\theta}^{(k)})}[\tau_j^{-1}]$, which are generally known or can be efficiently approximated [18]. For example, if τ follows a Gamma distribution, x follows a Student's t distribution [19].

Case Study III—The elliptically symmetric distribution

Recently, elliptically symmetric distributions have become popular in SP [16] because they generalize the multivariate normal distribution while preserving useful properties, such as ellipsoidal contours. They provide more flexibility for modeling data with outliers or heavy tails, with applications in finance, radar and array processing, biomedical applications, and computational imaging. Moreover, these distributions remain mathematically tractable, enabling straightforward adaptation of statistical methods developed for the Gaussian case and providing a robust framework when normality assumptions are overly restrictive. Consider a random vector $\boldsymbol{x} \sim ES(\boldsymbol{\mu}, \boldsymbol{\Sigma}, g)$ with an elliptically symmetric distribution, where g is the so-called *density generator*. The associated probability density function (pdf) reads

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} g((\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})).$$

By choosing different functions $g(\cdot)$, under certain mild conditions, this framework spawns a variety of specific distributions (see Table 4 for examples). With missing values, the E-step of the EM algorithm reduces to

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) \propto -\frac{n}{2}\log\det(\boldsymbol{\Sigma}) + \sum_j \mathbb{E}_{\boldsymbol{x}_{\mathrm{m},j}\sim f(\boldsymbol{x}_{\mathrm{m},j}|\boldsymbol{x}_{\mathrm{o},j};\boldsymbol{\theta}^{(k)})}[\log g((\boldsymbol{x}_j-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}))]. \tag{14}$$

Case Study IV—EM's stochastic variants

The expectation in (14) is not always tractable, particularly for general forms of the density generator. When the E-step is intractable, a solution is to use a stochastic variant of the EM algorithm. Rather

Table 4. Examples of common elliptically symmetric distributions. $\Gamma(\cdot)$ and $K(\cdot)$ denote the Gamma function and the modified Bessel function of the second kind, respectively.

	Gaussian	Student's t ν: degree of freedom	Generalized Gaussian s, b: shape, and scale	K-distribution ν: shape
$g(r)$	$\frac{\exp\left(-\frac{r}{2}\right)}{(2\pi)^{n/2}}$	$\frac{(1+r/\nu)^{-\frac{n+\nu}{2}}\Gamma\left(\frac{\nu+n}{2}\right)}{(\nu\pi)^{\frac{n}{2}}\Gamma(\nu/2)}$	$\frac{s\exp\left(-\frac{r^s}{2^s b}\right)\Gamma(n/2)}{(2\pi)^{n/2}b^{n/2s}\Gamma(n/2s)}$	$\frac{\nu^{n/2}(2\nu r)^{(2\nu-n)/4}}{2^{\nu-1}\pi^{n/2}\Gamma(\nu)}K_{\nu-n/2}(\sqrt{2\nu r})$

than computing the full expectation, it is approximated using random samples, often via Monte Carlo methods, allowing efficient parameter estimation while preserving convergence under certain conditions. Several stochastic EM (SEM) variants have been developed for such scenarios (see Table 5). In SEM, the full E-step is replaced by a sample-based approximation, using one or a few draws from the conditional distribution of the missing observations. This approach enables handling large or complex datasets where the exact expectation is impractical. Monte Carlo EM (MCEM) extends this by employing multiple Monte Carlo samples to approximate the E-step more accurately; sampling schemes such as Metropolis–Hastings [18] or Gibbs sampling [13] can be deployed. MCEM is particularly useful when analytical solutions are unavailable, but sampling is feasible [20]. Finally, stochastic approximation EM (SAEM) combines stochastic updates with a smoothing mechanism. Instead of recomputing the full expectation at each iteration, SAEM updates it incrementally according to

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k+1)}) = Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) + \gamma_k\left[\log f(\boldsymbol{X}_{\mathrm{o}},\boldsymbol{X}_{\mathrm{m}}^{(k)};\boldsymbol{\theta}) - Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)})\right]$$

where $\boldsymbol{X}_{\mathrm{m}}^{(k)} \sim f(\boldsymbol{X}_{\mathrm{m}} \mid \boldsymbol{X}_{\mathrm{o}};\boldsymbol{\theta}^{(k)})$, and $0 < \gamma_k < 1$ is a step-size sequence. This stochastic approximation improves numerical stability and convergence, even when only a small number of samples are used per iteration [3], [16].

The preceding analysis holds under an ignorable missing-data mechanism (cf. Definition 2 (Ignorable Missing-Data Mechanism). However, inference under ignorable assumptions can be restrictive in many practical SP applications. When the missingness mechanism is nonignorable, neglecting the dependence between missingness and unobserved values generally results in biased parameter estimates and invalid inference. In the following, we focus on estimation procedures that explicitly account for nonignorable missing-data mechanisms.

Statistical inference with nonignorable mechanism

As discussed in the "Basic Concepts of Missing Data" section, when data are MNAR, relying solely on the observed likelihood can lead to severely biased estimates. In this situation, it is necessary to consider the joint distribution $f(\boldsymbol{X}_{\mathrm{o}},\boldsymbol{M};\boldsymbol{\theta},\boldsymbol{\phi})$. Two key scenarios arise, depending on whether the missingness mechanism is nonignorable but known (e.g., censored data with a known threshold due to the operating range of a sensor) or nonignorable and unknown, in which case $\boldsymbol{\phi}$ must be estimated jointly with $\boldsymbol{\theta}$. In the following, we focus on the latter scenario, assuming independent observations for simplicity, so that $f(\boldsymbol{X},\boldsymbol{M};\boldsymbol{\theta},\boldsymbol{\phi}) = \prod_{i,j} f(X_{i,j},M_{i,j};\boldsymbol{\theta},\boldsymbol{\phi})$. Several models can be employed for inference under a nonignorable mechanism, depending on the parameters of interest. In the SP context, where the marginal distribution of the data is typically the primary target, the *selection model* is particularly suitable. Specifically, the selection model reads

$$f(X_{i,j}, M_{i,j};\boldsymbol{\theta},\boldsymbol{\phi}) = f(X_{i,j};\boldsymbol{\theta}) f(M_{i,j} \mid X_{i,j};\boldsymbol{\phi}). \tag{15}$$

This formulation requires specifying both the data model (e.g., a regression model for observed and missing outcomes) and the missingness mechanism (e.g., a logistic or probit model for the missingness probabilities). A common choice for the conditional distribution of the missingness pattern is

$$f(M_{i,j} \mid X_{i,j};\boldsymbol{\phi}) = (h(X_{i,j};\boldsymbol{\phi}))^{M_{i,j}}(1 - h(X_{i,j};\boldsymbol{\phi}))^{1-M_{i,j}}$$

where $h(\cdot)$ is any specific function mapping the data to the probability of missingness.

In the context of MNAR—specifically, considering the joint distribution $f(\boldsymbol{X}_{\mathrm{o}},\boldsymbol{M};\boldsymbol{\theta},\boldsymbol{\phi})$—the MLE is generally impractical, and closed-form expressions for $Q(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{\theta}^{(k)},\boldsymbol{\phi}^{(k)})$ in the EM algorithm are rarely available [20]. Again, stochastic variants of the EM (see Table 5) are therefore commonly employed in this scenario. As an example, the SEM reads

- Stochastic E-step (SE) step

$$\begin{aligned} Q^{\mathrm{SEM}}(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{\theta}^{(k)},\boldsymbol{\phi}^{(k)}) = \\ \sum_i\sum_j M_{i,j}(\log f(X_{i,j};\boldsymbol{\theta}) + \log f(M_{i,j} \mid X_{i,j};\boldsymbol{\phi})) \\ + (1 - M_{i,j})(\log f(X_{i,j}^{(k)};\boldsymbol{\theta}) + \log f(M_{i,j} \mid X_{i,j}^{(k)};\boldsymbol{\phi})) \end{aligned}$$

with $X_{i,j}^{(k)} \sim f(X_{i,j} \mid M_{i,j};\boldsymbol{\theta}^{(k)},\boldsymbol{\phi}^{(k)})$.

- M-step

$$\begin{aligned} \boldsymbol{\theta}^{(k+1)} &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\Omega_{\boldsymbol{\theta}}} \sum_i\sum_j M_{i,j}\log f(X_{i,j};\boldsymbol{\theta}) + (1-M_{i,j})\log f(X_{i,j}^{(k)};\boldsymbol{\theta}) \\ \boldsymbol{\phi}^{(k+1)} &= \operatorname*{argmax}_{\boldsymbol{\phi}\in\Omega_{\boldsymbol{\phi}}} \sum_i\sum_j M_{i,j}\log f(M_{i,j} \mid X_{i,j};\boldsymbol{\phi}) \\ &\quad + (1 - M_{i,j})\log f(M_{i,j} \mid X_{i,j}^{(k)};\boldsymbol{\phi}). \end{aligned}$$

We have shown that SEM variants can handle nonignorable missingness, delivering valid parameter estimates where standard EM fails. Their main drawback is the computational cost of sampling from the posterior distribution of the missing data. Recent advances address this issue by approximating the posterior in the E-step, cutting computational overhead while maintaining statistical accuracy [21].

Table 5. A comparison of E-step formulations in EM variants [20].

	E-Step description	Analytical equation
EM	Computes the exact conditional expectation of the complete-data log-likelihood	$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) = \mathbb{E}_{\boldsymbol{X}_{\mathrm{m}}\sim f(\boldsymbol{X}_{\mathrm{m}}\mid\boldsymbol{X}_{\mathrm{o}};\boldsymbol{\theta}^{(k)})}[\log f(\boldsymbol{X}_{\mathrm{o}},\boldsymbol{X}_{\mathrm{m}};\boldsymbol{\theta})]$
SEM	Approximates the E-step using a single sample from the posterior	$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) \approx \log f(\boldsymbol{X}_{\mathrm{o}},\boldsymbol{X}_{\mathrm{m}}^{(k)};\boldsymbol{\theta})$ where $\boldsymbol{X}_{\mathrm{m}}^{(k)} \sim f(\boldsymbol{X}_{\mathrm{m}} \mid \boldsymbol{X}_{\mathrm{o}};\boldsymbol{\theta}^{(k)})$
MCEM	Uses K samples to estimate the expectation in the E-step	$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) \approx \frac{1}{K}\sum_{c=1}^{K}\log f(\boldsymbol{X}_{\mathrm{o}},\boldsymbol{X}_{\mathrm{m}}^{(k),c};\boldsymbol{\theta})$ where $\boldsymbol{X}_{\mathrm{m}}^{(k),c} \sim f(\boldsymbol{X}_{\mathrm{m}} \mid \boldsymbol{X}_{\mathrm{o}};\boldsymbol{\theta}^{(k)})$
SAEM	Uses a recursive update combining current and past estimates	$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k+1)}) = Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) + \gamma_k[\log f(\boldsymbol{X}_{\mathrm{o}},\boldsymbol{X}_{\mathrm{m}}^{(k)};\boldsymbol{\theta}) - Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)})]$ $\boldsymbol{X}_{\mathrm{m}}^{(k)} \sim f(\boldsymbol{X}_{\mathrm{m}} \mid \boldsymbol{X}_{\mathrm{o}};\boldsymbol{\theta}^{(k)}), \quad 0 < \gamma_k < 1$

Estimation with prior information on the parameter space

In this section, we focus on parameter estimation from incomplete observations by leveraging two types of *prior information*, namely a statistical prior and a structural prior. Specifically, the framework estimates the parameter $\boldsymbol{\theta}$ by minimizing a cost function that balances data fidelity and parameter constraints

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\Omega_{\boldsymbol{\theta}}}{\operatorname{argmin}}\; d(\boldsymbol{\theta}, \boldsymbol{X}_{\mathrm{o}}) + \lambda R(\boldsymbol{\theta}), \tag{16}$$

where $\Omega_{\boldsymbol{\theta}}$ is the parameter space. Problem (16) encompasses two critical estimation paradigms in SP. The first uses *statistical priors*; by defining the distance $d(\boldsymbol{\theta}, \boldsymbol{X}_{\mathrm{o}})$ as the negative log-likelihood and $R(\boldsymbol{\theta})$ as the negative log-prior, the framework recovers the EM–maximum a posteriori (EM–MAP) estimator. The second leverages *structural priors*, constraining or parameterizing unknown parameters and/or using a structural regularizer $R(\boldsymbol{\theta})$ (e.g., the nuclear norm). This approach naturally extends to probabilistic principal component analysis (PPCA) and subspace tracking, both central for recovering dynamic low-dimensional signals.

Statistical prior

In SP applications, the statistical prior is typically modeled through a distribution on the unknown parameter $\boldsymbol{\theta}$, denoted $f(\boldsymbol{\theta})$. As an example, in **radar and surveillance**, the prior $f(\boldsymbol{\theta})$ can be centered on an expected angle of arrival, whereas in **array processing**, a Bernoulli-Gaussian mixture distribution is an effective choice for $f(\boldsymbol{\theta})$ when the number of active sources is small and unknown, leading to a sparse scenario [22]. In the presence of missing data, incorporating prior information can be efficiently achieved via the EM–MAP framework. The MAP estimator reads

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \underset{\boldsymbol{\theta}\in\Omega_{\boldsymbol{\theta}}}{\operatorname{argmax}} \sum_{j} \log \int f(\boldsymbol{x}_{\mathrm{o},j}, \boldsymbol{x}_{\mathrm{m},j} \mid \boldsymbol{\theta})\, d\boldsymbol{x}_{\mathrm{m},j} + \log f(\boldsymbol{\theta})$$

where $f(\boldsymbol{\theta})$ encodes the prior information. In this case, one may notice that the E-step remains unchanged as the expectation is taken only with respect to the latent variables $\boldsymbol{x}_{\mathrm{m},j}$. The M-step becomes

$$\boldsymbol{\theta}_{\mathrm{MAP}}^{(k+1)} = \underset{\boldsymbol{\theta}\in\Omega_{\boldsymbol{\theta}}}{\operatorname{argmax}}\; Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)}) + \log f(\boldsymbol{\theta}) \tag{17}$$

which corresponds to problem (16), where $d(\boldsymbol{\theta}, \boldsymbol{X}_{\mathrm{o}})$ reduces to the surrogate $Q(\cdot)$, and the regularization $R(\boldsymbol{\theta})$ represents the prior $f(\boldsymbol{\theta})$ with $\lambda = 1$.

Due to the presence of prior information, the M-step might be difficult to compute. In standard EM, this step involves a full joint maximization over all parameters, which can be computationally expensive or intractable for complex models. The Expectation Conditional Maximization (ECM) algorithm alleviates this issue by decomposing the M-step into a series of simpler conditional maximizations over parameter blocks. The ECM Either (ECME) algorithm builds upon this idea by replacing some conditional updates with direct maximization of the observed-data log-likelihood, often accelerating convergence. Lastly, the Generalized EM algorithm relaxes the requirement of full maximization, requiring only an increase in the surrogate function at each iteration [20]. This flexibility allows for the use of partial updates or gradient-based methods when exact optimization is not feasible. Such variants are particularly valuable in high-dimensional, constrained, or nonconvex settings where classical EM becomes impractical [3].

Structural prior

Many estimation algorithms can benefit from available structural prior knowledge pertaining to the unknown statistical parameter. This knowledge may stem from physical properties of the observed phenomenon under study or from dependencies among variables, which is often the case with large-scale high-dimensional data. In the missing-data case, $d(\boldsymbol{\theta}, \boldsymbol{X}_{\mathrm{o}})$ can be replaced by the surrogate function $Q(\cdot)$ or by fitting a parametric model. Here, problem (16) translates to a constrained optimization formulation

$$\begin{aligned} \hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\Omega_{\boldsymbol{\theta}}}{\operatorname{argmin}}\;\; & d(\boldsymbol{\theta}, \boldsymbol{X}_{\mathrm{o}}) + \lambda R(\boldsymbol{\theta}) \\ \text{s.t.}\quad\; & \Omega_{\boldsymbol{\theta}} = \{\boldsymbol{\theta} = \mathcal{F}(\boldsymbol{\phi}), \boldsymbol{\phi} \in \mathbb{R}^q\} \end{aligned} \tag{18}$$

where $\mathcal{F}(\cdot)$ is a known one-to-one mapping from $\boldsymbol{\phi}$ to $\boldsymbol{\theta}$.[4] To illustrate this framework, we consider two popular applications in SP: **PPCA** and **ST**.

Case Study I—PPCA: Let us first consider the PPCA formulation. In this setting, problem (18) becomes a constrained covariance matrix estimation problem, where the covariance matrix is assumed to be the sum of a low-rank component plus a scaled identity matrix—matching the well-known factor model [23]. Here, the parameter $\boldsymbol{\theta}$ reduces to the covariance matrix $\boldsymbol{\Sigma}$ and $\boldsymbol{\phi} = \{\sigma^2, \boldsymbol{R}\}$ such that

$$\Omega_{\boldsymbol{\theta}} = \{\boldsymbol{\Sigma} = \mathcal{F}(\boldsymbol{\phi}) \overset{\text{def}}{=} \sigma^2 \boldsymbol{I}_p + \boldsymbol{R},\; \boldsymbol{R} \succeq \boldsymbol{0},\; \operatorname{rank}(\boldsymbol{R}) \le r,\; \sigma^2 > 0\} \tag{19}$$

where $\boldsymbol{I}_p$ denotes the p-dimensional identity matrix, σ is a strictly positive scalar, and $\boldsymbol{R}$ is factored as $\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^{\top}$, with $\boldsymbol{U}$ being a $p \times r$ matrix and $\boldsymbol{S}$ being a $r \times r$ diagonal matrix. The solution to (18) under the structural constraint (19) can be obtained via the EM algorithm. At step $k+1$, it is given by

$$\hat{\boldsymbol{\Sigma}}^{(k+1)} = \hat{\sigma}^{2(k)} \boldsymbol{I}_p + \boldsymbol{U}^{(k)} \boldsymbol{\Lambda}^{(k)} \boldsymbol{U}^{\top(k)}$$

$$\text{with} \begin{cases} \hat{\sigma}^{2(k)} = \dfrac{1}{p-r} \displaystyle\sum_{i=r+1}^{p} \lambda_i^{(k)} \\ \hat{\boldsymbol{\Lambda}}^{(k)} = \operatorname{diag}(\lambda_1^{(k)} - \hat{\sigma}^{2(k)}, \cdots, \lambda_r^{(k)} - \hat{\sigma}^{2(k)}) \end{cases}$$

where $\hat{\boldsymbol{\Sigma}}^{(k)} \overset{\text{EVD}}{=} \sum_{i=1}^{p} \lambda_i^{(k)} \boldsymbol{u}_i^{(k)} \boldsymbol{u}_i^{\top(k)}$ $\lambda_1^{(k)} > \cdots > \lambda_p^{(k)}$ are the sorted eigenvalues of $\hat{\boldsymbol{\Sigma}}^{(k)}$, and $\boldsymbol{u}_1^{(k)}, \ldots, \boldsymbol{u}_p^{(k)}$ are the corresponding eigenvectors contained in $\boldsymbol{U}^{(k)}$. Note that $\boldsymbol{\Sigma}^{(0)}$ can be initialized using the sample covariance matrix computed from fully observed samples $\boldsymbol{x}_j$ in the data matrix $\boldsymbol{X}$.

Structure (19), together with the constrained optimization problem (18), arises in several SP applications. For example, in **radar processing**, the observations matrix $\boldsymbol{X}$ is often incomplete due to transmission-reception or sensor failures. In this

[4] In an alternative formulation, the unknown parameter $\boldsymbol{\theta}$ can be reparameterized such that $\hat{\boldsymbol{\theta}} = \mathcal{F}(\hat{\boldsymbol{\phi}})$, subject to $\hat{\boldsymbol{\phi}} = \operatorname{argmin}_{\boldsymbol{\phi}\in\mathbb{R}^q} d(\mathcal{F}(\boldsymbol{\phi}), \boldsymbol{X}_{\mathrm{o}}) + \lambda R(\mathcal{F}(\boldsymbol{\phi}))$.

case, $\boldsymbol{S}$ represents the diagonal source covariance matrix to be estimated, $\boldsymbol{U}$ the array manifold matrix, and σ^2 the thermal power of the noise. Model (19) also supports adaptive beamforming with monotone missing values in $\boldsymbol{X}$, where the covariance matrix must be estimated from incoming data to construct an adaptive beamformer (see [24] for alternative structured covariance models in radar processing applications). In **image analysis**, structure (19) has been used in [25] for covariance-based clustering of hyperspectral images under a scaled-Gaussian distribution. In this setting, the data consist of n pixels observed across p bands and may be incomplete due to dead or saturated pixels and band dropout, resulting in general missingness patterns.

Case Study II—ST: We now extend PPCA in the *streaming* scenario through robust ST (RST), which adaptively estimates a time-varying signal subspace [26]. RST concerns a wide range of applications involving high-dimensional data and dynamic systems, such as adaptive filtering, direction-of-arrival estimation, and blind source separation, among others. In the past 15 years, many techniques have flourished to adapt RST to missing data. Specifically, in scenarios where the data exhibit redundancy—so that high-dimensional observations lie near a low-dimensional subspace—incomplete data become particularly valuable as they may still contain sufficient information for recovery [27], [28]. In this context, the unknown signal at time t, $\boldsymbol{x}_t \in \mathbb{R}^p$, is assumed to belong to an *unknown* low-dimensional subspace $\boldsymbol{U} \in \mathbb{R}^{p\times r}$ ($r \ll p$), i.e., $\boldsymbol{x}_t = \boldsymbol{U}\boldsymbol{w}_t$, where $\boldsymbol{w}_t$ is an *unknown* weight vector. Many of the ST algorithms seek to optimize a recursively time-variant updated loss function $F_t(\cdot)$, e.g., the squared projection loss onto the subspace

$$F_t(\boldsymbol{U}) = \alpha_t F_{t-1}(\boldsymbol{U}) + \beta_t \left\| \boldsymbol{m}_t \odot (\boldsymbol{y}_t - \boldsymbol{U}\boldsymbol{w}_t) \right\|_2^2. \tag{20}$$

Here, $\boldsymbol{y}_t = \boldsymbol{m}_t \odot \boldsymbol{x}_t$ is the observed vector, $\boldsymbol{m}_t$ is the associated missing-data pattern, α_t and β_t control the tradeoff between convergence rate and tracking adaptability to subspace changes, and $\boldsymbol{U}$ is typically updated using the previous estimate. Estimating $\boldsymbol{U}$ amounts to solving

$$\boldsymbol{U}_t = \operatorname*{argmin}_{\boldsymbol{U}} F_t(\boldsymbol{U}) + \rho g(\boldsymbol{U}) \tag{21}$$

which is a specific case of (16), where $g(\boldsymbol{U})$ is an optional regularization function on $\boldsymbol{U}$ (e.g., a sparsity-promoting function in the presence of outliers [26]), and ρ controls the regularization level. One way to solve (20) and (21) is parallel estimation and tracking by recursive least squares (PETRELS). This approach is a second-order recursive least-squares algorithm [29] that employs second-order stochastic gradient descent (SGD) on the cost function. At each time t, the weight vector w_t and subspace $\boldsymbol{U}_t$ are alternately estimated by minimizing the projection residual on the previous subspace estimate $\boldsymbol{U}_{t-1}$, where $\boldsymbol{U}_0$ is a random subspace initialization. A probabilistic formulation of problem (21) has also been proposed in [30], which requires some assumptions on the data distribution. In this setting, $\boldsymbol{U}_t$ and $\boldsymbol{w}_t$ are estimated by maximizing the observed log-likelihood using a stochastic alternating EM approach. See "Application 3: Graph Learning for Spatiotemporal Signals with Missing Values" for more details about graph learning.

Takeaway

In the context of parameter estimation, likelihood-based frameworks, including EM and its stochastic variants (SEM, MCEM, and SAEM), are well suited for structured problems such as covariance matrix estimation, ST, or graph learning. These methods provide rigorous uncertainty quantification and explicitly account for the missingness mechanism but may encounter scalability challenges in high-dimensional regimes. Beyond these scalability limitations, utilizing the EM algorithm requires specific expertise in statistical modeling, and the need to tailor each EM variant to the specific application limits the feasibility of having a general unified software package. To overcome these issues, researchers are increasingly resorting to deep learning-based models for handling missing data, offering greater scalability and a more unified treatment across application domains.

Learning with missing data

This section highlights modern learning strategies for handling missing data, a domain still underexplored in SP applications. We encourage SP practitioners to adopt these advanced tools, either by tailoring them to their applications or by directly incorporating them into their methodologies.

A central question in learning with incomplete data concerns the intended downstream application, which fundamentally shapes the choice of modeling strategy. In this section, we classify these applications into two broad categories: imputation, often coupled with an underlying inference problem, and prediction. These tasks give rise to two major modeling paradigms: *generative* and *discriminative* approaches. We first discuss the imputation task, which typically relies on generative models that aim to learn the full data distribution $f(\boldsymbol{X})$. We then turn to prediction, which is more closely aligned with discriminative methods targeting, in some cases, the conditional distribution $f(\boldsymbol{y} \mid \boldsymbol{X}_o)$, where $\boldsymbol{y}$ is the target variable,[5] thereby avoiding the need to explicitly model $f(\boldsymbol{X})$.

Generative approaches for imputation

This section introduces the main families of deep learning-based imputation methods. We first present variational approaches, including variational autoencoder (VAE) and importance-weighted autoencoder (IWAE)-based models, which treat missingness through latent-variable modeling and reconstruct unobserved values by optimizing lower bounds on the incomplete-data likelihood. We then describe adversarial methods, such as generative adversarial networks (GANs), which formulate imputation as an adversarial game to produce realistic replacements for missing entries. Finally, we discuss optimal transport (OT)-based approaches, which view imputation as learning a mapping from incomplete to complete samples by minimizing a transport cost.

[5]Note that in this section, $\boldsymbol{y}$ does not represent a column of $\boldsymbol{Y} = \boldsymbol{M} \odot \boldsymbol{X}$, as introduced in the "Basic Concepts of Missing Data" section.

Application 3: Graph Learning for Spatiotemporal Signals with Missing Values

We earlier discussed graph signals and their numerous applications. The primary step in graph SP is to identify a graphical model that best captures the similarities or dependencies in the signal. This process is referred to as *graph learning*, for which there are various approaches, broadly classified into *undirected* and *directed* methods. An undirected graph models bilateral relationships, while a directed graph is used to represent unilateral dependencies. Let $\boldsymbol{X}$ denote the graph signal matrix, and let $\boldsymbol{\theta}$ be the parameters of the underlying graph model [S1], considered as random variables with prior $f(\boldsymbol{\theta})$. The MAP formulation for estimating $\boldsymbol{\theta}$ given the data yields the following:

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \underset{\in \Omega_{\boldsymbol{\theta}}}{\operatorname{argmax}} \log f(\boldsymbol{\theta} \mid \boldsymbol{X}) = \underset{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}}{\operatorname{argmin}} -\log f(\boldsymbol{X} \mid \boldsymbol{\theta}) - \log f(\boldsymbol{\theta}). \tag{S3}$$

This formulation is specified in Table S1 for two standard graph learning frameworks: the Gaussian Markov random field (GMRF) model [S2] for undirected graphs and the vector AR (VAR) model [S3] for directed graphs.

The graph learning framework based on (S3) assumes complete data statistics, while in practice, observations often contain missing values due to various factors, such as sensor failures or measurement errors. One may consider the missing signal samples as latent variables and use EM for the estimation of the unknown graph parameters. This approach, however, only infers the graph structure, requiring additional processing for the imputation of missing values.

An alternative approach is to simultaneously estimate the missing values and the graph parameters, enabling joint graph learning and signal imputation while reducing computational complexity. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{p \times n}$ represent the complete (ground-truth) data matrix. Consider the case where the observed data contain missing values and noise. Replacing the missing values with 0, the observation matrix can be modeled as $\boldsymbol{Y} = \boldsymbol{M} \odot (\boldsymbol{X} + \boldsymbol{N})$, where $\boldsymbol{M} \in \{0,1\}^{p \times n}$ is the missing-data pattern, and $\boldsymbol{N} \in \mathbb{R}^{p \times n}$ represents the measurement noise. Assuming that the missing-data mechanism is MCAR, the MAP estimation of the graph parameters $\boldsymbol{\theta}$ (see Table S1 for their definition) and the complete data matrix $\boldsymbol{X}$ given $\boldsymbol{Y}$ and $\boldsymbol{M}$ is formulated as

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\text{MAP}}, \hat{\boldsymbol{X}}_{\text{MAP}} &= \underset{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}, \boldsymbol{X}}{\operatorname{argmax}} \log f(\boldsymbol{\theta}, \boldsymbol{X} \mid \boldsymbol{M}, \boldsymbol{Y}), \\ &= \underset{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}, \boldsymbol{X}}{\operatorname{argmin}} -\log f(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{M}) - \log f(\boldsymbol{X} \mid \boldsymbol{\theta}) - \log f(\boldsymbol{\theta}). \end{aligned} \tag{S4}$$

Classical methods for graph learning, however, only capture either spatial correlations through an undirected graph structure (GMRF model) or temporal dependencies via a directed graph topology (VAR model). For spatiotemporal signals (i.e., networked time series data), both spatial and temporal dependencies are crucial. Recent approaches like [S4] have therefore introduced a multirelational graph model comprising both undirected and directed structures to simultaneously represent spatial and temporal dependencies. This model is described by the following:

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{\epsilon}_t, \qquad f(\boldsymbol{\epsilon}_t \mid \boldsymbol{L}) \propto \sqrt{\det{}^*\boldsymbol{L}} \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}_t^\top \boldsymbol{L} \boldsymbol{\epsilon}_t\right). \tag{S5}$$

Here, $\boldsymbol{x}_t$ and $\boldsymbol{\epsilon}_t$ are the signal and the innovations at time t, respectively; $\boldsymbol{A}$ represents the adjacency matrix of a directed graph capturing temporal dependencies, and $\boldsymbol{L}$ denotes the Laplacian matrix of an undirected graph encoding spatial correlations. The parameters of this model are $\boldsymbol{\theta} = (\boldsymbol{A}, \boldsymbol{L})$.

The spatiotemporal signal recovery and graph learning (STSRGL) method in [S4] (code available at https://github.com/javaheriamirhossein/STSRGL) solves problem (S4) for joint inference of this spatiotemporal model.

In Figure S6, we compare the STSRGL method with baseline algorithms for

- *spatial graph learning:* the combinatorial graph learning (CGL) [S2] method
- *temporal graph inference:* the joint inference of signals and graphs (JISG) [S3] method.

The evaluation uses spatiotemporal data defined over a 10×10 grid graph $(p = 100)$, featuring 50% missing

Table S1. Two standard models for undirected and directed graph learning.

Graph type	Model	Parameters $\boldsymbol{\theta}$	$-\log f(\boldsymbol{X} \mid \boldsymbol{\theta}) - \log f(\boldsymbol{\theta})$ (up to scale)
Undirected	GMRF $f(\boldsymbol{x}_j \mid \boldsymbol{L}) \propto \sqrt{\det{}^*\boldsymbol{L}} \exp\left(-\frac{\boldsymbol{x}_j^\top \boldsymbol{L} \boldsymbol{x}_j}{2}\right)$	$\boldsymbol{L}$ (Laplacian matrix) $\boldsymbol{L} \in \mathcal{S}_p^+, L_{ij \neq i} \leq 0, \boldsymbol{L}\mathbf{1} = \mathbf{0}$	$\operatorname{tr}\left(\boldsymbol{L} \sum_{j=1}^{n} \frac{\boldsymbol{x}_j \boldsymbol{x}_j^\top}{n}\right) - \log\det{}^*\boldsymbol{L} + \alpha \lVert \boldsymbol{L} \rVert_{1,\text{off}}$
Directed	Vector auto-regressive $\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \boldsymbol{I})$	$\boldsymbol{A}$ (adjacency matrix) $\boldsymbol{A} \in \mathbb{R}^{p \times p}$	$\sum_{t=2}^{n} \lVert \boldsymbol{x}_t - \boldsymbol{A}\boldsymbol{x}_{t-1} \rVert_F^2 + \alpha \lVert \boldsymbol{A} \rVert_{1,1}$

(*Continued*)

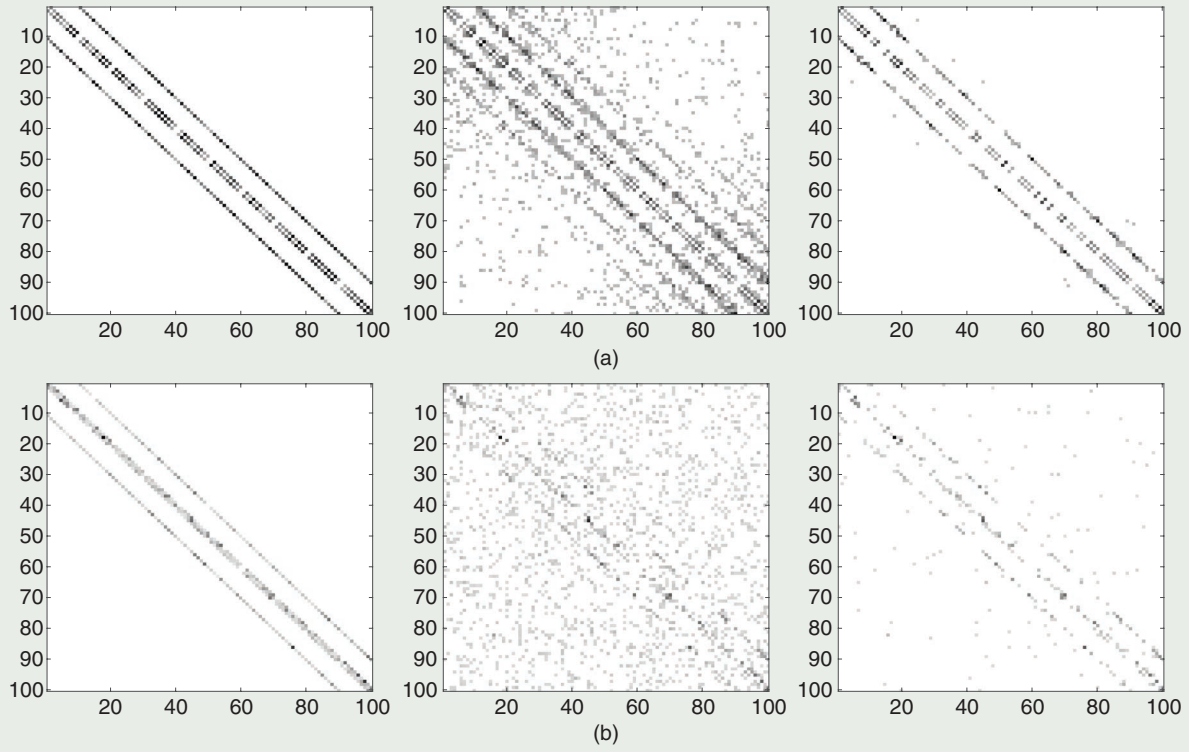

FIGURE S6. A visualization of (a) $|L_{i,j}|\,(i \neq j)$ and (b) $|A_{i,j}|$ learned from data with noisy and missing observations. (a) The ground-truth (left), the CGL (middle), and the STSRGL (right) spatial graphs. (b) The ground-truth (left), the JISG (middle), and the STSRGL (right) temporal graphs.

values and additive Gaussian noise ($\sigma_n^2 = 0.01$). As shown in the figures, the STSRGL more accurately identifies both the nonzero support (connections) and the entries (edge weights) of the graph matrices compared to methods that learn only spatial or temporal graphs.

References

[S1] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019, doi: 10.1109/MSP.2018.2887284.

[S2] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, Sep. 2017, doi: 10.1109/JSTSP.2017.2726975.

[S3] V. N. Ioannidis, Y. Shen, and G. B. Giannakis, "Semi-blind inference of topologies and dynamical processes over dynamic graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2263–2274, May 2019, doi: 10.1109/TSP.2019.2903025.

[S4] A. Javaheri, A. Amini, F. Marvasti, and D. P. Palomar, "Learning spatio-temporal graphical models from incomplete observations," *IEEE Trans. Signal Process.*, vol. 72, pp. 1361–1374, 2024, doi: 10.1109/TSP.2024.3354572.

Although grounded in different principles (variational inference, adversarial learning, or OT), all these techniques can be unified under a common generative perspective; their aim is to construct an imputation operator parameterized by a learnable deep neural network that models the conditional distribution of missing values given the observed data. Formally, this can be expressed as

$$\begin{aligned} \hat{\boldsymbol{X}}_{\mathrm{m}} &= I_{\hat{\zeta}}(\boldsymbol{X}_{\mathrm{o}}, \boldsymbol{M}) \\ \hat{\zeta} &= \operatorname*{argmin}_{\zeta} \mathbb{E}\left[\mathcal{D}_{\zeta}(\boldsymbol{X}_{\mathrm{o}}, \boldsymbol{M})\right] \end{aligned} \tag{22}$$

where $\hat{\boldsymbol{X}}_{\mathrm{m}}$ are the imputed values, $I_{\zeta}(\cdot)$ is the imputation algorithm (i.e., the estimator) indexed by learnable parameters $\hat{\boldsymbol{\zeta}}$, and $\mathcal{D}_{\zeta}(\cdot)$ is a specific divergence or discrepancy measure.

VAEs for imputation

Several methods employ deep latent variable models to estimate the distribution $f(\boldsymbol{x}) = \int f(\boldsymbol{x} \mid \boldsymbol{z}) f(\boldsymbol{z})\, d\boldsymbol{z}$ through a variational bound on the observed log-likelihood, which is typically intractable. Here, $\boldsymbol{z}$ is a latent variable following a prior distribution $f(\boldsymbol{z})$. In VAE-based schemes for missing data, a conditional variational distribution over $\boldsymbol{z}$ is introduced as a learnable proposal

distribution [35], [36], [37], [38], [39]. In this perspective, minimizing the discrepancy $\mathcal{D}_\zeta(\cdot)$ in (22) reduces to minimizing the Kullback-Leibler (KL) divergence between the true posterior distribution and the variational distribution. Specifically, by denoting the variational learnable distribution by $\tilde{f}(\cdot)$, the missing VAE bound (MVAE), also called *evidence lower bound*, under MAR and MCAR scenarios, reads

$$\begin{aligned}\mathcal{L}_{\text{MVAE}}(\boldsymbol{\theta},\boldsymbol{\gamma}) &= \sum_{j=1}^{n} \mathbb{E}_{z_j\sim\tilde{f}(z|\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})}\left[\log\frac{f(\boldsymbol{x}_{\text{o},j}\,|\,z_j;\boldsymbol{\theta})f(z_j)}{\tilde{f}(z_j\,|\,\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})}\right] \\ &= \ell(\boldsymbol{X}_{\text{o}};\boldsymbol{\theta}) - \text{KL}\left(\prod_{j=1}^{n}\tilde{f}(z_j|\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})\,\middle\|\,\prod_{j=1}^{n} f(z_j|\boldsymbol{x}_{\text{o},j};\boldsymbol{\theta})\right)\end{aligned} \tag{23}$$

where $\ell(\boldsymbol{X}_{\text{o}};\boldsymbol{\theta}) = \Sigma_{j=1}^{n}\log f(\boldsymbol{x}_{\text{o},j};\boldsymbol{\theta})$ denotes the observed log-likelihood. $\mathcal{L}_{\text{MVAE}}(\boldsymbol{\theta},\boldsymbol{\gamma})$ is obtained from a generative model that defines a stochastic mapping, parameterized by $\boldsymbol{\theta}$, from a latent variable $\boldsymbol{z}_j$ to each column $\boldsymbol{x}_j$ of the data matrix $\boldsymbol{X}$, together with a variational distribution $\tilde{f}(\boldsymbol{z}_j\,|\,\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})$ targeting the intractable $f(\boldsymbol{z}_j\,|\,\boldsymbol{x}_{\text{o},j};\boldsymbol{\theta})$

$$\boldsymbol{z}_j \sim \tilde{f}(\boldsymbol{z}_j\,|\,\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma}) \quad \text{and} \quad \boldsymbol{x}_j \sim f(\boldsymbol{x}_j\,|\,\boldsymbol{z}_j;\boldsymbol{\theta}). \tag{24}$$

The parameters γ and $\boldsymbol{\theta}$ are estimated via deep neural networks, commonly referred to as the *encoder* and *decoder*, respectively. Once γ and $\boldsymbol{\theta}$ are learned, several imputation scenarios can be considered as a particular case of (22). As an example, using model (24), the operator $I_{\hat{\theta},\hat{\gamma}}(\boldsymbol{X}_{\text{o}},\boldsymbol{M})$ can be instantiated by first sampling $\boldsymbol{z}_j \sim \tilde{f}(\boldsymbol{z}_j\,|\,\boldsymbol{x}_{\text{o},j};\hat{\boldsymbol{\gamma}})$ and then generating imputations from $\boldsymbol{x}_j \sim f(\boldsymbol{x}_j\,|\,\boldsymbol{z}_j;\hat{\boldsymbol{\theta}})$.

Deep generative modeling by IWAE

The MVAE framework was extended in [36], where the authors introduce the missing IWAE (MIWAE) to obtain a tighter variational bound to impute MCAR or MAR values. Again, the generative model is described by the stochastic mapping

$$\begin{aligned}\boldsymbol{z}_j &\sim \tilde{f}(\boldsymbol{z}_j\,|\,\boldsymbol{x}_{\text{o},j};\gamma) = \mathcal{N}(\boldsymbol{\mu}_\gamma(\boldsymbol{x}_{\text{o},j}),\boldsymbol{\Sigma}_\gamma(\boldsymbol{x}_{\text{o},j})) \\ \text{and}\ \ \boldsymbol{x}_j &\sim f(\boldsymbol{x}_j\,|\,\boldsymbol{z}_j;\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_j),\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{z}_j)).\end{aligned} \tag{25}$$

In practice, both the conditional and variational distributions are typically modeled as Gaussian (here restricted to continuous data for simplicity), with parameters γ and $\boldsymbol{\theta}$ learned via deep neural networks, as in MVAE. However, unlike MVAE, MIWAE maximizes the tighter importance-weighted bound

$$\begin{aligned}&\mathcal{L}^{K}_{\text{MIWAE}}(\boldsymbol{\theta},\boldsymbol{\gamma};\boldsymbol{x}_{\text{o},j}) = \\ &\sum_{j=1}^{n}\mathbb{E}_{z_j^{(1)},\ldots,z_j^{(K)}\sim\tilde{f}(z_j|\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})}\left[\log\frac{1}{K}\sum_{c=1}^{K}\frac{f(\boldsymbol{x}_{\text{o},j}\,|\,z_j^{(c)};\boldsymbol{\theta})f(z_j^{(c)})}{\tilde{f}(z_j^{(c)}\,|\,\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})}\right]\end{aligned} \tag{26}$$

where $K \in \mathbb{N}^*$ is the number of draws. This bound satisfies $\mathcal{L}^{K}_{\text{MIWAE}}(\boldsymbol{\theta},\boldsymbol{\gamma}) \le \ell(\boldsymbol{X}_{\text{o}};\boldsymbol{\theta})$. Note that $f(\boldsymbol{x}_{\text{o},j}\,|\,z_j^{(c)};\boldsymbol{\theta})$ is obtained by marginalizing the conditional joint distribution $f(\boldsymbol{x}_j\,|\,\boldsymbol{z}_j;\boldsymbol{\theta})$ in (25). Notably, MIWAE reduces to MVAE when $K = 1$, i.e., $\mathcal{L}^{1}_{\text{MIWAE}} = \mathcal{L}_{\text{MVAE}}$. The lower bound is then optimized using stochastic gradient-based algorithms. The imputation step can be performed either by sampling from the conditional distribution of the missing variables given the observed ones or, for a single imputation, by computing the conditional expectation

$$\begin{aligned}&\mathbb{E}[\boldsymbol{x}_{\text{m},j}\,|\,\boldsymbol{x}_{\text{o},j};\boldsymbol{\theta}] = \\ &\int\int \boldsymbol{x}_{\text{m},j}\frac{f(\boldsymbol{z}_j|\boldsymbol{x}_{\text{o},j};\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{z}_j|\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})}f(\boldsymbol{x}_{\text{m},j}|\boldsymbol{x}_{\text{o},j},\boldsymbol{z}_j;\boldsymbol{\theta})\,\tilde{f}(\boldsymbol{z}_j|\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})\,d\boldsymbol{z}_j d\boldsymbol{x}_{\text{m},j}.\end{aligned} \tag{27}$$

Since the encoder $\tilde{f}(\boldsymbol{z}_j|\boldsymbol{x}_{\text{o},j};\gamma)$ is a variational distribution, $\mathbb{E}[\boldsymbol{x}_{\text{m},j}|\boldsymbol{x}_{\text{o},j};\boldsymbol{\theta}]$ can be approximated using a Monte Carlo technique called *self-normalized importance sampling* [40, Section 9.2]

$$\hat{\boldsymbol{x}}_{\text{m},j} = \sum_{c=1}^{K} w_c \boldsymbol{x}_{\text{m},j}^{(c)}, \quad \text{where} \tag{28}$$

$$\begin{aligned}w_c &= \frac{r_c}{\sum_{c=1}^{K} r_c}, \\ r_c &= \frac{f(\boldsymbol{x}_{\text{o},j}\,|\,z_j^{(c)};\boldsymbol{\theta})f(z_j^{(c)})}{\tilde{f}(z_j^{(c)}\,|\,\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma})}, \\ (\boldsymbol{x}_{\text{m},j}^{(c)},z_j^{(c)}) &\overset{\text{iid}}{\sim} f(\boldsymbol{x}_{\text{m},j}|\boldsymbol{x}_{\text{o},j},z_j;\boldsymbol{\theta})\tilde{f}(z_j|\boldsymbol{x}_{\text{o},j};\boldsymbol{\gamma}).\end{aligned} \tag{29}$$

The limitation of MIWAE is that it is adapted only for the MAR and MCAR scenarios and fails when the mechanism is nonignorable. Recent extensions, called *not-MIWAE* and *AV-LR*, adapt the MIWAE framework to MNAR data [37], [39]. Since the missing-data mechanism is nonignorable, we cannot rely solely on the observed-data likelihood $f(\boldsymbol{x}_{\text{o},j};\boldsymbol{\theta})$. Concretely, the main change occurs in the variational lower bound (26); the joint distribution (the numerator of the terms in the sum) now involves the missing-data mechanism $f(\boldsymbol{x}_{\text{o},j},\boldsymbol{m}_j,\boldsymbol{z}_j;\boldsymbol{\theta},\boldsymbol{\phi}) = f(\boldsymbol{m}_j|\boldsymbol{x}_{\text{o},j},\boldsymbol{x}_{\text{m},j};\boldsymbol{\phi})f(\boldsymbol{x}_{\text{o},j}|\boldsymbol{z}_j;\boldsymbol{\theta})f(\boldsymbol{z}_j)$. While MVAE/MIWAE requires sampling only the latent variable $\boldsymbol{z}$, not-MIWAE requires sampling both $\boldsymbol{z}$ and the missing entries $\boldsymbol{x}_{\text{m}}$ using the same distribution as in (29).

GAN for Imputation

While MIWAE tackles imputation by explicitly approximating the data distribution via the evidence lower bound, GANs offer an alternative paradigm based on implicit modeling [41]. Let us describe GAN for Imputation (GAIN), an architecture dedicated to imputation [42].

Without missing data, a generator (G) tries to create realistic data from random noise to fool a discriminator (D), while D tries to distinguish between real data and fake data. Both G and D are fully connected neural nets. In the missing-data setting, only an incomplete dataset is available. Therefore, GAIN is defined such that G observes the real components of the data and attempts to impute the missing components. On the other hand, D attempts to determine which components of a completed vector were actually observed and which were imputed by G. Specifically,

G takes the observed part of the data $\boldsymbol{X}_{\text{o}}$ and the missingness mechanism $\boldsymbol{M}$ as input. Its output reads

$$\hat{\boldsymbol{x}}_j = (\mathbf{1}_n - \boldsymbol{m}_j) \odot G(\boldsymbol{x}_{\text{o},j}, \boldsymbol{m}_j) + \boldsymbol{m}_j \odot \boldsymbol{x}_j \tag{30}$$

where $\mathbf{1}_n$ is a vector of n components equal to one.

The final imputed vector, $\hat{\boldsymbol{x}}_j$, is constructed by keeping the originally observed values and replacing the missing ones with the generator's output. The discriminator D then receives the completed vector $\hat{\boldsymbol{x}}_j$. The main difference between the original GAN and GAIN is that the output is a single scalar (indicating real or fake status in GAN), whereas the GAIN discriminator outputs a probability vector of the same dimension as the data (i.e., $[D(\hat{\boldsymbol{x}}_j)]_i$ represents the probability that the ith element was observed rather than imputed). In addition to the generator and discriminator, the authors of [42] introduced a hint variable, $\boldsymbol{H}$. Its primary role is to regulate the amount of knowledge available to D regarding the missingness pattern $\boldsymbol{M}$. The authors theoretically establish that this control is critical; without providing D with sufficient hint information, or if $\boldsymbol{H}$ were omitted, the adversarial game would allow G to learn multiple distributions that are all optimal with respect to D. Specifically, the hint variable $\boldsymbol{H}$ is defined as a random matrix of dimension $p \times n$, as a function of $\boldsymbol{M}$, and taking values in a space $\mathcal{H}$. The authors set $\mathcal{H} = \{0, 0.5, 1\}$ and define

$$\forall i \in \{1, \ldots, p\} \text{ and } j \in \{1, \ldots, n\}, H_{i,j} = \begin{cases} M_{i,j} & \text{if } B_{i,j} = 1, \\ 0.5 & \text{otherwise.} \end{cases}$$

The matrix $\boldsymbol{B}$ is constructed such that $B_{i,j} = 1$ if $i \neq k$ and $B_{i,j} = 0$ otherwise, where the index k is sampled uniformly at random from $\{1, \ldots, p\}$. This implies that, for each sample, $\boldsymbol{H}$ reveals all but one of the components of $\boldsymbol{M}$ to D.

Training is formulated as a minimax game involving two loss functions, which can be directly expressed within the general framework (22). Specifically, both G and D are trained simultaneously through the following adversarial optimization problem:

$$\max_G \min_D - \sum_{i=1}^{p} \sum_{j=1}^{n} M_{i,j} \log \hat{M}_{i,j} + (1 - \hat{M}_{i,j}) \log (1 - \hat{M}_{i,j})$$

where $\hat{M}_{i,j} = D(\hat{X}_{i,j}, H_{i,j})$. The discriminator aims to accurately recover the missing-data pattern by minimizing the cross-entropy loss, while the generator maximizes this loss by producing imputations that are indistinguishable from observed values, challenging the discriminator. After training, the generator is applied to incomplete data with added noise to impute missing values.

OT

While VAEs focus on maximizing conditional likelihood and GANs on learning via discrimination, methods based on OT adopt a fundamentally different approach. They aim to find the most efficient transformation to align the distribution of the imputed data with the distribution of the observed data. OT is a mathematical framework used to calculate the most efficient way to transform one distribution of mass into another, and the Wasserstein distance represents the minimum amount of effort required for this move. If one wants to move from distribution μ to distribution ν, the squared Wasserstein distance reads [43]

$$W_2^2(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int \left\| \hat{\boldsymbol{X}}_L - \hat{\boldsymbol{X}}_K \right\|_F^2 d\pi(\hat{\boldsymbol{X}}_L, \hat{\boldsymbol{X}}_K) \tag{31}$$

where $\Pi(\mu, \nu)$ represents the set of all joint probability measures (transport plans) with marginals μ and ν, and $\hat{\boldsymbol{X}}_L$ and $\hat{\boldsymbol{X}}_K$ represent two batches extracted from the imputed matrix $\hat{\boldsymbol{X}}$. The set of indexes L and K is sampled randomly from $\{1, \ldots, n\}$, such that $L \cap K = \varnothing$ and $|L| = |K| < n$ (for example, if $L = \{1, 2, 3\}$, $\hat{\boldsymbol{X}}_L$ contains the three first samples of $\hat{\boldsymbol{X}}$, that is $\hat{\boldsymbol{X}}_L = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]$). The Wasserstein distance is, nevertheless, computationally expensive and not everywhere differentiable, limiting its use in gradient-based optimization. To address this, [43] introduce the entropically regularized OT cost

$$OT_\varepsilon(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int \left\| \hat{\boldsymbol{X}}_L - \hat{\boldsymbol{X}}_K \right\|_F^2 d\pi(\hat{\boldsymbol{X}}_L, \hat{\boldsymbol{X}}_K) - \varepsilon H(\pi)$$

where $H(\pi)$ is the entropy of the transport plan. Although this regularization ensures differentiability, it introduces bias. The Sinkhorn divergence

$$\mathcal{S}_\varepsilon(\mu, \nu) = OT_\varepsilon(\mu, \nu) - \frac{1}{2} OT_\varepsilon(\mu, \mu) - \frac{1}{2} OT_\varepsilon(\nu, \nu)$$

corrects this bias while retaining differentiability. Minimizing $\mathcal{S}_\varepsilon$ ensures convergence to the target distribution in the limit. Specifically, the algorithm initializes missing entries with the mean of observed values plus a small random perturbation to preserve marginals. At each iteration, batches $\hat{\boldsymbol{X}}_L$ and $\hat{\boldsymbol{X}}_K$ are sampled randomly from $\hat{\boldsymbol{X}}$, and their Sinkhorn divergence is minimized to obtain $\mathcal{S}_\varepsilon(\mu^*, \nu^*)$. Missing values are then updated via gradient descent by computing the gradient of the Sinkhorn divergence with respect to the imputed values, i.e., $\nabla_{\boldsymbol{X}_{\text{m}}} \mathcal{S}_\varepsilon(\mu^*, \nu^*)$. The imputation step is simply moving the missing values in the opposite direction of the gradient, i.e., $\hat{\boldsymbol{X}}_{L \cup K} \leftarrow \hat{\boldsymbol{X}}_{L \cup K} - \eta \nabla_{\boldsymbol{X}_{\text{m}}} \mathcal{S}_\varepsilon(\mu^*, \nu^*)$, where η is the learning rate. Essentially, this step moves the missing points in batch $\hat{\boldsymbol{X}}_L$ toward the distribution of batch $\hat{\boldsymbol{X}}_K$ and vice versa. The underlying assumption is that arbitrary subsets of the data share the same distribution, thereby excluding MNAR settings. This procedure is repeated until convergence.

Generative approaches—Takeaway

Deep learning has recently emerged as a powerful framework for addressing complex missing-data problems. Unlike traditional methods, deep models capture nonlinear dependencies and complex structures, making them particularly effective for high-dimensional or heterogeneous datasets. MVAE provides a principled generative approach, yielding coherent joint imputations, but it can be computationally intensive and sensitive to model design. MIWAE enhances VAE-based imputations through importance weighting, improving likelihood estimates and producing sharper reconstructions, while its extension, not-MIWAE, efficiently handles MNAR scenarios. In contrast, GAIN offers flexible, realistic imputations with minimal

assumptions, though its architecture requires careful tuning and is generally applicable only under MCAR. OT-based strategies provide geometric interpretability without strong parametric assumptions, but they can be computationally demanding and, like GAIN, are not suited for MNAR data. Overall, selecting among these approaches should consider data dimensionality, the assumed missingness mechanism, and available computational resources.

Label prediction with missing data

When the objective is to predict a target value y_{new} (e.g., a label) from a vector of new observations $\boldsymbol{x}_{\text{new}}$, several scenarios must be distinguished, depending on whether missing values occur in the training set $\boldsymbol{X}_{\text{train}}$, in the new observations $\boldsymbol{x}_{\text{new}}$, or in both. The target values in the training set, denoted as $\boldsymbol{y}_{\text{train}}$, are assumed to be fully observed.[6] The appropriate methodology depends on the specific configuration.

- *Only* $\boldsymbol{X}_{\text{train}}$ *contains missing values*: Missingness may occur exclusively in the training set. For instance, consider historical records from a sensor network measuring seismic variables. Some entries in the training set $\boldsymbol{X}_{\text{train}}$ may be missing due to sensor malfunction, temporary outages, or data corruption. By contrast, the new observation $\boldsymbol{x}_{\text{new}}$ is complete owing to more recent and reliable monitoring systems. In this setting, the objective is to learn the conditional relationship $f(\boldsymbol{y} \mid \boldsymbol{X})$.
- *Both* $\boldsymbol{X}_{\text{train}}$ *and* $\boldsymbol{x}_{\text{new}}$ *contain missing values*: When missing values occur in both the training set and the new observations, the challenge is to make predictions in the presence of incomplete data at both stages. In this context, the objective shifts to estimating the conditional distribution $f(\boldsymbol{y} \mid \boldsymbol{X}_{\text{o}})$, and it is not necessary to model the full distribution $f(\boldsymbol{X})$.

We now examine these two scenarios in detail, outlining the methodological approaches available for handling missing values in each setting.

Predict when only $\boldsymbol{X}_{\text{train}}$ is incomplete

When missing values occur only in $\boldsymbol{X}_{\text{train}}$, two main strategies can be considered. The first consists of applying an imputation procedure, $I_{\text{comp}}(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}})$, to obtain a fully imputed training dataset. This can be, for example, one of the model-based approaches discussed in the "Imputation" section. A standard prediction algorithm is then fitted to this completed dataset. Importantly, the imputation model should incorporate information from the target variable $\boldsymbol{y}_{\text{train}}$. Including the target variable in the imputation step helps preserve the dependence structure between $\boldsymbol{X}_{\text{train}}$ and $\boldsymbol{y}_{\text{train}}$ and can capture any relationship between the missing-data mechanism and the target variable [7].

The second strategy consists of directly estimating a parameter vector $\boldsymbol{\beta} \in \mathbb{R}^p$ in a model relating $\boldsymbol{y}$ and $\boldsymbol{X}$ despite the presence of missing values. For example, in the linear setting, the model can be written as $\boldsymbol{y} = \boldsymbol{X}^\top \boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \in \mathbb{R}^p$ denotes a noise term. Once $\boldsymbol{\beta}$ has been estimated from the training set, it can be used to predict $y_{\text{new}} = \boldsymbol{x}_{\text{new}}^\top \boldsymbol{\beta}$. In this framework, parameter estimation in the presence of missing data can be carried out using learning algorithms specifically adapted to incomplete data, such as EM-algorithms (see the "Estimation With Missing Data Under a Parametric Model Framework" section). Alternatively, one may adopt a simpler approach based on naive imputation, followed by bias correction. More specifically, if there is one target value prediction algorithm designed for complete data, called algorithm A, the steps to account for missing data are the following.

1) Apply a simple imputation method $I_{\text{simp}}(\cdot)$, such as constant imputation, to construct a completed training set $I_{\text{simp}}(\boldsymbol{X}_{\text{train}})$.
2) Algorithm A is subsequently modified to account for the bias induced by the imputation step. This debiased version of A is then fitted to the imputed dataset $I_{\text{simp}}(\boldsymbol{X}_{\text{train}})$.

This strategy has primarily been investigated in the context of linear regression with missing covariates [47], [48], that is, under the assumption that $\mathbb{E}[\boldsymbol{y} \mid \boldsymbol{X}] = h(\boldsymbol{X})$, where $h(\cdot)$ is a linear function. A representative example of this strategy is provided in [48], which considers ridge regression and adapts the SGD algorithm to accommodate MCAR values in $\boldsymbol{X}_{\text{train}}$. The objective is to estimate the regression parameter $\boldsymbol{\beta}$ in the linear model $y_j = \boldsymbol{x}_j^\top \boldsymbol{\beta} + \epsilon_j$, where $\epsilon_j \in \mathbb{R}$ is a noise term, by minimizing the risk $R(\boldsymbol{\beta}) = \mathbb{E}[(\boldsymbol{x}_j^\top \boldsymbol{\beta} - y_j)^2/2]$. Without missing values, the SGD update at iteration k is given by $\beta^{(k)} = \beta^{(k-1)} - \eta\, g_k(\beta^{(k-1)})$, where η is the learning rate, and $g_k(\beta^{(k-1)})$ is an unbiased gradient estimate of the risk, satisfying $\mathbb{E}[g_k(\beta^{(k-1)})] = \nabla R(\beta^{(k-1)})$. Specifically

$$g_k(\beta^{(k-1)}) = \boldsymbol{x}_k(\boldsymbol{x}_k^\top \beta^{(k-1)} - y_k) \tag{32}$$

where $\boldsymbol{x}_k$ and y_k denote the kth sample (i.e., the kth column of $\boldsymbol{X}$) and its corresponding target value. When covariates contain missing values, a simple approach consists of imputing them with zero, yielding the completed data matrix $\boldsymbol{M} \odot \boldsymbol{X}$. However, this naive imputation introduces bias. To correct for it, a debiased gradient estimator is constructed. The resulting update direction at iteration k becomes

$$\tilde{g}_k(\beta^{(k-1)}) = \frac{1}{p}(\boldsymbol{m}_k \odot \boldsymbol{x}_k)\left(\frac{1}{p}(\boldsymbol{m}_k \odot \boldsymbol{x}_k)^\top \beta^{(k-1)} - y_k\right) - \frac{1-p}{p^2}\operatorname{diag}((\boldsymbol{m}_k \odot \boldsymbol{x}_k)(\boldsymbol{m}_k \odot \boldsymbol{x}_k)^\top)\beta^{(k-1)} \tag{33}$$

where p denotes the probability for a value of being observed. This correction relies on the fact that, under the MCAR assumption, each entry of $\boldsymbol{M}$ follows a Bernoulli distribution with parameter p, so that $\mathbb{E}_{\boldsymbol{M}}[\boldsymbol{m}_k \odot \boldsymbol{x}_k] = p\boldsymbol{x}_k$. Consequently, to debias the term involving $\boldsymbol{x}_k$ in the original gradient expression (32), the quantity $(1/p)(\boldsymbol{m}_k \odot \boldsymbol{x}_k)$ is used in place of $\boldsymbol{m}_k \odot \boldsymbol{x}_k$ in (33). In this way, the procedure explicitly accounts for the zero-imputation step applied to the missing entries.

[6] Situations in which $\boldsymbol{y}_{\text{train}}$ is partially or entirely missing can instead be framed within unsupervised or semisupervised learning paradigms and are therefore beyond the scope of this tutorial. Relatively few studies address the unsupervised setting when the dataset $\boldsymbol{X}$ contains missing values. As an entry point using the tools presented here, we refer the reader to [44], which employs the EM algorithm to accommodate missing data. The semisupervised setting has only rarely been studied from a missing-data perspective; readers interested in this connection may consult [45] and [46].

Predict when $\boldsymbol{X}_{\text{train}}$ and $\boldsymbol{x}_{\text{new}}$ are incomplete

When missing values occur in both the training data and new observations, two main strategies can be employed [31]: 1) use a learning algorithm that directly handles missing data or 2) adopt a two-step approach; first impute the missing values to obtain a complete dataset, then train a predictive model on the imputed data. A crucial assumption underlying these strategies is that both the training set and test sets share not only the same data distribution but also the same distribution for missing data. For instance, if missingness is MCAR in the training set, it is assumed to be MCAR in the test set as well.

One-step strategy

This approach often refers to the Missing Incorporated in Attributes (MIA) method for handling missing values in tree-based algorithms [32]. In a standard decision tree, each node selects the splitting variable and threshold that minimize the error. When a variable contains missing values, MIA treats missingness as informative; for each candidate split, it determines not only the threshold but also the optimal direction (left or right) for routing missing values (see Figure 3(a)). This enables the tree to learn how to handle missing data during training, without requiring imputation or discarding observations. By integrating missingness directly into the tree, it can exploit patterns where the presence or absence of a value conveys information, making it well suited for MNAR scenarios.

Two-step strategy

The two-step strategy, or *impute-then-regress* approach, involves two distinct phases, imputation followed by prediction, as follows:

1) The training set $\boldsymbol{X}_{\text{train}}$ is imputed using a simple method $I_{\text{simp}}(\cdot)$.
2) A predictive model $\hat{d}$ is trained on the imputed dataset $(I_{\text{simp}}(\boldsymbol{X}_{\text{train}}), \boldsymbol{y}_{\text{train}})$.
3) The new observations $\boldsymbol{x}_{\text{new}}$ are predicted using the model from step (ii), i.e., $y_{\text{new}} = \hat{d}(I_{\text{simp}}(\boldsymbol{x}_{\text{new}}))$.

It is essential to utilize the same imputation procedure for both training and new data [31]. For example, if $I_{\text{simp}}(\cdot)$ is mean imputation, the mean of each variable i is computed from the training set $[\boldsymbol{X}_{\text{train}}]_{i,:}$ and used to impute missing values in both $[\boldsymbol{X}_{\text{train}}]_{i,:}$ and $[\boldsymbol{x}_{\text{new}}]_i$ ((see Figure 3(b)). This requirement highlights the need for an out-of-sample imputation method. Simple imputation techniques combined with a powerful learner at step 2, such as random forests, is often sufficient. More precisely, if the learning algorithm $\hat{d}$ is universally consistent when trained on complete data, it remains consistent under the impute-then-regress framework [31], even when simple imputation procedures are employed.

It is also commonly recommended to include the missing-data pattern $\boldsymbol{M}$ as additional input variables when training the predictive model $\hat{d}$, i.e., by concatenating the observed data matrix with the corresponding missing-data pattern (see Figure 3(a)). Intuitively, this reflects situations where the missing value itself carries predictive information. For instance, if sensor readings are frequently missing during extreme weather (e.g., sensor dropout during storms) and the objective is to forecast weather anomalies, this missingness can itself be exploited as informative signal [31], [33]. Overall, the choice of the imputation method becomes less critical when the predictor is sufficiently expressive and the missing-data pattern is included [34].

Label prediction—Takeaway

When the objective is to predict target values (labels), identifying which dataset contains missing values becomes crucial. If only the training set $\boldsymbol{X}_{\text{train}}$ is incomplete, missing data must be addressed carefully. This can be achieved either by applying advanced imputation techniques to obtain a complete dataset or by employing methods that estimate the relationship between $\boldsymbol{y}$ and $\boldsymbol{X}$ directly from the incomplete training set before applying the learned model to the test set. A simple approach consists of imputing the missing values naively and then adapting the prediction procedure to account for the imputation error. When both $\boldsymbol{X}_{\text{train}}$ and $\boldsymbol{x}_{\text{new}}$ contain missing values, imputation becomes less critical. In this case, the goal is to learn a model that predicts $\boldsymbol{y}$ from the observed components $\boldsymbol{X}_{\text{o}}$ of $\boldsymbol{X}$, acknowledging that future observations ($\boldsymbol{x}_{\text{new}}$) will also be incomplete. One possible strategy is to impute missing values in both the training and test sets using the same imputation model and then apply a standard learning algorithm to the resulting completed data. Finally, note that this tutorial does not address scenarios involving distributional shifts between training and test sets as they introduce additional challenges beyond its scope.

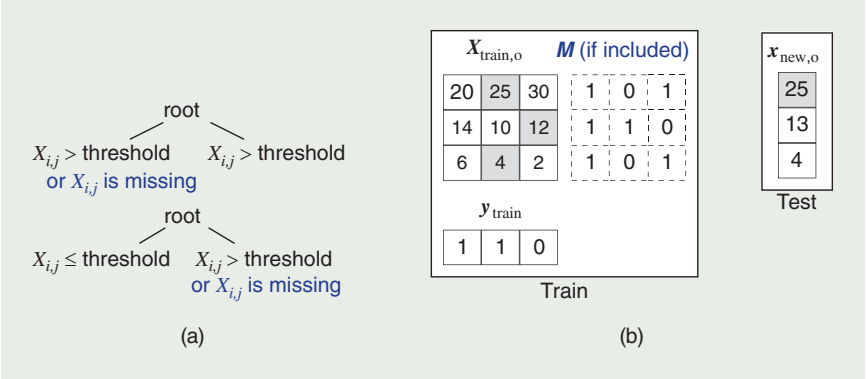

FIGURE 3. (a) An illustration of a split in the MIA method. The algorithm chooses the splitting variable $[\boldsymbol{X}]_{i,:}$, the threshold and the optimal direction to send missing values, the left or right side, represented by the graphs above. (b) The two-step strategy, when both $\boldsymbol{X}_{\text{train}}$ and $\boldsymbol{x}_{\text{new}}$ have missing entries. $\boldsymbol{X}_{\text{train,o}}$ and $\boldsymbol{x}_{\text{new,o}}$ denote the observed part of $\boldsymbol{X}_{\text{train}}$ and $\boldsymbol{x}_{\text{new}}$, respectively. The gray cases are missing values, and the blue entries are imputed values. For example, the variable $[\boldsymbol{X}_{\text{train}}]_{1,:}$ has been imputed by its mean, that is, 25. The same imputation model is applied for imputing the first variable $[\boldsymbol{x}_{\text{new}}]_1$ in the test set.

Conclusion and perspectives

This tutorial has provided a comprehensive treatment and unifying perspective of modern approaches organized around three core tasks—*imputation*, *estimation*, and *learning*—bridging classical statistical frameworks with contemporary algorithmic innovations.

Key insights. Throughout this tutorial, we have emphasized modern challenges where new methods and algorithms handling missing data have flourished. Effective handling of missing data requires a paradigm shift from simple and generic imputation to context-aware and model-based strategies that explicitly integrate domain-specific signal characteristics:

1) robustness to irregular and heterogeneous data through elliptically symmetric distributions and regularized optimization that maintain performance despite outliers and heavy tails
2) exploitation of low-dimensional structure via nuclear norm minimization, probabilistic PCA, and graph-regularized recovery, enabling both accurate imputation and computational efficiency in high dimensions.

Recent deep-learning approaches—notably VAEs—extend this further, offering greater flexibility for complex scenarios, including MNAR data, provided that prediction strategies are adapted to the specific missingness mechanism at hand.

Takeaway for practitioners. When tackling missing data in SP or ML, the adopted approach should align with both the nature of the data (data matrix, graph signal, or time series) and the analytical objective, whether it is imputation, parameter estimation, or predictive modeling (see each section's "Takeaway"). A good start is to characterize the quantity of missing data and the missingness mechanism as both fundamentally influence method selection. Specifically, determine whether the missingness is MCAR, MAR, or MNAR. Each mechanism carries distinct assumptions with direct implications for the validity of downstream analyses.

Once the analytical objective is chosen, key tradeoffs must be considered when selecting the appropriate methods. For imputation, model-based and deep learning methods offer robustness and flexibility for nonignorable mechanisms but often require greater computational resources and larger training datasets. For parameter estimation, likelihood-based procedures are well suited for structured problems, such as covariance estimation, ST, or graph learning, delivering rigorous uncertainty quantification and robust handling of missing data. However, they often face scalability issues in high dimensions. For prediction tasks, missing values in the training set require careful handling, either through principled model-based imputation or by explicitly modeling the link between predictors and response under missingness. When both training sets and test sets contain missing values, a simple imputation strategy combined with a strong prediction algorithm usually suffices. For the curious reader, the ground-truth S&P 500 index in Figure S5 corresponds to the first subplot—a testament to the quality of model-based MI.

Future directions. Numerous promising research directions, tied to the methods discussed in this tutorial, remain open. *Structure-aware approaches* should explore doubly robust algorithms that maintain validity under model misspecification and incorporate MNAR relationships grounded in domain-specific physical models. *Graph-based methods* require unified frameworks for MNAR graph learning, scaling to massive streaming networks, and handling heterogeneous graphs with mixed node types. *Uncertainty quantification* demands principled frameworks—leveraging self-supervised and semisupervised settings, modeling input uncertainty from conflicting sensors, and adapting conformal prediction to missing data—frameworks that are essential for high-stakes applications. Finally, *computational scalability* of deep generative models and SEM variants when applied to incomplete high-dimensional streaming data and large-scale multimodal data remains a critical and largely unsolved practical challenge.

Acknowledgment

This work was supported in part by the GlacioSim Project (ANR-25-CE56-3482-01) of the French National Research Agency (ANR), the DGA/AID Project 2025.65.0026, and in part by the Hong Kong GRF 16206123 research grant. We thank Çağatay Candan (associate editor) and the reviewers for their insightful feedback.

Authors

Alexandre Hippert-Ferrer (alexandre.hippert-ferrer@ign.fr) received his Ph.D. degree in remote sensing from the LISTIC Laboratory, University Savoie Mont-Blanc, Annecy, France, in 2020. Since 2022, he has been an associate professor in machine learning and remote sensing with IGN/ENSG, Université Gustave Eiffel, 77454 Champs-sur-Marne, France. His research interests include machine learning and statistical estimation with remote sensing data, including incomplete data.

Aude Sportisse (aude.sportisse@univ-grenoble-alpes.fr) received her Ph.D. degree in statistics from the LPSM Laboratory, Sorbonne University, in 2021. Since 2024, she has been a CNRS researcher at LIG, 38000 Grenoble, France. Her research interests include missing values, semisupervised learning, low-rank models, and clustering.

Amirhossein Javaheri (sajav@kth.se) received his Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology (HKUST) in January 2026. He is a postdoctoral researcher at the KTH Royal Institute of Technology, 11428 Stockholm, Sweden. His research interests include time-series analysis, graphical modeling, signal processing, optimization, machine learning, and financial data analytics.

Mohammed Nabil El Korso (mohammed.nabil.el-korso@centralesupelec.fr) received his Ph.D. degree from Paris-Sud XI University in 2011. He is a professor at the Paris Saclay University and Scientific Delegate at the CNRS, 91192 Gif-sur-Yvette, France. His research interests include robust statistical signal processing and learning, statistical analysis with missing values, and estimation theory with application to radio-interferometry, climate data analysis, and array processing.

Daniel P. Palomar (palomar@ust.hk) received his Ph.D. degree from the Technical University of Catalonia (UPC), Barcelona, Spain, in 2003 and was a Fulbright Scholar at

Princeton University during 2004–2006. He is a professor in the Department of Electronic & Computer Engineering at the Hong Kong University of Science and Technology (HKUST), Hong Kong 99999, China, which he joined in 2006. His research interests include optimization methods, high-frequency data, and deep learning in financial systems. He received the 2004, 2015, and 2020 Young Author Best Paper Awards from the IEEE Signal Processing Society. He is a Fellow of IEEE (since 2012) and EURASIP (since 2024).

References

[1] F. M. Lord, "Estimation of parameters from incomplete data," *J. Am. Stat. Assoc.*, vol. 50, no. 271, pp. 870–876, 1955, doi: 10.1080/01621459.1955.10501972.

[2] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976, doi: 10.1093/biomet/63.3.581.

[3] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, vol. 793. Hoboken, NJ, USA: Wiley, 2019.

[4] S. Seaman, J. Galati, D. Jackson, and J. Carlin, "What is meant by "missing at random"?" *Statist. Sci.*, vol. 28, no. 2, pp. 257–268, 2013, doi: 10.1214/13-STS415.

[5] J. W. Graham, "Missing data analysis: Making it work in the real world," *Annu. Rev. Psychol.*, vol. 60, no. 1, pp. 549–576, 2009, doi: 10.1146/annurev.psych.58.110405.085530.

[6] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, et al., "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001, doi: 10.1093/bioinformatics/17.6.520.

[7] S. Van Buuren, *Flexible Imputation of Missing Data*. Boca Raton, FL, USA: CRC Press, 2018.

[8] D. J. Stekhoven and P. Bühlmann, "MissForest—Non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012, doi: 10.1093/bioinformatics/btr597.

[9] J. Honaker, G. King, and M. Blackwell, "Amelia II: A program for missing data," *J. Statist. Softw.*, vol. 45, no. 7, pp. 1–47, 2011, doi: 10.18637/jss.v045.i07.

[10] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, Jul. 2017, doi: 10.1109/TSP.2017.2690388.

[11] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, Sep. 2015, doi: 10.1109/TSP.2015.2441042.

[12] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh, "Matrix completion and low-rank SVD via fast alternating least squares," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 3367–3402, 2015.

[13] J. Liu, S. Kumar, and D. P. Palomar, "Parameter estimation of heavy-tailed AR model with missing data via stochastic EM," *IEEE Trans. Signal Process.*, vol. 67, no. 8, pp. 2159–2172, Apr. 2019, doi: 10.1109/TSP.2019.2899816.

[14] M. Cherifi, X. Zhu, M. N. E. Korso, and A. Mesloub, "Maximum likelihood for logistic regression model with incomplete and hybrid-type covariates," *IEEE Signal Process. Lett.*, vol. 32, pp. 2808–2812, 2025, doi: 10.1109/LSP.2025.3583226.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977, doi: 10.1111/j.2517-6161.1977.tb01600.x.

[16] J.-P. Delmas, M. N. El Korso, S. Fortunati, and F. Pascal, *Elliptically Symmetric Distributions in Signal Processing and Machine Learning*. Cham, Switzerland: Springer-Verlag, 2024.

[17] A. Mennad, S. Fortunati, M. N. El Korso, A. Younsi, A. M. Zoubir, and A. Renaux, "Slepian-Bangs-type formulas and the related misspecified Cramér-Rao bounds for complex elliptically symmetric distributions," *Signal Process.*, vol. 142, pp. 320–329, Jan. 2018, doi: 10.1016/j.sigpro.2017.07.029.

[18] M. M. Cherifi, M. N. El Korso, S. Fortunati, A. Mesloub, and L. Ferro-Famil, "Robust inference with incompleteness for logistic regression model," *Signal Process.*, vol. 236, Nov. 2025, Art. no. 110027, doi: 10.1016/j.sigpro.2025.110027.

[19] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor, "Complex elliptically symmetric distributions: Survey, new results and applications," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5597–5625, Nov. 2012, doi: 10.1109/TSP.2012.2212433.

[20] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Hoboken, NJ, USA: Wiley, 2008.

[21] S. Allassonnière and J. Chevallier, "A new class of stochastic EM algorithms. Escaping local maxima and handling intractable sampling," *Comput. Statist. Data Anal.*, vol. 159, Jul. 2021, Art. no. 107159, doi: 10.1016/j.csda.2020.107159.

[22] P. Barbault, M. Kowalski, and C. Soussen, "LEMUR: Latent EM unsupervised regression for sparse inverse problems," *IEEE Trans. Signal Process.*, vol. 73, pp. 2087–2098, 2025, doi: 10.1109/TSP.2025.3565018.

[23] D. Ruppert and D. S. Matteson, *Statistics and Data Analysis for Financial Engineering*, vol. 13. Cham, Switzerland: Springer-Verlag, 2011.

[24] A. Aubry, A. De Maio, S. Marano, and M. Rosamilia, "Structured covariance matrix estimation with missing-(complex) data for radar applications via expectation-maximization," *IEEE Trans. Signal Process.*, vol. 69, pp. 5920–5934, 2021, doi: 10.1109/TSP.2021.3111587.

[25] A. Hippert-Ferrer, M. N. El Korso, A. Breloy, and G. Ginolhac, "Robust low-rank covariance matrix estimation with a general pattern of missing values," *Signal Process.*, vol. 195, Jun. 2022, Art. no. 108460, doi: 10.1016/j.sigpro.2022.108460.

[26] L. T. Thanh, N. V. Dung, N. L. Trung, and K. Abed-Meraim, "Robust subspace tracking with missing data and outliers: Novel algorithm with convergence guarantee," *IEEE Trans. Signal Process.*, vol. 69, pp. 2070–2085, 2021, doi: 10.1109/TSP.2021.3066795.

[27] L. Balzano, Y. Chi, and Y. Lu, "Streaming PCA and subspace tracking: The missing data case," *Proc. IEEE*, vol. 106, no. 8, pp. 1293–1310, Aug. 2018, doi: 10.1109/JPROC.2018.2847041.

[28] L. T. Thanh, N. V. Dung, N. L. Trung, and K. Abed-Meraim, "Robust subspace tracking algorithms in signal processing: A brief survey," *REV J. Electron. Commun.*, vol. 11, nos. 1–2, pp. 16–25, 2021.

[29] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETRELS: Parallel subspace estimation and tracking by recursive least squares from partial observations," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5947–5959, Dec. 2013, doi: 10.1109/TSP.2013.2282910.

[30] K. Gilman, D. Hong, J. A. Fessler, and L. Balzano, "Streaming heteroscedastic probabilistic PCA with missing data," 2025, *arXiv:2310.06277*.

[31] J. Josse, J. M. Chen, N. Prost, G. Varoquaux, and E. Scornet, "On the consistency of supervised learning with missing values," *Statist. Papers*, vol. 65, no. 9, pp. 5447–5479, 2024, doi: 10.1007/s00362-024-01550-4.

[32] B. E. Twala, M. Jones, and D. J. Hand, "Good methods for coping with missing data in decision trees," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 950–956, 2008, doi: 10.1016/j.patrec.2008.01.010.

[33] A. Perez-Lebel, G. Varoquaux, M. Le Morvan, J. Josse, and J.-B. Poline, "Benchmarking missing-values approaches for predictive models on health databases," *GigaScience*, vol. 11, Apr. 2022, Art. no. giac013, doi: 10.1093/gigascience/giac013.

[34] M. L. Morvan and G. Varoquaux, "Imputation for prediction: Beware of diminishing returns," in *Proc. Int. Conf. Learn. Representations*, 2025, pp. 41174–41213.

[35] A. Nazabal, P. M. Olmos, Z. Ghahramani, and I. Valera, "Handling incomplete heterogeneous data using VAEs," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107501, doi: 10.1016/j.patcog.2020.107501.

[36] P.-A. Mattei and J. Frellsen, "MIWAE: Deep generative modelling and imputation of incomplete data sets," in *Proc. Int. Conf. Mach. Learn.,* PMLR, 2019, pp. 4413–4423.

[37] N. B. Ipsen, P.-A. Mattei, and J. Frellsen, "not-MIWAE: Deep generative modelling with missing not at random data," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–18.

[38] C. Ma and C. Zhang, "Identifiable generative models for missing not at random data imputation," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 27,645–27,658.

[39] M. Cherifi, A. Sportisse, X. Zhu, M. N. E. Korso, and A. Mesloub, "Amortized variational inference for logistic regression with missing covariates," 2026, *arXiv:2603.21244*.

[40] A. B. Owen, "Monte Carlo theory, methods and examples," 2013. [Online]. Available: https://artowen.su.domains/mc/

[41] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., "Generative adversarial nets," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 2672–2680.

[42] J. Yoon, J. Jordon, and M. Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 5689–5698.

[43] B. Muzellec, J. Josse, C. Boyer, and M. Cuturi, "Missing data imputation using optimal transport," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 7130–7140.

[44] L. Hunt and M. Jorgensen, "Mixture model clustering for mixed data with missing information," *Comput. Statist. Data Anal.*, vol. 41, nos. 3–4, pp. 429–440, 2003, doi: 10.1016/S0167-9473(02)00190-1.

[45] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2004, pp. 529–536.

[46] H. Schmutz, O. Humbert, and P.-A. Mattei, "Don't fear the unlabelled: Safe semi-supervised learning via debiasing," in *Proc. 11th Int. Conf. Learn. Representations*, 2022, pp. 1–42.

[47] P.-L. Loh and M. J. Wainwright, "High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2011, pp. 2726–2734.

[48] A. Sportisse, C. Boyer, A. Dieuleveut, and J. Josse, "Debiasing averaged stochastic gradient descent to handle missing values," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 12,957–12,967.

SP