SPATIO-TEMPORAL DECOMPOSITION: A PLUG-IN INTERPRETABLE FRONT END FOR MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Forecasting models for spatio-temporal series usually either ignore cross-location structure or encode it with a graph that is fixed in advance and hard to read. We propose Spatio-Temporal Decomposition (STD), a module that sits in front of any forecasting backbone and splits each observation into four named components, namely local and global in space and high and low frequency in time. The split uses only means over time and over locations plus two sets of learned offsets, one per location and one per time step, so it introduces no additional hyperparameters beyond those of the backbone. We show that the per-location offset converges to the correction a piecewise-stationary trend requires, and that the per-time offset acts as an implicit set of spatial weights, so the plain average over locations becomes a learned neighborhood without a graph. Across TCN, Transformer and WaveNet backbones on BikeNYC, SKT, and Traffic, STD lowers MAE in 26 of 27 backbone, dataset and horizon combinations, cuts MAE summed over the three horizons by 19 to 29% on BikeNYC, and lifts graph-free backbones above MTGNN and StemGNN. On BikeNYC the global high-frequency part shows two commuting peaks on weekdays and one on weekends, indicating that STD surfaces readable structure that graph-based models encode only implicitly. Ablation experiments consistently show that removing spatial aggregation increases mean RMSE, highlighting the contribution of spatial information to forecasting accuracy. Robustness evaluations on Capital further show that STD maintains stable performance under Gaussian noise and achieves the lowest MAE and RMSE among competing models at 20% missingness.

Keywords: Spatio-temporal forecasting, Multivariate time series, Interpretable deep learning, Plug-in module

1 INTRODUCTION

As the scales and complexities of various data continue to grow, multivariate time series (MTS) forecasting has become increasingly vital across various fields, such as business (Chen et al., 2023), economics (Kalaitzi & Chamberlain, 2020), transportation (Cao et al., 2020), and environment (Wen et al., 2021).

To date, two major approaches have been employed for MTS forecasting: traditional statistical methods (Makridakis & Hibon, 1997; Toda, 1991; Chudik & Pesaran, 2016) and deep learning-based methods (Schuster & Paliwal, 1997; Hochreiter & Schmidhuber, 1997; Cho et al., 2014). However, traditional time series prediction algorithms, including ARIMA (Makridakis & Hibon, 1997), VAR (Toda, 1991), and GVAR (Chudik & Pesaran, 2016) rely on stationary and linear assumptions and struggle to handle large-scale, heterogeneous data effectively. In contrast, deep neural network-based methods have been proposed to address nonlinear, non-stationary, and high-dimensional time series forecasting (Long et al., 2022). However, these neural network models cannot capture spatial information correlations among variables.

To address this issue, a series of approaches have been proposed, such as convolutional neural network (CNN)-based (Oord et al., 2016; Bai et al., 2018), recurrent neural network (RNN)-based (Salinas et al., 2020; Lai et al., 2018), and graph-based methods (Wu et al., 2019; Yu et al., 2017;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Guo et al., 2019; Song et al., 2020; Cao et al., 2020). However, these models lack explainability, which is a crucial requirement for practical applications. Therefore, developing an interpretable MTS forecasting method capable of effectively exploring spatial and temporal information in time series remains a challenging problem.

This paper proposes a novel mechanism for interpreting MTS forecasting called Space-Temporal Decomposition (STD). STD uses a summation method to consider spatial and temporal information for comprehensive modeling. It comprises spatial and temporal structures with global and local components and high-frequency and low-frequency components, reflecting the potential mechanism of the time series. Furthermore, the proposed approach can be integrated into most MTS networks to enhance performance.

2 Related Work

The MTS prediction has achieved substantial improvements with the help of advances in deep learning over the past few decades (Cai et al., 2024; Qiu et al., 2025). Three common approaches, namely CNN-based, RNN-based, and graph-based methods, have been widely designed for MTS data. However, most MTS forecasting approaches utilizing deep learning methods are black-box models without interpretability, which may limit their use in high-stakes applications.

Interpretable time series forecasting is a challenging task that has been the focus of much research in recent years. Several post-hoc interpretability methods, such as LIME (Ribeiro et al., 2016), Integrated Gradients (Sundararajan et al., 2017), DeepLift (Shrikumar et al., 2017), and SHAP (Lundberg & Lee, 2017), have become mainstream due to their convenience, broad applicability, and ability to provide further insights into the model's decision-making process. Although there have been quantitative evaluations of these methods (Turbé et al., 2023), these post-hoc interpretability approaches are often computationally expensive and inefficient for large-scale time series data (Ozyegen et al., 2023). Another approach is to transfer attention methods to explain the relative importance of different time steps, but these methods are based on intermediate feature importance and ignore the spatio-temporal information (Serrano & Smith, 2019). Despite these efforts, including SCNN's structured component modeling (Deng et al., 2024), using temporal and spatial decompositions in both orders to construct interpretable inputs for different forecasting backbones remains underexplored. STD addresses this gap.

3 Methodology

3.1 Problem Formulation

We analyze a multivariate time series system with N spatial locations observed over T time steps, where each observation $X_{jt} \in \mathbb{R}$ represents the measured value at location j and time t.

3.2 Spatio-Temporal Decomposition Framework

Our framework decomposes multivariate time series through two fundamental operations: temporal decomposition (TD) and spatial decomposition (SD). The temporal decomposition separates each time series into high-frequency and low-frequency components. For location j at time t, we compute:

$$\begin{cases} X_{jt}^{\text{high}} = X_{jt} - \overline{X}_j^{\text{temp}} \\ X_j^{\text{low}} = \overline{X}_j^{\text{temp}} + \theta_j \end{cases}, \tag{1}$$

where $\overline{X}_j^{\text{temp}} = \frac{1}{T}\sum_{t=1}^{T} X_{jt}$ represents the temporal mean, capturing global patterns at location j, and θ_j is a trainable parameter that refines the low-frequency component. The high-frequency component X_{jt}^{high} captures short-term variations around this baseline.

The spatial decomposition, conversely, distinguishes between local and global patterns:

$$\begin{cases} X_{jt}^{\text{loc}} = X_{jt} - \overline{X}_t^{\text{spat}} \\ X_t^{\text{glo}} = \overline{X}_t^{\text{spat}} + \phi_t \end{cases}, \tag{2}$$

where $\overline{X}_t^{\text{spat}} = \frac{1}{N}\sum_{j=1}^{N} X_{jt}$ computes the spatial mean (representing global patterns) at time t, with ϕ_t serving as a trainable adjustment to the global component. The local component X_{jt}^{loc} captures location-specific deviations from these system-wide patterns.

These decomposition operations can be applied in different sequences to extract distinct component representations. When applying temporal decomposition followed by spatial decomposition (TD→SD), we first obtain the high-frequency and low-frequency components, then further decompose each spatially:

$$X_{jt}^{\text{high,loc}} = X_{jt} - \overline{X}_j^{\text{temp}} - \overline{X}_t^{\text{spat,high}} - \phi_t^{\text{high}}, \tag{3}$$

$$X_t^{\text{high,glo}} = \overline{X}_t^{\text{spat,high}} + \phi_t^{\text{high}}, \tag{4}$$

$$X_j^{\text{low,loc}} = \overline{X}_j^{\text{temp}} + \theta_j - \overline{X}_t^{\text{spat,low}} - \phi_t^{\text{low}}, \tag{5}$$

$$X_t^{\text{low,glo}} = \overline{X}_t^{\text{spat,low}} + \phi_t^{\text{low}}. \tag{6}$$

Conversely, applying spatial decomposition first (SD→TD) yields an alternative decomposition path:

$$X_{jt}^{\text{loc,high}} = X_{jt} - \overline{X}_t^{\text{spat}} - \overline{X}_j^{\text{temp,loc}} - \theta_j^{\text{loc}}, \tag{7}$$

$$X_j^{\text{loc,low}} = \overline{X}_j^{\text{temp,loc}} + \theta_j^{\text{loc}}, \tag{8}$$

$$X_t^{\text{glo,high}} = \overline{X}_t^{\text{spat}} + \phi_t - \overline{X}^{\text{temp,glo}} - \theta_j^{\text{glo}}, \tag{9}$$

$$X_j^{\text{glo,low}} = \overline{X}^{\text{temp,glo}} + \theta_j^{\text{glo}}. \tag{10}$$

3.3 Integrated Spatio-Temporal Forecasting

As illustrated in Figure 1, the feature construction begins by merging components from both TD→SD and SD→TD decomposition pathways. Each spatio-temporal point is represented by a 9-dimensional feature vector that comprehensively captures all frequency-locality combinations:

$$Z_{jt} = \left[X_{jt}^{\text{high,loc}}, X_t^{\text{high,glo}}, X_j^{\text{low,loc}}, X_t^{\text{low,glo}}, X_{jt}^{\text{loc,high}}, X_j^{\text{loc,low}}, X_t^{\text{glo,high}}, X_j^{\text{glo,low}}, X_{jt}\right]^\top \in \mathbb{R}^9. \tag{11}$$

The forecasting model then processes historical sequences of these feature vectors to generate predictions. Using $\{Z_{j,t-k}\}_{k=0}^{T_{\text{hist}}-1}$ as input, where T_{hist} is the look-back window length, the model outputs predictions for H future time steps:

$$\hat{Y}_{t+1:t+H} = f_\Theta(\{Z_{j,t-k}\}_{k=0}^{T_{\text{hist}}-1}). \tag{12}$$

The training minimizes a prediction loss function that measures forecast accuracy:

$$\mathcal{L}_{\text{pred}} = \frac{1}{H}\sum_{h=1}^{H}(\hat{Y}_{t+h} - Y_{t+h})^2. \tag{13}$$

And the framework is trained end-to-end by minimizing $\mathcal{L}_{\text{pred}}$ through gradient descent:

$$\Theta^*, \{\theta_j\}, \{\phi_t\} = \arg\min_{\Theta,\theta_j,\phi_t} \mathcal{L}_{\text{pred}}, \tag{14}$$

where both decomposition parameters (θ_j, ϕ_t) and model parameters (Θ) are jointly optimized.

4 Theoretical Foundations

4.1 Adaptive Trend Component Analysis

To rigorously characterize the role of θ_j, we formalize the data generation process through the following assumptions:

Assumption 1 (Semi-Stationary Trend)**.** *The observed time series decomposes as*

$$X_{jt} = \mathcal{T}_j(t) + S_{jt} + \epsilon_{jt}, \tag{15}$$

where $\mathcal{T}_j(t)$ is a piecewise-stationary trend satisfying $\mathbb{E}[\mathcal{T}_j(t)] = \mu_j^{(k)}$ over $t \in [t_k, t_{k+1})$ for stationary segments $k = 1, ..., K_j$, where K_j denotes the total number of stationary segments for series j, S_{jt} captures stationary seasonal/cyclic components, and $\epsilon_{jt} \sim \mathcal{N}(0, \sigma_j^2)$ is white noise.

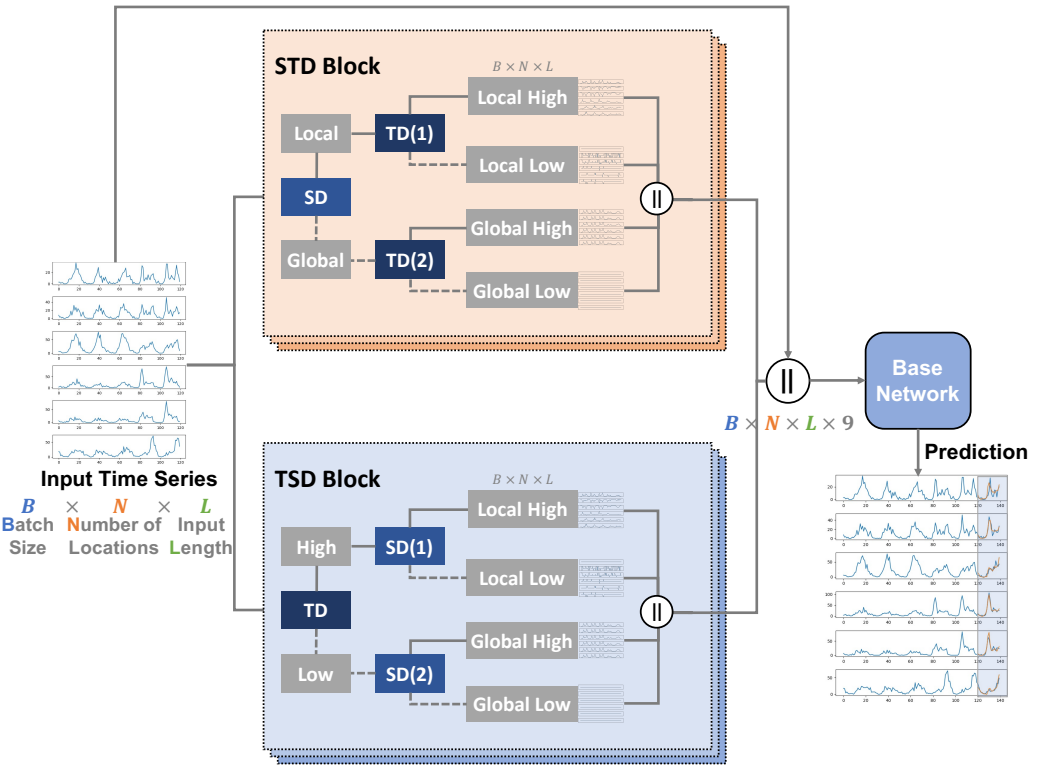

Figure 1: The framework of the STD for interpretively spatio-temporal prediction. The input sequences are split into a shape of $B \times N \times L$, and passed through an SD module to obtain local and global components. The local and global components are then processed by two TD modules to yield four components. The resulting eight channels are merged with the original sequence and inputted into the subsequent model network.

Assumption 2 (Forecast-Relevant Trend)**.** *The predictive relationship satisfies:*

$$Y_{t+h} = g\left(\{\mathcal{T}_j(t)\}_{j=1}^{N}\right) + \delta_{t+h}, \quad \delta_{t+h} \perp\!\!\!\perp \{X_{jt}\}, \tag{16}$$

where g is a Lipschitz-continuous function aggregating multi-location trends, and $\perp\!\!\!\perp$ denotes statistical independence between the error term δ_{t+h} and the predictors $\{X_{jt}\}$.

Under these assumptions, θ_j addresses two key limitations of the empirical temporal mean:

Theorem 1 (Trend Adaptation)**.** *Let $\hat{\mu}_j = \frac{1}{T}\sum_{t=1}^{T} X_{jt}$. For any location j with K_j stationary segments:*

$$\theta_j^* = \sum_{k=1}^{K_j} \frac{t_{k+1} - t_k}{T}(\mu_j^{(k)} - \hat{\mu}_j) + O\left(\sqrt{\frac{\log T}{T}}\right), \tag{17}$$

where θ_j^* minimizes the expected prediction error under Assumptions 1-2.

4.2 Spatial Dependency Encoding

To formalize the role of global components in capturing spatial dependencies, we establish the following modeling framework:

Assumption 3 (Spatial Interaction)**.** *The prediction target Y_{jt} depends on both local and neighboring signals through:*

$$Y_{jt} = f\left(X_{jt}, \sum_{k \in \mathcal{N}(j)} w_{jk} X_{kt}\right) + \epsilon_{jt}, \tag{18}$$

where f is an unknown differentiable function, $\mathcal{N}(j)$ denotes the neighborhood of location j, and w_{jk} are convex interaction weights satisfying $w_{jk} \geq 0$, $\sum_k w_{jk} = 1$, and $\epsilon_{jt} \sim \mathcal{N}(0, \sigma_j^2)$ is independent white noise.

The global component's capacity to encode spatial dependencies is characterized by:

Theorem 2 (Implicit Weight Learning)**.** *Let the global component be constructed as:*

$$X_t^{glo} = \frac{1}{N}\sum_{k=1}^{N} X_{kt} + \phi_t = \sum_{k=1}^{N}\left(\frac{1}{N} + \delta_{jk}(t)\right) X_{kt}. \tag{19}$$

Under Assumption 3, the global component can approximate the true spatial weights w_{jk} through trainable residual adjustment $\delta_{jk}(t)$ via ϕ_t, satisfying:

$$\sum_{k=1}^{N} \delta_{jk}(t) X_{kt} \approx \phi_t. \tag{20}$$

4.3 Optimization Convergence

The optimization framework operates on the trainable parameters

$$\omega = [\Theta, \{\theta_j\}_{j=1}^{N}, \{\phi_t\}_{t=1}^{T}] \in \mathcal{W} \subseteq \mathbb{R}^d, \tag{21}$$

where $\mathcal{W}$ denotes the compact parameter space constrained by L2 regularization. The stochastic optimization objective minimizes the expected loss:

$$\min_{\omega} \mathcal{L}(\omega) = \mathbb{E}_{\xi \sim \mathcal{D}}\left[\ell(\omega; \xi)\right]. \tag{22}$$

Here ξ represents randomly sampled spatiotemporal data points from distribution $\mathcal{D}$, $\ell(\omega; \xi)$ computes the loss for a single data point, and $\mathbb{E}_\xi$ denotes expectation over the data distribution.

Theorem 3 (Non-Convex Convergence)**.** *Consider the optimization problem $\min_{\omega \in \mathcal{W}} \mathcal{L}(\omega)$ where the gradient $\nabla\mathcal{L}$ is L-Lipschitz continuous, satisfying $\|\nabla\mathcal{L}(\omega_1) - \nabla\mathcal{L}(\omega_2)\| \leq L\|\omega_1 - \omega_2\|$ for all $\omega_1, \omega_2 \in \mathcal{W}$. Assume the stochastic gradient variance is bounded as $\mathbb{E}[\|\nabla\ell(\omega; \xi) - \nabla\mathcal{L}(\omega)\|^2] \leq \sigma^2$, and let $D_{\mathcal{L}}^2 = \mathbb{E}[\mathcal{L}(\omega_1) - \mathcal{L}^*]$ denote the initial optimality gap.*

Using step sizes $\eta_k = \eta/\sqrt{K}$ with $\eta < 1/(2L)$ and selecting R uniformly from $\{1, ..., K\}$, stochastic gradient descent (SGD) achieves the convergence rate:

$$\mathbb{E}\left[\|\nabla\mathcal{L}(\omega_R)\|^2\right] \leq \frac{D_{\mathcal{L}}^2 + \frac{L\eta^2}{2}\sigma^2}{\eta\sqrt{K} - \frac{L\eta^2}{2}} = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right). \tag{23}$$

This step size policy satisfies the Robbins-Monro conditions: $\sum_{k=1}^{\infty} \eta_k = \infty$ and $\sum_{k=1}^{\infty} \eta_k^2 < \infty$, guaranteeing convergence.

5 Experiment Results

5.1 Experiment Settings

We use three real datasets for the main forecasting comparison. Elaborate information on the network architectures and hyperparameters can be found in the Supplementary Material. The original STD experiments were conducted on an Nvidia Station workstation with a Tesla P100 GPU; hardware for the additional experiments is specified in Appendix F.1.

5.2 Simulation

We propose a method (Hewamalage et al., 2022) for generating spatiotemporal data. The simulation study encompasses three distinct spatiotemporal time series generating processes: Fourier terms, ARIMA(2,1,2), and Mackey-Glass equations, each contributing to the observed level x_{it} at spatial

location i and time t. Common parameters such as error variances and weighting coefficients are shared across all models.

All the series are scaled before they are integrated into $x_{j,t}$:

$$x_{j,t} = w_1 x_{j,t}^{\text{glo,high}} + w_2 x_{j,t}^{\text{glo,low}} + w_3 x_{j,t}^{\text{loc,high}} + w_4 x_{j,t}^{\text{loc,low}} + \varepsilon_{j,t}. \tag{24}$$

The process $\varepsilon_{j,t}$ represents typical Gaussian instrumental error $N(0,1)$, which is white noise in both space and time with variance σ_ε^2. The high frequency parts are generated by the three models while the low-frequency parts (global low frequency and local low frequency) follow a piecewise function, with each piece adhering to an ARIMA(2,1,2) model. By assigning different weights to each component, we simulate scenarios in which individual parts contribute differently to the spatio-temporal time series. The interpretability of the model is typically evaluated based on how closely the estimated series aligns with the simulated series.

5.2.1 Decomposition Analysis

In our experiments, we primarily evaluated the interpretability of STD through qualitative measures rather than quantitative metrics. The trends of the generated and estimated local high-frequency and global high-frequency components are displayed and compared respectively in Figure 4. It is evident that as the weight of either the local high or global high component among the four components increases, the resemblance between the estimated series and the actual one becomes more pronounced.

5.3 Experiments on Real-World Data

Experimental setup. The STD evaluation uses $L = 16$, $H = 3$ on BikeNYC, Traffic, and SKT (hourly for BikeNYC/Traffic; six-hourly for SKT). Archived STD training uses Adam, learning rate 10^{-4}, and batch size four. Dataset splits and lead-specific results appear in Appendices F.1 and D. The STD framework combines temporal and spatial decomposition with implementation-specific processing modules, including window normalization and learned softmax spatial pooling.

Baseline models. The comparison includes LSTM, GRU, LSTNet, MTGNN (Wu et al., 2020), StemGNN, and paired TCN, Transformer, and WaveNet predictors with and without STD. For the additional four-view implementation, Raw4 feeds one quarter of the normalized input to each matched prediction head. Appendix F.1 specifies the native baseline architectures and four-view predictors.

Performance comparison. Table 1 averages each metric over three leads. STD decreases both averages for all nine dataset–backbone pairs; BikeNYC MAE falls by 18.6%, 28.8%, and 23.9% for TCN, Transformer, and WaveNet. Across individual leads, MAE improves in 26/27 pairs: SKT TCN at 18 hours increases from 0.722 to 0.729. All 27 RMSE comparisons improve. At $L = 336$, $H = 3$, STD reduces the compact TCN's three-seed mean MAE/RMSE on BikeNYC by 4.10%/4.04% relative to Raw4. Complete STD short-horizon and $H = 192$ development comparisons appear in Appendix F.2.

Interpretability analysis. Across the nine archived dataset–backbone pairs in Table 2, STD lowers weekly-band, common, and local errors. Daily-band errors decrease on BikeNYC and Traffic. The Capital STD case in Figure 2 shows time-group separation of 34.43%, 69.12%, and 1.20% for input, common, and local representations. Appendix F.4 gives the definitions and full Capital comparisons; the BikeNYC component trajectories appear in Appendix E.

Ablation study. Sixty fits compare STD with temporal removal, spatial removal, uniform pooling, and removal of both axes. Spatial removal increases RMSE in all four dataset–head groups, while temporal removal has mixed effects. Uniform pooling performs best on Capital for MAE. Complete results appear in Appendix F.5.

Hyperparameter analysis. Across 75 configurations and 225 fits, Figure 3 varies history length, scorer width, head width, and dropout. With common training origins, STD improves Raw4 RMSE at all five histories on BikeNYC/Capital and four on Traffic. Appendix F.5 gives the factor grids and scoring protocol.

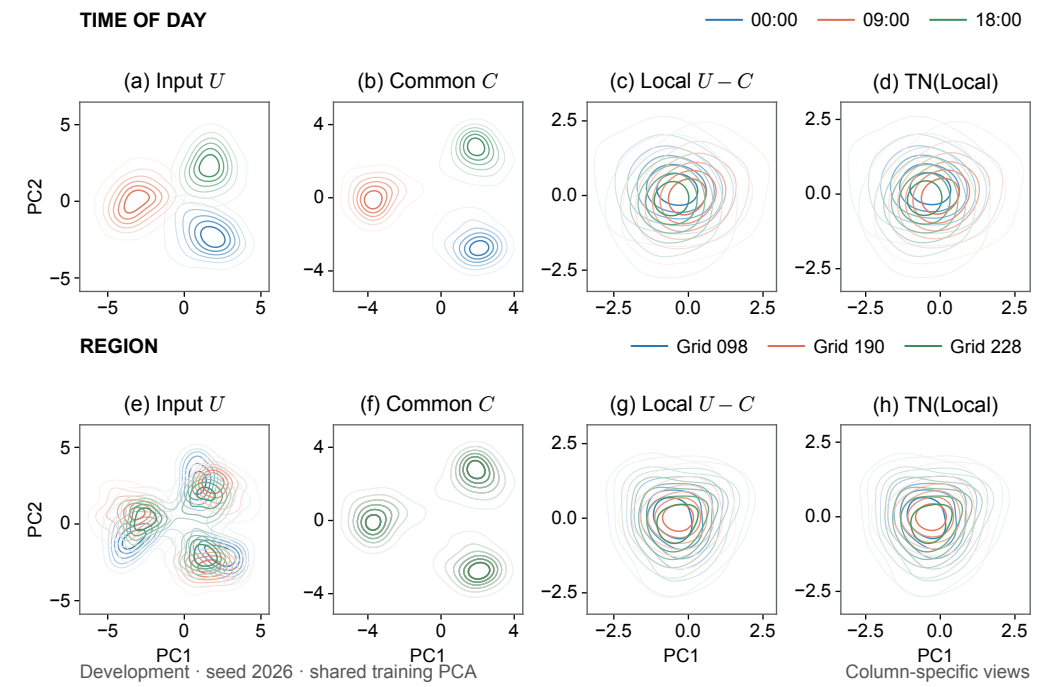

Figure 2: STD representations on Capital development data (seed 2026). Rows group time of day and region. Shared training PCA and fixed-bandwidth KDE; contours enclose 15–90% probability mass. Axis ranges differ across columns.

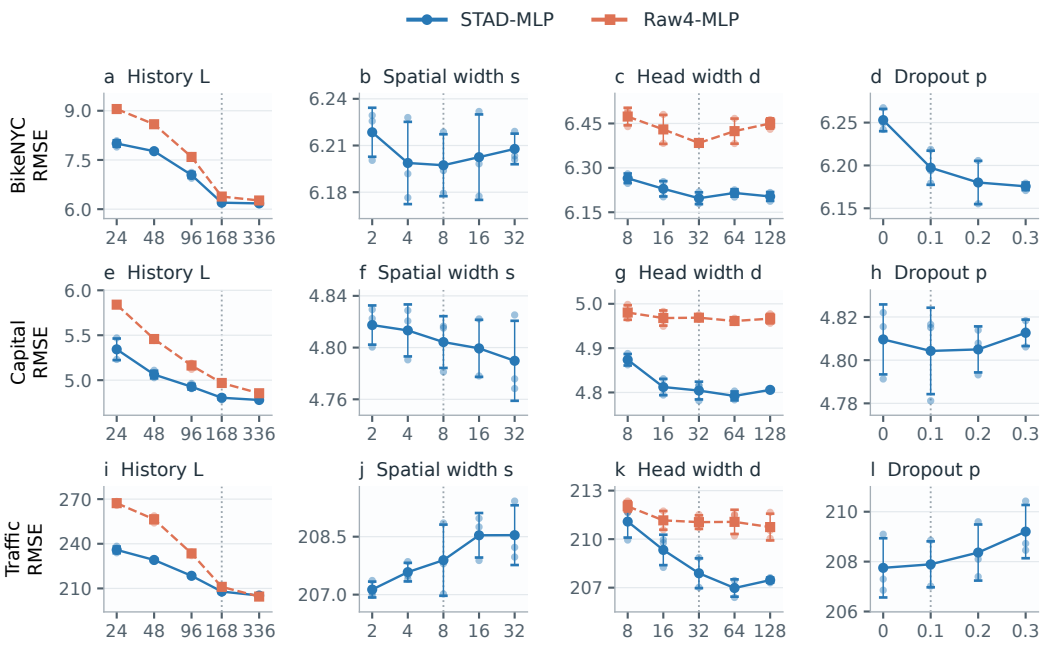

Figure 3: Complete sensitivity matrix: datasets by row and factors by column. Lines show three-seed mean RMSE; bars show sample standard deviations. Raw4 is paired on history and head-width sweeps. Axis ranges vary across panels.

Table 1: STD forecasting errors: arithmetic means of the three reported lead-specific errors. Leads are 1/2/3 hours for BikeNYC and Traffic, and 6/12/18 hours for SKT. Display multipliers follow the original tables. Bold marks the lowest column mean.

	BikeNYC		Traffic ($\times 10^{-2}$)		SKT ($\times 10^{2}$)	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM	5.193	10.810	1.750	3.243	0.953	1.523
GRU	5.077	10.877	1.667	3.197	0.755	1.273
LSTNet	4.043	9.267	1.707	3.293	3.077	4.487
MTGNN	3.533	7.800	1.285	2.673	0.825	1.257
StemGNN	3.520	7.713	1.310	2.603	0.751	1.287
TCN	3.873	8.167	1.433	2.857	0.682	1.183
TCN+STD	3.153	6.327	1.134	2.360	0.680	1.157
Transformer	4.033	8.640	1.460	2.967	0.707	1.203
Transformer+STD	**2.870**	**5.823**	1.123	2.377	0.655	1.137
WaveNet	3.890	8.347	1.349	2.767	0.681	1.180
WaveNet+STD	2.960	6.047	**1.083**	**2.313**	**0.652**	**1.127**

Table 2: Paired STD diagnostics ($L = 16$, $H = 3$) from archived reruns separate from Table 1. E_d/E_w: daily/weekly spectral error; M_g/M_l: common/local standardized MSE. Units: 10^{-3}. Lower is better; bold marks minima among these six models. See Appendix F.4.

	BikeNYC				Traffic				SKT			
Model	E_d	E_w	M_g	M_l	E_d	E_w	M_g	M_l	E_d	E_w	M_g	M_l
TCN	58.6	93.7	22.6	188.9	49.8	161.1	23.7	262.1	**3.5**	778.1	8.6	46.1
TCN+STD	16.3	82.1	14.5	112.6	27.0	97.1	15.9	178.3	4.3	746.3	8.1	45.8
Transformer	56.6	111.9	28.9	198.6	65.5	180.9	37.5	288.3	4.6	813.6	8.4	47.7
Transformer+STD	**11.1**	53.5	12.9	96.8	**21.6**	88.9	14.7	174.7	5.7	699.3	**7.2**	**42.6**
WaveNet	54.2	94.8	21.3	181.5	46.8	170.8	25.5	250.5	3.8	778.6	8.1	45.4
WaveNet+STD	12.3	**51.4**	**12.1**	**88.6**	24.4	**82.7**	**13.9**	**173.8**	5.9	**648.7**	7.4	42.9

Robustness analysis. Eight frozen Capital models share 360 origins and three corruption draws per level. Missing histories use causal filling; Gaussian noise has SD λ times the training SD. The STD-augmented model achieves the lowest MAE and RMSE at 20% missingness. Appendix F.6 reports all severity levels and the corruption protocol.

6 Conclusion

We present a novel module, STD, for spatio-temporal sequence prediction that incorporates decomposition mechanisms and interpretable modules. Besides, local and global components and high-frequency and low-frequency components are obtained, which reflect the underlying structure of the time series. Our experimental results on several benchmark datasets demonstrate that STD improves the prediction accuracy of the paired TCN, Transformer, and WaveNet backbones. Further forecasting results are presented in the Supplementary Material using the additional real datasets. Furthermore, the interpretability of STD offers valuable insights into how the model makes predictions, which can be leveraged to address various practical applications. The limitation of our work is the lack of theoretical proof to guarantee the identifiability of our approach, and the structure of STD can increase training time.

AI USE STATEMENT

We used GPT-5.4 to assist with code review and LaTeX checks. Code review included checking input–target alignment, data-split boundaries, normalization, metric calculations, and consistency between saved outputs and reported results. LaTeX checks helped correct equation layout, figure and table placement, and paragraph breaks, and verify citations, cross-references, and compliance with the official ICLR style and page limits. OpenAI Codex also assisted with experiment implementation, analysis of saved outputs, visualization, and manuscript editing. The authors are responsible for the final text, claims, data provenance, code, and reported results.

ETHICS STATEMENT

This work studies aggregate multivariate time series. Forecasting errors may differ across locations and periods, and deployment decisions should account for these differences. Decomposition patterns describe model representations and do not establish causal effects on locations or populations.

REPRODUCIBILITY STATEMENT

Section 3 defines the original method. Appendix B retains its supporting arguments; Appendix C gives the original training and dataset descriptions. The supplementary experimental protocols identify each additional implementation, split, seed count, and evaluation support. Development evaluations and the previously frozen independent-year study are reported separately. The anonymous code repository is available at `https://anonymous.4open.science/r/std-iclr2027-review-61F2/`.

REFERENCES

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11141–11149, 2024.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

Capital Bikeshare. System data, n.d. Trip records used in this study cover 2022–2025.

Hongjie Chen, Ryan A Rossi, Kanak Mahadik, Sungchul Kim, and Hoda Eldardiry. Graph deep factors for probabilistic time-series forecasting. *ACM Transactions on Knowledge Discovery from Data*, 17(2):1–30, 2023.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Alexander Chudik and M Hashem Pesaran. Theory and practice of gvar modelling. *Journal of Economic Surveys*, 30(1):165–197, 2016.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor Tsang. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 269–278, 08 2021. doi: 10.1145/3447548.3467330.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):7665–7680, 2023. doi: 10.1109/TKDE.2022.3218803.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Du Yin, Yi Yang, Xuan Song, and Ivor W. Tsang. Disentangling structured components: Towards adaptive, interpretable and scalable time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(8):3783–3800, 2024. doi: 10.1109/TKDE.2024.3371931.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Global models for time series forecasting: A simulation study. *Pattern Recognition*, 124:108441, 2022. ISSN 0031-3203. doi: 10.1016/j.patcog.2021.108441.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Athanasia S Kalaitzi and Trevor W Chamberlain. Merchandise exports and economic growth: multivariate time series analysis for the united arab emirates. *Journal of Applied Economics*, 23(1):163–182, 2020.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. CycleNet: Enhancing time series forecasting through modeling periodic patterns. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3373.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Lifan Long, Qian Liu, Hong Peng, Jun Wang, and Qian Yang. Multivariate time series forecasting method based on nonlinear spiking neural p systems and non-subsampled shearlet transform. *Neural Networks*, 152: 300–310, 2022.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, 2017.

Spyros Makridakis and Michele Hibon. Arma models and the box–jenkins methodology. *Journal of forecasting*, 16(3):147–163, 1997.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Ozan Ozyegen, Juyoung Wang, and Mucahit Cevik. Danlip: Deep autoregressive networks for locally interpretable probabilistic forecasting. *arXiv preprint arXiv:2301.02332*, 2023.

Qingyi Pan, Wenbo Hu, and Ning Chen. Two birds with one stone: Series saliency for accurate and interpretable multivariate time series forecasting. In *IJCAI*, pp. 2884–2891, 2021.

Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Jilin Hu, Chenjuan Guo, and Bin Yang. A comprehensive survey of deep learning for multivariate time series forecasting: A channel strategy perspective. *arXiv preprint arXiv:2502.10721*, 2025.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *KDD '16*, KDD '16, pp. 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

scikit-learn developers. HistGradientBoostingRegressor. scikit-learn 1.7 documentation, n.d.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1282.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2022.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153. PMLR, 2017.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 914–921, 2020.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328, 06–11 Aug 2017.

Ting Tian, Jianbin Tan, Wenxiang Luo, Yukang Jiang, Minqiong Chen, Songpan Yang, Canhong Wen, Wenliang Pan, and Xueqin Wang. The effects of stringent and mild interventions for coronavirus pandemic. *Journal of the American Statistical Association*, 116(534):481–491, 2021. doi: 10.1080/01621459.2021.1897015.

Hiroyuki Toda. *Vector autoregression and causality*. Yale University, 1991.

Hugues Turbé, Mina Bjelogrlic, Christian Lovis, and Gianmarco Mengaldo. Evaluation of post-hoc interpretability methods in time-series classification. *Nature Machine Intelligence*, 5:250–260, 2023. doi: 10.1038/s42256-023-00620-w.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

Jiabao Wen, Jiachen Yang, Bin Jiang, Houbing Song, and Huihui Wang. Big data driven marine environment information forecasting: a time series prediction network. *IEEE Transactions on Fuzzy Systems*, 29(1):4–18, 2021.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 753–763, 2020. doi: 10.1145/3394486.3403118.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

Hongyuan Yu, Ting Li, Weichen Yu, Jianguo Li, Yan Huang, Liang Wang, and Alex Liu. Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting. *arXiv preprint arXiv:2210.06126*, 2022.

A ADDITIONAL BACKGROUND AND REPRESENTATION

A.1 ADDITIONAL BACKGROUND

An accurate prediction of MTS not only promotes policy-making (Tian et al., 2021) and the optimization of resource allocation (Chen et al., 2023), but assists in the interpretation of the nature of several phenomena. Thus, it remains a major concentration in both industry and academia.

As one of the common networks, recursive neural networks (RNNs) (Schuster & Paliwal, 1997), long short-term memories (LSTMs) (Hochreiter & Schmidhuber, 1997), and gated recurrent units (GRUs) (Cho et al., 2014) utilize the cell structure to handle long-term dependencies. Additionally, the Transformer (Vaswani et al., 2017) improves basic neural network architectures by including the attention mechanism to access temporal information from the entire series, achieving better performance.

These approaches create a collection of MTS as a unified entity and leverage a relational global model with dependencies between different time series, taking advantage of spatial-temporal attributes.

A.2 ADDITIONAL RELATED WORK

Wavenet (Oord et al., 2016) and temporal convolution networks (TCNs) (Bai et al., 2018), as representatives of CNN-based models, improve the classical convolutional layer and extract temporal features to capture long-term dependencies and complex patterns.

Although RNN-based methods can capture temporal information well, they have a deficiency in the interaction of spatial-temporal relationships (Yu et al., 2022). To address this limitation, graph-based models have become increasingly popular for MTS.

STGCN (Yu et al., 2017), employing spatial and temporal convolution for forecasting, models the spatial-temporal dependencies between variables. ASTGCN (Guo et al., 2019) is a spatio-temporal GCN with spatial-temporal attention. STSGCN (Song et al., 2020) employs a synchronous modeling mechanism that accounts for the heterogeneities in localized spatial-temporal graphs, enabling the capture of intricate spatio-temporal patterns. By combining Graph Fourier Transform and Discrete Fourier Transform, StemGNN (Cao et al., 2020) enhances forecasting performance without predefined dependency graph priors. Additionally, for capturing the relationship between temporal and spatial information, ST-Norm (Deng et al., 2021) utilizes two separate normalization strategies for temporal and spatial dimensions, respectively. It outperforms several state-of-the-art methods for MTS forecasting, indicating the effectiveness of its temporal and spatial normalization approach.

A novel scheme of series saliency has been proposed for MTS forecasting, which can boost both forecasting accuracy and interpretability with feature and temporal patterns (Pan et al., 2021).

DANLIP (Ozyegen et al., 2023) is a novel deep learning-based interpretable parametric probabilistic time series forecasting model that can achieve comparable performance to state-of-the-art methods while maintaining high interpretability.

A.3 COMPONENT REPRESENTATION AND FORECASTING

This spatio-temporal process naturally exhibits distinct patterns along both temporal and spatial dimensions: temporally, the data contains high-frequency components (X_{jt}^{high}) capturing short-term variations and low-frequency components (X_{j}^{low}) representing long-term trends; spatially, it manifests local patterns (X_{jt}^{loc}) specific to individual locations and global patterns (X_{t}^{glo}) shared across the entire system. These dual characteristics suggest that the underlying dynamics can be decomposed along both frequency and spatial locality dimensions, motivating our four-component representation of the time series structure.

The framework's trainable parameters (θ_j, ϕ_t, and their variants) enable adaptive refinement during model training, allowing the decomposition to optimize for specific forecasting tasks while preserving the interpretability of each component.

Our unified forecasting architecture combines decomposed feature representation with neural network prediction in an end-to-end trainable architecture. The core components work together as follows.

This rich representation preserves both the original signal and its decomposed characteristics across different scales and localities.

The framework's modular design allows for seamless integration with different neural network architectures tailored to specific application needs.

B SUPPORTING ARGUMENTS FOR THE ORIGINAL THEORY

The optimality of θ_j emerges from systematic analysis of the temporal mean's bias structure and its interaction with the prediction objective. Beginning with the decomposition of the empirical temporal mean under Assumption 1, we express:

$$\begin{aligned}\hat{\mu}_j &= \frac{1}{T}\sum_{k=1}^{K_j}\sum_{t=t_k}^{t_{k+1}-1}[\mu_j^{(k)} + S_{jt} + \epsilon_{jt}] \\ &= \underbrace{\sum_{k=1}^{K_j}\frac{t_{k+1}-t_k}{T}\mu_j^{(k)}}_{\text{Segment-Weighted Trend}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}S_{jt}}_{\text{Stationary Mean}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\epsilon_{jt}}_{\text{Noise Term}}.\end{aligned} \tag{25}$$

The stationarity of S_{jt} ensures $\frac{1}{T}\sum S_{jt} \xrightarrow{a.s.} 0$ as $T \to \infty$, while the noise term exhibits $\mathbb{E}[(\frac{1}{T}\sum \epsilon_{jt})^2] = \sigma_j^2/T$. This decomposition reveals two critical bias sources: 1) structural mismatch from piecewise stationarity, and 2) noise contamination.

The gradient dynamics are governed by the prediction task through the chain rule differentiation:

$$\frac{\partial \mathcal{L}_{\text{pred}}}{\partial \theta_j} = \frac{2}{H}\sum_{h=1}^{H}(Y_{t+h} - \hat{Y}_{t+h})\frac{\partial g}{\partial X_j^{\text{low}}}, \tag{26}$$

where $X_j^{\text{low}} = \hat{\mu}_j + \theta_j$. Taking expectations over training data and substituting $Y = g(\{\mathcal{T}_j\}) + \delta$ with $\delta \perp\!\!\!\perp X_j^{\text{low}}$, we derive the convergence condition:

$$0 = \mathbb{E}\left[(g(\{\mathcal{T}_j\}) - g(\{\hat{\mu}_j + \theta_j\}))\frac{\partial g}{\partial X_j^{\text{low}}}\right]. \tag{27}$$

Lipschitz continuity of g enforces the bound:

$$|\mathbb{E}[g(\mathcal{T}_j) - g(\hat{\mu}_j + \theta_j)]| \le L\mathbb{E}|\mathcal{T}_j - (\hat{\mu}_j + \theta_j)| + O(\mathbb{E}[(\mathcal{T}_j - (\hat{\mu}_j + \theta_j))^2]). \tag{28}$$

Substituting the bias decomposition $\mathcal{T}_j - (\hat{\mu}_j + \theta_j) = \sum_{k=1}^{K_j}\frac{t_{k+1}-t_k}{T}(\mu_j^{(k)} - \hat{\mu}_j) - \theta_j + O_P(\sqrt{\log T/T})$ reveals that optimality requires cancellation of the leading structural bias term. The $O_P(\sqrt{\log T/T})$ term arises from applying Hoeffding's inequality, where $O(\cdot)$ denotes deterministic asymptotic upper bounds and $O_P(\cdot)$ represents probabilistic bounds holding with high probability. This forces the parameter convergence to:

$$\theta_j^* = \sum_{k=1}^{K_j}\frac{t_{k+1}-t_k}{T}(\mu_j^{(k)} - \hat{\mu}_j) + O\left(\sqrt{\frac{\log T}{T}}\right), \tag{29}$$

where the residual term originates from concentration inequalities applied to the noise aggregation $\frac{1}{T}\sum \epsilon_{jt}$.

Proof. The residual term ϕ_t learns true interaction weights through gradient updates:

$$\phi_t^{(n+1)} = \phi_t^{(n)} - \eta\frac{\partial \mathcal{L}}{\partial \phi_t}, \tag{30}$$

where the gradient decomposes as:

$$\frac{\partial \mathcal{L}}{\partial \phi_t} = \sum_{j=1}^{N} (Y_{jt} - \hat{Y}_{jt}) \cdot \frac{\partial f}{\partial X_t^{\text{glo}}}. \tag{31}$$

Starting from the loss function:

$$\mathcal{L} = \sum_{j,t} \left\| Y_{jt} - f(X_{jt}^{\text{loc}}, X_t^{\text{glo}}) \right\|^2, \tag{32}$$

we expand the gradient using the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \phi_t} = \sum_{j=1}^{N} \underbrace{(Y_{jt} - \hat{Y}_{jt})}_{\text{Error}} \cdot \underbrace{\frac{\partial f}{\partial X_t^{\text{glo}}}}_{\text{Sensitivity}} \cdot \underbrace{1}_{\partial \phi_t / \partial X_t^{\text{glo}}}. \tag{33}$$

The adjustment term δ_{jk} in the global component representation:

$$X_t^{\text{glo}} = \sum_{k=1}^{N} \left(\frac{1}{N} + \delta_{jk} \right) X_{kt}, \tag{34}$$

emerges naturally from the parametric relationship $\phi_t = \sum_k \delta_{jk} X_{kt}$. Its learning dynamics follow from:

$$\frac{\partial \phi_t}{\partial \delta_{jk}} = X_{kt} \implies \delta_{jk}^{(n+1)} = \delta_{jk}^{(n)} + \eta \Delta \phi_t \cdot X_{kt}, \tag{35}$$

where $\Delta \phi_t$ is the gradient update from the main optimization step. □

To establish convergence guarantees, we first verify three fundamental mathematical properties of the optimization landscape:

Lemma 4 (Measurable Gradient)**.** *The gradient decomposition satisfies:*

$$\nabla_\omega \mathcal{L} = \mathbb{E}_\xi \left[\frac{\partial \ell}{\partial f} \cdot \left(\frac{\partial f}{\partial \Theta} \oplus \frac{\partial f}{\partial \theta_j} \oplus \frac{\partial f}{\partial \phi_t} \right) \right], \tag{36}$$

where $\oplus$ denotes block-wise direct sum. This ensures gradient measurability through differentiable decomposition operations.

Proof. The gradient measurability follows from three key observations. First, all decomposition operations - including temporal means $\overline{X}_j^{\text{temp}} = T^{-1} \sum_t X_{jt}$, spatial means $\overline{X}_t^{\text{spat}} = N^{-1} \sum_j X_{jt}$, and parameter additions - are linear or affine transformations, guaranteeing differentiability everywhere.

Second, the direct sum $\oplus$ preserves distinct gradient flows for model parameters $\partial f / \partial \Theta$, location-specific terms $\partial f / \partial \theta_j$, and time-specific terms $\partial f / \partial \phi_t$ without information loss.

Finally, for all $\omega \in \mathcal{W}$ and $\xi \sim \mathcal{D}$, the loss $\ell(\omega; \xi)$ and its partial derivatives are Borel measurable by construction, and the expectation $\mathbb{E}_\xi$ maintains measurability via dominated convergence. Thus $\nabla_\omega \mathcal{L}$ remains measurable throughout the decomposition. □

The gradient measurability condition fundamentally requires that $\nabla_\omega \mathcal{L}$ must exist throughout the parameter space $\mathcal{W}$, be computable through a deterministic computation graph, and satisfy the integrability condition $\mathbb{E}_\xi[\|\nabla \ell(\omega; \xi)\|] < \infty$ to ensure well-defined expectations, thereby maintaining both mathematical rigor and practical optimizability through the decomposition framework.

Lemma 5 (Lipschitz Continuous Gradient)**.** *The gradient mapping $\nabla \mathcal{L}(\omega)$ satisfies the Lipschitz continuity condition:*

$$\|\nabla \mathcal{L}(\omega_1) - \nabla \mathcal{L}(\omega_2)\| \leq L \|\omega_1 - \omega_2\|, \quad \forall \omega_1, \omega_2 \in \mathcal{W}, \tag{37}$$

where the total Lipschitz constant L decomposes additively into three components:

$$L = \underbrace{L_f}_{\textit{Model}} + \underbrace{L_\theta}_{\textit{Location}} + \underbrace{L_\phi}_{\textit{Time}} = L_f + 2, \tag{38}$$

The model component L_f is determined by the neural network architecture:

$$L_f = \prod_{l=1}^{D} \|W_l\|, \tag{39}$$

where D is the network depth and $\|W_l\|$ denotes the operator norm of layer l's weight matrix. The constants $L_\theta = 1$ and $L_\phi = 1$ reflect the linear structure of location and time parameters respectively.

Proof. The Lipschitz continuity of the gradient can be established by examining each component of the parameter space. For the neural network parameters Θ, the submultiplicativity of operator norms in the D-layer architecture yields the bound

$$\left\| \frac{\partial f}{\partial \Theta}(\omega_1) - \frac{\partial f}{\partial \Theta}(\omega_2) \right\| \leq L_f \|\omega_1 - \omega_2\|, \tag{40}$$

where $L_f = \prod_{l=1}^{D} \|W_l\|$ captures the network's complexity.

The location parameters θ_j exhibit simpler behavior due to their linear relationship $X_j^{\text{low}} = \overline{X}_j^{\text{temp}} + \theta_j$, which leads to

$$\left\| \nabla_{\theta_j} \mathcal{L}(\omega_1) - \nabla_{\theta_j} \mathcal{L}(\omega_2) \right\| \leq \|\omega_1 - \omega_2\|, \tag{41}$$

establishing $L_\theta = 1$. An identical argument applies to the time parameters ϕ_t through the linear transformation $X_t^{\text{glo}} = \overline{X}_t^{\text{spat}} + \phi_t$, giving $L_\phi = 1$.

Combining these results through the summation of independent gradient variations, we obtain the composite Lipschitz constant $L = L_f + L_\theta + L_\phi = L_f + 2$ as claimed. □

This decomposition of the Lipschitz constant has important implications for optimization. The product form of L_f highlights the exponential sensitivity to network depth D, while the unit contributions from θ_j and ϕ_t reflect their well-conditioned linear nature. This characterization enables precise control of learning rates in gradient-based optimization algorithms.

Lemma 6 (Bounded Variance)**.** *The gradient variance reduces through decomposition:*

$$\mathbb{V}[g_B] = \frac{\sigma^2}{B} \cdot \underbrace{\frac{1}{T}}_{\textit{Temporal}} \cdot \underbrace{\frac{1}{N}}_{\textit{Spatial}}, \tag{42}$$

where $g_B(\omega) := \frac{1}{B}\sum_{i=1}^{B} \nabla \ell(\omega; \xi_i)$ denotes the mini-batch gradient estimate computed from B independent samples, $\sigma^2 := \mathbb{E}_\xi[\|\nabla \ell(\omega;\xi) - \nabla \mathcal{L}(\omega)\|^2]$ represents the population variance of single-sample gradients, and T (temporal window length) and N (number of spatial locations) characterize the decomposition dimensions.

Proof. The variance reduction emerges through three complementary mechanisms.

First, for a mini-batch of size B with independent samples $\{\xi_i\}_{i=1}^{B}$, the variance of the averaged gradient estimate $g_B(\omega)$ reduces as:

$$\mathbb{V}[g_B] = \mathbb{V}\left[\frac{1}{B} \sum_{i=1}^{B} \nabla \ell(\omega; \xi_i) \right] = \frac{\sigma^2}{B}, \tag{43}$$

where σ^2 is the population variance of individual gradients.

Second, temporal decomposition through the mean operation yields variance reduction for location parameters:

$$\overline{X}_j^{\text{temp}} = T^{-1} \sum_{t=1}^{T} X_{jt}, \mathbb{V}\left[\nabla_{\theta_j} \mathcal{L}\right] = \mathbb{V}\left[T^{-1} \sum_{t=1}^{T} \frac{\partial \ell}{\partial X_j^{\text{low}}}(\omega; \xi_t) \right] = \frac{\sigma^2}{T}. \tag{44}$$

Third, spatial averaging $\overline{X}_t^{\text{spat}} = N^{-1}\sum_{j=1}^{N} X_{jt}$ similarly reduces variance for time parameters:

$$\mathbb{V}\left[\nabla_{\phi_t}\mathcal{L}\right] = \mathbb{V}\left[N^{-1}\sum_{j=1}^{N}\frac{\partial \ell}{\partial X_t^{\text{glo}}}(\omega;\xi_j)\right] = \frac{\sigma^2}{N}. \tag{45}$$

By the independence of temporal and spatial averaging operations, the total variance reduction combines multiplicatively:

$$\mathbb{V}[g_B] = \frac{\sigma^2}{B}\cdot\frac{1}{T}\cdot\frac{1}{N}. \tag{46}$$

□

Lemma 6 shows how variance reduction factors multiplicatively across dimensions: batch size $(1/B)$, temporal window $(1/T)$, and spatial locations $(1/N)$. This decomposition enables precise noise control in optimization while σ^2 sets the fundamental noise floor. The result provides a principled way to balance computational cost and convergence stability.

Having established the key properties of measurable gradients (Lemma 4), Lipschitz continuity (Lemma 5), and bounded variance (Lemma 6), we now present our main convergence result for the spatiotemporal decomposition framework (Ghadimi & Lan, 2013; Bottou et al., 2018):

Proof. Starting from the Lipschitz continuity of gradients and Taylor expansion, we derive for each iteration k:

$$\mathcal{L}(\omega_{k+1}) \le \mathcal{L}(\omega_k) - \eta_k\langle\nabla\mathcal{L}(\omega_k), g_k\rangle + \frac{L\eta_k^2}{2}\|g_k\|^2, \tag{47}$$

where $g_k = \nabla\ell(\omega_k;\xi_k)$ is the stochastic gradient and $\langle\cdot,\cdot\rangle$ denotes the standard Euclidean inner product. Taking expectations and using $\mathbb{E}[g_k] = \nabla\mathcal{L}(\omega_k)$ with variance bound $\text{Var}(g_k) \le \sigma^2$, we obtain:

$$\mathbb{E}[\,\mathcal{L}(\omega_{k+1})\,] \le \mathbb{E}[\,\mathcal{L}(\omega_k)\,] - \eta_k\mathbb{E}\left[\,\|\nabla\mathcal{L}\|^2\,\right] + \frac{L\eta_k^2}{2}\left(\mathbb{E}\left[\,\|\nabla\mathcal{L}\|^2\,\right] + \sigma^2\right). \tag{48}$$

Summing over K iterations and letting $\Delta_k = \mathbb{E}[\,\mathcal{L}(\omega_k) - \mathcal{L}^*\,]$, we establish:

$$\sum_{k=1}^{K}\eta_k\mathbb{E}\left[\,\|\nabla\mathcal{L}\|^2\,\right] \le D_{\mathcal{L}}^2 + \frac{L}{2}\sum_{k=1}^{K}\eta_k^2\left(\mathbb{E}\left[\,\|\nabla\mathcal{L}\|^2\,\right] + \sigma^2\right) \tag{49}$$

Introducing a uniformly random index $R \sim \text{Uniform}\{1,\ldots,K\}$ and using the relation $\sum_{k=1}^{K}\eta_k = \eta\sqrt{K}$, we express:

$$\mathbb{E}\left[\,\|\nabla\mathcal{L}(\omega_R)\|^2\,\right] = \frac{1}{\eta\sqrt{K}}\sum_{k=1}^{K}\eta_k\mathbb{E}\left[\,\|\nabla\mathcal{L}\|^2\,\right] \tag{50}$$

Substituting into the summed inequality and using $\sum_{k=1}^{K}\eta_k^2 = \eta^2$, we derive:

$$\eta\sqrt{K}\mathbb{E}\left[\,\|\nabla\mathcal{L}(\omega_R)\|^2\,\right] \le D_{\mathcal{L}}^2 + \frac{L\eta^2}{2}\left(\mathbb{E}\left[\,\|\nabla\mathcal{L}(\omega_R)\|^2\,\right] + \sigma^2\right) \tag{51}$$

Rearranging terms and solving for the gradient norm expectation yields the final bound:

$$\mathbb{E}\left[\,\|\nabla\mathcal{L}(\omega_R)\|^2\,\right] \le \frac{D_{\mathcal{L}}^2 + \frac{L\eta^2}{2}\sigma^2}{\eta\sqrt{K} - \frac{L\eta^2}{2}} \tag{52}$$

□

C ORIGINAL EXPERIMENTAL PROTOCOLS

During the model training process, we implement a batch size of 4 for the training set and 48 for the testing set. The learning rate is set to 0.0001, and we incorporate data from the preceding 16 time points to predict the subsequent 3 time points in the multivariate time series (MTS). To optimize the model, we leverage the Adam algorithm and continue training until the error has converged.

In simulation experiments, the number of locations is varied across three settings: 50, 100, and 200, with a fixed time span of 8640. The global and local components are simulated using distinct methodologies. The global and local low components are generated using a piece-wise function, with each piece spanning 720 time units. The global low component is simulated once for the entire dataset, while the local low components are simulated individually for each location. The model employed for each piece is an ARIMA(2,1,2) with parameters:

- Global Low: $\phi = 0.5$, $\theta_1 = 0.5$, $\theta_2 = -0.5$, $\sigma = 1$.
- Local Low: $\phi = 0.3$, $\theta_1 = 0.5$, $\theta_2 = -0.2$, $\sigma = 1$.
- Fourier Terms
  - Global High: $T = 24$, $\delta = 0$.
  - Local High: $T = 24U$, $U \sim N(1, 1)$, $\delta \sim N(0, 1)$.
- ARIMA(2,1,2) Model
  Both global and local high components follow an ARIMA(2,1,2) model with parameters: $\phi = 0.1$, $\theta_1 = 0.3$, $\theta_2 = -0.2$, $\sigma = 1$.
- Mackey-Glass Equation
  The Mackey-Glass equation is utilized with the following parameters: - $\alpha = 0.2$, $\beta = 0.14$, $c = 10$, $\tau = 25$, $h = 1$.

Table 3: Dataset statistics.

Dataset	N	Timesteps	Time Granularity	Training Size	Validation Size	Testing Size
BikeNYC	128	4392	1 h	3240	384	768
Electricity	336	2208	1 h	1872	168	168
PeMSD7	228	2112	30 mins	1632	240	240
Solar	70	52560	10 mins	43200	4608	4752
Traffic	100	17544	1 h	12936	1536	3072
SP	100	8764	6 hours	6716	1024	1024
MSL	100	8764	6 hours	6716	1024	1024
SKT	100	8764	6 hours	6716	1024	1024
STL	100	8764	6 hours	6716	1024	1024

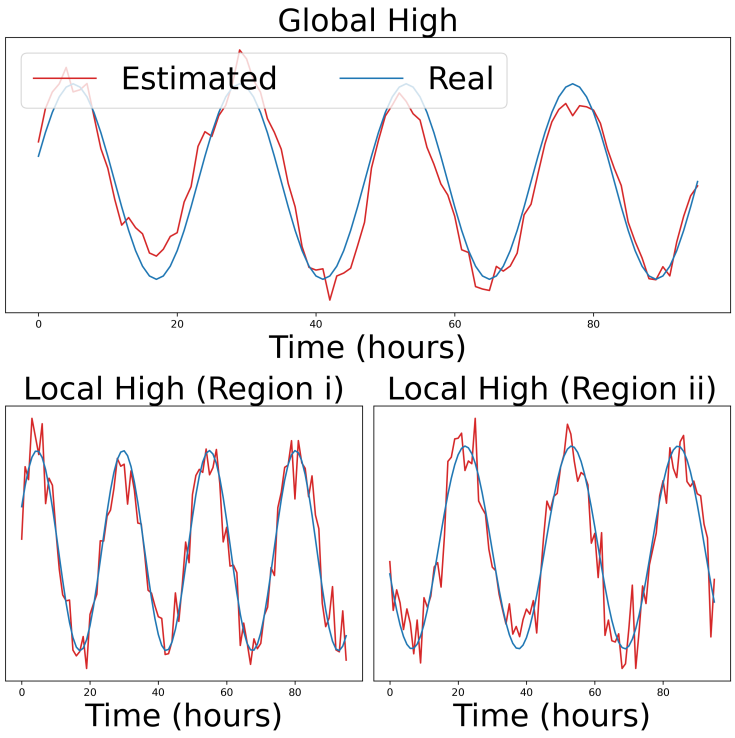

Figure 4: This figure provides insights into the fidelity of the model's estimation relative to the generated high-frequency components. The figure illustrates the series estimated by STD module. Both the generated and estimated local and global high-frequency components are displayed, with each component normalized to a range of 0 to 1. Specifically, for the local high-frequency components, two distinct regions are highlighted with their weight set to 10 ($w_3 = 10$), while the weights of the other components are set to 1 ($w_1 = w_2 = w_4 = 1$). Similarly, for the global high-frequency components, their weight is set to 10 ($w_1 = 10$), while the weights of the other components are set to 1 ($w_2 = w_3 = w_4 = 1$).

D COMPLETE ORIGINAL FORECASTING RESULTS

We have recorded the predictive accuracy of each model for each dataset at a three-subsequent time granularity and evaluated them using three distinct measures of predictive accuracy: mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE). In the original STD implementation, MAPE is computed only over targets greater than 10 in the stored data units; MAE and RMSE use all targets. Tables 4-6 contain four segments that show the comparison performances of five benchmark models, the TCN model with STD structures, the Transformer model with STD structures, and the WaveNet model with STD structures for BikeNYC, Traffic, and SKT datasets. The underlined outcomes indicate the best performance for that particular segment. The bolded outcomes in the table signify the overall lowest errors among all models for each dataset.

Table 4: Model performance on BikeNYC

	1-hour			2-hour			3-hour		
	MAPE %	MAE	RMSE	MAPE %	MAE	RMSE	MAPE %	MAE	RMSE
LSTM	34.4	4.01	8.53	44.3	5.38	11.3	47.8	6.19	12.6
GRU	33.9	3.88	8.13	44.7	5.13	11.3	49.4	6.22	13.2
LSTNet	26.9	3.57	8.21	38.0	4.12	9.39	40.3	4.44	10.2
MTGNN	25.8	2.85	6.11	31.8	3.58	8.01	37.0	4.17	9.28
StemGNN	26.9	3.04	6.54	27.3	3.54	7.90	**27.7**	3.98	8.70
TCN	28.7	3.11	6.50	35.2	3.92	8.44	38.4	4.59	9.56
TCN+STD	24.7	2.77	5.48	27.5	3.18	6.41	30.4	3.51	7.09
Transformer	29.7	3.28	6.83	36.3	4.17	8.99	40.2	4.65	10.1
Transformer + STD	**23.5**	**2.61**	**5.07**	**26.3**	**2.90**	**5.96**	28.0	**3.10**	**6.44**
WaveNet	28.9	3.12	6.66	36.9	4.02	8.67	40.1	4.53	9.71
WaveNet + STD	23.7	2.63	5.19	26.6	2.92	6.10	30.0	3.33	6.85

Table 5: Model performance on Traffic

	1-hour			2-hour			3-hour		
	MAPE %	MAE $\times 10^{-2}$	RMSE $\times 10^{-2}$	MAPE %	MAE $\times 10^{-2}$	RMSE $\times 10^{-2}$	MAPE %	MAE $\times 10^{-2}$	RMSE $\times 10^{-2}$
LSTM	31.4	1.23	2.47	46.2	1.80	3.36	61.3	2.22	3.90
GRU	24.1	1.12	2.40	45.4	1.73	3.29	60.7	2.15	3.90
LSTNet	35.5	1.50	2.93	41.3	1.78	3.43	43.3	1.84	3.52
MTGNN	18.0	0.886	2.08	31.3	1.33	2.80	43.3	1.64	3.14
StemGNN	21.0	1.04	2.17	33.5	1.34	2.68	38.8	1.55	2.96
TCN	21.2	1.01	2.22	33.3	1.48	2.96	44.8	1.81	3.39
TCN + STD	19.1	0.891	1.97	27.8	1.17	2.44	34.2	1.34	2.67
Transformer	19.7	1.03	2.30	32.4	1.51	3.07	44.1	1.84	3.53
Transformer + STD	18.3	0.889	1.98	25.8	1.16	2.48	31.3	1.32	2.67
WaveNet	18.6	0.957	2.15	29.1	1.39	2.87	40.8	1.70	3.28
WaveNet + STD	**17.6**	**0.848**	**1.91**	**24.0**	**1.13**	**2.41**	**27.0**	**1.27**	**2.62**

Table 6: Model performance on SKT

	6-hour			12-hour			18-hour		
	MAPE %	MAE $\times 10^2$	RMSE $\times 10^2$	MAPE %	MAE $\times 10^2$	RMSE $\times 10^2$	MAPE %	MAE $\times 10^2$	RMSE $\times 10^2$
LSTM	0.315	0.945	1.51	0.310	0.931	1.49	0.328	0.983	1.57
GRU	0.246	0.738	1.25	0.254	0.763	1.29	0.255	0.765	1.28
LSTNet	1.02	3.09	4.50	1.02	3.09	4.50	1.01	3.05	4.46
MTGNN	0.255	0.769	1.20	0.280	0.843	1.28	0.286	0.862	1.29
StemGNN	0.241	0.718	1.24	0.256	0.762	1.31	0.260	0.774	1.31
TCN	0.206	0.620	1.10	0.234	0.703	1.22	0.241	0.722	1.23
TCN + STD	0.206	0.618	1.06	0.232	0.693	1.20	0.243	0.729	1.21
Transformer	0.217	0.651	1.12	0.241	0.722	1.24	0.249	0.748	1.25
Transformer + STD	**0.198**	**0.593**	1.05	**0.224**	**0.673**	1.17	0.233	0.699	1.19
WaveNet	0.209	0.627	1.10	0.232	0.698	1.21	0.239	0.718	1.23
WaveNet + STD	**0.198**	**0.593**	**1.04**	**0.224**	**0.673**	**1.16**	**0.230**	**0.691**	**1.18**

Table 7: Performance on Electricity.

	1-hour			2-hour			3-hour		
	MAPE %	MAE	RMSE	MAPE %	MAE	RMSE	MAPE %	MAE	RMSE
LSTM	16.0	21.3	46.1	18.7	34.6	66.9	33.5	42.5	78.9
GRU	12.5	17.6	40.9	22.5	30.6	64.6	30.7	40.6	78.9
LSTNet	17.2	25.7	54.2	18.7	28.0	59.1	**19.5**	29.0	61.3
MTGNN	**9.61**	**14.9**	36.2	**14.7**	**22.5**	52.6	20.4	**27.7**	61.6
StemGNN	21.0	27.0	56.2	25.3	30.1	62.1	25.1	32.9	67.0
TCN	11.4	16.1	35.5	17.3	24.8	53.0	22.8	30.7	62.7
TCN + STD	15.1	19.3	40.4	20.6	26.6	54.6	24.9	30.6	62.6
Transformer	11.3	16.4	37.0	19.2	25.9	54.6	23.9	32.2	64.5
Transformer + STD	15.1	18.3	37.1	18.4	24.2	51.4	22.4	29.1	62.4
WaveNet	11.9	16.8	36.0	17.3	24.9	53.0	22.3	30.8	62.8
WaveNet + STD	12.3	16.6	**33.8**	17.6	23.3	**47.9**	23.1	28.6	**58.7**

Table 8: Performance on PeMSD7.

	30-min			60-min			90-min		
	MAPE %	MAE	RMSE	MAPE %	MAE	RMSE	MAPE %	MAE	RMSE
LSTM	6.84	2.92	5.24	11.6	4.63	7.91	15.2	5.92	9.49
GRU	6.64	2.85	5.18	11.6	4.67	7.91	15.3	6.02	9.57
LSTNet	6.71	3.05	7.64	10.1	4.72	9.50	10.7	6.01	10.3
MTGNN	6.41	3.01	6.77	10.1	4.67	10.9	12.9	6.03	14.4
StemGNN	8.03	3.39	5.41	10.8	4.37	7.10	14.2	5.40	8.52
TCN	6.54	2.82	5.13	11.3	4.63	7.81	15.0	6.03	9.43
TCN + STD	6.47	2.88	4.84	10.0	4.22	6.85	12.6	5.29	8.05
Transformer	6.52	2.80	5.18	10.9	4.42	7.72	14.2	5.56	9.27
Transformer + STD	6.29	**2.75**	4.72	**8.85**	**3.73**	**6.35**	**10.2**	**4.26**	**7.10**
WaveNet	6.38	2.76	5.02	10.8	4.43	7.55	14.1	5.62	9.11
WaveNet + STD	**6.19**	**2.75**	**4.67**	9.21	3.93	6.58	11.1	4.71	7.59

Table 9: Performance on Solar.

	10-min			20-min			30-min		
	MAPE %	MAE	RMSE	MAPE %	MAE	RMSE	MAPE %	MAE	RMSE
LSTM	4.88	0.349	1.07	8.68	0.600	1.65	11.3	0.861	2.09
GRU	4.97	0.343	1.07	8.65	0.633	1.66	11.3	0.896	2.11
LSTNet	5.46	0.495	1.29	8.36	0.742	1.85	10.9	0.956	2.31
MTGNN	4.53	**0.294**	0.972	7.84	0.498	1.47	9.45	0.672	1.83
StemGNN	4.55	0.351	1.02	7.32	0.546	1.48	9.35	0.682	1.79
TCN	4.51	0.319	1.01	7.82	0.560	1.55	10.3	0.803	1.96
TCN + STD	**4.35**	0.304	0.975	**7.28**	0.519	1.43	9.32	0.691	1.75
Transformer	4.54	0.357	1.02	7.78	0.554	1.55	10.3	0.774	1.96
Transformer + STD	4.51	0.309	0.981	7.46	0.542	1.45	9.47	0.734	1.78
WaveNet	4.58	0.314	1.01	7.91	0.543	1.53	10.3	0.734	1.93
WaveNet + STD	4.43	0.303	**0.966**	**7.28**	**0.494**	**1.42**	**9.29**	**0.660**	**1.74**

Table 10: Performance on SP.

	6-hour			12-hour			18-hour		
	MAPE $\times 10^{-2}$ %	MAE $(\times 10^4)$	RMSE $(\times 10^4)$	MAPE $\times 10^{-2}$ %	MAE $(\times 10^4)$	RMSE $(\times 10^4)$	MAPE $\times 10^{-2}$ %	MAE $(\times 10^4)$	RMSE $(\times 10^4)$
LSTM	8.92	0.867	1.07	9.52	0.926	1.16	9.44	0.918	1.15
GRU	9.01	0.874	1.08	9.03	0.878	1.10	8.80	0.856	1.07
LSTNet	15.2	1.48	1.86	15.2	1.48	1.86	15.2	1.48	1.87
MTGNN	12.1	1.17	1.48	12.8	1.24	1.57	12.9	1.27	1.60
StemGNN	6.68	0.639	0.822	7.12	0.683	0.872	7.98	0.765	0.987
TCN	5.45	0.530	0.677	6.25	0.608	0.775	6.97	0.679	0.857
TCN + STD	5.23	0.508	0.647	**6.00**	**0.584**	**0.744**	**6.66**	**0.648**	**0.820**
Transformer	6.78	0.660	0.836	7.43	0.723	0.915	7.64	0.744	0.938
Transformer + STD	**4.98**	**0.484**	**0.618**	6.30	0.613	0.780	6.74	0.657	0.830
WaveNet	5.33	0.519	0.662	6.34	0.617	0.786	7.12	0.693	0.874
WaveNet + STD	5.23	0.508	0.648	**6.00**	**0.584**	**0.744**	**6.66**	**0.648**	**0.820**

Table 11: Performance on MSL.

	6-hour			12-hour			18-hour		
	MAPE $\times 10^{-2}$ %	MAE $(\times 10^4)$	RMSE $(\times 10^4)$	MAPE $\times 10^{-2}$ %	MAE $(\times 10^4)$	RMSE $(\times 10^4)$	MAPE $\times 10^{-2}$ %	MAE $(\times 10^4)$	RMSE $(\times 10^4)$
LSTM	6.65	0.673	0.856	7.91	0.801	1.01	7.66	0.775	0.982
GRU	5.58	0.565	0.725	6.51	0.659	0.842	7.11	0.719	0.914
LSTNet	15.9	1.61	2.01	15.9	1.61	2.01	15.9	1.61	2.01
MTGNN	15.4	1.55	1.97	15.7	1.58	2.01	15.7	1.59	2.02
StemGNN	6.68	0.666	0.871	7.51	0.750	0.979	8.10	0.810	1.05
TCN	5.28	0.534	0.687	6.27	0.634	0.813	6.96	0.705	0.894
TCN + STD	4.89	0.495	0.639	5.85	0.592	0.760	6.61	0.669	0.850
Transformer	5.45	0.562	0.721	6.43	0.657	0.841	7.12	0.728	0.924
Transformer + STD	4.81	0.486	0.628	5.75	0.582	0.748	6.55	0.663	0.841
WaveNet	5.27	0.534	0.686	6.27	0.635	0.813	6.97	0.706	0.896
WaveNet + STD	**4.72**	**0.478**	**0.617**	**5.67**	**0.573**	**0.738**	**6.51**	**0.659**	**0.837**

Table 12: Performance on STL.

	6-hour			12-hour			18-hour		
	MAPE %	MAE ($\times 10^2$)	RMSE ($\times 10^2$)	MAPE %	MAE ($\times 10^2$)	RMSE ($\times 10^2$)	MAPE %	MAE ($\times 10^2$)	RMSE ($\times 10^2$)
LSTM	0.256	0.767	1.07	0.272	0.814	1.14	0.260	0.779	1.09
GRU	0.192	0.574	0.842	0.212	0.633	0.922	0.220	0.658	0.938
LSTNet	0.814	2.44	3.21	0.812	2.47	3.20	0.806	2.42	3.19
MTGNN	0.216	0.648	0.904	0.233	0.699	0.968	0.238	0.713	0.983
StemGNN	0.197	0.586	0.860	0.217	0.646	0.922	0.221	0.657	0.932
TCN	0.182	0.544	0.824	0.207	0.619	0.901	0.214	0.640	0.921
TCN + STD	0.176	0.528	0.784	0.203	0.608	0.877	0.212	0.634	0.903
Transformer	0.19	0.570	0.841	0.211	0.631	0.913	0.219	0.657	0.937
Transformer + STD	**0.171**	**0.511**	**0.764**	**0.198**	**0.594**	**0.856**	0.206	0.616	**0.878**
WaveNet	0.180	0.539	0.801	0.203	0.608	0.880	0.208	0.624	0.897
WaveNet + STD	0.179	0.537	0.778	**0.198**	**0.594**	0.860	**0.205**	**0.613**	0.879

Table 13: The number of parameters of TCN, Transformer, WaveNet Model, and these models with STD.

Dataset	TCN		Transformer		WaveNet	
	Original	with STD	Original	with STD	Original	with STD
BikeNYC	3955	6627	6355	8739	6755	9139
Traffic, SKT, SP, MSL, STL	3955	6515	6355	8627	6755	9027
Electricity	3955	7459	6355	9571	6755	9971
PeMS07	3955	7027	6355	9139	6755	9539
Solar	3955	6395	6355	8507	6755	8907

E ORIGINAL DECOMPOSITION INTERPRETATION

Referring to Figure 5, it can be observed that the global-high-frequency components exhibit a similar trend, whereas the local-high-frequency components display dissimilar trends. This observation aligns with our initial assumption. Specifically, during weekdays (from April 1st to April 4th), the global-high-frequency components display two peaks in a day, which correspond to the to-work and from-work trends. On weekends, however, only one peak is observed, which can be attributed to the high demand for shared bikes during daytime hours.

The limitation of our work is the lack of theoretical proof to guarantee the identifiability of our approach, and the structure of STD will slightly reduce the speed of the training process.

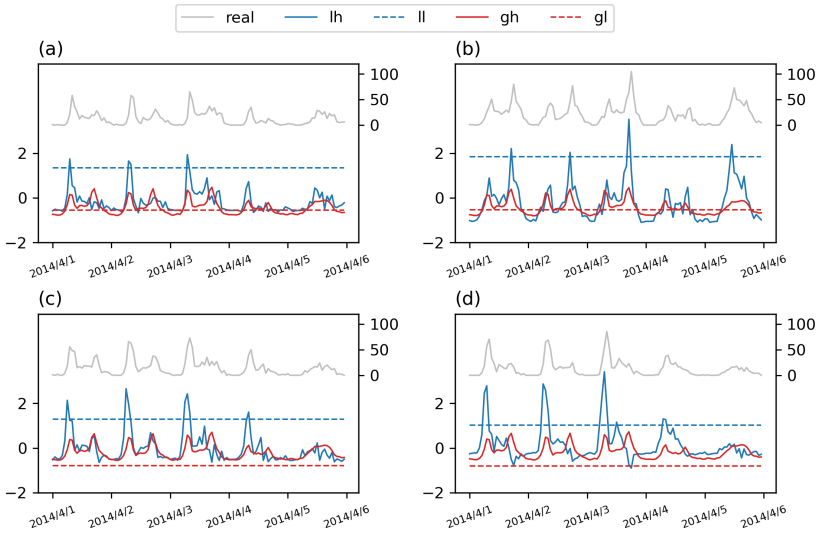

Figure 5: Decomposition on BikeNYC: The original series and four components obtained by the decomposition from April 1st to April 5th at four different regions. The original data underwent centering and standardization during the training phase of the deep learning model. The four extracted components were obtained directly from the network. Consequently, their absolute values hold no referential significance, though their relative trends remain interpretable and can serve as meaningful references.

F Additional experimental evidence

F.1 Data, forecasting interfaces, and training

Table 14: Dataset statistics for the STD forecasting evaluation. Split sizes count time points; $L = 16$, $H = 3$ for all panels.

Dataset	Locations	Interval	Training	Validation	Testing
BikeNYC	128	1 h	3,240	384	768
Traffic	100	1 h	12,936	1,536	3,072
SKT	100	6 h	6,716	1,024	1,024

Table 15 describes the panels used for STAD. BikeNYC retains the 81 locations with nonzero training variance; Traffic uses the released 100-sensor panel in its stored units. Capital aggregates public departure records into 79 regions selected using training activity (Capital Bikeshare, n.d.). It trains on 2022–2023 and uses 2024 for development. Ambiguous daylight-saving target hours are excluded; affected historical values are filled from the preceding week. Splits are chronological, and normalization statistics use training observations only. Reported errors retain zero targets.

Four-view implementation. With training-global statistics (μ_0, σ_0), each observed history first becomes $X = (Y - \mu_0)/\sigma_0$, then $U_{n,l} = (X_{n,l} - m_n)/s_n$, where m_n and $s_n^2 = \mathrm{Var}_l(X_{n,l}) + 10^{-5}$ use that location's visible window. Three independently parameterized spatial operators use $1 \to 8 \to 1$ tanh scoring networks and contemporaneous softmax pooling; three temporal operators have node offsets:

$$S_j(V)_{n,l} = \sum_i \frac{\exp g_j(V_{i,l})}{\sum_k \exp g_j(V_{k,l})} V_{i,l}, \qquad R_j(V) = \mathrm{mean}_l(V) - \beta_j. \tag{53}$$

Table 15: STAD panels and forecasting settings. Split sizes count time points per variable; L/H count sampling steps. Short-horizon fits use $L = 336, H = 3$; long fits use the shown L and $H = 192$. Diagnostics retain $L = 168, H = 3$ unless a factor is varied.

Dataset	Variables	Interval	Train	Development	Short L	Long L
BikeNYC	81	1 h	2,760	864	336	—
Capital	79	1 h	17,520	8,784	336	—
Traffic	100	1 h	10,512	3,504	336	720
Electricity	321	1 h	18,412	2,632	—	720
ETTh1	7	1 h	8,640	2,880	—	720
ETTh2	7	1 h	8,640	2,880	—	720
ETTm1	7	15 m	34,560	11,520	—	384
ETTm2	7	15 m	34,560	11,520	—	384
Weather	21	10 m	36,892	5,272	—	720

Writing $C = S_1(U)$, $A = U - C$, $Q = R_3(U)$, and $D = U - Q$, the two paths give

$$c^{\mathrm{ST}} = [A - R_1(A), R_1(A), C - R_2(C), R_2(C)], \tag{54}$$

$$c^{\mathrm{TS}} = [D - S_2(D), Q - S_3(Q), S_2(D), S_3(Q)], \tag{55}$$

$$c_k = (c_k^{\mathrm{ST}} + c_k^{\mathrm{TS}})/2. \tag{56}$$

The four views sum to U. MLP and compact TCN use four prediction heads shared across locations; Raw4 supplies $U/4$ to the same heads. An MLP head adds a linear history-to-horizon path to a width-32 two-layer GELU MLP with dropout 0.1. Each TCN head combines the linear path with six causal depthwise-separable blocks (width 32, kernel three, dilations $1, 2, 4, 8, 16, 32$). Joint iTransformer concatenates the four histories into a $4L$-length variate embedding of width 128. Native MLP/TCN use one input and one head; native iTransformer embeds the original length-L history. The STID diagnostic head uses 32-dimensional identities, three residual blocks, and nine channels: input, hour, weekday, four components, window mean, and window scale. Window and global scales are each inverted once; BikeNYC, Capital, and Traffic forecasts are clipped to zero, whereas ETT, Electricity, and Weather remain signed.

Optimization and scoring. AdamW uses learning rate 10^{-3}, weight decay 10^{-4}, batch 32, gradient clipping at five, training-standardized MSE, and a 30–160 epoch budget. Learning rate halves after six unimproved epochs; early-stopping patience is 20. Checkpoints minimize the development score $(\mathrm{MAE} + \mathrm{RMSE})/\sigma_0$, using pooled RMSE. Forecast origins have stride three; paired models share target cells and seeds. Scores average separately computed per-fit metrics, not ensemble predictions. The extension contains 75 completed fits (32 short, 26 long, 17 native); tables report the actual n. Seeds are 2026 for $n = 1$, 2026/2027 for $n = 2$, and 2026–2028 for $n = 3$. A single fit has no seed SD. Development scores concern the period used for checkpoint selection; the separately frozen Capital 2025 cohort below has an independent-year protocol.

The new adapters and native fits use RTX4090D, deterministic FP32, and disabled TF32. Archived $L = 336$ MLP pairs use the same origins and budget on P100 (BikeNYC/Capital) or RTX5090 (Traffic). Native-to-adapter differences also change architecture and, for MLP, hardware; matched Raw4/STAD contrasts isolate the input representation. Reference models include graph convolution, temporal patching, variate attention, identities, periodic templates, multiscale mixing, and boosted trees (Wu et al., 2019; Nie et al., 2023; Liu et al., 2024; Shao et al., 2022; Lin et al., 2024; Wang et al., 2024; scikit-learn developers, n.d.). Full configurations, source snapshots, predictions, per-seed metrics, and additional diagnostic tables remain in the reproducibility records. Experiment configurations, analysis scripts, saved aggregate results, and reproduction requirements are documented in the anonymous supplementary code.

F.2 Forecasting results across inputs and horizons

Table 16 includes all available short-horizon native, Raw4, and STAD groups. Gains depend on the dataset and backbone: the BikeNYC compact TCN improves MAE/RMSE by 4.10%/4.04%; joint

iTransformer improves both on BikeNYC and Traffic but worsens both on Capital. MLP improves BikeNYC/Capital RMSE while losing both metrics on Traffic. Partial two-seed and single-seed groups retain their actual repetition counts.

Table 16: All available short-horizon comparisons at $L = 336$, $H = 3$. Entries are seed means $\pm$ sample SD; $n = 1$ has no SD. Raw4 and STAD share the predictor interface within each pair. Native rows are separate architectures. MLP paired fits are the archived $L = 336$ cohort; native and new adapter fits belong to the extension.

Predictor	Input interface	n	MAE	RMSE
BikeNYC				
MLP	Native	1	3.789	6.250
MLP	Raw4	3	3.788 ± 0.002	6.264 ± 0.003
MLP	+STAD	3	3.747 ± 0.017	6.175 ± 0.021
TCN	Native	1	3.959	6.490
TCN	Raw4	3	3.947 ± 0.004	6.514 ± 0.004
TCN	+STAD	3	3.785 ± 0.017	6.251 ± 0.007
iTransformer	Native	3	3.778 ± 0.038	6.456 ± 0.021
Joint iTransformer	Raw4	3	3.782 ± 0.057	6.407 ± 0.080
Joint iTransformer	+STAD	3	3.704 ± 0.020	6.299 ± 0.052
Capital				
MLP	Native	3	2.468 ± 0.002	4.878 ± 0.006
MLP	Raw4	3	2.455 ± 0.002	4.854 ± 0.010
MLP	+STAD	3	2.403 ± 0.006	4.779 ± 0.008
TCN	Native	1	2.467	4.872
TCN	Raw4	2	2.502 ± 0.032	4.976 ± 0.077
TCN	+STAD	2	2.367 ± 0.014	4.668 ± 0.044
iTransformer	Native	3	2.345 ± 0.008	4.714 ± 0.025
Joint iTransformer	Raw4	3	2.330 ± 0.005	4.678 ± 0.020
Joint iTransformer	+STAD	3	2.341 ± 0.011	4.705 ± 0.013
Traffic				
MLP	Native	1	86.839	204.782
MLP	Raw4	3	86.362 ± 0.299	204.530 ± 0.520
MLP	+STAD	3	89.412 ± 0.165	205.227 ± 0.054
TCN	Native	1	90.426	208.407
TCN	Raw4	2	94.653 ± 2.323	213.487 ± 2.697
TCN	+STAD	2	94.426 ± 0.514	209.474 ± 0.458
iTransformer	Native	3	85.981 ± 0.448	202.734 ± 0.320
Joint iTransformer	Raw4	3	86.176 ± 0.756	201.953 ± 0.455
Joint iTransformer	+STAD	3	85.642 ± 0.344	201.227 ± 0.183

Extended histories. Table 17 retains every completed dataset pair, including the four single-seed pairs. Traffic has slightly lower STAD MSE but higher MAE; Raw4 has lower errors on the other six datasets. Each model is fitted once per seed at $H = 192$. Archived prefix scores for $h = 3, 24, 96$ reuse that prediction and checkpoint; they are not independently fitted short-horizon models. These findings distinguish local short-horizon gains from longer-horizon performance.

Archived horizon-specific comparison. The separate $L = 168$ study contains 180 fits: three datasets, four training horizons, five models, and three seeds. Table 18 uses 225 BikeNYC, 2,801 Capital, and 1,105 Traffic origins admissible for all horizons. Full horizon-specific support selects checkpoints. iTransformer and PatchTST use width 128, eight heads, feed-forward width 512, and two/three layers, respectively; PatchTST uses patch length 16, stride eight, dropout 0.2, and RevIN. STAD, Raw4, and the LSTM spatial-operator control keep width-32 MLP heads. BikeNYC's paired gains persist; Capital and Traffic retain mixed or unfavorable results. Seed SDs and full-support sensitivity remain in the archived complete tables.

Table 17: All seven extended-history MLP pairs, trained once for $H = 192$. Original-unit MSE/MAE are seed means $\pm$ sample SD; $n = 1$ denotes the completed seed-2026 pair without an SD estimate. Four single-seed pairs are distinguished from the three complete three-seed pairs.

Dataset	L	Input	n	MSE	MAE
Electricity	720	Raw4	1	2365295.1	191.257
		+STAD	1	2669077.6	203.830
Traffic	720	Raw4	3	53931.1 ± 139.0	99.316 ± 0.237
		+STAD	3	53876.5 ± 81.5	101.498 ± 0.018
ETTh1	720	Raw4	1	9.593	2.059
		+STAD	1	10.768	2.173
ETTh2	720	Raw4	3	15.215 ± 0.027	2.675 ± 0.002
		+STAD	3	18.887 ± 0.363	3.075 ± 0.027
ETTm1	384	Raw4	1	7.088	1.561
		+STAD	1	7.318	1.586
ETTm2	384	Raw4	3	10.470 ± 0.032	2.101 ± 0.009
		+STAD	3	10.844 ± 0.055	2.169 ± 0.008
Weather	720	Raw4	1	4586.3	21.980
		+STAD	1	4826.7	22.623

F.3 Frozen independent-year evaluation

Capital models train on 2022–2023, select on 2024, and freeze before accessing 2025. This $L = 168, H = 3$ cohort uses the same 79-region panel, three-hour stride, initial history from late 2024, and 2,919 origins (691,803 target cells, including 195,099 zeros). Table 19 retains all ten registered model/control groups and 22 fits. Raw4 preserves four heads; LSTM replaces spatial aggregation; the entity-matched MLP reads no other region and has STAD's 24,368 parameters. Six groups have three seeds; other models retain one fit. The period has now been evaluated and is not an untouched set for subsequent tuning.

The prespecified bootstrap uses 4,000 circular resamples and 7-/28-day blocks, with 95% simultaneous intervals over all nine comparator–two metric contrasts (Table 20). STAD–MLP improves Raw4 MAE/RMSE by 2.63%/1.93%, with positive bands at both block lengths. It also improves both metrics over TimeMixer and HGB. Native STID and iTransformer have the lowest MAE and RMSE, respectively. These bands quantify variation over evaluated days conditional on the frozen fits; single-fit references do not estimate training-seed variability.

F.4 Temporal and spatial representation diagnostics

Paired backbone diagnostics. The paired TCN, Transformer, and WaveNet rows in Table 2 use archived reruns, separate from Table 1's inherited scores. Each cell uses one fit, not a seed average.

The archived arrays contain all three leads and 768/3,072/1,024 origins on BikeNYC/Traffic/SKT. Each target was checked against the source series. The first two origins are dropped because the legacy window indexing places their earliest targets before the nominal evaluation boundary. The six archived models share 766/3,070/1,022 origins and 128/100/100 locations. Errors use the original scalar training-window standard deviation, retaining constant-training locations: $e_{o,h,n} = (\widehat{y} - y)/s_{\text{train}}$. We compute $M_g = \text{mean}_{o,h}(\bar{e}^2_{o,h})$ and $M_l = \text{mean}_{o,h,n}[(e_{o,h,n} - \bar{e}_{o,h})^2]$; their sum is total standardized MSE.

Temporal errors use consecutive complete 28-day windows from the first retained origin, giving 1/4/9 windows at sampling intervals 1/1/6 hours. Each regional series is demeaned and Hann-windowed. For each lead, $E_B = \sum_{w,n,k\in B} |P - Y|^2 / \sum_{w,n,k\in B} |Y|^2$, with daily bins 27–29 and weekly bins 3–5. The table averages the three lead-specific ratios. Final partial windows are

Table 18: Complete archived horizon-specific study: 180 fits, $L = 168$, three seeds, and five models at each horizon. Each cell reports mean MAE (upper line) and pooled RMSE (lower line) on common forecast origins. LSTM changes only spatial aggregation in the four-head MLP. Each H uses separately trained models.

Model	$H = 3$	$H = 24$	$H = 96$	$H = 192$
BikeNYC				
STAD–MLP	3.801	4.100	4.297	4.264
	6.284	6.895	7.216	7.202
Raw4–MLP	3.935	4.161	4.384	4.401
	6.509	7.017	7.402	7.450
LSTM	4.011	4.234	4.391	4.340
	6.640	7.115	7.365	7.319
iTransformer	3.725	4.136	4.334	4.367
	6.332	7.144	7.413	7.486
PatchTST	3.821	4.154	4.204	4.182
	6.327	7.003	7.149	7.130
Capital				
STAD–MLP	2.413	2.693	2.842	2.882
	4.760	5.544	5.959	6.076
Raw4–MLP	2.498	2.679	2.828	2.869
	4.909	5.503	5.908	6.043
LSTM	2.424	2.716	2.864	2.894
	4.825	5.597	5.995	6.105
iTransformer	2.346	2.624	2.805	2.849
	4.669	5.463	5.930	6.060
PatchTST	2.403	2.603	2.744	2.797
	4.727	5.325	5.732	5.901
Traffic				
STAD–MLP	90.557	103.734	109.907	110.175
	208.123	231.688	244.628	246.531
Raw4–MLP	90.677	101.963	106.554	106.705
	211.623	234.424	244.424	245.807
LSTM	90.514	103.993	110.848	111.161
	207.544	232.019	244.310	246.278
iTransformer	86.616	98.456	103.946	105.878
	205.427	225.282	233.320	236.494
PatchTST	82.321	92.443	98.051	98.891
	202.127	221.040	231.896	234.181

Table 19: Capital 2025 independent-year performance, $L = 168, H = 3$. Original-unit errors are averages of frozen-fit metrics; n is the number of fits.

Model or control	n	MAE	SD	RMSE	SD	MAE rank
STAD–MLP	3	2.6000	0.0083	5.1940	0.0151	4
MLP Raw4	3	2.6702	0.0057	5.2962	0.0054	9
LSTM aggregation–MLP	3	2.6170	0.0098	5.2703	0.0291	5
Entity-matched MLP	3	2.6331	0.0071	5.2193	0.0104	6
Graph WaveNet	3	2.5803	0.0154	5.1688	0.0210	3
CycleNet	3	2.6385	0.0065	5.2345	0.0153	7
iTransformer	1	2.5308	–	5.1006	–	2
STID	1	2.5206	–	5.1308	–	1
TimeMixer	1	2.7158	–	5.3438	–	10
HGB	1	2.6554	–	5.4218	–	8

Table 20: Capital 2025: all 18 prespecified STAD–MLP comparisons. Positive gain favors STAD. The 95% simultaneous intervals use 4,000 circular block bootstrap repetitions and jointly cover nine comparators and two metrics.

Comparator	MAE gain (%) Estimate	7-day interval	28-day interval	RMSE gain (%) Estimate	7-day interval	28-day interval
MLP Raw4	2.63	[1.62,3.63]	[1.65,3.60]	1.93	[0.24,3.59]	[0.51,3.33]
LSTM–MLP	0.65	[0.26,1.04]	[0.27,1.03]	1.45	[0.75,2.14]	[0.66,2.23]
Entity-matched	1.26	[0.25,2.26]	[0.32,2.19]	0.48	[-1.13,2.07]	[-0.89,1.84]
Graph WaveNet	-0.76	[-2.37,0.82]	[-3.00,1.43]	-0.49	[-2.88,1.85]	[-3.57,2.50]
CycleNet	1.46	[0.27,2.64]	[0.30,2.61]	0.77	[-1.06,2.57]	[-0.79,2.32]
iTransformer	-2.73	[-3.70,-1.77]	[-3.84,-1.64]	-1.83	[-3.53,-0.17]	[-3.29,-0.40]
STID	-3.15	[-4.40,-1.91]	[-4.42,-1.90]	-1.23	[-3.30,0.79]	[-3.05,0.55]
TimeMixer	4.26	[3.10,5.41]	[2.91,5.60]	2.80	[0.82,4.75]	[0.81,4.75]
HGB	2.09	[0.74,3.41]	[0.65,3.50]	4.20	[1.75,6.59]	[1.66,6.68]

omitted identically for all models. All nine pairs improve weekly, common, and local errors, while all three SKT pairs worsen daily-band errors. These paired comparisons use single archived fits.

Table 21: STAD on Capital, $L = 168$, $H = 3$. Temporal errors use lead one and eleven 28-day windows; spatial MSE pools 2,927 origins and three leads. n counts fits. Bold/underline rank the eight predictors; matched controls are separate.

Model	n	Temporal E_B (lead 1)↓ Daily	Half-day	Weekly	Mod.	Spatial MSE↓ Common	Local
STAD-MLP	3	**0.0387**	**0.1949**	<u>0.4176</u>	<u>0.5765</u>	0.0834	<u>0.8844</u>
STAD-STID	3	0.0463	0.2243	0.4365	0.5780	**0.0654**	**0.8564**
Graph WaveNet	3	0.0746	0.3648	0.4482	0.6208	<u>0.0657</u>	0.9340
CycleNet	3	0.0467	0.2214	0.4665	0.6175	0.0879	0.9039
iTransformer	1	<u>0.0437</u>	<u>0.2099</u>	**0.4150**	0.5830	0.0793	0.8850
STID	1	0.0560	0.2495	0.4876	0.5945	0.1121	0.8900
TimeMixer	1	0.0872	0.3072	0.4502	0.6088	0.0934	0.9144
HGB	1	0.0441	0.2223	0.4186	**0.5310**	0.0913	0.9045
Paired STID representation controls							
STID–Raw4	3	0.0456	0.2190	0.4766	0.5910	0.0847	0.8665
STID–LSTM	3	0.0473	0.2282	0.4321	0.5748	0.0726	0.8575

Capital representation diagnostics. Capital spectral diagnostics standardize each region using training statistics. At each lead, three-hourly rolling forecasts form eleven 28-day windows, each demeaned and Hann-windowed. For observed and predicted complex spectra Y and P, band error is $E_B = \sum_{w,n,k\in B} |P - Y|^2 / \sum_{w,n,k\in B} |Y|^2$. Daily, half-day, weekly, and modulation bands use bins 27–29, 55–57, 3–5, and 23–25∪31–33. Scores pool numerators/denominators before averaging per-fit ratios. STAD–MLP leads six external references on daily/half-day errors at all three leads, but its paired Raw4 band errors increase at leads two and three. Thus model-level periodic accuracy and the module's incremental gain are distinct.

For spatial errors, $e_{o,h,n} = (\widehat{y} - y)/\sigma_n$ uses training regional scales. Common MSE averages $\bar{e}^2_{o,h}$ with $\bar{e} = N^{-1} \sum_n e$; local MSE averages $(e - \bar{e})^2$. Their sum is total standardized MSE. STAD–STID lowers common/local error by 22.80%/1.17% versus its matched Raw4. Daily lead-one spectral error decomposes into amplitude/phase terms: STAD–MLP $0.0219/0.0167$, Graph WaveNet $0.0481/0.0265$. These are prediction-error projections, not attribution of errors to learned branches.

Figure 2 shows development representations using a shared PCA fitted to training input snippets (last 24 hours of each window). The sampling rule yields 235 training and 120 development days,

with origins at 00:00, 09:00, and 18:00; three regions are fixed by training-activity ranks. Full-dimensional between-group scatter for the input and intermediate common/local representations is $B = \sum_g n_g \|\mu_g - \mu\|^2 / \sum_i \|v_i - \mu\|^2$. Time groups explain 34.43%, 69.12%, and 1.20% for input, spatial-first common, and local representations. The common view is shared across regions by construction. Shape normalization removes each snippet's mean and RMS; local regional scatter then rises from 0.48% to 1.10%. The six contours enclose 15–90% KDE probability mass in 15% increments. Axis limits match within each column; plotted sizes are not comparable across representations. Contours depict densities, not confidence regions. These measurements describe learned structure, without establishing causal or uniquely identified physical components.

On 14 frozen nighttime origins, deleting the common-dynamic prediction branch raises STAD–MLP MAE from 0.6297 to 5.5306; deleting local dynamic, local reference, or common reference gives 1.4417, 0.8609, or 0.9581. This probes fitted-model reliance. A sum-preserving reference transfer instead reduces MAE by 4.40%, so reconstruction does not make the learned allocation uniquely necessary. The full per-lead spectral tables, component statistics, interventions, and derivative checks are retained in the reproducibility records.

F.5 Ablations and sensitivity

The ablation replaces temporal references by $R(V) = V/2$, spatial operators by $S(V) = V/2$, or both; uniform pooling substitutes contemporaneous node means. Each fit first initializes the full model and then replaces the selected operators, retaining predictor initialization. These controls preserve the four-view interface, but remove the corresponding parameters. Sixty retrained fits cover five variants for MLP on three datasets and STID on Capital (Table 22). Spatial bypass raises mean RMSE in all four groups; temporal bypass is mixed, and uniform pooling is strongest on Capital.

Table 22: All 60 retrained ablation fits at $L = 168$, $H = 3$. Each cell reports MAE (upper line) and RMSE (lower line), mean $\pm$ sample SD over seeds 2026–2028. Uniform uses contemporaneous spatial means; both bypasses give Raw4.

Configuration	BikeNYC / MLP	Capital / MLP	Traffic / MLP	Capital / STID
Full STAD	3.752 ± 0.014	2.413 ± 0.009	90.294 ± 0.506	2.297 ± 0.008
	6.160 ± 0.015	4.799 ± 0.021	207.653 ± 0.511	4.553 ± 0.016
Temporal bypass	3.768 ± 0.009	2.414 ± 0.002	90.354 ± 0.128	2.304 ± 0.012
	6.194 ± 0.014	4.797 ± 0.003	207.692 ± 0.124	4.537 ± 0.006
Spatial bypass	3.831 ± 0.010	2.461 ± 0.012	89.065 ± 0.746	2.355 ± 0.008
	6.302 ± 0.011	4.891 ± 0.021	208.445 ± 1.120	4.703 ± 0.016
Uniform pooling	3.774 ± 0.010	2.406 ± 0.006	93.913 ± 1.673	2.293 ± 0.007
	6.188 ± 0.010	4.782 ± 0.009	209.553 ± 1.528	4.540 ± 0.006
Both bypasses	3.882 ± 0.008	2.497 ± 0.008	90.441 ± 0.889	2.349 ± 0.002
	6.394 ± 0.016	4.943 ± 0.028	211.193 ± 1.346	4.694 ± 0.008

Factor grids. The sensitivity study varies $L \in \{24, 48, 96, 168, 336\}$, scorer width $s \in \{2, 4, 8, 16, 32\}$, head width $d \in \{8, 16, 32, 64, 128\}$, and dropout $p \in \{0, 0.1, 0.2, 0.3\}$ from $(168, 8, 32, 0.1)$. Raw4 is paired on history and head width. Reusing the shared default gives 75 unique configurations and 225 fits. All histories use training origins valid for $L = 336$ (808/5,726/3,392 for BikeNYC/Capital/Traffic), distinct from ablation training. BikeNYC/Capital improve RMSE over Raw4 at every history; Traffic loses at $L = 336$. Figure 3 summarizes the complete factor-wise sensitivity evaluations; full per-seed tables retain MAE, RMSE, parameter counts, and epochs.

F.6 Corruption robustness

Eight frozen Capital predictors share 360 origins, 79 regions, and 85,320 clean target cells. Independent windows receive 5/10/20% random missing entries, causally forward-filled with training-mean fallback, or Gaussian noise with SD $\lambda\sigma_{n,\mathrm{train}}$ for $\lambda = 0.05, 0.1, 0.2$, followed by nonnegative

input clipping. Three shared perturbation draws per level quantify corruption variability, not retraining variation. At 20% missingness STAD–STID has the lowest MAE/RMSE, 2.445/4.961; under $\lambda = 0.2$, native STID is lower (2.372/4.728 versus 2.383/4.777). A separately fixed high-change subset uses the training 95th percentile of 24-hour changes; Graph WaveNet remains better than STAD–STID there. Table 23 includes all models and severities; the full RMSE grids and perturbation variation remain in the stress-test records.

Table 23: Capital corruption grid: MAE at every severity and RMSE at the largest severity. One frozen fit per model; nonzero conditions average three shared perturbation draws. Missingness uses causal filling; Gaussian SD is λ times training regional SD.

	Clean		Missing history				Gaussian noise			
Model	MAE	RMSE	5%	10%	20%	RMSE (20%)	.05	.1	.2	RMSE (.2)
STAD-STID	2.351	4.756	2.366	2.385	2.445	4.961	2.356	2.363	2.383	4.777
STAD-MLP	2.479	4.857	2.515	2.556	2.672	5.307	2.485	2.494	2.523	4.888
Graph WaveNet	2.461	4.961	2.486	2.519	2.614	5.351	2.464	2.471	2.498	5.008
CycleNet	2.682	5.158	2.734	2.795	2.948	5.792	2.691	2.707	2.759	5.250
iTransformer	2.386	4.791	2.400	2.420	2.512	5.047	2.389	2.394	2.413	4.800
STID	2.348	4.695	2.377	2.411	2.499	5.110	2.349	2.354	2.372	4.728
TimeMixer	2.636	5.106	2.669	2.709	2.830	5.479	2.638	2.643	2.666	5.117
HGB	2.617	5.289	2.652	2.693	2.803	5.676	2.674	2.719	2.774	5.535