Deep Functional Factor Models: Forecasting High-Dimensional Functional Time Series via Bayesian Nonparametric Factorization

Yirui Liu[1], Xinghao Qiao[2], Yulong Pei[3], Liying Wang[4]

JP Morgan[1], University of Hong Kong[2], Eindhoven University of Technology[3], University of Liverpool[4]

July 2024

Introduction

Introduction to Functional Time Series

- Functional time series: sequential collection of functional objects with temporal dependence.
- Examples:
  - Annual age-specific mortality rates for different countries.
  - Daily energy consumption curves from various households.
  - Cumulative intraday return trajectories for hundreds of stocks.
- These datasets can be represented as p-dimensional functional time series $\boldsymbol{Y}_t(\cdot) = \left(Y_{t1}(\cdot), \ldots, Y_{tp}(\cdot)\right)^T$, where each $Y_{tj}(\cdot)$ is a random function defined on a compact interval $\mathcal{U}$.

Introduction

Challenges

- High-dimensionality: The number of functional variables p is comparable to, or even larger than, the number of temporally dependent observations n.
- Infinite-dimensional nature of curve data
- Temporal dependence

(a) Energy Consumption (after standardization)

(b) World-wide Mortality Rate (after standardization)

Figure 1: Examples of functional time series

Existing Methods

Statistical Methods

- Principal components-based dimension reduction (Guo and Qiao, 2023; Chang et al., 2023a)
- Factor model (Guo et al., 2021)
- Segmentation transformation (Chang et al., 2023b)

Limitations

- Assume linear and Markovian dynamics
- Fail to capture complex nonlinear or non-Markovian temporal dependence

Existing Methods

Deep Learning

- RNN: LSTM, GRU
- Transformer

Challenges

- Black-box nature lacks explainability
- Difficulty in handling cross-sectional and serial correlations
- Non-stationarity and large number of parameters

Motivation

- Develop a model capable of capturing complex, non-Markovian, and nonlinear temporal dynamics.
- Ensure the model remains explainable, providing insights into the relationships and dependencies within the data.
- Improve predictive accuracy over conventional deep learning models.

Model

Sparse Functional Factor Model

We propose a functional factor model from the Bayesian perspective:

$$\boldsymbol{Y}_t(\cdot) = (\boldsymbol{Z} \odot \boldsymbol{A})\boldsymbol{X}_t(\cdot) + \boldsymbol{\epsilon}_t(\cdot), \quad t = 1, \ldots, n. \tag{1}$$

- $\boldsymbol{Y}_t(\cdot)$: observed functional time series.
- $\boldsymbol{Z}$: binary matrix from the Indian buffet process, $\boldsymbol{Z} \sim \mathrm{IBP}(\alpha)$.
- $\boldsymbol{A}$: loading weight matrix, elements $A_{tr} \sim \mathrm{Normal}(0, \sigma_A^2)$.
- $\boldsymbol{X}_t(\cdot)$: latent functional factor time series.
- $\boldsymbol{\epsilon}_t(\cdot)$: Gaussian distributed white noise, scale σ_ϵ.

Model

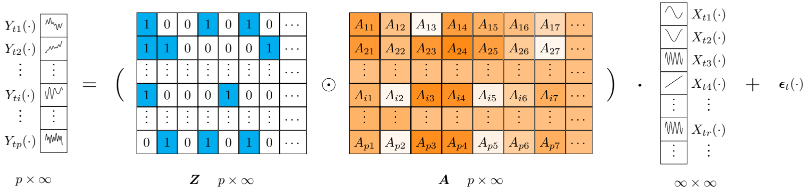


Figure 2: Factor Model

Indian Buffet Process

IBP

- IBP is a distribution over sparse binary matrices.
- Useful for models with an unknown number of latent features.
- Each row represents an observation, and each column represents a latent feature.
- The sparsity of the matrix is controlled by a parameter α.

IBP Sampling Process

- First customer samples Poisson(α) dishes.
- The i-th customer samples each previously chosen dish k with probability $\frac{m_k}{i}$, where m_k is the number of previous customers who have chosen dish k.
- The i-th customer then samples Poisson($\frac{\alpha}{i}$) new dishes.

Indian Buffet Process

Why IBP is used in factorization?

- In the context of the sparse functional factor model:
  - $\boldsymbol{Z} \sim \mathrm{IBP}(\alpha)$ creates a sparse binary matrix.
  - This matrix controls the inclusion of latent factors for each observation.
  - Promotes a parsimonious model by ensuring most factors are zero for each observation.
  - Helps in discovering a potentially infinite number of latent factors without overfitting.

Functional Gaussian Process Dynamical Model

Model Specification

- Let $\boldsymbol{X}_t(\cdot)$ be the latent functional factors.
- $\boldsymbol{X}_t(\cdot)$ follows a multi-task GP:

$$\boldsymbol{X}_t(\cdot) \sim \mathrm{MTGP}(\mathbf{0}, \kappa_{\mathcal{U}}(\cdot,\cdot), \kappa_{\mathcal{X}}(\cdot,\cdot)) \tag{2}$$

- The covariance structure is:

$$\mathrm{Cov}(X_{tr}(u), X_{sl}(v) \mid \mathcal{X}_{t-1}, \mathcal{X}_{s-1}) = \kappa_{\mathcal{X}}(\mathcal{X}_{t-1}, \mathcal{X}_{s-1})\kappa_{\mathcal{U}}(u,v)\mathbb{I}(r=l)$$

  where $\mathcal{X}_{t-1}$ indicates the set of historical information, $\kappa_{\mathcal{X}}$ is the temporal kernel and $\kappa_{\mathcal{U}}$ is the spatial kernel.
- Meanings of indices:
  - t, s: time indices
  - r, l: factor indices
  - u, v: spatial indices (points in the functional domain)
- This model is a functional variant of the Gaussian Process Dynamical Model (Wang et al. (2005))

Functional Gaussian Process Dynamical Model

Independence or not?

The model assumes independence across factors r and l, as indicated by $\mathbb{I}(r = l)$?

- Marginally dependent
- Conditionally independent

Non-Markovian Patterns

By incorporating a kernel function $\kappa_{\mathcal{X}}$ that depends on the entire history $\mathcal{X}_{t-1}$, the model can capture non-Markovian temporal dependencies.
Example:

$$\kappa(\mathcal{X}_{t-1}, \mathcal{X}_{s-1}) = \alpha_1 \int \mathbf{X}_{t-1}(u)^T \mathbf{X}_{s-1}(u)\, du + \alpha_2 \int \mathbf{X}_{t-2}(u)^T \mathbf{X}_{s-2}(u)\, du$$

Deep Temporal Kernels

Motivation

Deep kernels combine the flexibility of neural networks with the probabilistic properties of Gaussian Processes, to capture complex patterns and dependencies in temporal data.

Specification

- Let $\mathbf{h}_t$ be the hidden representation of the temporal data at time t.
- $\mathbf{h}_t$ is obtained through a neural network:

$$\mathbf{h}_t = H(F(\boldsymbol{X}_{t-1}), F(\boldsymbol{X}_{t-2}), \ldots) \tag{3}$$

- The temporal kernel is then constructed as:

$$\kappa_{\mathcal{X}}(\mathcal{X}_{t-1}, \mathcal{X}_{s-1}) = \kappa(\mathbf{h}_t, \mathbf{h}_s) \tag{4}$$

Deep Temporal Kernels

Deep Learning Modules

- **Mapping Function:** F maps infinite-dimensional Gaussian processes to d-dimensional vectors.
- **Neural Networks:** Various architectures can be used for H, such as LSTM, GRU, and attention mechanisms.
- **Non-Markovian Patterns:** Deep kernels can incorporate long-term dependencies, capturing non-Markovian patterns.
- **Example:** Using LSTM for H:

$$\mathbf{h}_t = \mathrm{LSTM}(\mathbf{x}_{1:t}) \tag{5}$$

Advantages

- Combines the flexibility of neural networks with the uncertainty quantification of GPs.
- Capable of modeling complex, nonlinear temporal dependencies.

The Imperative of Integration

Standard Deep Learning

- Directly applying deep learning to high-dimensional functional data is challenging due to:
  - High dimensionality of inputs.
  - Limited number of training time steps.
  - Risk of overfitting.
  - Loss of interpretability.

Role of Factorization

- Factorization reduces dimensionality by extracting latent factors:

$$\boldsymbol{Y}_t(\cdot) = (\boldsymbol{Z} \odot \boldsymbol{A})\boldsymbol{X}_t(\cdot) + \epsilon_t(\cdot)$$

- **Benefits:**
  - Enhances interpretability.
  - Reduces computational complexity.
  - Prevents overfitting: spectrum penalty

The Imperative of Integration

Integration with IBP and Deep Kernels

- **Indian Buffet Process (IBP):**
  - Provides a flexible, nonparametric approach to determine the number of latent factors.
  - Ensures sparsity in the factor loading matrix.
- **Deep Kernels:**
  - Incorporate non-Markovian and nonlinear dependencies.
  - Enhance the ability to capture complex temporal patterns.

Overall Framework

- The integration of factorization, IBP, and deep kernels results in a robust and explainable model:

$$\mathrm{DF^2M} = \text{Factor Model} + \text{IBP} + \text{Deep Temporal Kernels}$$

- This combination balances model complexity, interpretability, and predictive accuracy.

Sparse Variational Inference

Variational Inference

- Approximates the posterior distribution by maximizing the Evidence Lower Bound (ELBO).
- Minimizes the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior.

Sparse Variational Inference for Gaussian Processes

- Introduces a set of **inducing variables** to represent the function at a smaller set of points, $\boldsymbol{v} = (v_1, \ldots, v_K)$.
- Variational distribution for inducing variables:

$$q(\boldsymbol{X}(\boldsymbol{v})) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{S}) \tag{6}$$

- ELBO can be computed more efficiently by marginalizing over the inducing variables.

Sparse Variational Inference

Sparse Variational Inference for DF^{2}M

- Uses **common locations** for inducing variables across functional factors.
- Variational distribution for multi-task Gaussian process with inducing variables:

$$q(\boldsymbol{X}_r(\cdot)) = p\Big(\boldsymbol{X}_{1r}(\cdot), \ldots, \boldsymbol{X}_{nr}(\cdot) \mid \boldsymbol{X}_{1r}(\boldsymbol{v}), \ldots, \boldsymbol{X}_{nr}(\boldsymbol{v}), \kappa_{\mathcal{X}}, \kappa_{\mathcal{U}}\Big) \prod_{t=1}^{n} q(\boldsymbol{X}_{tr}(\boldsymbol{v})) \tag{7}$$

Sparse Variational Inference

ELBO for DF^{2}M

$$\begin{aligned}\text{ELBO} = &\sum_{t=1}^{n} \mathrm{E}_q \left[\log p(\boldsymbol{Y}_t(\cdot) \mid \boldsymbol{X}_t(\cdot), \boldsymbol{Z}, \boldsymbol{A}) \right] \\ &- \mathrm{KL}\big[q(\boldsymbol{Z}) \parallel p(\boldsymbol{Z} \mid \alpha)\big] \\ &- \mathrm{KL}\big[q(\boldsymbol{A}) \parallel p(\boldsymbol{A} \mid \sigma_A)\big] \\ &- \sum_{r\geq 1} \mathrm{KL}\Big[q(\boldsymbol{X}_r(\boldsymbol{v})) \parallel p(\boldsymbol{X}_r(\boldsymbol{v}) \mid \kappa_{\mathcal{X}}, \kappa_{\mathcal{U}})\Big]\end{aligned} \tag{8}$$

Closed Form

We derive a closed form of the last term as:

$$\begin{gathered} 2\mathrm{KL}\Big[q(\boldsymbol{X}_r(\boldsymbol{v})) \parallel p(\boldsymbol{X}_r(\boldsymbol{v}) \mid \kappa_{\mathcal{X}}, \kappa_{\mathcal{U}})\Big] \\ = \mathrm{trace}\Big((\boldsymbol{\Sigma}_{\mathcal{X}}^{-1} \otimes \boldsymbol{\Sigma}_{\mathcal{U}}^{vv^{-1}})(\boldsymbol{S}_r + \mathrm{vec}(\boldsymbol{\mu}_r)\mathrm{vec}(\boldsymbol{\mu}_r)^T) \Big) \\ + K \log |\boldsymbol{\Sigma}_{\mathcal{X}}| + n \log |\boldsymbol{\Sigma}_{\mathcal{U}}^{vv}| - \sum^{n} \log |\boldsymbol{S}_{tr}| - nK \end{gathered} \tag{9}$$

Key Theorems for Efficient Sampling

Theorem 1: Posterior Mean Independence

- The mean function of the posterior for $X_{tr}(\cdot)$ is solely dependent on the variational mean of $\mathbf{X}_{tr}(\mathbf{v})$, the inducing variables at time t.
- $$\mathbb{E}\left[X_{tr}(\mathbf{u})\right] = \Sigma_{\mathcal{U}}^{uv}(\Sigma_{\mathcal{U}}^{vv})^{-1}\mu_{tr}$$

Key Theorems for Efficient Sampling

Theorem 2: Posterior Variance Decomposition

- The variance function of the posterior for $\mathbf{X}_r(\cdot)$ consists of two parts.
- The first part is dependent on the variational variance of $\mathbf{X}_{tr}(\mathbf{v})$.
- The second part is independent of the variational distributions of all inducing variables.
- $$\begin{aligned}\mathrm{Var}_q\left[\mathrm{vec}\left(\mathbf{X}_r(\mathbf{u})\right)\right] &= \left(I \otimes \Sigma_{\mathcal{U}}^{uv}(\Sigma_{\mathcal{U}}^{vv})^{-1}\right)\mathrm{diag}(\mathbf{S}_{1r}, \ldots, \mathbf{S}_{nr}) \\ &\quad + \Sigma_{\mathcal{X}} \otimes \left(\Sigma_{\mathcal{U}}^{uu} - \Sigma_{\mathcal{U}}^{uv}(\Sigma_{\mathcal{U}}^{vv})^{-1}(\Sigma_{\mathcal{U}}^{uv})^{\top}\right)\end{aligned}$$

Key Theorems for Efficient Sampling

Theorem 3: Irrelevance to ELBO

- Sampling $\mathbf{X}_{tr}(\cdot)$ from the distribution of $\tilde{\mathbf{X}}_r^{(1)}(\cdot)$ does not change the variational mean.
- The corresponding ELBO is only modified by a constant term.
- $$\frac{1}{2\sigma_\epsilon^2}\|\mathbf{Z}\odot\mathbf{A}\|_F^2\,\mathrm{trace}\left[\Sigma_{\mathcal{X}}\right]\mathrm{trace}\left[\Sigma_{\mathcal{U}}^{uu}-\Sigma_{\mathcal{U}}^{uv}(\Sigma_{\mathcal{U}}^{vv})^{-1}(\Sigma_{\mathcal{U}}^{uv})^{\top}\right]$$

Training and Prediction

Training

- Utilize Automatic Differentiation Variational Inference (ADVI) to optimize the variational parameters.
- Compute the gradient of the Evidence Lower Bound (ELBO) with respect to the parameters.
- Iterate the following steps until ELBO converges:
  - Update variational distribution parameters μ_{tr} and S_{tr} for inducing variables $X_{tr}(\mathbf{v})$.
  - Update variational parameters for the Indian Buffet Process ($\{\tau_j^1, \tau_j^2\}_{1 \le j \le M}$ and $\{m_{tj}\}_{1 \le t \le n, 1 \le j \le M}$) and loading weight matrix ($\{\eta_{tj}, \sigma_{tj}^A\}_{1 \le t \le n, 1 \le j \le M}$).
  - Update the idiosyncratic noise scale σ_ϵ and parameters in the spatial kernel $\kappa_{\mathcal{U}}(\cdot, \cdot)$.

Training and Prediction

Prediction

- Once the model is trained, generate a posterior distribution based on the observed data up to time n.
- Make predictions for future time steps based on this distribution.
- **One-step ahead prediction:**

$$\bar{\mathbf{Y}}_{n+1}(\mathbf{u}) = (\bar{\mathbf{Z}} \odot \bar{\mathbf{A}})\bar{\mathbf{X}}_{n+1}(\mathbf{u})$$

where

$$\bar{X}_{n+1,r}(\mathbf{u}) = \Sigma_{\mathcal{U}}^{uv}(\Sigma_{\mathcal{U}}^{vv})^{-1}\mu_r\Sigma_{\mathcal{X}}^{-1}(\Sigma_{\mathcal{X}}^{n+1,1:n})^{\top}$$

Experiments

Datasets

We applied DF^{2}M to four real-world datasets consisting of high-dimensional functional time series:

- **Japanese Mortality**
  - Age-specific mortality rates for 47 Japanese prefectures.
  - Time span: 1975 to 2017 ($p = 47$, $n = 43$).
- **Energy Consumption**
  - Half-hourly measured energy consumption curves for London households.
  - Time span: December 2012 to January 2013 ($p = 40$, $n = 55$).
- **Global Mortality**
  - Age-specific mortality rates across 32 countries.
  - Time span: 1960 to 2010 ($p = 32$, $n = 50$).
- **Stock Intraday**
  - High-frequency price observations for the S&P 100 component stocks.
  - Time span: 2017, with ten-minute resolution prices ($p = 98$, $n = 45$).

Experiments Setup and Metrics

Experimental Setup

- The data is split into a training set with the first n_1 periods and a test set with the last n_2 periods.
- For each integer $h > 0$, we make the h-step-ahead prediction using the fitted model on the first n_1 periods.
- The process is repeated by moving the training window by one period, refitting the model, and making new predictions.

Experiments Setup and Metrics

Evaluation Metrics

We use two metrics to assess the predictive accuracy of the model:

- **Mean Absolute Prediction Error (MAPE)**

$$\mathrm{MAPE}(h) = \frac{1}{M} \sum_{j=1}^{p} \sum_{k=1}^{K} \sum_{t=n_1+h}^{n} \left| \hat{Y}_{tj}(u_k) - Y_{tj}(u_k) \right|$$

- **Mean Squared Prediction Error (MSPE)**

$$\mathrm{MSPE}(h) = \frac{1}{M} \sum_{j=1}^{p} \sum_{k=1}^{K} \sum_{t=n_1+h}^{n} \left[\hat{Y}_{tj}(u_k) - Y_{tj}(u_k) \right]^2$$

- Where:
  - $M = Kp(n_2 - h + 1)$ is the total number of predictions.
  - $\hat{Y}_{tj}(u_k)$ is the predicted value.
  - $Y_{tj}(u_k)$ is the actual value.

Experiments Setup and Metrics

DF^{2}M Variants

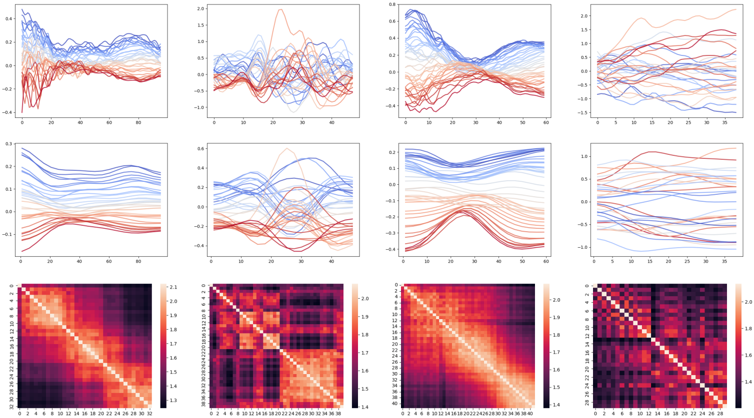
- DF^{2}M-LIN: Linear model
- DF^{2}M-LSTM: Long Short-Term Memory
- DF^{2}M-GRU: Gated Recurrent Unit
- DF^{2}M-ATTN: Attention Mechanism

Empirical Results: Explainability

Explainability of DF^2M

- **Temporal Dynamics of Largest Factors**
  - Observed a decreasing trend over time in the largest factors for the first three datasets.
  - Factors exhibit clear and smooth dynamics, aiding in robust predictions and understanding underlying changes.
- **Temporal Covariance Matrix ($\Sigma_{\mathcal{X}}$)**
  - Strong autocorrelation in the first three datasets compared to the *Stock Intraday* dataset.
  - Mortality datasets show strong autoregressive and blockwise patterns indicating change points in the 1980s.
  - *Energy Consumption* dataset reveals periodic patterns distinguishing weekdays and weekends during the first 21 days.

Empirical Results: Explainability



(a) *Japanese mortality* (b) *Energy consumption* (c) *Global mortality* (d) *Stock Intraday*

Figure 3: A visualization of real datasets with analysis. Row (1): raw functional time series. Row (2): the largest functional factor. Row (3): temporal covariance matrix. Rows (1) and (2) use a blue-to-red gradient to denote time progression. Blue for older and red for recent data. Row (3) employs brightness variations to represent covariance, with brighter areas indicating higher covariance.

Empirical Results: Predictive Accuracy

Predictive Accuracy of DF^{2}M

- DF^{2}M outperforms standard deep learning models in terms of both MSPE and MAPE across all datasets, except *Stock Intraday* where DF^{2}M-ATTN and ATTN achieve similar accuracy.
- **DF^{2}M-LSTM:**
  - Best performance on *Energy Consumption* and *Global Mortality* datasets.
- **DF^{2}M-ATTN:**
  - Lowest prediction error for *Japanese Mortality* dataset.
- **DF^{2}M-LIN:**
  - Outperforms DF^{2}M-LSTM and DF^{2}M-GRU on *Stock Intraday* dataset, suitable for financial data.

Comparison

- DF^{2}M achieves better or comparable results to standard deep learning models.

Empirical Results: Predictive Accuracy

Table 1: Comparison of DF^{2}M to Standard Deep Learning Models. For formatting reasons, MAPEs are multiplied by 10, and MSPEs are multiplied by 10^2, except for the *Energy Consumption* dataset.

(a) Comparison of DF^{2}M-LIN and LIN

		Japanese Mortality			Energy Consumption			Global Mortality			Stock Intraday		
h		1	2	3	1	2	3	1	2	3	1	2	3
DF^{2}M-LIN	MSPE	**4.707**	**4.567**	**5.623**	**10.29**	**17.58**	**17.64**	**10.78**	**9.300**	**9.706**	**99.58**	**101.2**	**89.82**
	MAPE	**1.539**	**1.446**	**1.635**	**2.334**	**3.060**	**3.100**	**2.319**	**2.041**	**2.106**	**6.424**	**6.505**	**6.269**
LIN	MSPE	7.808	8.774	9.228	16.16	18.95	20.27	16.84	18.05	19.93	137.5	127.8	139.1
	MAPE	2.092	2.227	2.313	2.939	3.214	3.342	2.783	2.949	3.174	7.896	7.491	7.924

(b) Comparison of DF^{2}M-LSTM and LSTM

		Japanese Mortality			Energy Consumption			Global Mortality			Stock Intraday		
h		1	2	3	1	2	3	1	2	3	1	2	3
DF^{2}M-LSTM	MSPE	**3.753**	**4.164**	**4.513**	**8.928**	**11.60**	**17.26**	**7.672**	**8.088**	**8.954**	**107.5**	**118.8**	**113.6**
	MAPE	**1.205**	**1.322**	**1.427**	**2.176**	**2.478**	**3.063**	**1.726**	**1.823**	**1.978**	**6.741**	**7.141**	**7.294**
LSTM	MSPE	4.989	5.597	6.501	13.51	19.71	24.61	13.28	16.29	17.08	193.3	176.0	213.8
	MAPE	1.447	1.523	1.684	2.635	3.278	3.759	2.332	2.572	2.680	9.281	9.283	10.20

(c) Comparison of DF^{2}M-GRU and GRU

		Japanese Mortality			Energy Consumption			Global Mortality			Stock Intraday		
h		1	2	3	1	2	3	1	2	3	1	2	3
DF^{2}M-GRU	MSPE	**4.092**	**4.395**	**4.898**	**9.132**	**8.714**	**9.730**	**8.741**	**8.714**	**9.730**	**102.5**	**117.3**	**95.49**
	MAPE	**1.318**	**1.402**	**1.537**	**2.204**	**1.951**	**2.110**	**1.967**	**1.951**	**2.110**	**6.675**	**7.339**	**6.649**
GRU	MSPE	8.800	8.552	10.41	15.55	24.02	17.53	14.12	15.33	17.53	414.0	445.9	427.2
	MAPE	1.691	1.809	1.865	2.872	3.518	2.597	2.211	2.403	2.597	14.12	14.66	14.07

(d) Comparison of DF^{2}M-ATTN and ATTN

		Japanese Mortality			Energy Consumption			Global Mortality			Stock Intraday		
h		1	2	3	1	2	3	1	2	3	1	2	3
DF^{2}M-ATTN	MSPE	**3.608**	**3.839**	**3.985**	**14.22**	18.70	19.03	**14.22**	**18.70**	**19.03**	104.2	103.4	93.93
	MAPE	**1.119**	**1.203**	**1.264**	**2.741**	**3.141**	**3.163**	**2.741**	**3.141**	**3.163**	6.695	6.646	6.427
ATTN	MSPE	13.44	14.85	16.17	17.03	**17.79**	**18.24**	39.52	41.83	43.95	**103.4**	**98.39**	**91.21**
	MAPE	3.166	3.363	3.546	3.130	3.216	3.268	5.332	5.506	5.643	**6.579**	**6.392**	**6.257**

Conclusion

- Introduced DF^2M, a deep Bayesian nonparametric approach for high-dimensional functional time series.
- Combines Indian Buffet Process, Factor Model, Gaussian Process, and Deep Neural Networks.
- Captures non-Markovian and nonlinear dynamics while maintaining explainability.
- Superior predictive performance compared to conventional deep learning models.
- Achieves explainability in neural network utilization.
- Efficient computational approach with proposed inference algorithm.

References

Chang, J., Chen, C., Qiao, X., and Yao, Q. (2023a). An autocovariance-based learning framework for high-dimensional functional time series. *Journal of Econometrics*.

Chang, J., Fang, Q., Qiao, X., and Yao, Q. (2023b). On the modelling and prediction of high-dimensional functional time series. *Working Paper*.

Guo, S. and Qiao, X. (2023). On consistency and sparsity for high-dimensional functional time series with application to autoregressions. *Bernoulli*, 29(1):451–472.

Guo, S., Qiao, X., and Wang, Q. (2021). Factor modelling for high-dimensional functional time series. *arXiv:2112.13651*.

Wang, J., Hertzmann, A., and Fleet, D. J. (2005). Gaussian process dynamical models. In *Advances in Neural Information Processing Systems*, volume 18.

Acknowledgements

This paper was prepared for informational purposes by the CDAO group of JPMorgan Chase & Co and its affiliates ("J.P. Morgan") and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.