Time Series Analysis via Matrix Estimation

Anish Agarwal
MIT
anish90@mit.edu

Devavrat Shah
MIT
devavrat@mit.edu

Muhammad Jehangir Amjad
MIT
mamjad@mit.edu

Dennis Shen
MIT
deshen@mit.edu

ABSTRACT
We propose an algorithm to impute and forecast a time series by transforming the observed time series into a matrix, utilizing matrix estimation to recover missing values and de-noise observed entries, and performing linear regression to make predictions. At the core of our analysis is a representation result, which states that for a large model class, the transformed matrix obtained from the time series via our algorithm is (approximately) low-rank. This, in effect, generalizes the widely used Singular Spectrum Analysis (SSA) in the time series literature, and allows us to establish a rigorous link between time series analysis and matrix estimation. The key to establishing this link is constructing a matrix with non-overlapping entries rather than with the Hankel matrix as is done in the literature, including in SSA. We provide finite sample analysis for imputation and prediction leading to the asymptotic consistency of our method. A salient feature of our algorithm is that it is model agnostic both with respect to the underlying time dynamics as well as the noise model in the observations. Being noise agnostic makes our algorithm applicable to the setting where the state is hidden and we only have access to its noisy observations a la a Hidden Markov Model, e.g., observing a Poisson process with a time-varying parameter without knowing that the process is Poisson, but still recovering the time-varying parameter accurately. As part of the forecasting algorithm, an important task is to perform regression with noisy observations of the features a la an error-in-variable regression. In essence, our approach suggests a matrix estimation based method for such a setting, which could be of interest in its own right. Through synthetic and real-world datasets, we demonstrate that our algorithm outperforms standard software packages (including R libraries) in the presence of missing data as well as high levels of noise.

ACM Reference format:
Anish Agarwal, Muhammad Jehangir Amjad, Devavrat Shah, and Dennis Shen. 2016. Time Series Analysis via Matrix Estimation. In Proceedings of xxx, xxx, xxx, 19 pages.
DOI: 10.1145/nnnnn.nnnnnn

1 INTRODUCTION
Time series data is of enormous interest across all domains of life: from health sciences and weather forecasts to retail and finance, time dependent data is ubiquitous. Despite the diversity of applications, time series problems are commonly confronted by the same two pervasive obstacles: interpolation and extrapolation in the presence of noisy and/or missing data. Specifically, we consider a discrete-time setting with \( t \in \mathbb{Z} \) representing the time index and \( f : \mathbb{Z} \to \mathbb{R} \) representing the latent discrete-time series of interest. For each \( t \in [T] = \{1, \ldots, T\} \) and with probability \( p \in (0,1] \), we observe the random variable \( X(t) \) such that \( E[X(t)] = f(t) \). While the underlying mean signal \( f \) is strongly correlated, we assume the per-step noise is independent across \( t \) and has uniformly bounded variance. Under this setting, we have two objectives: (1) interpolation, i.e., estimate \( f(t) \) for all \( t \in [T] \); (2) extrapolation, i.e., forecast \( f(t) \) for \( t > T \). Our interest is in designing a generic method for interpolation and extrapolation that is applicable to a large model class while being agnostic to time dynamics and noise distribution.

We develop an algorithm based on matrix estimation, a topic which has received widespread attention, especially with the advent of large datasets. In matrix estimation, there is a “parameter” matrix \( M \) of interest, and we observe a sparse, corrupted signal matrix \( X \) where \( E[X] = M \); the aim then is to recover the entries of \( M \) from noisy and partial observations given in \( X \). For our purposes, the attractiveness of matrix estimation derives from the property that these methods are fairly model agnostic in terms of structure of \( M \) and distribution of \( X \) given \( M \): effectively begging to be utilized to develop model and noise agnostic time series imputation and forecasting algorithms. And we precisely do that in this work.

1.1 Overview of contributions.

Time series becomes a matrix. We transform the time series of observations \( X(t) \) for \( t \in [T] \) into a matrix by placing contiguous segments of size \( L > 1 \) (an algorithmic hyper-parameter) of the time series into non-overlapping columns: see Figure 1 for a caricature of this transformation.

As the key contribution, we establish that in expectation, this generated matrix is either exactly or approximately low-rank for a large class of models \( f \). Specifically, \( f \) can be from the following families:

Linear Recurrent Formulae (LRF): \( f(t) = \sum_{g=1}^{G} f(t-g) \).

---

1We denote \( \mathbb{R} \) as the field of real numbers and \( \mathbb{Z} \) as the integers.
we apply a matrix estimation (ME) algorithm to

Sublinear: \( f(t) = g(t), g : \mathbb{R} \rightarrow \mathbb{R}, \left| \frac{dg(x)}{dx} \right| \leq Cs^{-\alpha}, \alpha, C > 0, \forall s \in \mathbb{R} \)

Over the past decade, the matrix estimation community has developed a plethora of methods to recover an exact or approximately low-rank matrix from its noisy, partial observations in a noise model agnostic manner. Therefore, by applying such a matrix estimation method to this transformed matrix, we can recover the underlying mean matrix (and thus \( f(t) \) for \( t \in [T] \)) accurately. In other words, we can interpolate and de-noise the original corrupted and incomplete time series without any knowledge of its time dynamics or noise distribution. Theorem 4.1 and Corollary 4.1 provide finite-sample analyses for this method and establish the consistency property of our algorithm, as long as the underlying \( f \) satisfies Property 4.1 and the matrix estimation method satisfies Property 2.1. In Section 5, we show that Property 4.1 holds for any additive mixture of the three function classes listed above. Effectively, Theorem 4.1 establishes a statistical reduction between time series imputation and matrix estimation.

It is clear that for LRF, the mean transformed matrix is such that its last row can be expressed as a linear combination of the other rows. We generalize this notion by arguing that for the other two model classes, an approximate LRF relationship holds. Therefore, we can forecast \( f(t) \), say for \( t = T + 1 \), as follows: apply matrix estimation to the transformed data matrix as done in imputation; then, linearly regress the last row with respect to the other rows in the matrix; finally, compute the inner product of the learnt regression vector with the vector containing the previous \( L - 1 \) values that were estimated via the matrix estimation method. Theorem 4.2 and Corollary 4.2 imply that the mean-squared error in forecasting under this algorithm decays to zero provided the matrix estimation method satisfies Property 2.2 and the underlying model \( f \) satisfies Property 4.2. In Section 5, we show that Property 4.2 holds for the three function classes listed above.

Noisy regression. The forecasting algorithm as described above effectively performs regression when observations of features are missing as well as noisy. In literature, this is known as error-in-variables regression. There has been exciting recent progress to understand it especially in the high-dimensional setting. Our algorithm provides a solution for the high-dimensional setting through matrix estimation: perform matrix estimation on noisy, missing observations of features and then perform least squares. If the feature matrix can be approximated by a low-rank matrix, then it leads to a consistent estimator. This, however, requires utilizing a non-standard error for Matrix Estimation, the max row sum error (MRSE) (see Property 2.2).

Which models satisfy the requirement. As mentioned above, the algorithm enjoys strong performance guarantees provided the underlying mean matrix induced by the time series \( f \) satisfies certain structural properties (Properties 4.1, 4.2). We argue that a broad class of time series models that are commonly used meet the requirements of the three function classes listed above.

LRFs include the following important family of time series - \( f \) which can be represented as a finite sum of products of exponentials (\( \exp(at) \)), harmonics (\( \cos(2\pi\omega t + \phi) \)) and finite degree polynomials (\( P_m(t) \)) [27], i.e. \( f(t) = \sum_{g=1}^{G} \exp(a_g t) \cos(2\pi\omega_g t + \phi_g)P_{m_g}(t) \). Further, since stationary processes and \( L_2 \) integrable functions are well approximated by a finite summation of harmonics (i.e. sin and cos), LRFs encompass a vitally important family of models. For this model, we show that indeed the structural properties required from the time series matrix for both imputation and forecasting are satisfied.

However, there are many important time series models that do not admit a finite order LRF representation. A few toy examples of such \( f \) include, \( \cos(\sin(t)) \), \( \exp[\sin^2(t)] \), \( \log t \). Now a model \( f \) composed of a finite summation of functions that exhibit periodicity (e.g. \( \cos(\sin(t)) \), \( \exp[\sin^2(t)] \)) do indeed have compact support and we argue that due to our approximately low-rank representation result, all the desired structural properties hold. Further, a model \( f \) composed of a finite summation of functions that do not scale (super-)linearly (e.g. \( \log t \), \( \sqrt{t} \)) are indeed sub-linear, and again the desired properties hold. More importantly, we argue that the finite mixture of the above processes possesses the necessary structural properties.
Recovering the hidden state. Our algorithm, being noise and time-dynamics agnostic, makes it relevant to recover the hidden state from its noisy, partial observations as in a Hidden Markov-like Model. For example, imagine having access to partial observations of a time-varying Poisson process without knowledge that the process is Poisson. By applying our algorithm for imputation, we can recover the time-varying parameters of this process accurately and thus recover this hidden state. If we were to apply an Expectation-Maximization (EM) like algorithm, it would require knowledge of the underlying model being Poisson, and even then theoretical guarantees are not clear for such an approach.

Sample complexity. Given its generality and the model agnostic nature of our algorithm, it is expected that its sample complexity for a specific model class will be worse than model aware optimal algorithms. Interestingly, our finite sample analysis suggests that for the model classes stated above, the performance loss incurred due to this generality is minor. The details are in Section 5.6.

Experiments. Using synthetic and real-world datasets, our experiments establish that our method outperforms existing standard software packages (including R) for the tasks of interpolation and extrapolation in the presence of noisy and missing observations. When the data is generated synthetically, we "help" the existing software package by choosing the correct parametric model and algorithm while our algorithm remains oblivious to the underlying model; despite this disadvantage, our algorithm continues to outperform the standard packages with missing data.

Our empirical studies demonstrate that our imputation algorithm accurately recovers the hidden state, verifying our theoretical imputation guarantees (see Theorem 4.1). All experimental findings can be found in Section 6.

1.2 Related works

There are two related topics: matrix estimation and time series analysis. Both have a rich literature. Naturally, we cannot do justice in providing a full overview. Instead, we provide a high-level summary of known results with references that provide details.

Matrix estimation. Matrix estimation is the problem of recovering a data matrix from an incomplete and noisy sampling of its entries. This has become of great interest due to its connection to recommendation systems (cf. [17–19, 23, 32–35, 38]), social network analysis (cf. [1–3, 7, 30]) and graph learning (graphon estimation) (cf. [4, 13, 14, 50]). The key realization of this rich literature is that one can do estimation of the true underlying matrix from noisy, partial observations by simply taking a low-rank approximation of the observed data. We refer an interested reader to recent works such as [13, 18] and references there in.

Time series analysis. The question of time series analysis is potentially as old as civilization in some form. Few textbook style references include [15, 16, 28, 40]. At the highest level, time series modeling primarily involves viewing a given time series as a function indexed by time (integer or real values) and the goal of model learning is to identify this function from observations (over finite intervals). Given that the space of such functions is complex, the task is to utilize function form (i.e. “basis functions”) so that for the given setting, the time series observation can fit a sparse representation. For example, in communication and signal processing, the harmonic or Fourier representation of a time series has been widely utilized, due to the fact that signals communicated are periodic in nature. The approximation of stationary processes via harmonics or ARIMA has made them a popular model class to learn stationary-like time series, with domain specific popular variations, such as “Autoregressive Conditional Heteroskedasticity” (ARCH) in finance. To capture non-stationary or “trend-like” behavior, polynomial bases have been considered. There are rich connections to the theory of stochastic processes and information theory (cf. [21, 26, 39, 44]). Popular time series models with latent structure are the Hidden Markov Model (HMM) in its probabilistic form, (cf. [9, 31]) and the Recurrent Neural Network (RNN) in its deterministic form, (cf. [41]).

The question of learning time series models with missing data has received comparatively less attention. A common approach is to utilize HMMs or general State-Space-Models to learn with missing data, (cf. [24, 45]). To the best of the authors’ knowledge, most work in this literature is restricted to such class of models, (cf. [25]). Recently, building on the literature in online learning, sequential approaches have been proposed to address prediction with missing data, (cf. [8]).

Time series and matrix estimation. The use of matrix structure for time series analysis has roughly two streams of prior works: the Singular Spectrum Analysis (SSA) for a single time series as in our setting, and the use of multiple time series. We discuss relevant results for both of these topics.

Singular Spectrum Analysis (SSA) of time series has been around for some time. Generally, it assumes access to time series data that is not noisy and fully observed. The core steps of SSA for a given time series are as follows: (1) create a Hankel matrix from the time series data, (2) perform a Singular Value Decomposition (SVD) of it; (3) group the singular values based on user belief of the model that generated the process; (4) perform diagonal averaging for the “Hankelization” of the grouped rank-1 matrices outputted from the SVD to create a set of time series; learn a linear model for each "Hankelized" time series for the purpose of forecasting.

At the highest level, SSA and our algorithm have similarities. But there are key differences: (i) the matrix transformation is different — SSA uses Hankel structure, we use non-overlapping structure; (ii) matrix processing is different — SSA does standard SVD, we utilize matrix estimation (which could be singular value thresholding, but that is not necessary), and (iii) linear regression — SSA assumes access to noiseless features, we do not.

These differences are key for deriving theoretical results. For example, there have been few recent works that attempt to apply matrix estimation method to the Hankel structure matrix inspired by SSA for imputation without any theoretical guarantees [42, 43, 46]. In effect, the Hankel structure creates strong correlation of noise in the matrix which creates an impediment for proving theoretical results. Our use of non-overlapping matrix overcomes this challenge and we argue that in doing so we still retain the underlying structure in the matrix.

It is also worth noting that the classical literature on SSA seem to be lacking finite sample analysis in the presence of noisy observations to the best of authors’ knowledge.
Multiple time series viewed as matrix. In some of the recent works [5, 6, 20, 37, 47, 49], multiple time series have been viewed as a matrix with the goal of primarily imputing the missing values or de-noising them. Some of these works also require prior model assumptions on the underlying time series. For example in [49], as stated in Section 1, the second step of their algorithm changes based on the user’s belief in the model that generated the data along with the multiple time series requirement.

In summary, to the best of our knowledge, ours is the first work to give rigorous theoretical guarantees for a matrix estimation inspired algorithm, for a single univariate time series.

Recovering the hidden state. The question of recovering the hidden state from noisy observations is quite prevalent and a workhorse of classical systems theory. For example, most of the system identification literature focuses on recovering model parameters of a Hidden Markov Model. While Expectation-Maximization or Baum-Welch are the go-to approaches, there is limited theoretical understanding of it in generality (for example, see a recent work [48] for an overview) and knowledge of the underlying model is required. In [12], for instance, an optimization based statistically consistent estimation method was proposed. However, the optimization “objective” encoded the knowledge of the precise underlying model.

It is worth comparing our method with a recent work [5] where the authors attempt to recover the hidden time-varying parameter of classical systems theory. For example, most of the system identification algorithms require access without knowing the underlying model; this can be easily accomplished by scaling X in a data-centric manner.

1.3 Notation.
For any positive integer N, let \([N] = \{1, \ldots, N\}\). For any vector \(v \in \mathbb{R}^n\), we denote its Euclidean \((\ell_2)\) norm by \(\|v\|_2\), and define \(\|v\|_2^2 = \sum_{i=1}^n v_i^2\). In general, the \(\ell_p\) norm for a vector \(v\) is defined as \(\|v\|_p = \left(\sum_{i=1}^n |v_i|^p\right)^{1/p}\).

For a \(m \times n\) real-valued matrix \(A = [A_{ij}]\), its spectral/operator norm, denoted by \(\|A\|_2\), is defined as \(\|A\|_2 = \max_{1 \leq |j| \leq k} |\sigma_i|\), where \(k = \min(m, n)\) and \(\sigma_i\) are the singular values of \(A\) (assumed to be in decreasing order and repeated by multiplicities). The Frobenius norm, also known as the Hilbert-Schmidt norm, is defined as \(\|A\|^2_F = \sum_{i=1}^m \sum_{j=1}^n A_{ij}^2 = \sum_{i=1}^k \sigma_i^2\). The max-norm, or sup-norm, is defined as \(\|A\|_{\max} = \max_{i, j} |A_{ij}|\). The Moore-Penrose pseudoinverse \(A^\dagger\) of \(A\) is defined as

\[
A^\dagger = \sum_{i=1}^k (1/\sigma_i) y_i x_i^T,
\]

where \(x_i\) and \(y_i\) being the left and right singular vectors of \(A\), respectively.

Let \(f\) and \(g\) be two functions defined on the same space. We say that \(f(x) = O(g(x))\) and \(f(x) = \Omega(g(x))\) if and only if there exists a positive real number \(M\) and a real number \(x_0\) such that for all \(x \geq x_0\),

\[
|f(x)| \leq M|g(x)| \quad \text{and} \quad |f(x)| \geq M|g(x)|,
\]

respectively.

1.4 Organization
In Section 2, we list the desired properties needed from matrix estimation methods to achieve our theoretical guarantees for imputation and forecasting. In Section 3, we formally describe the matrix estimation based algorithms we utilize for time series analysis. In Section 4, we identify the required properties of time series models \(f\) under which we can provide finite sample analysis for imputation and forecasting performance. In Section 5, we list out a broad set of time series models that satisfy the properties in Section 4 and we analyze the sample complexity of our algorithm for each of these models. Lastly, in Section 6, we corroborate our theoretical findings with detailed experiments.

2 MATRIX ESTIMATION
2.1 Problem setup
Consider an \(m \times n\) matrix \(M\) of interest. Suppose we observe a random subset of the entries of a noisy signal matrix \(X\), such that \(E[X] = M\). For each \(i \in [m]\) and \(j \in [n]\), the \((i, j)\)-th entry \(X_{ij}\) is a random variable that is observed with probability \(p \in (0, 1)\) and is missing with probability \(1-p\), independently of all other entries. Given \(X\), the goal is to produce an estimator \(\hat{M}\) that is “close” to \(M\). We use two metrics to quantify the estimation error:

1. Entry-wise mean-squared error,

\[
\text{MSE}(\hat{M}, M) := \mathbb{E} \left[ \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (\hat{M}_{ij} - M_{ij})^2 \right];
\]

2. Max row sum error,

\[
\text{MRSE}(\hat{M}, M) := \max_{i \in [m]} \mathbb{E} \left[ \frac{1}{n} \sum_{j=1}^n (\hat{M}_{ij} - M_{ij})^2 \right]^{1/2}.
\]

Here, \(\hat{M}_{ij}\) and \(M_{ij}\) denote the \((i, j)\)-th elements of \(\hat{M}\) and \(M\), respectively. We now give a definition of a matrix estimation algorithm, which will be used in the following sections.

Definition 2.1. A matrix estimation algorithm, denoted as \(\text{ME} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}\), takes as input a noisy matrix \(X\) and outputs an estimator \(\hat{M}\).

2.2 Required properties of matrix estimation algorithms
As aforementioned, our algorithm (Section 3.3) utilizes matrix estimation as a pivotal “blackbox” subroutine, which enables model agnostic accurate imputation and prediction. Over the past decade, the field of matrix estimation has spurred tremendous theoretical and empirical research interest, leading to the emergence of a myriad of algorithms such as spectral, convex optimization, and nearest neighbor based methods. Consequently, as the field continues to advance, our algorithm will continue to improve in parallel. We now state the properties needed of a matrix estimation algorithm \(\text{ME}(\cdot)\) to achieve our theoretical guarantees (formalized through Theorems 4.1 and 4.2). See Section 1.3 for matrix norm definitions.

Property 2.1. Let \(\text{ME}\) satisfy the following: Define \(Y = [Y_{ij}]\) where \(Y_{ij} = X_{ij}\) if \(X_{ij}\) is observed, and \(Y_{ij} = 0\) otherwise\(^3\). Then for all \(x \geq x_0\),

\[
|f(x)| \leq M|g(x)| \quad \text{and} \quad |f(x)| \geq M|g(x)|,
\]

\(^3\) We assume the entries of \(X\) are bounded between \([-1, 1]\); this can be easily accomplished by scaling \(X\) in a data-centric manner.
We argue the two quantities in Property 2.1, \( \| Y - pM \|_2 \) and \( \| M \|_\infty \), are natural. \( \| Y - pM \|_2 \) quantifies the amount of noise corruption on the underlying signal matrix \( M \); for many settings, this norm concentrates well (e.g., a matrix with independent zero-mean sub-Gaussian entries scales as \( \sqrt{m} + \sqrt{r} \) with high probability). \( \| M \|_\infty \) is a measure of the inherent model complexity of the latent signal matrix; this norm is well behaved for an array of situations, including low-rank and Lipschitz matrices (e.g., for low-rank matrices, \( \| M \|_\infty \) scales as \( \sqrt{rm} \) where \( r \) is the rank of the matrix, see [18] for bounds on \( \| M \|_\infty \) under various settings). We note the universal singular value thresholding algorithm proposed in [18] is one such algorithm that satisfies Property 2.1.

**Property 2.2.** Let \( ME \) satisfy the following: For all \( p \geq p^*(m, n) \), the produced estimator \( \hat{M} = ME(X) \) satisfies

\[
\text{MSE}(\hat{M}, M) \leq \delta_3(m, n)
\]

where \( \lim_{m, n \to \infty} \delta_3(m, n) = 0 \).

Property 2.2 requires the normalized max row sum error to decay to zero as we collect more data. While spectral thresholding and convex optimization methods accurately bound the average mean-squared error, minimizing norms akin to the normalized max row sum error require matrix estimation methods to utilize "local" information, e.g., nearest neighbor type methods. For instance, [50] satisfies Property 2.2 for generic latent variable models (which include low-rank models) with \( p^*(m, n) = 1 \); [34] also satisfies Property 2.2 for \( p^*(m, n) \gg \text{min}(m, n)^{-1/2} \); [13] establishes this for low-rank models as long as \( p^*(m, n) \gg \text{min}(m, n)^{-1} \).

### 3 ALGORITHM

#### 3.1 Notations and definitions.

Recall that \( X(t) \) denotes the observation at time \( t \in [T] \) where \( \mathbb{E}[X(t)] = f(t) \). We shall use the notation \( X'[s : t] = [X(s), \ldots, X(t)] \) for any \( s \leq t \). Furthermore, we define \( L > 1 \) to be an algorithmic hyperparameter and \( N = \lfloor T/L \rfloor - 1 \). For any \( L \times N \) matrix \( A \), let \( \bar{A} = [\bar{A}_{ij}]_{1 \leq i \leq N} \) represent the the last row of \( A \). Moreover, let \( \bar{A} = [\bar{A}_{ij}]_{1 \leq i \leq L, 1 \leq j \leq N} \) denote the \((L - 1) \times N\) submatrix obtained by removing the last row of \( A \).

#### 3.2 Viewing a univariate time series as a matrix.

We begin by introducing the crucial step of transforming a single, univariate time series into a matrix. Given time series data \( X[1 : T] \), we construct \( L \) different \( L \times N \) matrices \( X^{(k)} \), defined as

\[
X^{(k)} = X'_{[1 : T]} = X'(i + (j - 1)L + (k - 1))_{1 \leq L, j \leq N},
\]

where \( k \in [L] \). In words, \( X^{(k)} \) is obtained by dividing the time series into \( N \) non-overlapping contiguous intervals each of length \( L \), thus constructing \( N \) columns; for \( k \in [L] \), \( X^{(k)} \) is the \( k \)-th shifted version with starting value \( X(k) \). For the purpose of imputation, we shall only utilize \( X^{(1)} \). In the case of forecasting, however, we shall utilize \( X^{(k)} \) for all \( k \in [L] \). We define \( M^{(k)} \) analogously to \( X^{(k)} \) using \( f(t) \) instead of \( X(t) \).

#### 3.3 Algorithm description.

We will now describe the imputation and forecast algorithms separately (see Figure 1).

**Imputation.** Due to the matrix representation \( X^{(1)} \) of the time series, the tasks of imputing missing values and de-noising observed values translates to that of matrix estimation.

1. Transform the data \( X[1 : T] \) into the matrix \( X^{(1)} \) via the method outlined in Subsection 3.2.
2. Apply a matrix estimation method (as in Definition 2.1) to \( \hat{X}^{(1)} = ME(X^{(1)}) \).
3. Produce estimate: \( \hat{f}_t(i + (j - 1)L) = \hat{M}_{ij}^{(1)} \) for \( i \in [L] \) and \( j \in [N] \).

**Forecast.** In order to forecast future values, we first de-noise and impute via the procedure outlined above, and then learn a linear relationship between the the last row and the remaining rows through linear regression.

1. For \( k \in [L] \), apply the imputation algorithm to produce \( \hat{M}^{(k)} \) from \( \hat{X}^{(k)} \) (recall from Section 3.2 that \( \hat{A} \) refers to the submatrix of the first \( L - 1 \) rows of \( A \)).
2. For \( k \in [L] \), define \( \hat{M}^{(k)} = \arg \min \mathcal{C}_{r \in [L] - 1} \left\| X^{(k)}_{L} - (\hat{M})^T \mathcal{C}_{r \in [L] - 1} \right\|_2^2 \)
3. Produce the estimate at time \( t > T \) as follows:
   
   (i) Let \( \mathcal{C}_t := [X(t - L + 1 : X(t - 1)] \) and \( k = (t \mod L) + 1 \).
   
   (ii) Let \( \mathcal{C}_{t}^\text{proj} = \arg \min \mathcal{C}_{r \in [L] - 1} \left\| \mathcal{C}_{r} - (\hat{M})^T \mathcal{C}_{t} \right\|_2^2 \)
   
   (iii) Produce estimate: \( \hat{f}_t(i) = (\mathcal{C}_{t}^\text{proj} \hat{M})_i \).

**Choosing the number of rows \( L \):** Theorems 4.1 and 4.2 suggest \( L \) should be as large as possible, with the requirement \( L = o(N) \). Thus, it suffices to let \( N = L^{1+\delta} \) for any \( \delta > 0 \), e.g., \( N = L^2 = T^{2/3} \).

### 4 MAIN RESULTS

#### 4.1 Properties.

We now introduce the required properties for the matrices \( X^{(k)} \) and \( M^{(k)} \) to identify the time series models \( f \) for which our algorithm provides an effective method for imputation and prediction. Under these properties, we state Theorems 4.1 and 4.2, which establish the efficacy of our algorithm. The proofs of these theorems can be found in Appendices B and C respectively. In Section 5, we argue these properties are satisfied for a large class of time series models.

**Property 4.1.** Let matrices \( X^{(1)} \) and \( M^{(1)} \) satisfy the following:

\( A \). For each \( i \in [L] \) and \( j \in [N] \):

1. **Precisely**, we define \( \hat{p} = \max \left( \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} \text{ observed} \cdot \frac{1}{mn} \right) \).
1. $X_{ij}^{(1)}$ are independent RVs satisfying $\mathbb{E}[X_{ij}^{(1)}] = M_{ij}^{(1)}$ and $\text{Var}(X_{ij}^{(1)}) \leq \sigma^2$.
2. $X_{ij}^{(2)}$ is observed with probability $p \in (0, 1]$, independent of other entries.

B. There exists a matrix $M_{(r)}$, of rank $r$ such that for $\delta_1 \geq 0$,
\[
\left\| M^{(1)} - M_{(r)} \right\|_{\text{max}} \leq \delta_1.
\]

Property 4.2. For $k \in [L]$, let matrices $X^{(k)}$ and $M^{(k)}$ satisfy the following:

A. For each $i \in [L]$ and $j \in [N]$:
1. $X_{ij}^{(k)} = M_{ij}^{(k)} + \epsilon_{ij}$, where $\epsilon_{ij}$ are independent sub-Gaussian RVs satisfying $\mathbb{E}[\epsilon_{ij}] = 0$ and $\text{Var}(\epsilon_{ij}) \leq \sigma^2$.
2. $X_{ij}^{(2)}$ is observed with probability $p \in (0, 1]$, independent of other entries.

B. There exists a matrix $M^{(k)} \in \mathbb{R}^{L \times 1}$ with $\|\beta^{(k)}\|_2 = C\beta^{(k)}$ for some constant $C\beta^{(k)} > 0$ and $\delta_2 \geq 0$ such that
\[
\left\| M^{(k)}_L - (\hat{M}^{(k)})^T \beta^{(k)} \right\|_2 \leq \delta_2.
\]

4.2 Imputation.

The imputation algorithm produces $\hat{f}_t$ as the estimate for the underlying time series $f$. We measure the imputation error through the mean-squared error:
\[
\text{MSE}^{(\hat{f}_t, f)} := \mathbb{E}\left[ \frac{1}{T} \sum_{t=1}^{T} (\hat{f}_t(t) - f(t))^2 \right].
\]

Theorem 4.1. Assume Property 4.1 holds and ME satisfies Property 2.1, with $p \geq \max(L,N)^{1-\xi}$ for some $\xi \in (0, 1]$. Then for some $C_1, C_2, C_3, C_4 > 0$,
\[
\text{MSE}^{(\hat{f}_t, f)} \leq C_1 \left( \frac{\delta_1}{\sqrt{p}} + \sqrt{\frac{\xi}{pL}} \right) + C_2 \left( \frac{1}{Lp} + C_3 \delta^4 \right).
\]

Theorem 4.2 states that any matrix estimation subroutine ME that satisfies Property 2.1 will accurately filter noisy observations and recover missing values. This is achieved provided that the rank of $M_{(r)}$ and our low-rank approximation error $\delta_1$ are not too large. Additionally, Theorem 4.2 implies the following consistency property of $\hat{f}_t$.

Corollary 4.1. Let the conditions for Theorem 4.1 hold. Suppose $\delta_1 \leq C_6 L^{-\epsilon_1}$, $r \leq C_6 L^{-\epsilon_2}$ for some $\epsilon_1, \epsilon_2 \in (0, 1)$ and $C_5, C_6 > 0$. Then for $p \gg L^{-\min\{\epsilon_1, \epsilon_2\}}$
\[
\lim_{T \to \infty} \text{MSE}^{(\hat{f}_t, f)} = 0.
\]

As argued in Section 5, the conditions of Corollary 4.1 hold for a large class of time series models.

4.3 Forecasting.

Recall $\hat{f}_t(t)$ can only utilize information till time $t - 1$. For $k \in [L]$, our forecasting algorithm learns $\hat{\beta}^{(k)}$ with the previous $L - 1$ time steps. We measure the forecasting error through:
\[
\text{MSE}^{(\hat{f}_t, f)} := \mathbb{E}\left[ \frac{1}{T-L+1} \sum_{t=L}^{T} (\hat{f}_t(t) - f(t))^2 \right].
\]

Theorem 4.2. Assume Property 4.2 holds and ME satisfies Property 2.2, with $p \geq p^*(L,N)^{\gamma}$. Let $\hat{r} := \max_{k \in [L]} \hat{r}^{(k)}$. Then,
\[
\text{MSE}^{(\hat{f}_t, f)} \leq \frac{1}{N-1} \left( \delta_2 + \sqrt{C_\beta N \delta_3}^2 + 2\sigma^2 \hat{r} \right).
\]

Note that $\hat{r}$ is bounded by $L = o(N)$ (see Section 3). If the underlying matrix, $M$, is low-rank then ME algorithms such as the universal singular value thresholding algorithm (cf. [18]) will output a $\hat{M}$ with a corresponding $\hat{r}$ that is small. However, we provide bounds for general ME algorithms and so explicitly state the dependence on $\hat{r}$. Theorem 4.2 states that any matrix estimation subroutine ME that satisfies Property 2.2 will produce accurate forecasts from noisy, missing data. This is achieved provided the linear model approximation error $\delta_2$ is not too large (recall $\delta_3 = o(1)$ by Property 2.2). Additionally, Theorem 4.2 implies the following consistency property of $\hat{f}_t$.

Corollary 4.2. Let the conditions for Theorem 4.2 hold. Suppose $\delta_2 = C_1 N^{-1-\epsilon_1}$ for any $\epsilon_1, \epsilon_2 \in (0, 1)$ and $N = L^{1+\epsilon}$ for any $\delta > 0$. Then for $p \geq p^*(L,N)$, such that $\lim_{L \to \infty} \delta_3(L,N) = 0$ for $p^*(L,N)$,
\[
\lim_{T \to \infty} \text{MSE}^{(\hat{f}_t, f)} = 0.
\]

As argued in Section 5, the conditions of Corollary 4.2 hold for a large class of time series models. Learning a simple linear relationship among the singular vectors of the de-noised matrix is sufficient to drive the empirical error to zero for a broad class of time series models. The simplicity of this linear method suggests that our estimator will have low generalization error, but we leave that as future work.

5 FAMILY OF TIME SERIES THAT FIT OUR FRAMEWORK

In this section, we list out a broad set of time series models that satisfy Properties 4.1 and 4.2, which are required for the results stated in Section 4. The proofs of these results can be found in Appendix D. To that end, we shall repeatedly use the following model types for our observations.

Model Type 1. For any $t \in \mathbb{Z}$, let $X(t)$ be independent RVs such that $\mathbb{E}[X(t)] = 0$ and $\text{Var}(X(t)) \leq \sigma^2$. Note the noise on $f(t)$ is generic (e.g. non-additive).

Model Type 2. For $t \in \mathbb{Z}$, let $X(t) = f(t) + \epsilon(t)$, where $\epsilon(t)$ are sub-Gaussian independent RVs with $\mathbb{E}[\epsilon(t)] = 0$ and $\text{Var}(\epsilon(t)) \leq \sigma^2$.

6Recall that this condition only requires the per-step noise to be independent; the underlying mean time series $f$ remains highly correlated.
5.1 Linear Recurrent Functions (LRFs).

For \( t \in \mathbb{Z} \), let

\[
    f^{\text{LRF}}(t) = \sum_{g=1}^{G} a_g f(t-g).
\]

(1)

Proposition 5.1.

(i) Under Model Type 1, \( f^{\text{LRF}} \) satisfies Property 4.1 with \( \delta_1 = 0, r = \mathcal{L} \).

(ii) Under Model Type 2, \( f^{\text{LRF}} \) satisfies Property 4.2 with \( \delta_2 = 0 \) and \( C_{p(k)} = \mathcal{C} G \) for \( k \in \{L\} \) where \( \mathcal{C} \) is an absolute constant.

By Proposition 5.1, Theorems 4.1 and 4.2 give the following corollaries:

**Corollary 5.1.** Under Model Type 1, let the conditions of Theorem 4.1 hold. Let \( N = L^{1+\delta} \) for any \( \delta > 0 \). Then for some \( C > 0 \), if

\[
    T \geq C \left( \frac{G}{\delta_{\text{error}}} \right)^{2+\delta}
\]

we have \( \text{MSE}(\hat{f}, f^{\text{LRF}}) \leq \delta_{\text{error}} \).

**Corollary 5.2.** Under Model Type 2, let the conditions of Theorem 4.2 hold. Let \( N = L^{1+\delta} \) for any \( \delta > 0 \). Then for some \( C > 0 \), if

\[
    T \geq C \left( \frac{\sigma^2}{\delta_{\text{error}} - \mathcal{C} \left( \frac{2 \pi}{L} + \delta_3 \right)^{1/2}} \right)^{2+\delta}
\]

we have \( \text{MSE}(\hat{f}, f^{\text{LRF}}) \leq \delta_{\text{error}} \).

**Proposition 5.2.** Let \( P_{m_a} \) be a polynomial of degree \( m_a \). Then,

\[
    f(t) = \sum_{a=1}^{A} \exp(\alpha_a t) \cos(2\pi \omega_a t + \phi_a) P_{m_a}(t)
\]

admits a representation as in (1). Further the order \( \mathcal{G} \) of \( f(t) \) is independent of \( T \), the number of observations, and is bounded by

\[
    \mathcal{G} \leq \mathcal{A}(m_{\text{max}} + 1)(m_{\text{max}} + 2)
\]

where \( m_{\text{max}} = \max_{a \in \mathcal{A}} m_a \).

5.2 Functions with Compact Support.

For \( t \in \mathbb{Z} \), let

\[
    f^{\text{Compact}}(t) = g(\varphi(t))
\]

where \( \varphi : \mathbb{Z} \rightarrow [0, 1] \) and \( g : [0, 1] \rightarrow \mathcal{F} \) is \( \mathcal{L} \)-Lipschitz.

**Proposition 5.3.** For any \( \varepsilon \in (0, 1) \),

(i) Under Model Type 1, \( f^{\text{Compact}} \) satisfies Property 4.1 with \( \delta_1 = \frac{2\pi}{L}, r = \mathcal{L} \).

(ii) Under Model Type 2, \( f^{\text{Compact}} \) satisfies Property 4.2 with \( \delta_2 = 2\delta_1 \sqrt{\mathcal{N}} \) and \( C_{p(k)} = 1 \) for \( k \in \{L\} \).

Using Proposition 5.3, Theorems 4.1 and 4.2 immediately lead to the following corollaries.

**Corollary 5.3.** Under Model Type 1, let the conditions of Theorem 4.1 hold. Let \( N = L^{1+\delta} \) for any \( \delta > 0 \). Then for some \( C > 0 \) and any \( \varepsilon \in (0, 1) \) if

\[
    T \geq \left( \frac{1}{\delta_{\text{error}}} \right)^{1/2} + \left( \frac{L}{\delta_{\text{error}}} \right)^{1/2}
\]

we have \( \text{MSE}(\hat{f}, f^{\text{LRF}}) \leq \delta_{\text{error}} \).

5.3 Finite sum of Sublinear Trends.

Consider \( f^{\text{Trend}}(t) \) such that

\[
    \frac{df^{\text{Trend}}(t)}{dt} \leq C_{t} t^{-\alpha}
\]

for some \( \alpha, C_{t} > 0 \).

**Proposition 5.4.** Let \( \left| \frac{df^{\text{Trend}}(t)}{dt} \right| \leq C_{t} t^{-\alpha} \) for some \( \alpha, C_{t} > 0 \). Then for any \( \varepsilon \in (0, 1) \),

(i) Under Model Type 1, \( f^{\text{Trend}} \) satisfies Property 4.1 with \( \delta_1 = \frac{C_{t}}{L^{1/\alpha}}, r = \mathcal{L}^{1/\alpha} + L^{1-L/\alpha} \).

(ii) Under Model Type 2, \( f^{\text{Trend}} \) satisfies Property 4.2 with \( \delta_2 = 2\delta_1 \sqrt{\mathcal{N}} \) and \( C_{p(k)} = 1 \) for \( k \in \{L\} \).

By Proposition 5.5 and Theorems 4.1 and 4.2, we immediately have the following corollaries on the finite sample performance guarantees of our estimators.

**Corollary 5.5.** Under Model Type 1, let the conditions of Theorem 4.1 hold. Let \( N = L^{1+\delta} \) for any \( \delta > 0 \). Then for some \( C > 0 \) if

\[
    T \geq C \left( \frac{1}{\delta_{\text{error}}} \right)^{1/2} + \left( \frac{L}{\delta_{\text{error}}} \right)^{1/2}
\]

we have \( \text{MSE}(\hat{f}, f^{\text{LRF}}) \leq \delta_{\text{error}} \).

**Corollary 5.6.** Under Model Type 2, let the conditions of Theorem 4.2 hold. Let \( N = L^{1+\delta} \) for any \( \delta > 0 \). Then for some \( C > 0 \) and for any \( \varepsilon \in (0, 1) \) if

\[
    T \geq C \left( \frac{\sigma^2}{\delta_{\text{error}} - \mathcal{C} \left( \frac{2 \pi}{L} + \delta_3 \right)^{1/2}} \right)^{2+\delta}
\]

we have \( \text{MSE}(\hat{f}, f^{\text{LRF}}) \leq \delta_{\text{error}} \).

The “fundamental period” \( \mathcal{N} \) of \( \{\omega_1, \ldots, \omega_q\} \) is the smallest value such that \( x_{\mathcal{N}}(g_a/p_a) \) is an integer for all \( a \in \mathcal{A} \) and \( \mathcal{L} \) is the least common multiple (LCM) of \( \{p_1, \ldots, p_c\} \). Rewriting \( S = \{q_a \cdot p_m/p_a : g \in G\} \), \( G = \mathcal{N} \cdot \mathcal{N}_{\text{m}} \), we have the set of numerators, \( \{q_a \cdot p_m/p_a : g \in G\} \) are all integers and we define their LCM as \( \mathcal{L}_{m} \). It is easy to verify that \( \mathcal{L}_{m} = \mathcal{N} \cdot \mathcal{N}_{m} \) is indeed a fundamental period. As an example, consider \( x = \{n, n/2, n/3, \ldots, n/n-1\} \), in which case the above computation results in \( \mathcal{N}_{m} = n \).
we have $\text{MSE}(\hat{f}, f^{\text{LRF}}) \leq \delta_{\text{error}}$.

**Proposition 5.6.** For $t \in Z$ with $\alpha_b < 1$ for $b \in [B]$, 
\[ f^{\text{Trend}}(t) = \sum_{b=1}^{B} \gamma_b \alpha_b + \sum_{q=1}^{Q} \log(\gamma_q t). \]

admits a representation as in (3).

**5.4 Additive Mixture of Dynamics.**

We now show that the imputation results hold even when we consider an additive mixture of any of the models described above. For $t \in Z$, let 
\[ f^{\text{Mixture}}(t) = \sum_{q=1}^{Q} \rho_q f_q(t). \]

Here, each $f_q$ is such that under Model Type 1 with $E[X(t)] = f_q(t)$, Property 4.1 is satisfied with $\delta_1 = \delta_q$ and $r = r_q$ for $q \in [Q]$.

**Proposition 5.7.** Under Model Type 1, $f^{\text{Mixture}}$ satisfies Property 4.1 with $\delta_1 = \sum_{q=1}^{Q} \rho_q \delta_q$ and $r = \sum_{q=1}^{Q} r_q$.

Proposition 5.7 and Corollary 4.1 imply the following.

**Corollary 5.7.** Under Model Type 1, let the conditions of Theorem 4.1 hold. For each $q \in [Q]$, let $\delta_q \leq C_q \epsilon^{-\delta}$ and $r_q = o(L)$ for some $\epsilon, C_q > 0$. Then, $\lim_{T \to \infty} \text{MSE}(\hat{f}, f^{\text{Mixture}}) = 0$.

In summary, Corollaries 5.1, 5.3, 5.5 and 5.7 imply that for any additive mixture of time series dynamics coming from $f^{\text{LRF}}$, $f^{\text{Compact}}$ and $f^{\text{Trend}}$, the algorithm in Section 3.2 produces a consistent estimator for an appropriate choice of $L$.

**5.5 Hidden State.**

A common problem of interest is to uncover the hidden dynamics of latent variables given noisy observations. For example, consider the problem of estimating the true weekly demand rate of umbrellas at Target given its weekly sales of umbrellas. This can be mathematically described as uncovering the underlying parameters of a time varying Poisson process whose samples are the weekly sales reports, (cf. [5]). In general, previous methods to learn the hidden states either require multiple time series as inputs or require that the underlying noise model is known (refer to Section 1.2 for a detailed overview).

In contrast, by viewing $f(t)$ as the time-varying latent variables (see Figure 2), we are well equipped to handle more generic noise distributions and complicated hidden dynamics. Specifically, our imputation and forecast algorithms can uncover the latent dynamics if: (i) per-step noise has bounded variance (additive noise is needed for forecasting); (ii) $E[X(t)] = f(t)$. Moreover, our algorithm is model and noise agnostic, robust to missing entries, and comes with strong theoretical consistency guarantees (Theorems 4.1 and 4.2). Given these findings, our approach is likely to become a useful gadget in the toolkit for dealing with scenarios pertinent to uncovering latent states a la Hidden Markov-like models. We corroborate our findings through experiments in Section 6.

**5.6 Sample complexity.**

As discussed, our algorithm operates for a large class of models – it is not tailored for a specific model class (e.g. sum of harmonics). In particular, for a variety of model classes, our algorithm provides consistent estimation for imputation while the forecasting MSE scales with the quality of the matrix estimation algorithm (i.e. $\delta_1$). Naturally, it is expected that to achieve accurate performance, the number of samples (i.e. $T$) required will scale relatively poorly compared to model specific optimal algorithms. Corollaries 5.1 to 5.6 provide finite sample analysis, which quantify this "performance loss", which indicate that this performance loss is minor. As an example, consider imputation for any periodic time series with periods between $\left[ n \right]$. By Proposition 5.2, it is easy to see that the order, $G$, of such a time series is $2n$. Thus corollary 5.1 indicates that the MSE goes to 0 with $T \sim n^{2+\delta}$ for any $\delta > 0$ as $n \to \infty$. For such a time series, one expects such a result to require $T \sim n \log n$ even for a model aware optimal algorithm.

**6 EXPERIMENTS**

We conduct experiments on real-world and synthetic datasets to study the imputation and forecasting performance of our algorithm for mixtures of time series processes, as we vary the amount of missing data. Additionally, we present the applicability of our algorithm to the hidden state setting (see Section 5.5).

**Mixtures of Time Series Processes.** For the synthetically generated datasets, we utilize mixtures of harmonics, trend, and auto-regressive (AR) processes with Gaussian additive noise – since AR is effectively a noisy version of a LRF. When using real-world datasets, we do not know the underlying time series processes – however, they appear to display periodicity, trend, and auto-regression.

**Comparisons.** For forecasting, we compare our algorithm to the state-of-the-art time series forecasting library of R, which decomposes a time series into stationary auto-regressive, seasonal, and trend components. The library learns each component separately and combines them to produce forecasts. Given that our synthetic and real-world datasets involve additive mixtures of these processes, this serves as a strong baseline to compare our algorithm against. We note that we do not outperform optimal model-aware methods for single model classes with all data present, at least as implemented in R-package. However, these methods are not necessarily optimal with missing data and/or when data is generated by mixture of multiple model types, which is the setting in which we see our model agnostic method outperform. For imputation experiments, we compare our algorithm against AMELIA II ([29]), which is another R-based package widely believed to produce excellent...
imputation performance.

**Metric of Evaluation.** Our metric of comparison is the root mean-squared error (RMSE).

**Algorithmic Parameters.** For both imputation and forecasting, we apply the Singular Value Thresholding (SVT) algorithm used in [18]. We use a data-driven approach to choose the singular value threshold, $\mu$, and the number of rows in the time series matrix, $L$, in our algorithm. Specifically, we reserve 30% of our training data for cross-validation to pick $\mu$ and $L$.

**Summary of Results.** Details of all experiments are provided below. Recall $p$ is the probability of observation of each datapoint.

**Synthetic Data -** For forecasting, we determine the forecast RMSE of our algorithm vs. R’s forecast library (see above for how synthetic was generated). Our experimental results show we outperform the R’s forecast library, especially under high levels of missing data and noise. For imputation, we outperform the imputation library AMELIA under all levels of missing data.

**Real-World Data -** We test against two real world datasets: (i) Bitcoin price dataset from March 2016 at 30s intervals; (ii) Google flu trends data for Peru from 2003-2012. In both cases, we introduce randomly missing data and then use our algorithm and R’s forecast library to forecast into the future. Corroborating the results from the synthetic data experiments, our algorithm’s forecast RMSE is better than the R library’s.

**Hidden State Model -** We produce a time series according to a Poisson process with time-varying parameters, which are hidden. These parameters evolve according to a mixture of time series processes, i.e. sum of harmonics and trends. Our interest is in estimating these time-varying hidden parameters, using one realization of integer observations of which several are randomly missing. For $p$ ranging from 0.3 to 1.0 the imputation RMSE is always $< 0.2$ while the $R^2$ is always $> 0.8$, which should be considered excellent, thus illustrating the versatility of our algorithm in solving a diverse set of problems.

### 6.1 Synthetically Generated Data.

We generate a mixture process of harmonics, trend and auto-regressive components. The first 70% of the data points are used to learn a model (training) and point-predictions, i.e. forecasts, are performed on the remaining 30% of the data. In order to study the impact of missing data, each entry in the training set is observed independently with probability $p \in (0, 1]$.

**Forecasts.** Figures 3a-3c visually depict the predictions from our algorithm when compared to the state-of-the-art time series forecasting library in R. We provide the R library the number of lags of the AR component to search over, in effect making its job easier. It is noticeable that the forecasts from the R library always experience higher variance. As $p$ becomes smaller, the R library’s forecasts also contain an apparent bias. These visual findings are confirmed in Figure 4b which shows that our algorithm produces better RMSE’s than the R forecasting library when working with mixtures of AR, harmonic and trend processes; our algorithm’s RMSE ranges from $[0.03, 0.11]$ vs $[0.09, 0.16]$ for R’s forecasting library.

**Imputation.** Figure 4a shows that our algorithm outperforms the state-of-the-art AMELIA library for multiple time series imputation for all levels of missing data. The RMSE of our algorithm ranged between $[0.09, 0.13]$ vs $[0.14, 0.24]$ for AMELIA. Note that AMELIA is much better than the baseline, i.e. imputing all missing entries with the mean.

Note that this experiment involved multiple time series where the outcome variable of interest and the log of its squared power were also included. The additional time series components were included to help AMELIA impute missing values because it is unable to impute missing entries in a single time series. However, our algorithm did not use these additional time series, and instead just used the original time series with missing, noisy observations.

### 6.2 Real-World Data.

We use two real-world datasets to evaluate the performance of our algorithm in situations where the identities of the time series
processes are unknown. This set of experiments is intended to highlight the versatility of our algorithm and applicability to practical scenarios involving time series forecasting. Note that for these datasets, we do not know the true mean processes. Therefore, it is not possible to generate the metric of interest, i.e. RMSE, using the means. Instead, we use the observations themselves as the reference to compute the metric.

**Bitcoin.** Figures 5a and 5b show the forecasts for Bitcoin prices (in Yuans) in March 2016 at regular 30s time intervals which demonstrates classical auto-regressive properties. We provide a week’s data to learn and forecast over the next two days. Figure 5a shows that our algorithm and the R library appear to do an excellent job of predicting the future even with 50% data missing. Figure 5b shows the RMSE of the predictions as a function of $p$ indicating outperformance of our algorithm under all levels of missing data; our algorithm had RMSE’s in range [0.8, 1.75] vs [0.9, 2.6] for the R library, with $p$ ranging from 1.0 to 0.5 (note prices are not normalized).

**Google Flu Trends (Peru).** Figures 6a and 6b show the forecasts for Google flu search-trends in Peru which shows significant seasonality. We provide weekly data from 2003-2012 to learn and then forecast for each week in the next three years. Figure 6a shows that our algorithm outperforms R when predicting the future with 30% data missing. Figure 6b shows the RMSE of the predictions as a function of $p$ indicating outperformance of our algorithm under all levels of missing data; our algorithm had RMSE’s in range [8.0, 17.5] vs [9.0, 26.0] for the R library, with $p$ ranging from 1.0 to 0.5 (note prices are not normalized).

### 6.3 Hidden State.

We generate a time series from a Poisson process with time-varying parameters, which are hidden. These parameters evolve according to a mixture of sums of harmonics and logarithmic trend components. Each $X(t)$ is then observed independently with probability $p$, to produce a random variable $Y(t)$. We normalize all parameters and observations to lie in $[-1, 1]$. Observe that $E[Y(t)] = p\lambda(t)$. Note that this is similar to the settings described earlier in this work but instead of additive noise we have imposed a generic noise model. However, our goal...
is the same: estimate the mean time series process under randomly missing data profiles.

Figures 7a-7b show the mean time series process can be estimated via imputation using the algorithm proposed in our work. These two plots show the original time series (with randomly missing data points set to 0), the true means and our estimation. With only 1% missing data, our algorithm is able to impute the means accurately with the performance degrading slightly with 10% missing data. However, note that these are relatively small datasets with only 25,000 points. Figure 7d shows the same process under 10% missing data but for 50,000 data points. As expected, with access to a greater number of data points, our algorithm performs better.

Figure 7c shows plots of RMSE and $R^2$ for the imputed means of the process. Note these apply to the smaller time series of 25,000 data points. The metrics are computed only on the data points which were missing. We note that the $R^2$ rises while the RMSE falls as $p$ increases. Both of these profiles confirm our intuition that the imputation improves as a function of $p$. However, note that performance is fairly robust (RMSE < 0.2 and $R^2 > 0.8$) for all levels of missing data plotted.

Figure 7: Imputation of the means of an Poisson time series. The first three plots are for the time series which has 25,000 data points and the resulting matrix is of dimensions: $50 \times 500$. The last one is for the same process but with twice as much data and matrix dimensions of $100 \times 500$. Note that the observations randomly missing are set to 0 which the entire process is normalized to lie in the range $[-1, 1]$. 
7 CONCLUSION

In this paper, we introduce a novel algorithm for time series imputation and forecasting using matrix estimation methods, which allows us to operate in a model and a noise agnostic setting. Further, we identify generic conditions on the time series model class under which the algorithm provides consistent estimation. We establish that many popular model classes and their mixtures satisfy these generic conditions. Using synthetic and real-world data, we exhibit the efficacy of our algorithm with respect to the state-of-art software implementation available through R. Our finite sample analysis agrees with these experimental results. Lastly, we demonstrate that our method can provably recover the hidden state of dynamics, which could be of interest in its own right.

REFERENCES

[1] Emmanuel Abbe and Colin Sandon. 2015. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on. IEEE, 670–688.

[2] Emmanuel Abbe and Colin Sandon. 2015. Recovering communities in the general stochastic block model without knowing the parameters. In Advances in neural information processing systems.

[3] Emmanuel Abbe and Colin Sandon. 2016. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap. Advances in neural information processing systems (2016).

[4] Edo M Airoldi, Thago Costa, and Stanley H Chan. 2013. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In Advances in Neural Information Processing Systems. 692–700.

[5] Muhammad J Amjad and Devavrat Shah. 2017. Censored Demand Estimation in Retail. Proceedings of the ACM on Measurement and Analysis of Computing Systems 1, 2 (2017), 31.

[6] Muhammad Jehangir Amjad, Devavrat Shah, and Dennis Shen. 2017. Robust synthetic control. arXiv preprint arXiv:1711.06940 (2017).

[7] Animashree Anandkumar, Rong Ge, Daniel Hsu, and Sham Kakade. 2013. A tensor spectral approach to learning mixed membership community models. In Conference on Learning Theory. 867–881.

[8] Oren Anava, Eldad Hazan, and Assaf Zeevi. 2015. Online Time Series Prediction with Missing Data. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15). Blei and Francis Bach (Eds.) JMLR Workshop and Conference Proceedings, 2191–2199. http://jmlr.org/proceedings/papers/v37/anava15.pdf

[9] Leonard E Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state Markov chains. The Annals of Mathematical Statistics 37, 6 (1966), 1545–1563.

[10] Alexandre Belloni, Matthew Rosenbaum, and Alexandre B Tsybakov. 2017. Linerar and non-linear programming estimators in high dimensional errors-in-variables models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79, 3 (2017), 935–956.

[11] Sergei Bernstein. 1946. The Theory of Probabilities. Gostehizdat Publishing House.

[12] Dimitris Bertsimas, David Gamarnik, and John N Tsitsiklis. 1999. Estimation of time-varying parameters in statistical models: an optimization approach. Machine Learning 35, 3 (1999), 225–245.

[13] Christian Borgs, Jennifer Chayes, Christina E Lee, and Devavrat Shah. 2017. Thy Friend is My Friend. Iterative Collaborative Filtering for Sparse Matrix Estimation. In Advances in Neural Information Processing Systems. 4718–4728.

[14] Christian Borgs, Jennifer T Chayes, Henry Cohn, and Shashua and Ganguly. 2015. Consistent nonparametric estimation for heavy-tailed sparse graphs. arXiv preprint arXiv:1508.06675 (2015).

[15] Jenkins Box and Reinsel. 1994. Time Series Analysis, Forecasting and Control (3rd ed.). Prentice Hall, Englewood Cliff, NJ.

[16] Peter J Brockwell and Richard A Davis. 2013. Time series: theory and methods. Springer Science & Business Media.

[17] Emmanuel J Candes and Terence Tao. 2010. The power of convex relaxation: Near-optimal matrix completion. IEEE Transactions on Information Theory 56, 5 (2010), 2053–2080.

[18] Sourav Chatterjee. 2015. Matrix estimation by universal singular value thresholding. The Annals of Statistics 43, 1 (2015), 177–214.

[19] Yudong Chen and Martin J Wainwright. 2015. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. arXiv preprint arXiv:1509.03025 (2015).

[20] Zhe Chen and Andrej Cichocki. 2005. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. In Laboratory for Advanced Brain Signal Processing, RIKEN. Tech. Rep.

[21] Thomas M Cover. 1966. BEHAVIOR OF SEQUENTIAL PREDICTORS OF BINARY SEQUENCEs. Technical Report. DTIC Document.

[22] Abharrup Datta, Hui Zou, et al. 2017. Cocolo for high-dimensional error-in-variables regression. The Annals of Statistics 45, 6 (2017), 2409–2426.

[23] Mark A Davenport, Yaniv Plan, Ewout van den Berg, and Mary Wootters. 2014. 1-bit matrix completion. Information and Inference 3, 3 (2014), 189–223.

[24] William Dunsmuir and PM Robinson. 1981. Estimation of time series models in the presence of missing data. J Amer. Statist. Assoc. 76, 375 (1981), 560–568.

[25] James Durbin and Siem Jan Koopman. 2012. Time series analysis by state space methods. Vol. 38. OUP Oxford.

[26] Mein Feder, Neri Merhav, and Michael Gutman. 1992. Universal prediction of individual sequences. Information Theory, IEEE Transactions on 38, 4 (1992), 1258–1270.

[27] Nina Golyandina, Vladimir Nekrutkin, and Anatoly A Zhigljavsky. 2001. Analysis of time series structure: SSA and related techniques. Chapman and Hall/CRC.

[28] James Douglas Hamilton. 1994. Time series analysis. Vol. 2. Princeton university press Princeton.

[29] James Honaker, Gary King, and Matthew Blackwell. 2015. AMELIA II: A Program for Missing Data. https://cran.r-project.org/web/packages/Amelia/vignettes/amelia.pdf

[30] Samuel B Hopkins and David Steurer. 2017. Efficient Bayesian estimation from few samples: community detection and related problems. In Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on. IEEE, 379–390.

[31] Ludolph Emil Kalman et al. 1960. A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82, 1 (1960), 35–45.

[32] Raghunandan H Keshavan, Andrea Montanari, and Seshong Oh. 2010. Matrix completion from a few entries. IEEE Transactions on Information Theory 56, 6 (2010), 2900–2938.

[33] Raghunandan H Keshavan, Andrea Montanari, and Seshong Oh. 2010. Matrix completion from noisy entries. Journal of Machine Learning Research 11, Jul (2010), 2057–2078.

[34] Christina E Lee, Yihua Li, Devavrat Shah, and Dogyoon Song. 2016. Blind regression: Nonparametric Regression for Latent Variable Models via Collaborative Filtering. In Advances in Neural Information Processing Systems 29: 2155–2163.

[35] Sahand Negahban and Martin J Wainwright. 2011. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. The Annals of Statistics (2011), 1069–1097.

[36] Loh Po-lin and Martin J Wainwright. 2012. High-dimensional regression with noisy and missing data: provable guarantees without convexity. The Annals of Statistics 40 (2012), 1637–1664.

[37] Swati Rallapalli, Lili Qiu, Yin Zhang, and Yi-Chao Chen. 2010. Exploiting temporal stability and low-rank structure for localization in mobile networks. In Proceedings of the sixteenth annual international conference on Mobile computing and networking. ACM, 161–172.

[38] Benjamin Recht. 2011. A simpler approach to matrix completion. Journal of Machine Learning Research 12, Dec (2011), 3413–3430.

[39] Jörgen Schönfelder. 1992. Learning complex, extended sequences using the principle of history compression. Neural Computation 4, 2 (1992), 234–242.

[40] David H Schoolenheimer. 2001. Singular spectrum analysis for time series with missing data. Geophysical Research Letters 28, 16 (2001), 3187–3190.

[41] Y Shen, F Peng, and B Li. 2015. Improved singular spectrum analysis for time series with missing data. Nonlinear Processes in Geophysics 22, 4 (2015), 371–376.

[42] Paul C Shields. 1998. The interactions between ergodic theory and information theory. In IEEE Transactions on Information Theory. Citeseer.

[43] Robert H Shumway and David S Stoffer. 1982. An approach to time series smoothing and forecasting using the EM algorithm. Journal of time series analysis 3, 4 (1982), 253–264.

[44] Grigorios Tsagatakis, Balafern Beferru-Lorano, and Panagiotis Tsakalides. 2016. Singular spectrum-based matrix completion for time series recovery and prediction. EURASIP Journal on Advances in Signal Processing 2016, 1 (2016), 66.

[45] Christopher Xie, Alex Talk, and Emily Fox. 2015. A Unified Framework for Missing Data and Cold Start Prediction for Time Series Data. In Advances in Neural Information Processing Systems 28: 847–855.

[46] Yuan Zhang, Elizaveta Levina, and Ji Zhu. 2015. Estimating network edge probabilities by neighborhood smoothing. Information and Inference: A Journal of the IMA 4, 3 (2015), 351–379.
A USEFUL THEOREMS

Theorem A.1. Bernstein’s Inequality. [11]
Suppose that X₁, . . . , Xₙ are independent random variables with zero mean, and M is a constant such that |Xᵢ| ≤ M with probability one for each i. Let S := Σᵢ=1ⁿ Xᵢ and ν := Var(S). Then for any t ≥ 0,
\[ \mathbb{P}(|S| ≥ t) ≤ 2 \exp \left( - \frac{3t^2}{2ν^2} \right). \]

Theorem A.2. Theorem 3.4 of [18]
Take any two numbers m and n such that 1 ≤ m ≤ n. Suppose that A = [Aᵢⱼ]₁≤ₖ≤m,₁≤ₖ≤n is a matrix whose entries are independent random variables that satisfy, for some δ² ∈ (0, 1],
\[ E[|Aᵢⱼ|^2] ≤ δ², \quad \text{and} \quad |Aᵢⱼ| ≤ 1 \quad \text{a.s.} \]
Suppose that δ² ≥ n⁻¹⁺ζ for some ζ > 0. Then, for any η ∈ (0, 1),
\[ \mathbb{P}(|\|A\||₂ ≥ (2 + η)δ√n) ≤ C(ζ)e⁻ᶜδ²n, \]
where C(ζ) depends only on η and ζ, and c depends only on η. The same result is true when m = n and A is symmetric or skew-symmetric, with independent entries on and above the diagonal, all other assumptions remaining the same. Lastly, all results remain true if the assumption δ² ≥ n⁻¹⁺ζ is changed to δ² ≥ n⁻¹⁺(log n)²⁺ζ.

Remark A.1. The proof of Theorem A.2 can be found in [18] under Theorem 3.4.

B IMPOSITION ANALYSIS

Lemma B.1. Let X = [Xᵢⱼ]₁≤ₖ≤L,₁≤ₖ≤N with |Xᵢⱼ| ≤ 1 and Var(Xᵢⱼ) ≤ σ². Similarly, let M = [Μᵢⱼ]₁≤ₖ≤L,₁≤ₖ≤N with |Μᵢⱼ| ≤ 1. Let 1 ≤ L ≤ N. Define Y = [Yᵢⱼ]₁≤ₖ≤L,₁≤ₖ≤N where
\[ Yᵢⱼ = \begin{cases} Xᵢⱼ & \text{w.p. } p, \\ 0 & \text{w.p. } 1 - p. \end{cases} \]
Let \( \hat{p} = \max \left( \frac{1}{N^L} \sum_{i=1}^{L} \sum_{j=1}^{N} 1_{Xᵢⱼ}, \frac{1}{N^L} \right) \). Define E₁ and E₂ as
\[ E₁ := \{ |\hat{p} - p| ≤ kp/20 \}, \]
\[ E₂ := \{ \|Y - pM\|₂ ≤ (2 + k/2)√(Np) \}, \]
for some choice of k ∈ (0, 1) where p ≥ N⁻¹⁺ζ for some ζ > 0. Then,
\[ \mathbb{P}(E₁) ≥ 1 - 2e⁻𝒄L²Np - (1 - p)^L²N, \]
\[ \mathbb{P}(E₂) ≥ 1 - C₁e⁻ᶜL²Np, \]
for some positive constants c₁, c₂, and c₃ that depend on k.

Proof. Under this model, we have that
\[ \mathbb{E}[Yᵢⱼ] = pMᵢⱼ. \]
As a result,
\[ \text{Var}(Yᵢⱼ) ≤ \mathbb{E}[Yᵢⱼ^2] = pE[Xᵢⱼ^2] ≤ p. \]
Let \( \hat{p}_0 = \frac{1}{\sqrt{N}} \sum_{i=1}^{T} \sum_{j=1}^{N} 1_{Xᵢⱼ}, \) observed, which implies \( \hat{p}_0 = p. \) We define the event \( E₃ := \{ \hat{p}_0 = \hat{p} \}. \) Thus, we have that
\[ \mathbb{P}(E₃^c) = \mathbb{P}(E₃^c \cap E₁) + \mathbb{P}(E₃^c \cap E₂) \]
\[ = \mathbb{P}(\hat{p}_0 - p| ≥ kp/20) + \mathbb{P}(E₃^c \cap E₂) \]
\[ ≤ \mathbb{P}(\hat{p}_0 - p| ≥ kp/20) + \mathbb{P}(E₃^c) \]
where the final equality follows by the independence of observations assumption and the fact that \( \hat{p}_0 \neq \hat{p} \) only if we do not have any observations. By Bernstein’s Inequality, we have that
\[ \mathbb{P}(\hat{p}_0 - p| ≤ kp/20) ≥ 1 - 2e⁻ᶜL²Np, \]
for an appropriately defined constant c₁ that depends on κ. By Theorem A.2,
\[ \mathbb{P}(E₂) ≥ 1 - C₂e⁻ᶜ₂L²Np, \]
for positive constants C₂ and c₂, which depend on κ.

Corollary B.1. Let \( L = E₁ \cap E₂. \) Then,
\[ \mathbb{P}(E₁^c) ≤ C₁e⁻ᶜL²N, \]
\[ \mathbb{P}(E₂^c) ≤ C₁e⁻ᶜL²N, \]
where C₁ and c₂ are positive constants independent of L and N.

Proof. By DeMorgan’s Law and the Union Bound, we have that
\[ \mathbb{P}(E₁^c) = \mathbb{P}(E₁^c \cup E₂^c) \]
\[ ≤ \mathbb{P}(E₁^c) + \mathbb{P}(E₂^c) \]
\[ ≤ C₁e⁻ᶜL²N, \]
where C₁, c₂ > 0 are appropriately defined and depend on κ, but are independent of L and N.

Lemma B.2. Let \( M^{(1)} \) be defined as in Section 4.1 and satisfy Property 4.1. Then,
\[ \|M^{(1)}\|₁ ≤ L\sqrt{N}δ₁ + \sqrt{N}L. \]

Proof. By the definition of \( M^{(1)} \) and the triangle inequality property of nuclear norms,
\[ \|M^{(1)}\|₁ ≤ \|M^{(1)} - M(r)\|₁ + \|M(r)\|₁ \]
\[ ≤ \sqrt{N} \|M^{(1)} - M(r)\|₁ + \|M(r)\|₁ \]
\[ ≤ \sqrt{N} \|M(r)\|₁ \]
Note that (a) makes use of the fact that \( \|Q\|₁ ≤ \sqrt{\text{rank}(Q)}\|Q\|₁ \) for any real-valued matrix Q and (b) utilizes Property 4.1 since rank\( (M(r)) = r \), we have \( \|M(r)\|₁ ≤ \sqrt{N}\|L\|₁ \).

Theorem 4.1. Assume Property 4.1 holds and ME satisfies Property 2.1, with \( p ≥ \max(L, N)^{-1+ζ} \) for some ζ ∈ [0, 1]. Then for some \( C₁, C₂, C₃, c₄ > 0 \),
\[ \text{MSE}(\hat{f}, f) ≤ C₁ \left( \frac{δ₁}{\sqrt{p}} + \sqrt{\frac{T}{pL}} + C₂ \frac{(1 - p)}{LNp} + C₃ e⁻ᶜ₄N. \right) \]

Proof. Observe that we can rewrite \( \text{MSE}(\hat{f}, f) \) in the following manner:
\[ \text{MSE}(\hat{f}, f) := E \left[ \sum_{t=1}^{T} (\hat{f}(t) - f(t))^2 \right] = E \left[ \frac{1}{T} \|M^{(1)} - M^{(1)}\|₁ \right] \]

(10)
For notational simplicity, let us drop the superscripts on \( \hat{M}^{(1)} \) and \( M^{(1)} \). Let \( E := E_1 \cap E_2 \), where \( E_1 \) and \( E_2 \) are defined as in Lemma B.1. By the law of total probability and \( \mathbb{P}(E) \leq 1 \), we have that

\[
\mathbb{E}\left[ \| \hat{M} - M \|_F^2 \right] \leq \mathbb{E}\left[ \| \hat{M} - M \|_F^2 \mid E \right] + \mathbb{E}\left[ \| \hat{M} - M \|_F^2 \mid E^c \right] \mathbb{P}(E^c). \tag{11}
\]

We begin by bounding the first term on the right-hand side of (11). By Property 2.1 and assuming \( \mathcal{E} \) occurs, we have that

\[
\期待 \| \hat{\beta} \hat{M} - pM \|_F^2 \leq C_1\| Y - pM \|_2^2\| pM \|_2,
\]

where \( C_2 \) depends on the choice of \( \kappa \) (defined in Lemma B.1). Therefore,

\[
\| \hat{\beta} \hat{M} - pM \|_F^2 \leq C_1\| \hat{\beta} \hat{M} - pM \|_F^2 + C_3\| \hat{\beta} - \beta \|_2^2\| \hat{M} \|_2,
\]

\[
\leq C_1\| \hat{\beta} \hat{M} - pM \|_F^2 + C_3\| \hat{\beta} - \beta \|_2^2\| M \|_2^2 + C_3\| \hat{\beta} - \beta \|_2^2L^2N,
\]

for an appropriately defined \( C_4 \). Observe that \( \| \hat{\beta} - \beta \|_2^2 = p(1 - np/L)N \). Combining the above results and using Corollary B.1, we obtain

\[
\mathbb{E}\left[ \| \hat{M} - M \|_F^2 \right] \leq C_4N^{1/2}p^{-1/2}\| M \|_\infty + C_3p^{-1}(1 - np/LN) + C_4n\mathbb{e}^{-c_3N}.
\]

Normalizing by \( T = LN \) gives

\[
\text{MSE}(\hat{f}_T, f) \leq \frac{C_4\| M \|_\infty}{L\sqrt{T}N} + \frac{C_3(1 - np/LN)}{pLN} + C_4n\mathbb{e}^{-c_3N}.
\]

We invoke Lemma 2.2 to bound \( \| M \|_\infty \), to obtain

\[
\text{MSE}(\hat{f}_T, f) \leq \frac{C_4\sqrt{N}\sqrt{\kappa_1} + \sqrt{\kappa_2}LN}{L\sqrt{T}N} + \frac{C_3(1 - np/LN)}{pLN} + C_4n\mathbb{e}^{-c_3N} = C_4\left( \sqrt{\frac{1}{\beta}} + \sqrt{\frac{T}{pL}} \right) + \frac{C_3(1 - np/LN)}{pLN} + C_4n\mathbb{e}^{-c_3N},
\]

for appropriately redefined \( C_1, C_2, C_3, \) and \( C_4 > 0 \).

\[ \square \]

C. Forecasting Analysis

Let us begin by analyzing the forecasting error for any \( k \in [L] \).

Lemma C.1. For each \( k \in [L] \), assume Property 4.2 holds and \( ME(k) \) satisfies Property 2.2. Then,

\[
\mathbb{E}\left[ \left( \hat{f}_T(t) - f(t) \right)^2 \right] \leq (\delta_2 + \sqrt{C_3N\delta_2^2})^2 + 2\sigma^2 \hat{\kappa}_k. \tag{12}
\]

Here, \( S_k : = \{ t \in [T] : (t \mod L) + 1 = k \} \) and \( \hat{\kappa}_k := \text{rank}(M^{(k)}) \).

Proof. Observe that

\[
\mathbb{E}\left[ \left( \hat{f}_T(t) - f(t) \right)^2 \right] = \mathbb{E}\left[ \sum_{t \in S_k} \left( \hat{f}_T(t) - f(t) \right)^2 \right]. \tag{13}
\]

For notational simplicity, let \( \hat{Q} := (\hat{M}^{(k)})^T \) and \( \hat{Q} := (\hat{M})^T \). Similarly, we will drop all superscripts \( (k) \) throughout this analysis for notational ease. Recall \( X_L = M_L + \epsilon_L \). Then note that by the definition of the optimization in step 2 of the forecast algorithm,

\[
\left\| X_L - \hat{Q} \hat{\beta} \right\|_2^2 \leq \left\| X_L - \hat{Q} \hat{\beta}_L \right\|_2^2 = \left\| M_L - \hat{Q} \hat{\beta}_L \right\|_2^2 + \left\| \epsilon_L \right\|_2^2 + 2\epsilon_L^T (M_L - \hat{Q} \hat{\beta}_L). \tag{14}
\]

Moreover,

\[
\left\| X_L - \hat{Q} \hat{\beta} \right\|_2^2 \leq \left\| M_L - \hat{Q} \hat{\beta}_L \right\|_2^2 + \left\| \epsilon_L \right\|_2^2 + 2\epsilon_L^T (\hat{Q} \hat{\beta}_L - M_L). \tag{15}
\]

Combining (14) and (15) and taking expectations, we have

\[
\mathbb{E}\left[ \left\| X_L - \hat{Q} \hat{\beta} \right\|_2^2 \right] \leq \mathbb{E}\left[ \left\| M_L - \hat{Q} \hat{\beta}_L \right\|_2^2 \right] + 2\mathbb{E}[\epsilon_L^T (\hat{Q} \hat{\beta}_L - M_L)]. \tag{16}
\]

Let us bound the final term on the right-hand side of (16). Under our independence assumptions, observe that

\[
\mathbb{E}[\epsilon_L^T (\hat{Q} \hat{\beta}_L - M_L)] = \mathbb{E}[\epsilon_L^T (\hat{Q} \hat{\beta}_L - M_L)]. \tag{17}
\]

Recall \( \hat{\beta} = \hat{Q}^T X_L = \hat{Q}^T M_L + \hat{Q}^T \epsilon_L \). Using the cyclic and linearity properties of the trace operator (coupled with similar independence arguments), we further have

\[
\mathbb{E}[\epsilon_L^T (\hat{Q} \hat{\beta})] = \mathbb{E}[\epsilon_L^T (\hat{Q} \hat{\beta} - M_L)] = \mathbb{E}[\epsilon_L^T (\hat{Q} \hat{\beta} - M_L)] = \mathbb{E}[\epsilon_L^T (\hat{Q} \hat{\beta} - M_L)] = \sigma^2 \mathbb{E}\left[ \text{Tr}\left( \hat{Q} \hat{\beta} - M_L \right) \right]. \tag{18}
\]

Let \( \hat{Q} = USV^T \) be the singular value decomposition of \( \hat{Q} \). Then

\[
\hat{Q}^T \hat{Q} = USV^T \hat{U} \hat{V}^T = \hat{U} \hat{V}^T. \tag{19}
\]

Here, \( \hat{I} \) is a block diagonal matrix where its nonzero entries on the diagonal take the value 1. Plugging in (19) into (18), and using the fact that the trace of a square matrix is equal to the sum of its eigenvalues,

\[
\sigma^2 \mathbb{E}\left[ \text{Tr}\left( \hat{Q} \hat{\beta} - M_L \right) \right] = \sigma^2 \mathbb{E}[\text{rank}(\hat{Q})]. \tag{20}
\]

We now turn our attention to the first term on the right-hand side of (16). By Property 4.2, we obtain

\[
\left\| M_L - \hat{Q} \hat{\beta}_L \right\|_2^2 = \left\| M_L - (Q - \hat{Q}) \hat{\beta}_L \right\|_2^2 \geq \left\| M_L - \hat{Q} \hat{\beta}_L \right\|_2^2 + \left\| (Q - \hat{Q}) \hat{\beta}_L \right\|_2^2 
\].

Thus we have that

\[
\mathbb{E}\left[ \left\| (Q - \hat{Q}) \hat{\beta} \right\|_2^2 \right] \leq \mathbb{E}\left[ \left\| (M - \hat{M}) \hat{\beta} \right\|_2^2 \right] \tag{21}
\]

\[
\leq \sum_{t=1}^L \left\| \hat{\beta}_t \right\|_2 \mathbb{E}\left[ \left( \sum_{j=1}^L (\hat{M}_{ij} - M_{ij})^2 \right)^{1/2} \right] \tag{22}
\]

\[
\leq \left\| \hat{\beta}_t \right\|_2 \max_{1 \leq t < \hat{L}} \mathbb{E}\left[ \left( \sum_{j=1}^L (\hat{M}_{ij} - M_{ij})^2 \right)^{1/2} \right] \tag{23}
\]
Putting everything together, we obtain our desired result.  \( \square \)

**Theorem (4.2).** Assume Property 4.2 holds and ME satisfies Property 2.2, with \( p \geq p^*(L, N) \). Let \( \hat{f} := \max_{k \in [L]} \text{rank}(M^{(k)}) \). Then,

\[
\text{MSE} (\hat{f}_F, f) \leq \frac{1}{N-1} (\delta_2 + \sqrt{C \beta N \delta_1})^2 + 2\sigma^2 \hat{r}.
\]

**Proof.** For simplicity, define \( \delta(k) := (\delta_2 + \sqrt{C \beta N \delta_1})^2 + 2\sigma^2 \hat{r} \). By Lemma C.1, for all \( k \in [L] \) we have

\[
E \left[ \sum_{t \in S_k} \left( f_F(t) - f(t) \right)^2 \right] \leq \delta(k).
\]

Let \( \delta_{\max} := (\delta_2 + \sqrt{C \beta N \delta_1})^2 + 2\sigma^2 \hat{r} \). Recall \( S_k := \{ t \mid T + (t \mod L) + 1 = k \} \). Then, it follows that

\[
\text{MSE} (\hat{f}_F, f) \leq \frac{\delta_{\max}}{N-1}.
\]

**D MODEL ANALYSIS**

We first define a somewhat technical Property D.1, which will aid us in proving that the various models in Section 5 satisfy Property 4.1 and 4.2. Define \( \eta : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \) such that

\[
\eta(\theta_i, \rho_j) := M(i + (j - 1)L + (k - 1)), \quad \theta_i = i, \quad \rho_j = (j - 1)L + (k - 1).
\]

We thus have \( [M_i^{(k)}] = [\eta(\theta_i, \rho_j)] \) by construction of \( M^{(k)} \). Effectively, the latent parameters \( \theta_i, \rho_j \) encode the amount of shift in the argument to \( M(t) \), so as to obtain the appropriate entry in the matrix, \( M^{(k)} \).

**Property D.1.** Let matrices \( X^{(k)} \) and \( M^{(k)} \) satisfy the following:

**A.** For each \( i \in [L], j \in [N] \):

1. \( X^{(k)}_{ij} \) are independent across \( i, j \), and each have variance at most \( \sigma^2 \).
2. \( X^{(k)}_{ij} \) is observed with probability \( p \in (0, 1] \), independently.

**B.** There exists \( p : [L] \rightarrow [L] \) such that:

1. Define \( P(\theta) := \{ p(\theta) : i \in [L] \} \). Let it be such that \( |P(\theta)| = r_4 < L \).
2. Define \( L \times N \) matrix \( M(r) \), as \( [M^{(r)}] = [\eta(p(\theta_i), \rho_j)] \). Then for \( \delta_4 \geq 0 \),

\[
\|M^{(k)} - M(r)\|_{\max} \leq \delta_4.
\]

We begin with Proposition D.1, which motivates the use of linear methods in forecasting.

**Proposition D.1.** Let \( M \) be defined as in Section 4.1 and obey Property D.1. Then there exists \( \beta^* \) such that

\[
\|M_L^{(k)} - (\hat{M})_T^T \beta^*\|_2 \leq 2\delta_4 \sqrt{N},
\]

where \( \|\beta^*\|_0 = 1 \).

**Proof.** We drop the superscript \( k \) from \( M^{(k)} \) for notational convenience. Furthermore, we prove it for the case of \( k = 1 \) since the proofs for general \( k \) follow identical arguments by first making an appropriate shift in the entries of the matrix of interest. Assume we have access to data from \( X[1:T + r_4 - 1] \). Let us first construct a matrix with overlapping entries, \( \hat{M} = [\hat{M}_{ij}] = [ f(i + j - 1) \], of dimension \( L \times (T + r_4 - 1) \). We have \( \hat{M}_{ij} = \eta(\hat{\theta}_i, \hat{\rho}_j) \) with \( \hat{\theta}_i = i \) and \( \hat{\rho}_j = [j - 1, y] \), where \( \eta \) is as defined in (26). By construction, the skew-diagonal entries from left to right of \( \hat{M} \) are constant, i.e.,

\[
\hat{M}_{k+i,j} := \{ \hat{M}_{k+i,j} : 1 \leq k \neq j \leq L, 1 \leq i + j \leq T + r_4 - 1 \}.
\]

Under this setting, we note that the columns of \( M \) are subsets of the columns of \( \hat{M} \). Specifically, for all \( 0 \leq j < N \) and \( k \leq L \),

\[
\hat{M}_{k+i,j} := \{ \hat{M}_{k+i,j} : 1 \leq k \neq j \leq L, 1 \leq i + j \leq T + r_4 - 1 \}.
\]

By construction, observe that every entry within \( \hat{M} \) exists within \( M \). Hence, \( \hat{M}_{ij} = M_{i', j'} \) for some \( i', j' \), and

\[
|\eta(\hat{\theta}_i, \hat{\rho}_j) - \eta(\hat{\theta}_i', \hat{\rho}_j')| = |\eta(\hat{\theta}_i', \hat{\rho}_j') - \eta(p(\hat{\theta}_i), \hat{\rho}_j')| \leq \|\hat{M} - M(r)\|_{\max} \leq \delta_4,
\]

where the inequality follows from Condition B.2 of Property D.1. In light of this, just as we defined \( M(r) \) with respect to \( \hat{M} \), we define \( \hat{M}(r) \) from \( M \). Specifically, the \( (i, j) \)-th element of \( \hat{M}(r) \) is

\[
\hat{M}_{ij}(r) = \eta(p(\hat{\theta}_i), \hat{\rho}_j).
\]

By Condition B.1 of Property D.1 and applying the Pigeonhole Principle, we observe that within the last \( r_4 + 1 \) rows of \( M^{(r)} \), at least two rows are identical. Without loss of generality, let these two rows be denoted as \( M^{(r)}_{L-r_4,i} = [M^{(r)}_{L-r_4,i}]_{1 \leq i \leq N} \) and \( M^{(r)}_{L-r_2,i} = [M^{(r)}_{L-r_2,i}]_{1 \leq i \leq N} \), respectively, where \( r_4 \in \{1, \ldots, r_4 - 1\} \), \( r_2 \in \{2, \ldots, r_4\} \), and \( r_1 < r_2 \). Since \( \hat{\theta}_i = i \), \( \hat{\rho}_j \), we trivially have that \( p(\hat{\theta}_i') = p(\hat{\theta}_i) \). Consequently, it must be the case that the same two rows in \( M^{(r)} \) are also identical, i.e. for all \( i \leq T + r_4 - 1 \),

\[
\| \hat{M}^{(r)}_{L-r_4,i} - \hat{M}^{(r)}_{L-r_2,i} \|_{\max} \leq \delta_4.
\]

Using this fact, we have that for all \( i \leq T + r_4 - 1 \),

\[
\|M_{L-r_4,i} - \hat{M}_{L-r_4,i}\|_{\max} \leq \|M_{L-r_4,i} - \hat{M}^{(r)}_{L-r_4,i}\|_{\max} + \|\hat{M}^{(r)}_{L-r_4,i} - \hat{M}^{(r)}_{L-r_2,i}\|_{\max} \leq 2\delta_4.
\]

where the last inequality follows from (30) and the construction of \( \hat{M}(r) \). Additionally, by the skew-diagonal property of \( \hat{M} \) as described above by (27), we necessarily have the following two equalities:

\[
\|M_{L-r_4,i} - \hat{M}_{L-r_4,i} - \hat{M}_{L-r_2,r_4+i}\|_{\max} \leq \delta_4.
\]

where \( \Delta_r = r_2 - r_1 \). Thus, by (31), (33), and (34), we obtain for all \( i \leq T \),

\[
\|M_{L-i} - \hat{M}_{L-\Delta_r,i} - \hat{M}_{L-r_2,r_4+i}\|_{\max} \leq \delta_4.
\]
Thus, applying (28) and (35), we reach our desired result, i.e. for all \( i \leq N \),
\[
| M_{Li} - M_{L - A_r, i} | \leq 2\delta_i. 
\]
(36)
Recall \( \bar{M} = [M_{ij}]_{1 \leq i, j \leq N} \) excludes the last row of \( M \). From above, we know that there exists some row \( \ell := L - A_r < L \) such that \( \| M_{L} - M_{\ell} \|_2 \leq 2\delta_i \sqrt{N} \). Clearly, we can express
\[
M_{\ell} = \bar{M}^T \beta^*,
\]
where \( \beta^* \in \mathbb{R}^{L-1} \) is a 1-sparse vector with a single nonzero component of value 1 in the \( \ell \)th index. This completes the proof.
\[\square\]

**Corollary D.1.** Let \( M^{(k)} \) be defined as in Section 4.1 and obey Property D.1 with \( \delta_i, r_k \). Then \( M^{(k)} \) obeys,

(i) Under Model Type 1, Property 4.1 is satisfied with \( \delta_1 = \delta_4, r = \delta_4 \).

(ii) Under Model Type 2, Property 4.2 is satisfied with \( \delta_2 = 2\delta_4 \sqrt{N} \).

**Proof.** Condition A of both Property 4.1 and 4.2 is satisfied by definition. (i) Condition B.1, B.2 of Property D.1 together imply Condition B of Property 4.1 for the same \( \delta_1, r_k \). (ii) Proposition D.1 implies Condition B of Property 4.2 by scaling \( \delta_4 \) with \( 2\sqrt{N} \). \[\square\]

**D.1 Proof of Proposition 5.1**

**Proposition (5.1),**

(i) Under Model Type 1, \( f^{LRF} \) satisfies Property 4.1 with \( \delta_1 = 0, r = G \).

(ii) Under Model Type 2, \( f^{LRF} \) satisfies Property 4.2 with \( \delta_2 = 0 \) and \( C_p = C \cdot G \) where \( C \) is an absolute constant.

**Proof.** Let \( f(t) = f^{LRF} \). By definition of \( f(t) \), we have that for \( \forall i \in G + 1, \ldots , L, j \in 1, \ldots , N \),
\[
M^{(k)}_{ij} = f(i + (j - 1)L) + (k - 1) = \sum_{g=1}^{G} \alpha_g f((i-g) + (j - 1)L) + (k - 1)) = \sum_{g=1}^{G} \alpha_g M^{(k)}_{i-g, j}. 
\]
In particular, \( M^{(k)}_{ij} = \sum_{g=1}^{G} \alpha_g M^{(k)}_{L - g, j} \) \( \forall j \in 1, \ldots , N \), and so we immediately have condition (ii) of the Proposition with \( C = \max_{g \in G} \alpha_g \). Since every row from \( G + 1, \ldots , L \) is a linear combination of the rows above, the rank of \( M^{(k)} \) is at most \( G \). Ergo, we have condition (i) of the Proposition. \[\square\]

**Corollary (5.1).** Under Model Type 1, let the conditions of Theorem 4.1 hold. Let \( N = L^{1+\delta} \) for any \( \delta > 0 \). Then for some \( C > 0 \), if
\[
T \geq C \left( \frac{G}{\delta_{error}} \right)^{2+\delta},
\]
we have \( \text{MSE}(f^{LRF}) \leq \delta_{error} \).

**Proof.** By Proposition 5.1, we have for some \( C_1, C_2, C_3, c_4 > 0 \)
\[
\text{MSE}(\hat{f}, f^{LRF}) \leq C_1 \sqrt{\frac{G}{pL}} + C_2 \frac{(1-p)}{LNp} + c_4 e^{-\delta_{error} N}. 
\]
We require the r.h.s of the term above to be less than \( \delta_{error} \). We have,
\[
C_1 \sqrt{\frac{G}{pL}} + C_2 \frac{1-p}{LNp} + c_4 e^{-\delta_{error} N} \overset{(a)}{\leq} C \left( \sqrt{\frac{G}{T}} + \frac{1}{LN} + e^{-\delta_{error} N} \right) \overset{(b)}{\leq} C \left( \frac{G}{\delta_{error}} \right)^{2+\delta}.
\]
where (a) follows for appropriately defined \( C > 0 \) and by absorbing \( p \) into the constant; (b) follows since \( \frac{1}{LN} \leq \frac{G}{T} \) for sufficiently large \( L, N \), and by redefining \( C \).

Hence it suffices that \( \delta_{error} \geq C \left( \frac{G}{\delta_{error}} \right)^{2+\delta} \).
\[\square\]

**D.2 Proof of Proposition 5.2**

**Proposition (5.2).** Let \( P_{max} \) be a polynomial of degree \( m_a \). Then,
\[
f(t) = \sum_{a=1}^{A} \exp(\alpha_a t) \cos(2\pi \omega_a t + \phi_a) P_{max}(t)
\]
admits a representation as in (1). Further the order \( G \) of \( f(t) \) is independent of \( T \), the number of observations, and is bounded by
\[
G \leq A(m_{max} + 1)(m_{max} + 2)
\]
where \( m_{max} = \max_{a \in A} m_a \).

**Proof.** This proof is adapted from [27]. We state it here for completeness.
First, observe that if there exists latent functions $\psi_l : \{1, \ldots, L\} \to \mathbb{R}$ and $\rho_l : \{1, \ldots, N\} \to \mathbb{R}$ for $l \in [G]$ such that we can write,

$$f(i + j) = \sum_{l=1}^{G} \psi_l(i) \rho_l(j), \quad i \in [L], j \in [N]$$

then each $M(k)$ (induced by $f$ for $k \in [L]$) has rank at most $G$.

Second, observe that time series that admit a representation of the form in (38) form a linear space which is closed with respect to term-by-term multiplication, i.e.,

$$f(i + j) = f^{(1)} \circ f^{(2)} = \left( \sum_{l=1}^{G} \psi_l^{(1)}(i) \rho_l^{(1)}(j) \right) \left( \sum_{l=1}^{G} \psi_l^{(2)}(i) \rho_l^{(2)}(j) \right)$$

where $G_1$ and $G_2$ are the orders of the $f^{(1)}$ and $f^{(2)}$ respectively.

Given the two observations above, it suffices to show separately that $f^{(1)}(t) = \exp(at) \cos(2\pi \omega t + \phi)$ and $f^{(2)}(t) = P_m(t)$ have a representation of the form in (38).

We begin with $f^{(1)}(t) = \exp(at) \cos(2\pi \omega t + \phi)$. For $i \in [L], j \in [N]$,

$$f^{(1)}(i + j) = \exp[(a(i + j)] \cos[2\pi \omega (i + j) + \phi]$$

where $(a)$ we have used the trigonometric identity $\cos(a + b) = \cos(a) \cos(b) - \sin(a) \sin(b)$. Thus for $f^{(1)}(t), G = 2$.

For $f^{(2)}(t) = P_m(t)$, with $i \in [L], j \in [N]$, we have $P_m(i + j) = \sum_{l=1}^{m} c_l(i + j)$. By expanding $(i + j)^L$, it is easily seen (using the Binomial theorem) that there are $L + 1$ unique terms involving powers of $i$ and $j$. Hence for $f^{(2)}(t), G = \sum_{l=1}^{L+1} 1 = \frac{(m+1)(m+2)}{2}$.

Now we bound $G$ for $(t) = \sum_{a=1}^{A} \exp[\alpha_a t] \cos(2\pi \omega_a t + \phi_a)P_m(t)$.

For $f^{(1)}(t) = \exp(at) \cos(2\pi \omega t + \phi)$, we have $G^{(1)} = 2$. For $f^{(2)}(t) = P_m(t), G^{(2)} = L$. Let $G^{(3)} = \min(L, \min_j (m+1)(m+2))$. Since there are $A$ different terms, it follows immediately that for $f(t), G = \min(G^{(1)}, G^{(2)})$, which completes the proof.

D.3 Proof of Proposition 5.3

Proposition 5.3. For any $\epsilon \in (0, 1)$,

(i) Under Model Type 1, $f^{\text{compact}}$ satisfies Property 4.1 with $\delta_1 = \frac{L}{\epsilon}$, $r = L^\epsilon$.

(ii) Under Model Type 2, $f^{\text{compact}}$ satisfies Property 4.2 with $\delta_2 = 2\delta_1 \sqrt{N}$ and $C_2 = \frac{\sqrt{L}}{\sqrt{\tilde{L}}}$.\footnote{To build intuition, consider $f(t) = t^2$, in which case $f(i + j) = i^2 + j^2 + 2(ij) = \psi(i)\rho(i) + \psi(j)\rho(j) = \psi(i)\rho(j) + \psi(j)\rho(i)$, here, $G = 3$.}

Proof. Recall that $\eta$ as defined in (26) has row and column parameters $\{\eta_0, \ldots, \eta_{T}\}$ and $\eta_0, \ldots, \eta_{N}\}$, which denote shifts in the time index. Let $\tilde{R}$ and $\tilde{D}$ refer to the set of row and column parameters respectively. Since $f \in \mathcal{G}(\eta(t))$ where $\eta : \mathbb{Z} \to [0, L]$ and $\mathcal{G}$ is $L$-Lipschitz, rather than considering the set of time steps $\{1, \ldots, T\}$, it suffices to consider instead the set $\{g(\eta(0)), \ldots, g(\eta(T))\} \subseteq [0, \tilde{L}]$. Because $\tilde{R}$ is simply an index to a time step, it is sufficient to consider an alternate, compact set of row parameters, $\tilde{R}$, where $\theta_i \in \tilde{R} \subseteq [0, L]$. Crucially, we highlight that $\tilde{R}$ is independent of $L$.

From here onwards, the arguments we make follow directly from arguments in [18]. We provide it here for completeness. For any $\delta > 0$, we first define a partition $\mathcal{P}(\delta)$ of $\tilde{R}$ where the following holds: for any $A \in \mathcal{P}(\delta)$, whenever $\theta, \theta'$ are two points such that $\theta, \theta' \in A$, we have $|\eta(\theta) - \eta(\theta')| \leq \delta$ for all $j \in [N]$.

Due to the Lipschitz property of the function $\eta$ (with Lipschitz constant $\tilde{L}$) and the compactness of $[0, x_{\text{term}}]$, it can be shown that $|\eta(\theta)| \leq x_{\text{term}}L^{-1}$.

Let $T$ be a subset of $\tilde{R}$ that is constructed by selecting exactly one element from each partition in $\mathcal{P}(\delta)$, i.e. $|T| = |\mathcal{P}(\delta)|$. Let $p : [L] \to \{0, \ldots, \tilde{L} - 1\}$ be the mapping from $\tilde{R}$ to $T$. Therefore, it follows that for each $\theta \in \tilde{R}$, we can find $\theta(\theta) \in T$ so that $\theta$ and $\theta(\theta)$ belong to the same partition of $\mathcal{P}(\delta)$. Let $M(\theta)$ be the matrix whose $(i,j)$th element is $f(\theta(i), \theta(j))$. Then by construction, for $k \in [K]$, we have

$$\left\| M(k) - M(\theta) \right\|_{\text{max}} \leq \delta.$$

Now, if $\theta_i$ and $\theta_j$ belong to the same element of $\mathcal{P}(\delta)$, then $\theta(i)$ and $\theta(j)$ are identical. Therefore, there are at most $|\mathcal{P}(\delta)|$ distinct rows in $M(\theta)$.

Let $\mathcal{P}(\theta) := \{\theta(i) : i \in [L]\} \subseteq \{\theta_0, \ldots, \theta_{N}\}$. By construction, we have $|\mathcal{P}(\theta)| = |\mathcal{P}(\delta)|$. Choosing $\delta = L\tilde{L}^{-\epsilon}$, then $|\mathcal{P}(\theta)| = L^\epsilon$.

Hence Property D.1 is satisfied with $\delta_1 = \frac{L}{\epsilon}, r = L^\epsilon$. By Corollary D.1, we have: under Model Type 1, Property 4.1 is satisfied with $\delta_1 = \delta_2 = 2\delta_1 \sqrt{N}$. This completes the proof. $\square$

Corollary (5.3). Under Model Type 1, let the conditions of Theorem 4.1 hold. Let $N = L^{1+\delta}$ for any $\delta > 0$. Then for some $C > 0$ and any $\epsilon \in (0, 1)$ if

$$T \geq \left( \frac{1}{\delta_{\text{error}}} \right)^{1/2} + \left( \frac{L}{\delta_{\text{error}}} \right)^{1/2} + \frac{1}{2},$$

we have $\text{MSE}(\tilde{f}_1, f^{\text{LRF}}) \leq \delta_{\text{error}}$.

Proof. By Proposition 5.3, for any $\epsilon \in (0, 1)$ and some $C_1, C_2, C_3, C_4 > 0$,

$$\text{MSE}(\tilde{f}_1, f^{\text{compact}}) \leq \frac{C_1}{L} \left( \frac{C_2}{\sqrt{\tilde{L}}} \right)^{1/2} \left( \frac{C_3}{L^2} \right)^{1/2} \leq \frac{C_4}{L^2} \leq C_{\text{error}} N.$$
require $L$ to be
\[
L \geq C \left( \frac{1}{\delta_{\text{error}}} \right)^2 + \left( \frac{L}{\delta_{\text{error}}} \right)^{1/2},
\]
\[
\Rightarrow T \geq C \left( \frac{1}{\delta_{\text{error}}} \right)^2 + \left( \frac{L}{\delta_{\text{error}}} \right)^{1/2},
\]

Corollary (5.4). Under Model Type 2, let the conditions of Theorem 4.2 hold. Let $N = L^{1+\delta}$ for any $\delta > 0$. Then for some $C > 0$ and any $\epsilon \in (0, 1)$ if
\[
T \geq C \left( \frac{1}{\delta_{\text{error}}} \right)^{2\alpha} \left( \frac{L}{\delta_{\text{error}}} \right)^{1/2},
\]
we have $\text{MSE}(\hat{f}_{fc}, f^{\text{LRF}}) \leq \delta_{\text{error}}$.

Proof. By Proposition 5.3, for any $\epsilon \in (0, 1)$,
\[
\text{MSE}(\hat{f}_{fc}, f^{\text{Compact}}) \leq \frac{1}{N-1} \left( \frac{L}{L^*} + \delta_3 \right)^2 N + 2\sigma^2 \right).$

We require the r.h.s. of the term above to be less than $\delta_{\text{error}}$. Since $\frac{1}{N^{\sigma^2} \epsilon^2} \leq \frac{1}{L^*} \sigma^2$, it suffices that
\[
\delta_{\text{error}} \geq C \left( \frac{L_1 + \delta_3}{L} \right)^2 + \frac{1}{L^*} \sigma^2
\]
\[
\Rightarrow L^* \leq C \left( \delta_{\text{error}} - \left( \frac{L_1 + \delta_3}{L} \right)^2 \right)^{-1} \sigma^2
\]
\[
\Rightarrow T \geq C \left( \delta_{\text{error}} - \left( \frac{L_1 + \delta_3}{L} \right)^2 \right)^{1/2},
\]
where (a) and (b) follow for appropriately defined $C > 0$. □

Proposition (5.4).
\[
f^{\text{Harmonic}}(t) = \sum_{g=1}^{G} \varphi_g \sin(2\pi \omega_g t + \phi)
\]
where $\varphi_g$ is $L_g$-Lipschitz and $\omega_g$ is rational, admit a representation as in (2). Let $x_{\text{lcm}}$ denote the fundamental period. Then the Lipschitz constant, $L$, of $f^{\text{Harmonic}}(t)$ is bounded by
\[
L \leq 2\pi \cdot \max_{g \in G} (L_g) \cdot \max_{g \in G} (\omega_g) \cdot x_{\text{lcm}}.
\]

Proof. That $f^{\text{Harmonic}}$ has a representation as in (2) is immediate. It remains to show the explicit dependence of $L$ on the parameters of $f^{\text{Harmonic}}$. Observe that
\[
f^{\text{Harmonic}}(t) = f^{\text{Harmonic}}(\psi(t))
\]
where $\psi(t) = t \mod x_{\text{lcm}}$.

By bounding the derivative of $f^{\text{Harmonic}}(t)$, it is easy to see that
\[
L \leq 2\pi \cdot \max_{g \in G} (L_g) \cdot \max_{g \in G} (\omega_g) \cdot x_{\text{lcm}}.
\]

D.4 Proof of Proposition 5.5

Proposition (5.5). Let $|f^{\text{Trend}}(t)| \leq C_\alpha t^{-\alpha}$ for some $\alpha, C_\alpha > 0$.

Then for any $\epsilon \in (0, \alpha)$,

(i) Under Model Type 1, $f^{\text{Trend}}$ satisfies Property 4.1 with $\delta_1 = \frac{C_\alpha}{L^* \epsilon^{1/\alpha}}$, $r = L^\epsilon / \alpha + L_r - L^\epsilon / \alpha$.

(ii) Under Model Type 2, $f^{\text{Trend}}$ satisfies Property 4.2 with $\delta_2 = 2\delta_1 \sqrt{N}$ and $C_\beta = 1$.

Proof. Let $f(t) = f^{\text{Trend}}$. We construct our mapping $p : [L] \rightarrow [L]$ in two steps:

Step 1: For $j < L^\epsilon / \alpha$, with $\epsilon \in (0, \alpha)$, let $p(j) = j$ (i.e. $j$-th row of $M_{\nu_j}$) be equal to the $j$-th row of $M(k)$.

Step 2: For $j \geq L^\epsilon / \alpha$, the mapping we construct is an adaptation of the argument in [18]. Let $R$ and $D$ refer to the set of row and column parameters of the sub-matrix of $M(k)$ corresponding to its last $L - j + 1$ rows, $\{L^\epsilon / \alpha \cdots 1\}$ and $\{1 \cdots p_j\}$ respectively.

Through a straightforward application of the Mean Value Theorem, observing that the derivative is decreasing in $t$, and the condition $|\frac{df(t)}{dt}| \leq C_\alpha t^{-\alpha}$, we have that for all $\theta_1, \theta_2 \in [R]$ (with appropriately defined constants)
\[
|\eta(\theta_1, \rho_j) - \eta(\theta_2, \rho_j)| \leq \eta'(|\theta_1 - \theta_2|)
\]
\[
\leq f'(L^\epsilon / \alpha) \cdot |\theta_1 - \theta_2|
\]
\[
\leq C_\alpha (L^\epsilon / \alpha)^{-\alpha} \cdot |\theta_1 - \theta_2|
\]
\[
\leq C_\alpha L^{-\alpha} \cdot |\theta_1 - \theta_2|,
\]
where $\eta'$ and $f'$ are the derivatives with respect to $\theta$ and $t$, respectively, and $\delta_3 \in (\theta_1, \theta_2)$.

Define a partition, $P(e)$, of $R$ into continuous intervals of length $L^\epsilon / \alpha$. Observe that since $\theta_1 = i$, for any $A \in P(e)$, whenever $\theta_2 \in A$, we have $|\theta_2 - \theta_1| \leq L^\epsilon / \alpha$. It follows that $|P(e)| = (L - L^\epsilon / \alpha) / L^\epsilon / \alpha = L^{1-\epsilon / \alpha} = L(1 - 1 / \alpha)$.

Let $T$ be a subset of $R$ that is constructed by selecting exactly one element from each partition in $P(e)$. That is, $|T| = |P(e)|$. For each $\theta \in R$, let $p(\theta)$ be the corresponding element from the same partition in $T$. Therefore, it follows that for each $\theta \in R$, we can find $p(\theta) \in T$ so that $\theta$ and $p(\theta)$ belong to the same partition of $P(e)$.

Hence, we can define the $(i,j)$-th element of $M_{ij}$ in the following way: (1) for all $j < L^\epsilon / \alpha$, let $p(\theta_j) = \theta_j$ such that $M_{ij} = \eta(\theta_j, \rho_j)$; (2) for $j \geq L^\epsilon / \alpha$, let $M_{ij} = \eta(\theta_j, \rho_j)$. Consequently $\forall k$,
\[
\|M(k) - M(e)\| \leq \max_{i \in [L]} |\eta(\theta_j, \rho_j) - \eta(\theta_i, \rho_j)|
\]
\[
\leq \max_{i \in [L]} |\eta(\theta_j, \rho_j) - \eta(\theta_i, \rho_j)|
\]
\[
\leq \max_{i \in [L]} |\theta_i - \rho_j| L^{-\alpha} C_\alpha
\]
\[
\leq C_\alpha L^{-\epsilon / \alpha}.
\]

Now, if $\theta_j$ and $\theta_i$ belong to the same element of $P(e)$, then $p(\theta_j)$ and $p(\theta_i)$ are identical. Therefore, there are at most $|P(e)|$ distinct rows in the last $L - L^\epsilon / \alpha$ rows of $M(e)$ where $|P(e)| = L^{1-\epsilon / \alpha} = L(1 - 1 / \alpha)$. Let $P(\theta) := \{p(\theta_i) : i \in [L]\} \subset \{\theta_1, \ldots, \theta_L\}$. By construction, since $\epsilon \in (0, \alpha)$, we have that $|P(\theta)| = L^\epsilon / \alpha + |P(e)| = o(L)$. 
Hence Property D.1 is satisfied with \( \delta_1 = \frac{C}{L^{\gamma_1}}, \gamma = L^{\gamma/2} + L^{\gamma/2} \). By Corollary D.1, we have: under Model Type 1, Property 4.1 is satisfied with \( \delta_1 = \delta_4 \) and \( r = r_4 \); under Model Type 2, Property 4.2 is satisfied with \( \delta_2 = 2\delta_1 \sqrt{N} \). This completes the proof.

**Corollary (5.5).** Under Model Type 1, let the conditions of Theorem 4.1 hold. Let \( N = L^{1+\delta} \) for any \( \delta > 0 \). Then for some \( C > 0 \) if

\[
T \geq C \left( \delta_{\text{error}} \right)^{2+\delta}
\]

we have \( \text{MSE}(\hat{f}_T, f^{\text{LRF}}) \leq \delta_{\text{error}} \).

**Proof.** By Proposition 5.5, for any \( \epsilon \in (0, \alpha) \) and some \( C_1, C_2, C_3, C_4 > 0 \),

\[
\text{MSE}(\hat{f}_T, f^{\text{Trend}}) \leq C_1 \left( \frac{L}{\epsilon} \right)^{1/2} \left( \frac{1}{\epsilon} \right)^{1/2} + C_2 \left( \frac{1}{L} \right)^{1/2} + C_3 e^{-\epsilon c_1 N}.
\]

We require the r.h.s of the term above to be less than \( \delta_{\text{error}} \). We have,

\[
C_1 \left( \frac{1}{\epsilon} \right)^{1/2} \left( \frac{1}{\epsilon} \right)^{1/2} + C_2 \left( \frac{1}{L} \right)^{1/2} + C_3 e^{-\epsilon c_1 N}
\]

(a) follows for appropriately defined \( C > 0 \) and by absorbing \( p \) into the constant; (b) follows since \( \frac{1}{L^{\gamma/2}} \leq \frac{1}{L^{\gamma/2}}, \quad e^{-\epsilon c_1 N} \leq \frac{1}{L^{\gamma/2}} \) for sufficiently large \( L \), and by redefining \( C \).

Setting \( \frac{1}{\epsilon} = \frac{\alpha}{\alpha_1} \), we get \( \epsilon = \frac{\alpha_1}{\alpha} < \alpha \), hence satisfying the condition that \( \epsilon \in (0, \alpha) \) in Proposition 5.5.

Hence it suffices that \( \delta_{\text{error}} \geq C \frac{\alpha}{\alpha_1} \Rightarrow T \geq C \left( \frac{\alpha}{\alpha_1} \right)^{2+\delta} \).

**Corollary (5.6).** Under Model Type 2, let the conditions of Theorem 4.2 hold. Let \( N = L^{1+\delta} \) for any \( \delta > 0 \). Then for some \( C > 0 \) and for any \( \epsilon \in (0, \alpha) \) if

\[
T \geq C \left( \delta_{\text{error}} \right)^{2+\delta}
\]

we have \( \text{MSE}(\hat{f}_T, f^{\text{LRF}}) \leq \delta_{\text{error}} \).

**Proof.** By Proposition 5.5, for any \( \epsilon \in (0, \alpha) \),

\[
\text{MSE}(\hat{f}_T, f^{\text{Trend}}) \leq \frac{1}{N-1} \left( \frac{C}{L^{\gamma/2}} + \frac{\alpha_1^2}{2} N + 2\sigma^2 r \right).
\]

We require the r.h.s of the term above to be less than \( \delta_{\text{error}} \). Since \( \frac{1}{N} \sigma^2 r \leq \frac{1}{L^{\gamma/2}} \), it suffices that

\[
\delta_{\text{error}} \geq C \left( \frac{1}{L^{\gamma/2}} + \frac{\alpha_1^2}{L^{\gamma/2}} \right)^{2+\delta}
\]

where (a) and (b) follow for appropriately defined \( C > 0 \).

**Proposition (5.6).** For \( t \in Z \) with \( \alpha_6 < 1 \) for \( b \in [B] \),

\[
f_{\text{Trend}}(t) = \sum_{b=1}^{B} \gamma_b a_b + \sum_{q=1}^{Q} \log(y_q t).
\]

admits a representation as in (3).

**Proof.** The proof is immediate from the definition of \( f_{\text{Trend}} \).

**D.5 Proof of Proposition 5.7**

**Proposition (5.7).** Under Model Type 1, \( f^{\text{Mixture}} \) satisfies Property 4.1 with \( \delta_1 = \sum_{q=1}^{Q} \rho_q \delta_q \) and \( r = \sum_{q=1}^{Q} r_q \).

**Proof.** Let \( M_y^{(1)} \) refer to the underlying mean matrix induced by each \( X_y(t) \). Similarly, as defined in Property 4.1, let \( M_{y,(r)} \) be the low rank matrix associated with \( M_y^{(1)} \). We have

\[
M^{(1)} = \sum_{g} \alpha_g M_y^{(1)}.
\]

Let us define \( M_{(r)} \) in the following way,

\[
M_{(r)} = \sum_{g} \alpha_g M_{y,(r)}.
\]

We then have \( \text{rank}(M_{(r)}) \leq \sum_{g} \alpha_g \), and

\[
\| M^{(1)} - M_{(r)} \|_{\max} = \sum_{g} \alpha_g \| M^{(1)} - \sum_{g} \alpha_g M_{y,(r)} \|_{\max} \leq \sum_{g} \alpha_g \| M^{(1)} - M_{y,(r)} \|_{\max} = \sum_{g} \alpha_g \delta_g.
\]

This completes the proof.