

Supervised Dynamic Dimension Reduction with Deep Neural Network

Zhanye Luo¹, Yuefeng Han², Xiufan Yu²

¹University of Chicago

²University of Notre Dame

zhanye@uchicago.edu, yuefeng.han@nd.edu, xiufan.yu@nd.edu

Abstract

This paper studies the problem of dimension reduction, tailored to improving time series forecasting with high-dimensional predictors. We propose a novel Supervised Deep Dynamic Principal component analysis (SDDP) framework that incorporates the target variable and lagged observations into the factor extraction process. Assisted by a temporal neural network, we construct target-aware predictors by scaling the original predictors in a supervised manner, with larger weights assigned to predictors with stronger forecasting power. A principal component analysis is then performed on the target-aware predictors to extract the estimated SDDP factors. This supervised factor extraction not only improves predictive accuracy in the downstream forecasting task but also yields more interpretable and target-specific latent factors. Building upon SDDP, we propose a factor-augmented nonlinear dynamic forecasting model that unifies a broad family of factor-model-based forecasting approaches. To further demonstrate the broader applicability of SDDP, we extend our studies to a more challenging scenario when the predictors are only partially observable. We validate the empirical performance of the proposed method on several real-world public datasets. The results show that our algorithm achieves notable improvements in forecasting accuracy compared to state-of-the-art methods.

Code — <https://github.com/hhxsxlzy/SDDP>

Introduction

Dimension reduction stands as a pivotal technique in modern data analysis to address the complexities arising from increasing data dimensionality in today’s data-rich environment (Van der Maaten, Postma, and Van den Herik 2007; Sánchez Sorzano, Vargas, and Montano 2014). The core idea of dimension reduction is to transform high-dimensional data into a lower-dimensional embedding while preserving essential informative features, aiming for improved model interpretability and enhanced computational efficiency. Among various dimension reduction techniques that have emerged over the decades, *factor models* (Bai and Ng 2002; Lam and Yao 2012) stand out as one of the most popular and widely adopted approaches. Factor models reduce dimensionality by capturing the underlying structure of

high-dimensional data through a smaller set of unobserved latent variables (a.k.a., factors), which are assumed to account for the commonality among the observed variables. In supervised learning tasks with a large number of predictors, factor models are often employed as a first step to extract useful predictive information from high-dimensional predictors before applying suitable learning algorithms (Bair et al. 2006; Paul et al. 2008; Fan, Xue, and Yao 2017; Yu et al. 2024; Chen, Han, and Yu 2025; Luo, Han, and Yu 2025).

In the context of high-dimensional time series analysis, one particularly influential method is the *diffusion-index forecasting model* (Bai and Ng 2006; Stock and Watson 2002a,b), a factor-augmented regression that applies principal component analysis (PCA) to estimate latent factors from high-dimensional predictors, which are then used as inputs in a linear regression model to forecast the target variable. Another important direction is the *sufficient forecasting approach* (Fan, Xue, and Yao 2017; Yu, Yao, and Xue 2022; Luo et al. 2022), which uses sufficient dimension reduction (SDR) techniques (Li 2018) to construct sufficient predictive indices for predicting the target variable. In this approach, PCA serves as an important first step to extract latent factors for subsequent estimation of the sufficient predictive indices. By contrast, classical SDR methods are designed to capture both linear and nonlinear relationships between predictors and the response variable in i.i.d. settings, as seen in foundational works such as (Li 1991; Cook and Li 2002; Lee, Li, and Chiaromonte 2013; Tang and Li 2024).

PCA is a well-established technique that has stood the test of time. Despite originating decades ago, it continues to serve as a cornerstone of many modern methods and applications, widely used and deeply valued for its simplicity, practicality, and interpretability. Nevertheless, traditional PCA operates under the assumption of a linear factor model and extracts components through a static, unsupervised decomposition of the data covariance matrix, which presents certain inherent limitations. First, *the linear assumption* overlooks potential nonlinear relationships that may underlie the data structures, thereby restricting its efficacy in capturing intricate data patterns (Yalcin and Amemiya 2001; Schölkopf, Smola, and Müller 1997; Hinton and Salakhutdinov 2006; Gu, Kelly, and Xiu 2021; Zhou, Han, and Yu 2025a). Second, *the static analysis*, focusing on the cross-sectional structure of a contemporaneous panel, neglects the

potential informational content embedded within the temporal dependencies of lagged observations, which can be particularly insightful in time series analysis (Bernanke, Boivin, and Eliasziw 2005; Ashraf et al. 2023; Gao and Tsay 2024). Third, *the unsupervised nature* directs its focus solely towards maximizing variance without considering the target variable. Consequently, this may lead to the overlooking of feature directions that, despite exhibiting low variance, are highly predictive for a specific task. These limitations motivate the development of more advanced methods and refined variants of PCA (Huang et al. 2022; Gao and Tsay 2024).

In this paper, we propose a Supervised Deep Dynamic PCA (SDDP) framework that efficiently constructs low-dimensional representations from high-dimensional predictors, specifically tailored for time series forecasting. Compared to traditional PCA, SDDP is nonlinear, dynamic, and supervised. SDDP factor extraction not only improves predictive accuracy in downstream forecasting tasks but also yields more interpretable and target-specific latent factors. Our contributions can be summarized as follows.

- SDDP explicitly incorporates the target variable and lagged observations into the training process, refining the factor extraction by aligning it more closely with the forecasting objective. We construct a panel of **target-aware predictors**, which scales the original predictors by their predictive power, specifically, with larger weights allocated to those predictors exhibiting stronger forecasting performance. By extracting factors in a **dynamic and supervised** manner, SDDP enables more effective factor extraction tailored for time series analysis.
- SDDP employs advanced deep learning architectures to effectively capture the **complex nonlinear relationships** and **temporal dependencies** inherent in the data. Through a nonlinear factor model and a temporal neural network, SDDP can identify intricate nonlinear patterns that conventional linear models often overlook, contributing to more accurate and insightful predictions.
- Building upon SDDP, we propose a **factor-augmented nonlinear dynamic forecasting model** that **unifies a broad family of factor-model-based forecasting approaches**. By varying the underlying factor structure and selecting different link functions within the forecasting equation, the SDDP-forecasting model subsumes the classical diffusion-index model (Stock and Watson 2002a,b) and various extensions thereof (Bair et al. 2006; Huang et al. 2022; Gao and Tsay 2024), as special cases.
- Furthermore, we extend SDDP in a different direction to accommodate scenarios where the predictors are only **partially observed**. With a minor adjustment to handle missing entries, SDDP can be **adapted to covariate completion tasks** and remains effective in extracting latent factors despite the incomplete observability of the predictors.

Problem Setup. Suppose we observe a time series dataset consisting of T temporal samples $\{(\mathbf{x}_t, y_t), 1 \leq t \leq T\}$, where $\mathbf{x}_t \in \mathbb{R}^N$ denotes the predictor vector and $y_t \in \mathbb{R}$ is the response variable. The goal is to extract supervised dynamic factors from the predictors that are most relevant for accurately forecasting the future response y_{T+h} .

Methodology

Nonlinear Factor Model

For a high-dimensional observed time series predictor vector $\mathbf{x}_t = (x_{1,t}, \dots, x_{N,t})^\top \in \mathbb{R}^N$, the objective is to forecast y_{t+h} at a horizon of h , using the available information $\{\mathbf{x}_j : j = 1, \dots, t\}$. Each component $x_{i,t}$ serves as a relevant but noisy proxy for the target, making it unlikely that a small subset alone can adequately capture the target’s underlying dynamics. However, using all predictors in a conventional multivariate regression suffers from the curse of dimensionality, often leading to overfitting in-sample and poor performance out-of-sample. A common solution is to impose a factor structure on the predictors and extract a lower-dimensional set of latent factors.

We adopt a nonlinear factor model and consider a factor-augmented nonlinear dynamic forecasting model: at time t , the covariates \mathbf{x}_t and the future target y_{t+h} satisfy

$$x_{i,t} = x_{i,t}^* + u_{i,t} = h_i^*(\mathbf{f}_t) + u_{i,t} = h_i^*(\mathbf{g}_t, \boldsymbol{\zeta}_t) + u_{i,t}, \quad (1)$$

$$y_{t+h} = \phi(\mathbf{g}_{t-q+1}, \dots, \mathbf{g}_t) + \epsilon_{t+h}, \quad (2)$$

where $\mathbf{f}_t = (\mathbf{g}_t^\top, \boldsymbol{\zeta}_t^\top)^\top \in \mathbb{R}^K$ denotes the full set of latent factors. Among them, $\mathbf{g}_t \in \mathbb{R}^{K_1}$ are the relevant factors directly related to the target y_{t+h} , while $\boldsymbol{\zeta}_t \in \mathbb{R}^{K-K_1}$ are irrelevant. The observed predictor $x_{i,t}$ consists of a common component $x_{i,t}^*$, modeled via a possibly nonlinear loading function $h_i^*(\cdot): \mathbb{R}^K \rightarrow \mathbb{R}$ applied to the latent factors, and an idiosyncratic noise term $u_{i,t}$. The noise vector $\mathbf{u}_t = (u_{1,t}, \dots, u_{N,t})$ is assumed to be uncorrelated with the forecasting process. The function $\phi(\cdot)$ captures the nonlinear relationship between the target y_{t+h} and the past q lags of the relevant factors \mathbf{g}_t . When $h_i^*(\cdot) = b_i$ is a linear mapping, model (1) reduces to a linear factor model, with $B = (b_1^\top, \dots, b_p^\top) \in \mathbb{R}^{N \times K}$ denoting the factor loading matrix.

The factor-augmented regression model (Bai and Ng 2006; Stock and Watson 2002a,b) can be viewed as a special case of (1)-(2) under linear specifications. In their formulation, all components of \mathbf{f}_t are assumed to have predictive power for y_{t+h} . In contrast, our framework assumes that only a subset \mathbf{g}_t is relevant to the target, which may better reflect practical scenarios. Compared with sufficient forecasting methods (Fan, Xue, and Yao 2017; Yu, Yao, and Xue 2022), model (2) incorporates temporal dependence by allowing the response y_{t+h} to depend on the past q periods of the latent factors, and by permitting a nonlinear mapping $h_i^*(\cdot)$ from the factors to the covariates.

In the non-time-series setting with $q = 1$, model (2) has been extensively studied in the literature on sufficient dimension reduction. Building on this model, Cook and Li (2002); Lee, Li, and Chiaromonte (2013); Tang and Li (2024) formulated the goal of sufficient mean dimension reduction as identifying a transformation $\psi: \mathbb{R}^N \rightarrow \mathbb{R}^{K_1}$, possibly linear or nonlinear, such that $y_{t+h} \perp\!\!\!\perp \mathbf{x}_t \mid \psi(\mathbf{x}_t)$, where $\perp\!\!\!\perp$ denotes statistical independence. That is, the conditional distribution of y_{t+h} given \mathbf{x}_t depends only on $\psi(\mathbf{x}_t)$. In this sense, our model can be interpreted as a dynamic extension of sufficient dimension reduction to the time series setting.

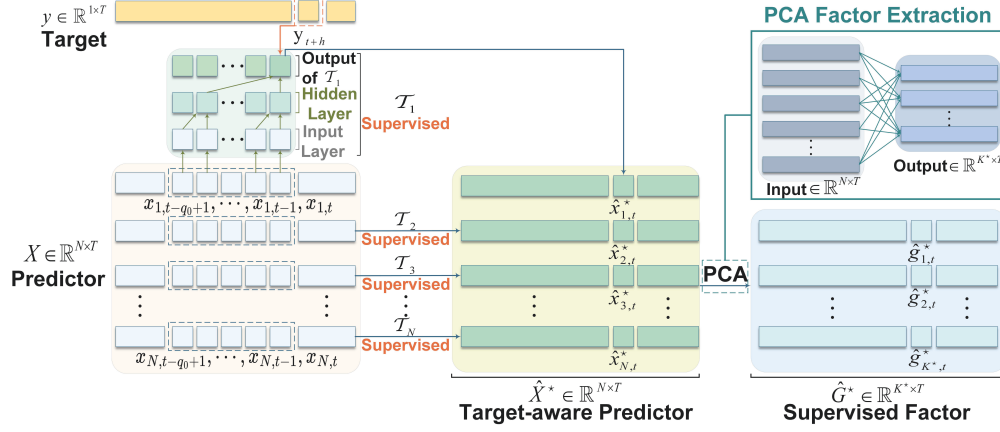


Figure 1: An illustration of the proposed Supervised Deep Dynamic PCA (SDDP) algorithm. Inputs are observed predictors $X = (\mathbf{x}_1, \dots, \mathbf{x}_T) \in \mathbb{R}^{N \times T}$, target $y = (y_1, \dots, y_T)^\top \in \mathbb{R}^T$. Output is the estimated supervised dynamic factors \hat{G}^* .

Supervised Deep Dynamic PCA for Time Series

Feature Reconstruction and Dimension Reduction Based on Deep Neural Network Given the underlying factor structure, a common approach to estimating the latent factors \mathbf{f}_t is to use PCA in linear settings, or autoencoders in nonlinear ones. However, under models (1)-(2), both PCA and autoencoders suffer from a key limitation: they do not incorporate information from the target variable during factor extraction. Specifically, when the factors are strong, these methods cannot distinguish between target-relevant and irrelevant latent components, offering no guarantee that the top K_1 factors are optimal for forecasting the outcome. When the factors are weak, they may fail to separate signal from noise, resulting in biased forecasts even when using all extracted factors. To address these shortcomings, we propose a **supervised deep dynamic PCA (SDDP)** approach that (i) explicitly incorporates the target variable into the factor extraction process, (ii) enables more effective dynamic “sufficient” dimension reduction tailored for time series prediction, and (iii) leverages flexible model architectures, using powerful neural networks to efficiently capture complex nonlinear relationships.

As illustrated in Figure 1, our method consists of two key components. First, we construct a panel of **target-aware predictors** $\hat{\mathbf{x}}_{i,t}^*$, where each $\hat{\mathbf{x}}_{i,t}^*$ is obtained by fitting a temporal DNN that regresses the future outcome y_{t+h} on the individual predictors $\{x_{i,1}, \dots, x_{i,t}\}$. Second, we apply conventional PCA to this panel to extract supervised factor features. This approach generalizes traditional vector-based factor models used for prediction (Sen, Yu, and Dhillion 2019; Fan, Xue, and Yao 2017; Luo et al. 2022; Yu, Yao, and Xue 2022, 2024; Fan and Gu 2023), as it explicitly incorporates target information during feature construction. By doing so, it preserves predictive signals more effectively while simultaneously achieving dimension reduction, which substantially lowers computational cost.

In the first step, for each $i = 1, \dots, N$, we estimate a temporal DNN that regresses the future target variable y_{t+h}

on the i -th predictor and its past lagged variables:

$$y_{t+h} \approx \mathcal{T}_i(x_{i,(t-q_0+1)\vee 1}, \dots, x_{i,t}; \hat{\theta}_i), \quad (3)$$

where \mathcal{T}_i denotes a pre-specified temporal DNN architecture (e.g., Temporal Convolutional Network, TCN), q_0 is the window size with $q_0 \geq q$, and $\hat{\theta}_i$ represents the learned parameters for the i -th neural network. The parameters $\hat{\theta}_i$ are obtained by minimizing the least squares loss

$$\hat{\theta}_i = \arg \min_{\theta_i} \sum_t (y_{t+h} - \mathcal{T}_i(x_{i,(t-q_0+1)\vee 1}, \dots, x_{i,t}; \theta_i))^2.$$

After training, we use the fitted networks to construct a panel of **target-aware predictors**, denoted by $\hat{\mathbf{x}}_t^* = (\hat{\mathbf{x}}_{1,t}^*, \dots, \hat{\mathbf{x}}_{N,t}^*)^\top$, where each component is defined as

$$\hat{\mathbf{x}}_{i,t}^* = \mathcal{T}_i(x_{i,(t-q_0+1)\vee 1}, \dots, x_{i,t}; \hat{\theta}_i).$$

This transformation embeds predictive information about the target into each feature, enhancing its relevance for downstream supervised factor extraction.

In the second step, we apply PCA to the transformed predictors $\hat{\mathbf{x}}_t^*$ to obtain the estimated dynamic factors $\hat{\mathbf{g}}_t^*$. Specifically, we compute the sample covariance matrix $\hat{\Sigma} = T^{-1} \sum_t \hat{\mathbf{x}}_t^* \hat{\mathbf{x}}_t^{*\top} \in \mathbb{R}^{N \times N}$, and extract the factor loading matrix $\hat{B}^* \in \mathbb{R}^{N \times K^*}$ as the top K^* eigenvectors¹ of $\hat{\Sigma}$ scaled by \sqrt{N} . The estimated factors are then given by $\hat{\mathbf{g}}_t^* = N^{-1} \hat{B}^{*\top} \hat{\mathbf{x}}_t^*$. These extracted factors capture information from the original relevant latent factors \mathbf{g}_t , as well as from their lagged values that are predictive of the future target y_{t+h} . Intuitively, $\hat{\mathbf{g}}_t^*$ serves as an estimator for the concatenated vector $(\mathbf{f}_t^\top, \dots, \mathbf{f}_{t-q+1}^\top)^\top$ in models (1)-(2), under suitable conditions. It is important to emphasize that, due to the temporal dependence structure in model (2), relying solely on contemporaneous predictors $x_{i,t}$ to approximate y_{t+h} in the DNN fitting step can lead to biased parameter

¹Numerous methods have been developed for determining the number of factors in factor analysis and PCA. Here, we adopt the approach by Fan, Guo, and Zheng (2022) as our default choice.

estimates. This, in turn, results in inefficient recovery of the relevant latent factors \mathbf{g}_t . Incorporating lagged predictors is therefore crucial for accurate dynamic factor estimation.

A key advantage of SDDP is its ability to filter out irrelevant predictors by assigning them shrinking weights. This is critical because, unlike strong factors, weak factors often have signals that are not clearly distinguishable from noise. Without such a signal-enhancing mechanism, conventional dimension reduction methods may struggle to recover these subtle signals from the overwhelming presence of noise.

The following proposition establishes the consistency of the estimated factors in a linear setting.

Assumption 1 *Suppose a linear structure of models (1)-(2)*

$$\mathbf{x}_t = B\mathbf{f}_t + u_t, \quad y_{t+h} = \beta_1^\top \mathbf{g}_t + \dots + \beta_q^\top \mathbf{g}_{t-q+1} + \epsilon_{t+h},$$

where $B = (b_1^\top, \dots, b_p^\top) \in \mathbb{R}^{N \times K}$ and $1 \leq t \leq T$. The terms u_t and ϵ_{t+h} are i.i.d. sub-Gaussian noises with variances σ_u and σ_ϵ , respectively. We assume independence among u_t , ϵ_{t+h} and \mathbf{f}_t . Additionally, we assume $\max_i \|b_i\|^4 \leq C$, and that the eigenvalues of the scaled factor loading covariance matrix $\Sigma_b = N^{-\nu} \sum_{i=1}^N b_i b_i^\top$ are bounded above and below by positive constants, where $\nu \in (0, 1]$ controls factor strength, with $\nu = 1$ corresponding to strong factors and smaller ν indicating weaker factors. Finally, we assume the latent factors satisfy $\mathbb{E}\|\mathbf{f}_t\|^4 \leq C$.

Proposition 1 *Suppose Assumption 1 holds. Assume q and K are fixed. Let $\mathbf{g}_t^* = (\mathbf{f}_t^\top, \dots, \mathbf{f}_{t-q+1}^\top)^\top$. Then there exist rotation matrices $R \in \mathbb{R}^{qK \times qK}$ such that*

$$\|\hat{\mathbf{g}}_t^* - R\mathbf{g}_t^*\|_2 = O_p(N^{-\nu} + T^{-1/2}N^{-\nu/2} + N^{1-3\nu/2}T^{-3/2}).$$

The well-known Wiener–Kolmogorov prediction theory (Kolmogorov 1941; Wiener 1964) is a foundational result in time series analysis. Among its key findings, it shows that under mild conditions, any weakly stationary time series can be represented as an infinite-order linear autoregressive (AR) process driven by white noise. This perspective suggests that our theoretical results, though developed in a linear framework, may be extended to more general nonlinear time series models. However, such extensions would involve substantial analytical challenges.

Target Time Series Regression Based on Supervised Factors In the previous step, we accomplished our first objective: extracting supervised dynamic factors through the SDDP method, thereby achieving supervised dynamic dimension reduction. The next step is to use these estimated latent factors $\hat{\mathbf{g}}_t^*$, along with their lagged values, to forecast the future outcome y_{t+h} . The target y_{t+h} is estimated using:

$$\hat{y}_{t+h} = \mathcal{H}(\hat{\mathbf{g}}_{(t-q_0+1)\vee 1}^*, \dots, \hat{\mathbf{g}}_t^*, y_{(t-q_0+1)\vee 1}, \dots, y_t),$$

where $\mathcal{H}(\cdot)$ is a flexible nonlinear mapping capturing complex dependencies. In our empirical analysis, we explore several forecasting models for $\mathcal{H}(\cdot)$, including classical deep learning architectures such as Temporal Convolutional Networks (TCN) (Bai, Kolter, and Koltun 2018), Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997), and DeepAR (Salinas et al. 2020), as well as recent models like DeepGLO (Sen, Yu, and Dhillon 2019), Autoformer (Wu et al. 2021), Crossformer (Zhang and Yan 2023),

Algorithm 1: SDDP Method: Supervised Deep Dynamic PCA with Time Series Regression

Input: Time series predictor $X = (x_{i,t}) \in \mathbb{R}^{N \times T}$, target variable $\{y_t\}_{t=1}^T$, rolling window size q_0 , temporal DNN regression model \mathcal{H} .

Output: Estimated \hat{y}_{T+h} .

1: **Step 1: Construction of Target-aware Predictors**

2: **for** each variable $i = 1, \dots, N$ **do**

3: Estimate parameters θ_i by solving $\hat{\theta}_i = \arg \min_{\theta_i} \sum_t (y_{t+h} - \mathcal{T}_i(x_{i,(t-q_0+1)\vee 1}, \dots, x_{i,t}; \theta_i))^2$.

4: Construct target-aware predictors $\hat{\mathbf{x}}_{i,t}^* = \mathcal{T}_i(x_{i,(t-q_0+1)\vee 1}, \dots, x_{i,t}; \hat{\theta}_i)$.

5: **end for**

6: **Step 2: Extract Latent Factors**

7: Form target-aware predictors $\hat{\mathbf{x}}_t^* = (\hat{\mathbf{x}}_{1,t}^*, \dots, \hat{\mathbf{x}}_{N,t}^*)^\top$ for $t = 1, \dots, T$.

8: Compute sample covariance matrix $\hat{\Sigma} = T^{-1} \sum_t \hat{\mathbf{x}}_t^* \hat{\mathbf{x}}_t^{*\top} \in \mathbb{R}^{N \times N}$.

9: Determine the number of factors K^* using $\hat{\Sigma}$ or its normalized correlation counterpart.

10: Compute the top K^* eigenvectors \hat{U}^* of $\hat{\Sigma}$.

11: Set the factor loading matrix $\hat{B}^* = \sqrt{N}\hat{U}^*$ and compute supervised factors $\hat{\mathbf{g}}_t^* = N^{-1}\hat{B}^{*\top}\hat{\mathbf{x}}_t^*$.

12: **Step 3: Train Forecasting Target Model with DNN**

13: Fit the temporal DNN model on the training data using an appropriate loss function

$$\hat{y}_{t+h} = \mathcal{H}(\hat{\mathbf{g}}_{(t-q_0+1)\vee 1}^*, \dots, \hat{\mathbf{g}}_t^*, y_{(t-q_0+1)\vee 1}, \dots, y_t).$$

and TimesNet (Wu et al. 2023). Empirical results demonstrate that our framework consistently enhances forecasting accuracy across these models, highlighting its robustness and versatility. Although the supervised factors from the previous subsection embed relevant temporal information and could be used without additional lags, we include both lagged factors and past outcomes in the forecasting model to further improve predictive performance. The complete algorithm is presented in Algorithm 1. We introduced SDDP with one target variable, but it can be extended to multiple targets by aggregating their approximations in Step 1.

Supervised Deep Dynamic PCA with Incomplete Data

In real world applications, it is common to encounter missing covariate data due to situations such as sensor malfunctions, incomplete reporting, or data corruption. These missing entries pose a significant challenge for our tasks, especially when high-quality covariate information is essential for accurate dimension reduction and forecasting. To address this, we develop a dimension reduction framework tailored for scenarios with incomplete covariate data, building upon the SDDP architecture introduced in previous subsections.

In addition to the observed covariate \mathbf{x}_t , we introduce a binary mask vector $\mathbf{w}_t = (w_{1,t}, \dots, w_{N,t})^\top \in \{0, 1\}^N$, where

$$w_{i,t} = \begin{cases} 1, & \text{if the covariate } x_{i,t} \text{ is observed,} \\ 0, & \text{if the covariate } x_{i,t} \text{ is missing.} \end{cases}$$

Method	Climate		Energy		FinC		Light		Weather	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
SDDP-TCN	2.936	<u>3.673</u>	41.506	74.279	0.0446	0.0564	2.725	4.965	2.679	3.369
sdPCA-TCN	3.169	3.912	40.165	<u>71.045</u>	0.0451	0.0559	3.000	4.888	2.704	3.409
PCA-TCN	3.569	4.430	59.604	93.505	0.0484	0.0653	3.555	6.344	3.832	4.782
Vanilla-TCN	3.956	4.838	62.362	95.822	0.0484	0.0654	3.494	6.333	3.991	4.974
SDDP-LSTM	<u>2.946</u>	3.628	40.677	73.514	0.0476	0.0598	2.706	5.404	<u>2.689</u>	<u>3.375</u>
sdPCA-LSTM	3.439	4.254	40.268	69.940	0.0476	0.0597	3.138	5.731	3.114	3.907
PCA-LSTM	4.494	5.535	59.803	93.170	0.0484	0.0654	3.583	6.504	4.434	5.509
Vanilla-LSTM	4.518	5.547	59.716	93.295	0.0484	0.0654	3.538	6.464	4.472	5.544
SDDP-DeepAR	3.188	3.953	52.697	85.275	0.0476	0.0599	2.912	5.888	2.954	3.725
sdPCA-DeepAR	3.658	4.524	50.916	83.427	0.0536	0.0668	3.341	6.059	3.460	4.362
PCA-DeepAR	4.513	5.570	59.680	92.882	0.0484	0.0653	3.629	6.513	4.445	5.526
Vanilla-DeepAR	4.532	5.575	58.979	92.850	0.0493	0.0661	3.548	6.459	4.487	5.565
SDDP-TimesNet	4.781	6.057	39.821	71.133	0.0444	<u>0.0564</u>	2.827	5.462	3.629	4.579
sdPCA-TimesNet	4.621	5.914	<u>39.094</u>	71.210	0.0446	0.0565	2.780	5.296	3.559	4.515
PCA-TimesNet	4.645	5.929	38.954	71.192	0.0457	0.0578	<u>2.692</u>	5.343	3.558	4.509
Vanilla-TimesNet	4.753	6.107	43.980	73.926	0.0468	0.0592	2.776	5.389	3.506	4.424
AEAR	3.267	4.112	45.424	74.489	0.0469	0.0635	3.027	4.894	2.985	3.754
DeepGLO	3.016	3.881	43.337	79.443	0.0785	0.0960	2.025	4.405	2.943	3.739
Autoformer	4.024	4.946	45.123	74.067	0.0546	0.0687	2.941	4.815	3.806	4.795
Crossformer	4.259	5.445	46.346	90.817	0.0555	0.0682	3.110	<u>4.653</u>	3.947	5.089
ARIMA	3.315	4.153	49.166	81.603	0.0497	0.0666	3.035	5.387	3.079	3.816

Table 1: Comparison of prediction accuracy (MAE and RMSE over the testing data) across various methods and datasets. Results are averaged over 100 repetitions, with the associated confidence intervals reported in Tables S.3 in the Appendix. **Bold** indicates the best performance in each column, and underlined indicates the second best.

The first step in the SDDP framework involves supervised training of a temporal DNN for each predictor. For each $i = 1, \dots, N$, we estimate the model parameters θ_i by solving the following weighted least squares problem

$$\hat{\theta}_i = \arg \min_{\theta_i} \sum_t w_{i,t} (y_{t+h} - \mathcal{T}_i(x_{i,(t-q_0+1)\vee 1}, \dots, x_{i,t}; \theta_i))^2,$$

where $\mathcal{T}_i(\cdot; \theta_i)$ denotes the temporal DNN associated with the i -th predictor, and $q_0 \geq q$ specifies the window size. The mask $w_{i,t}$ ensures that only observed entries are used during training. Once trained, we use the fitted networks to construct a panel of target-aware predictors, denoted by $\hat{\mathbf{x}}_t^* = (\hat{x}_{1,t}^*, \dots, \hat{x}_{N,t}^*)^\top$, where each entry is computed as

$$\hat{x}_{i,t}^* = \mathcal{T}_i(\tilde{x}_{i,(t-q_0+1)\vee 1}, \dots, \tilde{x}_{i,t}; \hat{\theta}_i),$$

and each $\tilde{x}_{i,t}$ is defined by $\tilde{x}_{i,t} = \hat{x}_{i,t}^*$ if $w_{i,t} = 0$, and $\tilde{x}_{i,t} = x_{i,t}$ if $w_{i,t} = 1$. That is, when the covariate $x_{i,t}$ is missing, we replace it with the DNN (\mathcal{T}_i) based imputation $\hat{x}_{i,t}^*$; otherwise, we retain the original observed value.

In the second step, mirroring previous subsection, we apply PCA to the transformed predictor panel $\hat{\mathbf{x}}_t^*$ to extract the estimated latent factors $\hat{\mathbf{g}}_t^*$. These estimated factors integrate both observed and imputed covariate information, yielding a refined representation that compensates for missing entries.

Unlike traditional matrix completion approaches that focus solely on modeling the correlation structure among covariates, our method leverages supervision from the target variable y_{t+h} throughout the reconstruction process. This design aligns directly with our forecasting objective,

accurately predicting y_{t+h} based on the covariate history $\{\mathbf{x}_1, \dots, \mathbf{x}_t\}$. The SDDP algorithm under covariate missingness closely follows Steps 1 and 2 of Algorithm 1, and is therefore omitted for brevity.

Despite its conceptual simplicity, our dynamic SDDP approach proves to be highly effective in practice. Empirical results demonstrate that it consistently outperforms baseline methods in downstream prediction tasks, especially under high levels of covariate missingness.

Empirical Studies

We demonstrate the empirical performance of the proposed SDDP across five prediction tasks using real-world high-dimensional time series datasets. Descriptions of the datasets, data sources, and preprocessing steps are provided in Table S.1 of the Appendix. Each dataset is split chronologically with 80% for training and 20% for testing.

Forecasting with SDDP. Our primary goal is to evaluate whether the proposed SDDP can extract factors that are more predictively powerful than other methods. We compare the performance of four DNN forecasting algorithms (TCN (Bai, Koltner, and Koltun 2018), LSTM (Hochreiter and Schmidhuber 1997), DeepAR (Salinas et al. 2020), and TimesNet (Wu et al. 2023)) using factors extracted in three ways: traditional unsupervised principal component analysis (**PCA**), the supervised dynamic PCA (**sdPCA**) (Gao and Tsay 2024), and the proposed supervised deep dynamic PCA (**SDDP**). We benchmark these methods against

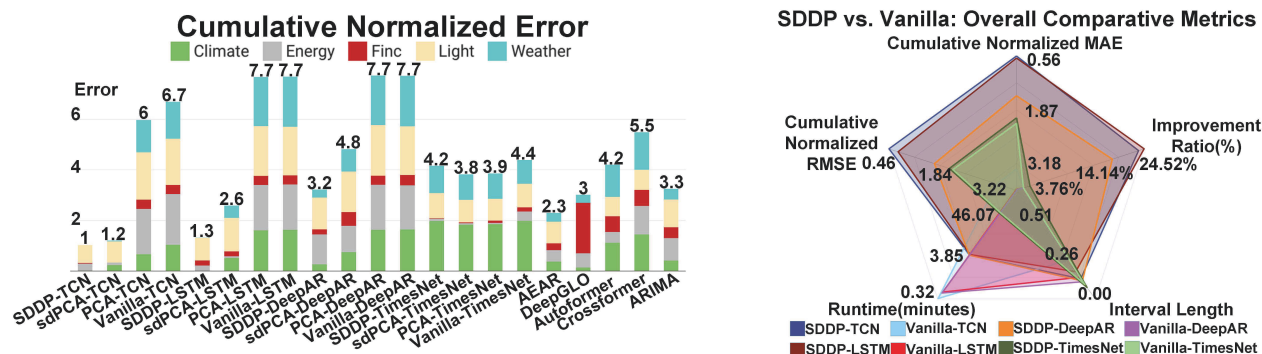


Figure 2: (a) Cumulative normalized error for each method; (b) radar chart comparing the overall performance between SDDP methods and vanilla baselines in five aspects. Detailed values and the normalization procedure are provided in the Appendix.

the **Vanilla** approach that feeds high-dimensional covariates directly into DNNs without any dimension reduction. We include two factor-augmented deep learning algorithms: Autoencoder Augmented Regression (AEAR) (Le, Patterson, and White 2018) and DeepGLO (Sen, Yu, and Dhillon 2019), and two Transformer-based algorithms: Autoformer (Wu et al. 2021) and Crossformer (Zhang and Yan 2023) as well as traditional ARIMA, as additional benchmarks.

Table 1 summarizes the comparisons of prediction accuracy using different methods. We use the mean absolute error (MAE) and root mean squared error (RMSE) over the testing data as evaluation metrics. Results are reported based on 100 repetitions, with the associated 95% confidence intervals in Table S.3 (in the Appendix). Since datasets vary in scale, raw MAEs and RMSEs are not directly comparable across datasets. To overcome this limitation, we provide min-max normalized MAEs and RMSEs in Table S.2, with details of the normalization procedure introduced in the Appendix. We also record the average runtime in Table S.4 (in the Appendix) to further assess the practicality of various methods.

As shown in Table 1, among all the methods considered, SDDP methods consistently achieve the best performance in 4 out of 5 datasets (except for the Light data, in which DeepGLO excels). Additionally, when comparing different dimension reduction strategies within the same DNN forecasting algorithm, the proposed SDDP outperforms the sdPCA, unsupervised PCA, and Vanilla methods by a large margin in most learning algorithms (except for TimesNet, in which all dimension reduction approaches yield comparable performance). The numerical comparisons demonstrate that the dynamic, supervised, and nonlinear nature of SDDP equips it with enhanced ability to effectively capture the informative structure embedded in high-dimensional time series data, thereby boosting the model’s predictive power.

We present three plots to better visualize the comparisons among various approaches. Figure 2(a) plots the cumulative normalized error across datasets to demonstrate each model’s overall forecasting performance. The plot shows among all algorithms considered (TCN, LSTM, DeepAR, TimesNet), the SDDP-based method achieves a smaller cumulative normalized error than the corresponding sdPCA,

PCA, and Vanilla methods, demonstrating evident advantages of SDDP over other dimension reduction approaches. Figure S.1 (in the Appendix) displays the relative improvements (in percentage) of each method compared to its corresponding Vanilla baseline, revealing a consistent improvement (around 10%–30%) of using SDDP methods compared to Vanilla methods without performing dimension reduction on high-dimensional predictors. Figure 2(b) graphs a radar chart to provide a holistic comparison of SDDP and Vanilla methods with the four DNN forecasting algorithms in five aspects: cumulative normalized MAE, cumulative normalized RMSE, relative improvement (in percentage), runtime, and length of the confidence interval. The plot shows SDDP-TCN and SDDP-LSTM appear to achieve the overall best or near-best results in terms of predictive accuracy in the tasks in this study. The plot further emphasizes substantial advantages of SDDP over Vanilla methods in prediction accuracy, while possibly incurring slight increases in runtime.

SDDP with Partially Observed Covariates. Another promising application of SDDP lies in settings with partially observed covariates, specifically, when the covariate matrix contains missing values. Our goal is still to leverage the target time series to perform supervised dimension reduction and construct effective downstream forecasting models. To explore this, we use the Climate and Weather datasets and introduce random missingness into the covariate matrix, with missing rates ranging from 12.5%, 25%, 37.5%, 50%. Based on the results in fully observed predictors, where TCN outperforms LSTM and DeepAR on these datasets, we focus comparison on TCN and TimesNet, using vanilla AEAR as the benchmark. We evaluate forecasting performance using SDDP with incomplete covariates, benchmarked against the Vanilla method. We also evaluate whether imputing missing covariates first affects subsequent factor-based prediction performance using two preprocessing techniques: Multiple Imputation by Chained Equations (MICE) (White, Royston, and Wood 2011) and MissForest (Stekhoven and Bühlmann 2012) as well as TimeDiff (Shen and Kwok 2023).

Table 2 summarizes the comparison of out-of-sample MAE and RMSE using different methods, based on 100 repetitions. Figure S.2 (in the Appendix) displays the rel-

Dataset	Method	Missing Rate									
		0%		12.5%		25%		37.5%		50%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Climate	SDDP-TCN	2.936	3.673	2.938	3.645	2.929	3.642	2.993	3.693	3.004	3.703
	SDDP-TimeDiff-TCN	–	–	2.992	3.732	3.066	3.789	3.006	3.724	3.055	3.803
	SDDP-MICE-TCN	–	–	2.932	3.667	2.948	3.688	2.964	3.691	2.977	3.708
	SDDP-MissForest-TCN	–	–	2.935	3.676	2.941	3.689	2.964	3.702	2.974	3.710
	Vanilla-TCN	3.956	4.838	3.979	4.873	3.946	4.850	3.948	4.848	3.812	4.695
	Vanilla-TimeDiff-TCN	–	–	3.919	4.811	3.771	4.686	3.765	4.717	3.757	4.689
	Vanilla-MICE-TCN	–	–	4.006	4.897	3.962	4.845	3.964	4.844	3.955	4.834
	Vanilla-MissForest-TCN	–	–	3.973	4.854	3.990	4.875	3.967	4.845	3.960	4.839
	SDDP-TimesNet	4.781	6.057	4.621	5.899	4.657	5.953	4.675	5.951	4.676	5.981
	Vanilla-TimesNet	4.753	6.107	4.829	6.204	4.834	6.201	4.829	6.192	4.803	6.146
	AEAR	3.267	4.112	3.373	4.244	3.337	4.189	3.410	4.293	3.422	4.286
	Weather	SDDP-TCN	2.679	3.369	2.654	3.334	2.647	3.334	2.666	3.356	2.662
SDDP-TimeDiff-TCN		–	–	2.644	3.313	2.672	3.341	2.705	3.397	2.679	3.366
SDDP-MICE-TCN		–	–	2.659	3.346	2.653	3.327	3.663	3.356	2.671	3.356
SDDP-MissForest-TCN		–	–	2.665	3.348	2.661	3.331	2.672	3.359	2.714	3.406
Vanilla-TCN		3.991	4.974	3.954	4.936	3.887	4.847	3.923	4.876	3.940	4.906
Vanilla-TimeDiff-TCN		–	–	3.886	4.864	3.898	4.875	3.970	4.938	3.802	4.772
Vanilla-MICE-TCN		–	–	3.994	4.979	3.996	4.987	3.991	4.969	4.006	4.995
Vanilla-MissForest-TCN		–	–	3.984	4.968	3.997	4.986	3.987	4.963	3.967	4.943
SDDP-TimesNet		3.629	4.579	3.600	4.570	3.630	4.611	3.668	4.665	3.712	4.731
Vanilla-TimesNet		3.506	4.424	3.599	4.545	3.618	4.574	3.707	4.706	3.739	4.741
AEAR		2.985	3.754	3.093	3.890	3.092	3.872	3.225	4.055	3.116	3.897

Table 2: Comparison of prediction accuracy with randomly introduced covariate missingness. Results are averaged over 100 repetitions. **Bold** indicates the best performance for each metric within each dataset. “–” denotes fully observed data requiring no preprocessing. “MICE”, “MissForest” or “TimeDiff” indicates the respective imputation before prediction.

ative improvements (in percentage) of TCN and TimesNet compared to its corresponding Vanilla baseline under partially observed covariates. As shown in Table 2, forecasting with SDDP-TCN consistently outperforms other dimension reduction techniques across all levels of missingness. Figure S.2 further illustrates that, compared to vanilla TCN without dimension reduction, SDDP-TCN yields a consistent improvement in predictive accuracy, typically around 20%-30%. The performance of SDDP-TimesNet is very similar to its vanilla counterpart, suggesting that transformer-based models may already capture complex structures, making additional dimension reduction less impactful. On the Climate and Weather datasets, injecting random missingness into the covariates causes only minor MAE and RMSE fluctuations, typically within 3%, indicating that the temporal DNN component effectively imputes missing values under these conditions. Furthermore, SDDP-TCN with MICE, MissForest, or TimeDiff imputed data performs very similarly to original SDDP-TCN, demonstrating that the superior performance stems from the factor extraction process rather than imputation. This confirms that SDDP’s performance improvement comes primarily from the framework itself rather than the imputation step.

Discussion of Limitations and Extensions

In this paper, we have introduced SDDP, a unifying framework that marries the strengths of DNNs with classical factor-model ideas to produce target-aware, low-dimensional dynamic representations for high-dimensional time series predictors. Empirical studies across five diverse datasets demonstrate that SDDP consistently outperforms unsupervised PCA, sdPCA, and a range of benchmark deep learning baselines, both in the fully observed and partially observed covariate settings. However, SDDP does demand greater computational resources than unsupervised PCA, since training a separate deep network for each predictor can become prohibitive at large scale.

Looking ahead, the SDDP framework can be extended to handle more complex data modalities, such as tensor-valued, image, or network data, by incorporating higher-order factorization techniques (e.g., tensor decompositions) to extract supervised, dynamic features (Han, Chen, and Zhang 2022; Han and Zhang 2023; Chen, Han, and Yu 2024; Han et al. 2024a,b,c; Kong et al. 2024; Luo, Han, and Yu 2025; Zhou, Han, and Yu 2025b). On the theoretical front, we aim to establish consistency results for nonlinear factor models, and methodologically we will investigate advanced imputation strategies (e.g., variational autoencoders) for structured missingness alongside online updating schemes to support real-time streaming applications.

Acknowledgments

We thank the anonymous reviewers for helpful suggestions. The work of Yuefeng Han is supported in part by National Science Foundation Grants DMS-2412578. The work of Xiufan Yu is supported in part by National Institutes of Health grant R01GM152812.

References

- Ashraf, M.; Anowar, F.; Setu, J. H.; Chowdhury, A. I.; Ahmed, E.; Islam, A.; and Al-Mamun, A. 2023. A Survey on Dimensionality Reduction Techniques for Time-Series Data. *IEEE Access*, 11: 42909–42923.
- Bai, J.; and Ng, S. 2002. Determining the Number of Factors in Approximate Factor Models. *Econometrica*, 70(1): 191–221.
- Bai, J.; and Ng, S. 2006. Confidence Intervals for Diffusion Index Forecasts and Inference for Factor-Augmented Regressions. *Econometrica*, 74(4): 1133–1150.
- Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv preprint arXiv:1803.01271.
- Bair, E.; Hastie, T.; Paul, D.; and Tibshirani, R. 2006. Prediction by Supervised Principal Components. *Journal of the American Statistical Association*, 101(473): 119–137.
- Bernanke, B. S.; Boivin, J.; and Elias, P. 2005. Measuring the effects of monetary policy: a factor-augmented vector autoregressive (FAVAR) approach. *The Quarterly Journal of Economics*, 120(1): 387–422.
- Chen, B.; Han, Y.; and Yu, Q. 2024. Estimation and Inference for CP Tensor Factor Models. arXiv preprint arXiv:2406.17278.
- Chen, B.; Han, Y.; and Yu, Q. 2025. Diffusion Index Forecast with Tensor Data. Available at SSRN 5202316.
- Cook, R. D.; and Li, B. 2002. Dimension Reduction for Conditional Mean in Regression. *The Annals of Statistics*, 30(2): 455–474.
- Fan, J.; and Gu, Y. 2023. Factor Augmented Sparse Throughput Deep ReLU Neural Networks for High Dimensional Regression. *Journal of the American Statistical Association*, 118(543): 2680–2694.
- Fan, J.; Guo, J.; and Zheng, S. 2022. Estimating Number of Factors by Adjusted Eigenvalues Thresholding. *Journal of the American Statistical Association*, 117(538): 852–861.
- Fan, J.; Xue, L.; and Yao, J. 2017. Sufficient Forecasting Using Factor Models. *Journal of Econometrics*, 201(2): 292–306.
- Gao, Z.; and Tsay, R. S. 2024. Supervised Dynamic PCA: Linear Dynamic Forecasting with Many Predictors. *Journal of the American Statistical Association*, 120(550): 869–883.
- Gu, S.; Kelly, B.; and Xiu, D. 2021. Autoencoder Asset Pricing Models. *Journal of Econometrics*, 222(1): 429–450.
- Han, Y.; Chen, R.; Yang, D.; and Zhang, C.-H. 2024a. Tensor Factor Model Estimation by Iterative Projection. *The Annals of Statistics*, 52(6): 2641–2667.
- Han, Y.; Chen, R.; and Zhang, C.-H. 2022. Rank Determination in Tensor Factor Model. *Electronic Journal of Statistics*, 16(1): 1726–1803.
- Han, Y.; Chen, R.; Zhang, C.-H.; and Yao, Q. 2024b. Simultaneous Decorrelation of Matrix Time Series. *Journal of the American Statistical Association*, 119(546): 957–969.
- Han, Y.; Yang, D.; Zhang, C.-H.; and Chen, R. 2024c. CP Factor Model for Dynamic Tensors. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(5): 1383–1413.
- Han, Y.; and Zhang, C.-H. 2023. Tensor Principal Component Analysis in High-Dimensional CP Models. *IEEE Transactions on Information Theory*, 69(2): 1147–1167.
- Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786): 504–507.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural computation*, 9(8): 1735–1780.
- Huang, D.; Jiang, F.; Li, K.; Tong, G.; and Zhou, G. 2022. Scaled PCA: A New Approach to Dimension Reduction. *Management Science*, 68(3): 1678–1695.
- Kolmogorov, A. 1941. Interpolation and Extrapolation of Stationary Random Sequences. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 5: 3.
- Kong, L.; Chen, E.; Chen, Y.; and Han, Y. 2024. TEAFormers: Tensor-Augmented Transformers for Multi-Dimensional Time Series Forecasting. arXiv preprint arXiv:2410.20439.
- Lam, C.; and Yao, Q. 2012. Factor Modeling for High-Dimensional Time Series: Inference for the Number of Factors. *The Annals of Statistics*, 40(2): 694–726.
- Le, L.; Patterson, A.; and White, M. 2018. Supervised Autoencoders: Improving Generalization Performance with Unsupervised Regularizers. In *Advances in Neural Information Processing Systems*, 107–117.
- Lee, K.-Y.; Li, B.; and Chiaromonte, F. 2013. A General Theory for Nonlinear Sufficient Dimension Reduction: Formulation and Estimation. *The Annals of Statistics*, 41(1): 221–249.
- Li, B. 2018. *Sufficient Dimension Reduction: Methods and Applications with R*. Chapman and Hall/CRC.
- Li, K.-C. 1991. Sliced Inverse Regression for Dimension Reduction. *Journal of the American Statistical Association*, 86(414): 316–327.
- Luo, W.; Xue, L.; Yao, J.; and Yu, X. 2022. Inverse Moment Methods for Sufficient Forecasting Using High-Dimensional Predictors. *Biometrika*, 109(2): 473–487.
- Luo, Z.; Han, Y.; and Yu, X. 2025. Factor Augmented Supervised Learning with Text Embeddings. arXiv preprint arXiv:2508.06548.
- Paul, D.; Bair, E.; Hastie, T.; and Tibshirani, R. 2008. “Preconditioning” for Feature Selection and Regression in High-Dimensional Problems. *The Annals of Statistics*, 36(4): 1595–1618.

- Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *International Journal of Forecasting*, 36(3): 1181–1191.
- Schölkopf, B.; Smola, A.; and Müller, K.-R. 1997. Kernel Principal Component Analysis. In *International Conference on Artificial Neural Networks*, 583–588.
- Sen, R.; Yu, H.-F.; and Dhillon, I. 2019. Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting. In *Advances in Neural Information Processing Systems*, 4837–4846.
- Shen, L.; and Kwok, J. T. 2023. Non-Autoregressive Conditional Diffusion Models for Time Series Prediction. In *Proceedings of the 40th International Conference on Machine Learning*, 30100–30113.
- Stekhoven, D. J.; and Bühlmann, P. 2012. MissForest: Non-Parametric Missing Value Imputation for Mixed-Type Data. *Bioinformatics*, 28(1): 112–118.
- Stock, J. H.; and Watson, M. W. 2002a. Forecasting Using Principal Components From a Large Number of Predictors. *Journal of the American Statistical Association*, 97(460): 1167–1179.
- Stock, J. H.; and Watson, M. W. 2002b. Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, 20(2): 147–162.
- Sánchez Sorzano, C. O.; Vargas, J.; and Montano, A. P. 2014. A Survey of Dimensionality Reduction Techniques. *arXiv preprint arXiv:1403.2877*.
- Tang, Y.; and Li, B. 2024. Belted and Ensembled Neural Network for Linear and Nonlinear Sufficient Dimension Reduction. *arXiv preprint arXiv:2412.08961*.
- Van der Maaten, L.; Postma, E.; and Van den Herik, J. 2007. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research*, 10(66): 66–71.
- White, I. R.; Royston, P.; and Wood, A. M. 2011. Multiple Imputation Using Chained Equations: Issues and Guidance for Practice. *Statistics in medicine*, 30(4): 377–399.
- Wiener, N. 1964. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press.
- Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.
- Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 1717–1728.
- Yalcin, I.; and Amemiya, Y. 2001. Nonlinear Factor Analysis as a Statistical Method. *Statistical Science*, 16(3): 275–294.
- Yu, R.; Chen, R.; Xiao, H.; and Han, Y. 2024. Dynamic Matrix Factor Models for High Dimensional Time Series. *arXiv preprint arXiv:2407.05624*.
- Yu, X.; Yao, J.; and Xue, L. 2022. Nonparametric Estimation and Conformal Inference of the Sufficient Forecasting with a Diverging Number of Factors. *Journal of Business & Economic Statistics*, 40(1): 342–354.
- Yu, X.; Yao, J.; and Xue, L. 2024. Power Enhancement for Testing Multi-Factor Asset Pricing Models via Fisher’s Method. *Journal of Econometrics*, 239(2): 105458.
- Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *International Conference on Learning Representations*.
- Zhou, G.; Han, Y.; and Yu, X. 2025a. Covariate-Adjusted Deep Causal Learning for Heterogeneous Panel Data Models. *arXiv preprint arXiv:2505.20536*.
- Zhou, G.; Han, Y.; and Yu, X. 2025b. Factor Augmented Tensor-on-Tensor Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 22928–22936.