FSC: BRIDGING PERIODIC PATTERNS AND LOCAL DYNAMICS IN MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series forecasting is widely used in domains such as traffic, energy, and finance, where effective forecasting requires capturing both stable periodic patterns and rapidly evolving local dynamics. However, under distribution shifts, periodic priors learned from historical data can become mismatched with fast local dynamics, leading to prediction lag. To address this issue, we introduce Fast-to-Slow Correction (FSC), a lightweight and plug-and-play framework that can be seamlessly integrated into existing forecasting models. FSC explicitly decouples fast and slow features via parallel pathways, and employs a learnable refinement module to correct slow periodic features using fast local dynamics as adaptive references. This design effectively mitigates prediction lag in periodic modeling without modifying the backbone architecture. Extensive experiments on 12 real-world datasets show that FSC achieves state-of-the-art results on 9 datasets, with an average 2.51% MSE reduction over strong periodic forecasting methods. FSC also achieves these improvements with minimal computational overhead, introducing only $\sim$1.26M additional parameters. We will release code and detailed configurations upon acceptance to ensure reproducibility.

1 INTRODUCTION

Multivariate Time Series Forecasting (MTSF) is widely applied across a variety of real-world scenarios (Wu et al., 2021; Huang et al., 2025; Qiu et al., 2025a; Zhou et al., 2022a;b), such as energy forecasting (Chengqing et al., 2023), traffic flow prediction (Khan et al., 2023), human motion analysis (Chen et al., 2026), and Industrial Internet of Things (IIoT) monitoring (Sun et al., 2023). In these applications, forecasting performance typically depends on both short-term dynamics in recent observations and long-term temporal structures, such as periodic patterns, which have been extensively studied in prior work (Wu et al., 2021; Lin et al., 2024b). While periodic structures provide stable structural priors, short-term dynamics reflect rapidly evolving variations. However, in real-world deployments, the available look-back window is often much shorter than the intrinsic periodicity of the data (e.g., yearly, seasonal, or monthly cycles) (Lin et al., 2024a), making it difficult to observe complete cycles, which in turn makes it challenging for models to infer global patterns from incomplete observations while adapting to fast-changing local dynamics. As a result, existing models struggle to reconstruct global periodicity from limited windows while adapting to rapid temporal variations, ultimately degrading forecasting performance.

To mitigate the limitations of restricted input horizons, recent studies have explored various modeling strategies to enhance periodic pattern modeling within local windows. A common line of work decomposes the input series into multiple components and models them separately to capture trend and seasonal patterns (Zhou et al., 2022b; Wang et al., 2024b). Another direction introduces learnable periodic representations to explicitly encode global cyclic structures beyond the observed window (Lin et al., 2024a). More recent approaches further strengthen periodic awareness by leveraging inter-channel dependencies, frequency-domain transformations, or enlarged receptive fields to better capture long-range temporal dependencies (Cao et al., 2026; Lin et al., 2025a; Zhang et al., 2026a). Despite their architectural differences, these methods share a key assumption: periodic representations are extracted from historical data and treated as stable priors for future forecasting.

However, this assumption often breaks under temporal distribution shifts, where global periodic patterns learned from historical data become misaligned with real-time evolving dynamics. As shown

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.

in Figure 1, periodic priors are learned during training (Figure 1(a)) to capture stable slow-varying patterns and remain fixed during inference (Figure 1(b)). While effective under stable conditions, these priors struggle to adapt to rapidly changing local dynamics. This misalignment introduces temporal lag between static periodic representations and real-time observations. As a result, models struggle to accurately align long-term periodicity with short-term variations, leading to degraded forecasting performance.

To overcome this limitation, we propose a locally-driven Fast-to-Slow Correction (FSC) framework, a lightweight and plug-and-play module. Unlike prior work that treats periodic features as stationary priors or adapts them implicitly, FSC explicitly models the temporal misalignment (lag) between slow periodic and fast dynamic features and introduces a learnable fast-to-slow correction mechanism to adapt periodic representations online without modifying the backbone architecture. Specifically, FSC decouples the input into slow periodic and fast dynamic features through two parallel branches. The slow branch captures stable long-range periodic regularities through window-specific periodic embeddings, while the fast branch extracts local temporal variations and channel-specific structural information using Adaptive Channel Encoding (ACE). To bridge the temporal misalignment between the two temporal scales, the *FSC-Refiner* uses fast dynamic features as adaptive references to correct the slow periodic representations, enabling them to better track evolving temporal dynamics. Extensive experiments demonstrate that FSC achieves state-of-the-art performance across multiple real-world datasets with low computational overhead, effectively reducing the mismatch between periodic regularities and current dynamics.

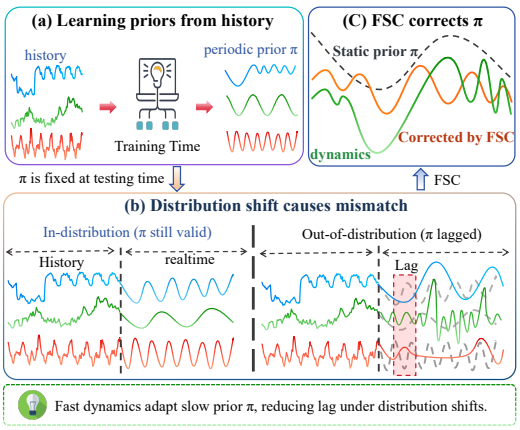

Figure 1: **Motivation of FSC.** (a) periodic priors π are learned from historical data. (b) Under distribution shifts, the fixed prior π becomes lagged behind current dynamics. (c) FSC uses fast local dynamics to correct π, reducing temporal lag.

Our main contributions are summarized as follows:

- We identify a key issue in MTSF: under distribution shifts, there exists a temporal misalignment between slow periodic patterns and fast-evolving local dynamics, leading to lagging predictions.
- We propose Fast-to-Slow Correction (FSC), a lightweight, plug-and-play, and model-agnostic framework that explicitly models temporal misalignment by decoupling slow periodic and fast dynamic features, and introduces a learnable fast-to-slow correction mechanism for adaptive alignment.
- Extensive experiments on multiple real-world benchmarks show that FSC consistently improves diverse forecasting backbones and achieves state-of-the-art performance with minimal computational overhead.

2 Related Work

Recent advances in deep learning have led to substantial improvements in long-term time series forecasting. Existing methods are primarily built on a range of architectural backbones, including recurrent neural networks (RNNs) (Lin et al., 2025b; Wang et al., 2024a), convolutional neural networks (CNNs) (Liu et al., 2022; Luo & Wang, 2024), multilayer perceptrons (MLPs) (Das et al., 2023; Lin et al., 2024a), Transformers (Liu et al., 2023; Wang et al., 2024c), and more recent Mamba-based architectures (Wang et al., 2025; Ma et al., 2025). Beyond architectural design, recent methods mainly focus on three aspects: improving robustness to temporal distribution shifts, capturing channel dependencies and modeling temporal periodicity.

Temporal Distribution Shifts in MTSF. Time series forecasting is often affected by temporal distribution shifts (TDS) (Liu et al., 2024; Dai et al., 2024), where data distributions change over time.

Prior work has approached this problem from several directions. Normalization-based methods, such as DAIN (Passalis et al., 2019) and RevIN (Kim et al., 2021), reduce non-stationarity by adjusting input statistics. Methods like AdaRNN (Du et al., 2021) instead model distribution changes through feature-level adaptation. Another line of work improves adaptability by introducing time-varying parameters, as seen in Triformer (Cirstea et al., 2022). Building on these strategies, DUET (Qiu et al., 2025b) further enhances adaptability by modeling heterogeneous temporal patterns via a Temporal Clustering Module. These studies highlight the importance of modeling temporal distribution changes for improving forecasting performance.

Channel Modeling in MTSF. Multivariate time series forecasting (MTSF) relies on effective modeling of inter-channel dependencies. Existing work mainly follows two directions. One line of work models channel dependencies through channel ordering. For instance, MambaTS (Cai et al., 2024) formulates the ordering as an asymmetric traveling salesman problem, while S-Mamba (Wang et al., 2025) shows its impact via bidirectional scanning. The other approach models dependencies through cross-channel interactions, either by stabilizing global relationships with channel embeddings (e.g., EMAformer (Zhang et al., 2026b)), or by capturing multi-scale 2D variations in the frequency domain (e.g., TimesNet (Wu et al., 2022)). Despite these efforts, most channel modeling methods are still driven by short-term dynamics and provide limited modeling of global temporal patterns, particularly periodic structures.

Modeling Periodicity in MTSF. Periodic patterns are an important component in time series forecasting, and have been extensively studied in prior work. Early works, such as Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022b), DLinear (Zeng et al., 2023), and the TimeMixer family (Wang et al., 2024b), rely on seasonal–trend decomposition to separate periodic components from raw sequences, resulting in implicit periodic features. More recent work moves toward explicit modeling of periodic features. CycleNet (Lin et al., 2024a) parameterizes periodic features via learnable embeddings, while TQNet (Lin et al., 2025a) builds on this idea by incorporating global periodic features as queries in attention. Subsequent extensions focus on improving modeling capacity: FreqCycle (Zhang et al., 2026a) captures cross-scale couplings among nested periodicities via multi-scale frequency enhancement, and GTR (Cao et al., 2026) expands the receptive field through global retrieval to capture very long-range periodicity. Although recent methods have made progress, most still assume that periodic features are stable over time. However, under temporal distribution shifts, such stable periodic features may fail to adapt to evolving local dynamics, leading to a misalignment between learned periodic representations and current temporal dynamics.

3 METHODS

Consider a multivariate time series with C channels. We adopt a channel-first representation and denote the input sequence as $\mathbf{x} \in \mathbb{R}^{C \times L}$, where L is the input sequence length. The goal is to learn a mapping function $g : \mathbb{R}^{C \times L} \to \mathbb{R}^{C \times H}$ that predicts the future sequence $\mathbf{Y} \in \mathbb{R}^{C \times H}$, where H is the forecasting horizon.

3.1 OVERVIEW OF FSC

Fast-to-Slow Correction (FSC) addresses a key issue in multivariate time series forecasting: under distribution shifts, periodic slow features often lag behind rapidly changing local dynamics. In contrast to prior methods that treat periodic features as stationary priors, FSC models this lag as a learnable misalignment between slow periodic features and fast dynamic features. To address this issue, FSC uses fast features as adaptive anchors and applies correction that pulls slow features toward fast features, encouraging slow features to better reflect current temporal variations rather than relying solely on historical periodic structure. This interaction between fast features and slow features forms the core principle of the FSC framework.

As shown in Figure 2, FSC consists of three stages: representation construction, fast-slow decoupling, and FSC-Refiner. The first stage constructs the periodic prior π via template lookup and the channel-aware ACE representation. The second stage maps π and the ACE-enhanced input to slow and fast features, respectively. The **FSC-Refiner** serves as the core module, using fast local dynamics to correct slow features and produce a corrected feature h_{fsc}, thereby reducing lag.

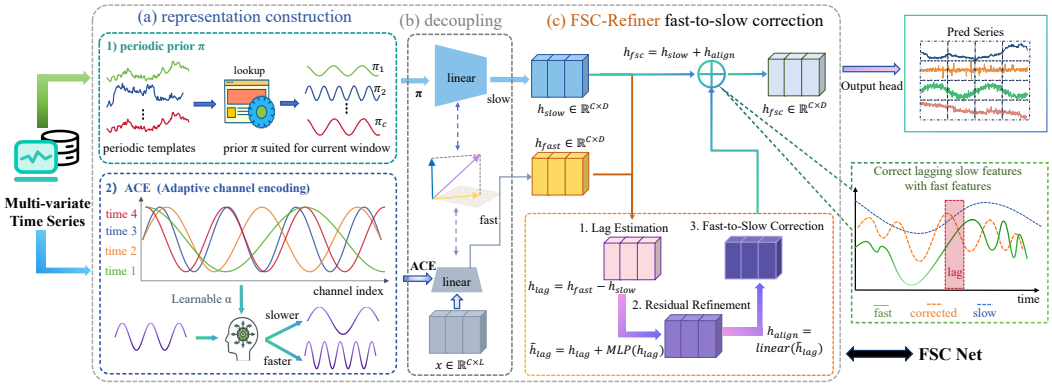

Figure 2: **Overview of the FSC framework.** FSC consists of three stages: **(a) Representation Construction:** extracting the periodic prior π via template lookup and constructing ACE for channel-wise spatial encoding; **(b) Fast-Slow Decoupling:** mapping the periodic prior and ACE-enhanced input into decoupled slow and fast features. **(c) FSC-Refiner:** the core component that uses fast features to correct slow features, producing corrected features that reduces lag.

3.2 Representation Construction

To capture global periodic patterns beyond the local receptive window, we adopt the periodic embedding mechanism of CycleNet (Lin et al., 2024a). Given a predefined period length P, we construct a learnable periodic embedding matrix $\mathbf{Q} \in \mathbb{R}^{C \times P}$, which is globally shared across the temporal dimension. For an input sequence starting at time index t, periodic embeddings are retrieved according to their phase indices, yielding the window-aligned periodic prior:

$$\boldsymbol{\pi} = \text{Lookup}\left(\mathbf{Q}, [(t+\tau) \bmod P]_{\tau=0}^{L-1}\right) \in \mathbb{R}^{C \times L}. \tag{1}$$

To enhance local feature representations, we introduce a lightweight Adaptive Channel Encoding (ACE), as illustrated in Figure 2(a). Given an embedded representation $\mathbf{z} \in \mathbb{R}^{B \times C \times D}$, ACE encodes each channel $v \in \{0, \ldots, C-1\}$ using sinusoidal functions: $\text{ACE}(v)_{2k} = \sin(v\tilde{\omega}_k)$ and $\text{ACE}(v)_{2k+1} = \cos(v\tilde{\omega}_k)$, where $\tilde{\omega}_k = \alpha_k \omega_k$, $\omega_k = \exp(-2k\ln(10000)/D)$, and $k = 0, \ldots, D/2-1$. Here, $\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_{D/2-1}]$ is a learnable frequency scaling vector that adaptively modulates each frequency component. Stacking all channel-wise encodings yields $\mathbf{ACE} \in \mathbb{R}^{C \times D}$, which is added to the input representation:

$$\mathbf{z}' = \mathbf{z} + \mathbf{ACE}. \tag{2}$$

3.3 Fast-Slow Decoupling

As shown in Figure 2(b), we project the periodic prior $\boldsymbol{\pi}$ and the current dynamics $\mathbf{x} + \mathbf{ACE}$ into two separate representation spaces:

$$\mathbf{h}_{\text{slow}} = \text{Linear}_{\text{slow}}(\boldsymbol{\pi}), \qquad \mathbf{h}_{\text{fast}} = \text{Linear}_{\text{fast}}(\mathbf{x} + \mathbf{ACE}). \tag{3}$$

With independent projections, $\mathbf{h}_{\text{slow}}$ emphasizes slowly-varying periodic information, while $\mathbf{h}_{\text{fast}}$ captures fast-varying local dynamics, reducing the information coupling between the two representations.

3.4 Fast-to-Slow Correction (FSC)

Most forecasting models rely on slow-varying periodic features to capture long-term temporal patterns. While such representations are effective under relatively stable distributions, they may become misaligned with rapidly evolving local dynamics under distribution shifts, causing the periodic representation to lag behind the current temporal state. To address this issue, we introduce the **FSC-Refiner**, the core component of our framework, which explicitly models the discrepancy between slow periodic patterns and fast local dynamics and uses it to adapt the slow representation.

Lag Modeling and Fast-to-Slow Correction. Given the dual-pathway features $\mathbf{h}_{\text{slow}}, \mathbf{h}_{\text{fast}} \in \mathbb{R}^{C\times D}$ produced by the decoupling module, we first estimate the discrepancy between the slow-varying periodic representation and the current local dynamics. Specifically, their feature-wise difference provides an explicit lag representation:

$$\mathbf{h}_{\text{lag}} = \mathbf{h}_{\text{fast}} - \mathbf{h}_{\text{slow}} \in \mathbb{R}^{C\times D}. \tag{4}$$

The estimated lag may involve nonlinear discrepancies as the current dynamics deviate from historical periodic patterns under distribution shifts. We therefore refine the lag feature with a two-layer MLP equipped with ReLU activation. A residual connection is adopted to preserve the original lag information while enabling nonlinear refinement: $\tilde{\mathbf{h}}_{\text{lag}} = \mathbf{h}_{\text{lag}} + \text{MLP}(\mathbf{h}_{\text{lag}})$.

The refined lag feature is then converted into a correction signal for the slow representation. Rather than directly fusing the fast and slow features, we use a learnable transformation to project the lag information into the representation space of $\mathbf{h}_{\text{slow}}$:

$$\mathbf{h}_{\text{align}} = \text{Linear}_{\text{fs}}\left(\text{Dropout}(\tilde{\mathbf{h}}_{\text{lag}})\right). \tag{5}$$

The resulting correction is added to the slow representation, allowing the periodic representation to adapt to current local dynamics while preserving its underlying slow-varying structure:

$$\mathbf{h}_{\text{fsc}} = \mathbf{h}_{\text{slow}} + \mathbf{h}_{\text{align}} \in \mathbb{R}^{C\times D}. \tag{6}$$

In this way, FSC does not directly replace or fuse the slow and fast representations. Instead, it converts their discrepancy into a corrective signal that updates the slow periodic representation, thereby reducing temporal lag while retaining long-term periodic information.

Finally, the corrected representation is projected to the forecasting horizon through an output head:

$$\hat{\mathbf{Y}} = \text{Linear}_{\text{out}}\left(\text{Dropout}(\mathbf{h}_{\text{fsc}})\right) \in \mathbb{R}^{C\times H}, \tag{7}$$

where H denotes the forecasting horizon.

3.5 Loss Function

To encourage the correction representation to provide information beyond the slow periodic representation, we introduce an orthogonal decoupling loss alongside the forecasting objective. Specifically, the loss encourages $\mathbf{h}_{\text{align}}$ to be complementary to $\mathbf{h}_{\text{slow}}$ by penalizing their feature-wise inner product:

$$\mathcal{L}_{\text{ortho}} = \frac{1}{B\times C}\sum_{b=1}^{B}\sum_{c=1}^{C}\left(\mathbf{h}_{\text{slow}}^{(b,c)}\cdot\mathbf{h}_{\text{align}}^{(b,c)}\right)^2, \tag{8}$$

where B and C denote the batch size and channel number, respectively, and $\cdot$ denotes the dot product along the feature dimension D. By suppressing redundant components shared with $\mathbf{h}_{\text{slow}}$, this regularization encourages $\mathbf{h}_{\text{align}}$ to retain complementary information that is not captured by the slow periodic representation, thereby making the correction more informative for adapting the slow features.

The overall framework is trained end-to-end with the objective $\mathcal{L} = \mathcal{L}_{\text{forecast}} + \lambda\mathcal{L}_{\text{ortho}}$. where $\mathcal{L}_{\text{forecast}} = \text{MSE}(\hat{\mathbf{Y}}, \mathbf{Y})$ is the forecasting loss and λ controls the strength of the orthogonal decoupling regularization.

4 Experiments

4.1 Experimental Details

Datasets. We evaluate the proposed method on 12 widely used multivariate time series forecasting benchmarks, covering 8 long-term forecasting datasets (ETTh1, ETTh2, ETTm1, ETTm2 (Zhou

et al., 2021), Electricity, Weather, Traffic (Wu et al., 2022), and Solar-Energy (Lin et al., 2024a)) and 4 short-term PEMS (Liu et al., 2022) traffic datasets (PEMS03, PEMS04, PEMS07, and PEMS08). Detailed descriptions of these datasets are provided in the Appendix A.1.

Dataset Distribution Characteristics. To characterize the statistical properties of these benchmarks, we report the Augmented Dickey-Fuller (ADF) (Said & Dickey, 1984) statistic to assess their stationarity. During testing, we further measure the Pearson correlation between the historical periodic prior π and the current real-time dynamics x to quantify their consistency under the test distribution, where a smaller Pearson coefficient indicates a larger distribution gap between π and x. The resulting statistics are summarized in Table 1.

Table 1: Statistical properties of benchmark datasets. ADF reflects the overall distribution and stationarity (more negative values indicate stronger stationarity). *pearson.corrs* denotes the correlation between historical periodic priors and real-time dynamics during testing (details in Appendix A.4).

Metric	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Traffic	Solar	PEMS03	PEMS04	PEMS07	PEMS08
ADF	-5.91	-4.13	-14.98	-5.66	-26.68	-8.44	-15.02	-37.23	-19.05	-15.66	-20.60	-16.04
pearson.corrs	0.090	0.073	0.116	0.220	0.152	0.130	0.080	-0.056	-0.064	-0.077	0.056	0.026

Baselines. We compare our method with a diverse set of representative state-of-the-art baselines, including Transformer-based models (iTransformer (Liu et al., 2023)), periodic modeling methods (CycleNet (Lin et al., 2024a), TQNet (Lin et al., 2025a), GTR (Cao et al., 2026), and FreqCycle (Zhang et al., 2026a)), channel modeling approaches (S-Mamba (Wang et al., 2025) and DLinear (Zeng et al., 2023)), lightweight forecasting methods (FITS (Xu et al., 2024) and SparseTSF (Lin et al., 2024b)), distribution-aware frameworks (DUET (Qiu et al., 2025b)), and hybrid architectures (SOFTS (Han et al., 2024)).

Setup. All experiments are implemented in PyTorch and conducted on a single NVIDIA RTX 3090 GPU with 24GB memory. To ensure a fair comparison, we follow the experimental setup of iTransformer (Liu et al., 2023). We report mean absolute error (MAE) and mean squared error (MSE) as evaluation metrics. The input length is fixed to $L = 96$. For the PEMS datasets, the prediction horizon is set to $H \in \{12, 24, 48, 96\}$, while for the remaining datasets, $H \in \{96, 192, 336, 720\}$.

4.2 Main Results

Extensive experiments on 12 real-world datasets show that FSC achieves the best performance on 9 datasets, reducing MSE by 2.51% on average compared with the best non-FSC methods.

Long-term Forecasting. Table 2 reports the performance of FSR on eight multivariate long-term forecasting benchmarks, with detailed results provided in Table 11 of the Appendix C.1. Note that FreqCycle reports results on only 7 datasets and is therefore compared within this subset.

Table 2: Long-term forecasting results (input length: 96), averaged over $H \in \{96, 192, 336, 720\}$. Best and second-best results are marked in **bold** and underlined, respectively. Lower is better.

model	FSC (Ours)		GTR (2026)		FreqCycle (2026)		DUET (2025)		TQNet (2025)		S-Mamba (2025)		CycleNet (2024)		SOFTS (2024)		FITS (2024)		SparseTSF (2024)		iTransformer (2024)		DLinear (2023)	
metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.417**	0.432	0.439	0.434	0.425	**0.427**	0.443	0.436	0.441	0.434	0.455	0.450	0.457	0.441	0.449	0.443	0.483	0.463	0.450	0.426	0.454	0.448	0.456	0.452
ETTh2	**0.368**	0.399	0.372	0.400	0.371	0.399	0.372	**0.397**	0.378	0.402	0.381	0.405	0.388	0.409	0.373	0.400	0.385	0.405	0.385	0.398	0.383	0.407	0.559	0.515
ETTm1	**0.360**	**0.385**	0.367	0.389	0.372	0.389	0.390	0.393	0.377	0.393	0.398	0.405	0.379	0.396	0.393	0.402	0.406	0.391	0.408	0.393	0.407	0.410	0.403	0.407
ETTm2	0.268	0.313	0.268	0.315	**0.263**	**0.311**	0.280	0.324	0.277	0.323	0.288	0.332	0.266	0.314	0.287	0.330	0.287	0.331	0.281	0.320	0.288	0.332	0.350	0.401
Weather	**0.238**	**0.267**	0.239	0.268	0.243	0.270	0.251	0.273	0.242	0.269	0.251	0.276	0.243	0.271	0.255	0.278	0.258	0.277	0.261	0.278	0.258	0.278	0.265	0.317
Electricity	**0.162**	0.259	0.166	0.260	0.168	0.259	0.172	**0.258**	0.164	0.259	0.170	0.265	0.168	0.259	0.171	0.264	0.226	0.304	0.216	0.281	0.178	0.270	0.212	0.300
Traffic	0.462	0.277	0.470	0.280	0.448	**0.261**	0.451	0.269	0.445	0.276	0.414	0.276	0.472	0.301	**0.409**	0.267	0.691	0.427	0.589	0.322	0.428	0.282	0.625	0.383
Solar	**0.194**	0.241	**0.194**	0.245	-	-	0.237	**0.233**	0.198	0.256	0.240	0.273	0.210	0.261	0.229	0.256	-	-	0.303	0.282	0.233	0.262	0.330	0.401
1st Cnt	8		1		4		3		0		0		0		1		0		0		0		0	

Overall, FSC achieves the best results on 6 out of 8 datasets. Its gains are particularly pronounced on datasets with lower Pearson correlations, where the larger mismatch between periodic priors and real-time dynamics calls for fast-to-slow correction. This mechanism is validated by comparing datasets with different degrees of distribution shifts. For instance, on the Electricity dataset (*pearson.corrs* = 0.130), FSC effectively aligns the shifted features, achieving the lowest MSE of 0.162. This yields a relative improvement of $\sim$1.2% over TQNet and up to $\sim$3.6% over FreqCycle and CycleNet. Furthermore, on datasets with even smaller Pearson coefficients such as **[Dataset 2, e.g., Solar]**

(*pearson.corrs* = **[Value, e.g., -0.056]**), the distribution discrepancy becomes more severe. In this highly dynamic scenario, FSC's **fast-to-slow correction** demonstrates an even more pronounced advantage. In summary, FSC significantly enhances robustness against temporal distribution shifts.

Short-term Forecasting. Table 3 presents the results on four short-term forecasting benchmarks, with detailed results provided in Table 11 of the Appendix C.1. Note that FreqCycle, DUET, FITS, and SparseTSF are not included due to the lack of publicly available results or training code. Overall, FSC achieves the best results in 7 out of 8 metrics (Table 3), demonstrating consistent improvements over existing methods. PEMS07, with the highest Pearson correlation, also shows the smallest performance gain.

Table 3: Short-term forecasting results (input length: 96), averaged over $H \in \{12, 24, 48, 96\}$. Best and second-best results are marked in **bold** and underlined, respectively. Lower is better.

model	FSC (Ours)		GTR (2026)		TQNet (2025)		S-Mamba (2025)		CycleNet (2024)		SOFTS (2024)		iTransformer (2024)		DLinear (2023)	
metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	**0.087**	**0.186**	**0.087**	0.189	0.097	0.203	0.122	0.228	0.118	0.226	0.104	0.210	0.113	0.222	0.278	0.375
PEMS04	**0.084**	**0.184**	0.087	0.189	0.091	0.197	0.103	0.211	0.119	0.232	0.102	0.208	0.111	0.221	0.295	0.388
PEMS07	0.076	**0.168**	0.076	0.169	**0.075**	0.171	0.089	0.188	0.113	0.214	0.086	0.184	0.101	0.204	0.329	0.396
PEMS08	**0.103**	**0.191**	0.142	0.222	0.142	0.230	0.148	0.224	0.150	0.246	0.138	0.220	0.150	0.226	0.379	0.416
1st Cnt	7		1		1		0		0		0		0		0	

4.3 Ablation Study

We conduct ablation studies to evaluate the contributions of ACE and FSC-Refiner (Table 4). Removing either component consistently degrades performance across datasets, demonstrating the complementary benefits of adaptive channel encoding and fast-to-slow correction.

Table 4: Ablation analysis (input length: 96). Results are averaged over $H \in \{12, 24, 48, 96\}$ for PEMS and $H \in \{96, 192, 336, 720\}$ for other datasets. Best results are marked in **bold**; full results are in the Appendix C.2.

Dataset	Metric	w/o FSR	w/o RM	w/o orth	w/o ACE	Re-Time	ACE-mean	Re-C1	lag-fast	lag-add	lag-cat	only-S	only-A	**FSC**
ETTh1	MSE	0.427	0.441	0.421	0.426	**0.416**	0.423	0.427	0.422	0.418	0.423	0.485	0.419	0.417
	MAE	0.432	0.439	0.434	0.436	**0.431**	0.435	0.437	0.435	0.433	0.437	0.482	0.434	0.432
Solar	MSE	0.204	0.198	**0.194**	0.195	0.198	**0.194**	0.195	**0.194**	0.195	0.203	0.306	0.195	**0.194**
	MAE	0.249	0.245	**0.241**	0.243	0.245	0.242	0.242	0.244	0.242	0.252	0.380	0.246	**0.241**
Weather	MSE	0.242	0.239	**0.238**	0.240	0.239	0.240	0.239	**0.238**	**0.238**	0.239	0.268	0.239	**0.238**
	MAE	0.269	0.268	**0.267**	0.269	0.269	0.269	0.268	**0.267**	**0.267**	**0.267**	0.291	0.268	**0.267**
PEMS08	MSE	0.132	0.127	**0.103**	0.119	0.105	0.119	0.105	0.119	**0.103**	0.106	0.405	0.115	**0.103**
	MAE	0.223	0.219	0.193	0.205	0.196	0.208	0.195	0.207	0.193	0.197	0.389	0.203	**0.191**

The ablation results show that different variants exhibit dataset-dependent performance. Re-Time achieves the best results on ETTh1, while the proposed FSC remains competitive and performs better on the other datasets. Removing or simplifying individual components generally degrades performance, particularly for w/o FSR and the simplified ACE variants. Overall, the results demonstrate the complementary contributions of ACE and the FSC-Refiner.

FSC Ablation Variants. **w/o FSC** eliminates the FSR-Refiner and sets $h_{\text{fsr}} = h_{\text{fast}} + h_{\text{slow}}$. **w/o ACE** removes the entire CPE module. **w/o RM** bypasses MLP refinement, i.e., $\tilde{h}_{\text{lag}} = h_{\text{lag}}, h_{\text{align}} = \text{Linear}(h_{\text{lag}})$. **w/o orth** removes the orthogonal loss. **Re-Time** utilizes sinusoidal functions to capture sequential dependencies along the temporal dimension. **ACE-mean** averages CPE across channels and broadcasts the shared encoding. **Re-C1** replaces sinusoidal encoding with RoPE (Su et al., 2024). **lag-fast** directly sets $h_{\text{lag}} = h_{\text{fast}}$. **lag-add** replaces subtraction with addition, i.e., $h_{\text{lag}} = h_{\text{fast}} + h_{\text{slow}}$. **lag-cat** derives the lag feature via linear projection of concatenated representations, i.e., $h_{\text{lag}} = \text{Linear}(\text{Cat}(h_{\text{fast}}, h_{\text{slow}}))$. **only-S** sets $h_{\text{fsr}} = h_{\text{slow}}$. **only-A** sets $h_{\text{fsr}} = h_{\text{align}}$.

4.4 Model Analysis

Effectiveness of FSC. To evaluate the effectiveness and generalizability of FSC, we conduct experiments on three benchmark datasets with prominent periodic patterns using three representative

backbones, as shown in Figure 3. On the Electricity dataset, FSC significantly reduces the average MSE by 6.8%, 15.8%, and 20.9% for iTransformer, PatchTST (Nie et al., 2023), and DLinear, respectively, with corresponding average memory overheads of 6.6%, 8.5%, and 19.7%. Meanwhile, similar and even more pronounced accuracy improvements are consistently observed on the other evaluated datasets (e.g., PEMS03 and PEMS04). Overall, FSC provides a general plug-and-play enhancement across different architectures and forecasting settings, boosting predictive accuracy with manageable computational overhead.

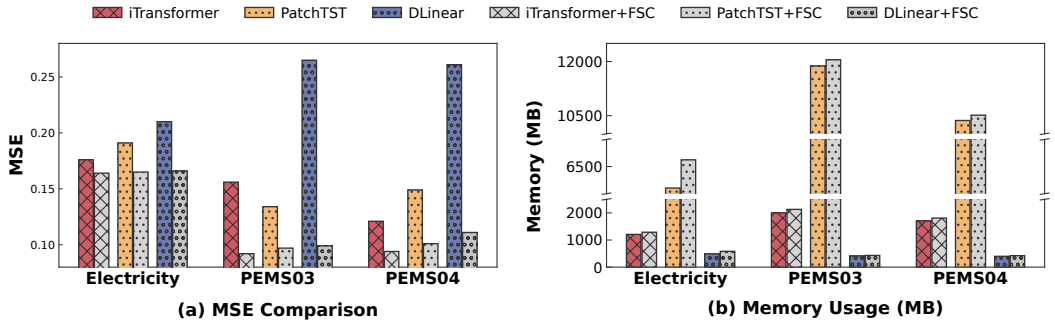

Figure 3: Effectiveness and generalizability of FSC (input length: 96). Results are averaged over $H \in \{96, 192, 336, 720\}$ for Electricity and $H \in \{12, 24, 48, 96\}$ for PEMS. Memory denotes peak training GPU usage.Detailed results are provided in Table 13 of the Appendix C.3.

Robustness Analysis. We evaluate FSC across four random seeds (2024–2027) on ETTh1, Weather, and PEMS04, reporting the mean ± standard deviation of MSE in Figure 4, with detailed results in Table 14 in Appendix C.4. FSC consistently achieves the lowest MSE and the smallest standard deviation across all three datasets, demonstrating both improved forecasting accuracy and stability across random seeds.

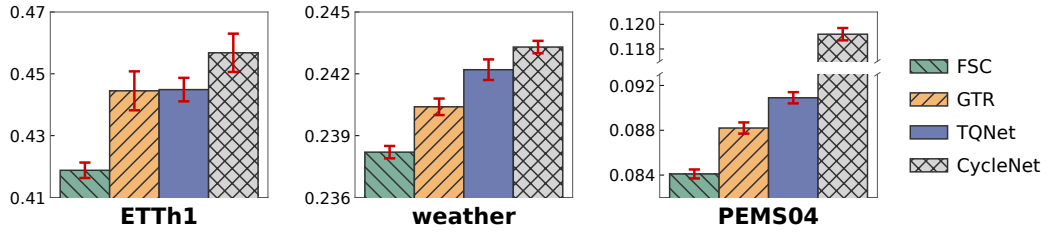

Figure 4: Robustness across random seeds. Results are averaged over four forecasting horizons per seed and reported as mean ± standard deviation across four seeds (2024–2027).

4.5 Mechanism Analysis of FSC

We analyze the mechanism of FSC from two perspectives: prior correction and feature decoupling. First, we examine whether FSC effectively corrects the periodic prior toward the underlying dynamics. As shown in Fig. 5, the overall Pearson correlation structure of the periodic prior π becomes closer to that of the ground truth after passing through FSC. This indicates that FSC effectively corrects the misaligned periodic representation toward the current underlying dynamics.

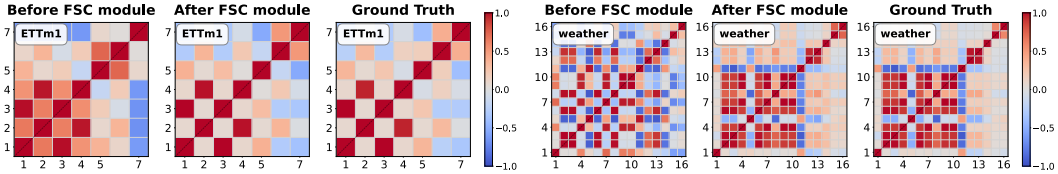

Figure 5: Pearson Correlation analysis of the periodic prior before and after FSC. The overall correlation structure of π becomes closer to that of the ground truth after FSC.

To further understand how FSC achieves this correction, Fig. 6 examines the overall relationships among the representations involved in FSC. Before FSC, the periodic prior π and the current input x exhibit noticeable correlation, with mean absolute Pearson correlations of 0.199 and 0.108 on ETTm1 and Weather, respectively. After the linear projections, the slow and fast representations show substantially weaker correlations, decreasing to 0.062 and 0.026, respectively. This suggests that the projections separate the relatively stable periodic information from rapidly evolving local dynamics. Furthermore, the lag representation exhibits even weaker correlation with the slow representation, with mean absolute Pearson correlations of 0.021 and 0.018. This indicates that the lag representation captures complementary information rather than simply duplicating the slow representation, providing a more isolated correction signal for the fast-to-slow correction.

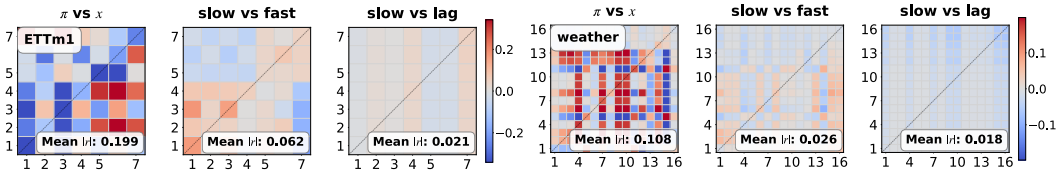

Figure 6: Pearson Correlation analysis of the representations involved in FSC. The linear projections substantially reduce the correlation between the slow and fast representations, while the lag representation exhibits an even weaker correlation with the slow representation. These results indicate that FSC separates periodic and dynamic information and isolates complementary lag information for fast-to-slow correction.

Model Complexity. We compare FSC with representative forecasting methods in terms of parameter count, training time, and forecasting accuracy, as shown in Table 5. FSC achieves the lowest MSE of 0.199 while maintaining a moderate computational cost. Extremely lightweight models such as FITS and SparseTSF require only 13.9K and 4.5K parameters, respectively, with training times of 17.4s and 13.7s per epoch; however, their forecasting accuracy is substantially lower, with MSEs of 0.275 and 0.252. In comparison, among periodic modeling methods, including CycleNet, TQNet, FreqCycle, and GTR, FSC achieves the best forecasting accuracy, reducing MSE to 0.199, although its parameter count (1.26M) and training time (14.8s/epoch) are not the lowest. Notably, FSC still incurs substantially lower training cost than Transformer-based methods such as PatchTST and Autoformer, while achieving better forecasting accuracy. These results demonstrate that FSC provides a favorable balance between forecasting accuracy and computational efficiency.

Table 5: Efficiency of FSC vs. baselines on Electricity (input: 96, H: 720). Training time is averaged per epoch.

Model	Params	Training Time	MSE
Autoformer (2021)	12.2M	107.7s	0.254
DLinear (2023)	139K	18.1s	0.245
PatchTST (2023)	10.7M	129.5s	0.246
iTransformer (2023)	5.15M	35.1s	0.225
FITS (2024)	13.9k	17.4s	0.275
SparseTSF (2024b)	4.5k	13.7s	0.252
CycleNet (2024a)	0.41M	12.0s	0.212
TQNet (2025a)	0.92M	14.4s	0.201
FreqCycle (2026a)	105K	12.5s	0.210
GTR (2026)	0.93M	14.0s	0.208
FSC (ours)	**1.26M**	**14.8s**	**0.199**

5 Conclusion

Distribution shifts can cause historical periodic priors to lag behind rapidly evolving local dynamics in multivariate time series forecasting. We introduce FSC, a backbone-agnostic framework that explicitly models the fast–slow discrepancy and converts it into a correction signal for slow periodic features, enabling adaptive alignment while preserving stable periodic information. Extensive experiments on 12 real-world datasets show that FSC achieves the best performance on 9 datasets, with an average 2.51% MSE reduction over strong periodic forecasting methods and only $\sim$1.26M additional parameters. Further analyses demonstrate that FSC effectively corrects misaligned periodic representations and decouples slow periodic and fast dynamic features, providing a lightweight and general enhancement for existing forecasting models.

AI USE STATEMENT

We used Large Language Models (LLMs) solely as auxiliary tools for language editing, text polishing, and LaTeX formatting. They had no influence on the research design, experiments, analysis, or scientific claims. All research contributions and conclusions were developed and verified independently by the authors.

ETHICS STATEMENT

This work uses only publicly available datasets and does not involve human subjects or personally identifiable information. All datasets were used in accordance with their publicly stated terms of use.

REPRODUCIBILITY STATEMENT

The model architecture, training objectives, evaluation metrics, and datasets are detailed in the paper and supplementary materials. The complete source code will be released upon acceptance.

REFERENCES

Xiuding Cai, Yaoyao Zhu, Xueyao Wang, and Yu Yao. Mambats: Improved selective state space models for long-term time series forecasting. *arXiv preprint arXiv:2405.16440*, 2024.

Fanpu Cao, Lu Dai, Jindong Han, and Hui Xiong. Enhancing multivariate time series forecasting with global temporal retrieval. *arXiv preprint arXiv:2602.10847*, 2026.

Hui Chen, Zhuo Wang, Fangliang Yang, Xiangyang Wang, Chunjie Chen, and Xinyu Wu. Enhancing terrain recognition with a transformer-based model: Integrating imus for motion intent detection. *IEEE Transactions on Human-Machine Systems*, 56(1):22–31, 2026. doi: 10.1109/THMS.2025.3631855.

Yu Chengqing, Yan Guangxi, Yu Chengming, Zhang Yu, and Mi Xiwei. A multi-factor driven spatiotemporal wind power prediction model based on ensemble deep graph attention reinforcement learning networks. *Energy*, 263:126034, 2023.

Razvan-Gabriel Cirstea, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting–full version. *arXiv preprint arXiv:2204.13767*, 2022.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Xue Yuerong, Shu-Tao Xia, and Zexuan Zhu. Ddn: Dual-domain dynamic normalization for non-stationary time series forecasting. *Advances in Neural Information Processing Systems*, 37:108490–108517, 2024.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 402–411, 2021.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems*, 37: 64145–64175, 2024.

Qihe Huang, Zhengyang Zhou, Yangze Li, Kuo Yang, Binwu Wang, and Yang Wang. Many minds, one goal: Time series forecasting via sub-task specialization and inter-agent cooperation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Anwar Khan, Mostafa M Fouda, Dinh-Thuan Do, Abdulaziz Almaleh, and Atiq Ur Rahman. Short-term traffic prediction using deep learning long short-term memory: Taxonomy, applications, challenges, and future trends. *IEEE access*, 11:94371–94391, 2023.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37:106315–106345, 2024a.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*, 2024b.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. *arXiv preprint arXiv:2505.12917*, 2025a.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *IEEE Internet of Things Journal*, 2025b.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in neural information processing systems*, 35:5816–5828, 2022.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-tao Xia. Timebridge: Non-stationarity matters for long-term time series forecasting. *arXiv preprint arXiv:2410.04442*, 2024.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *ICLR*, 2024.

Xiaowen Ma, Zhenliang Ni, Shuai Xiao, and Xinghao Chen. Timepro: Efficient multivariate long-term time series forecasting with variable-and time-aware hyper-state. *arXiv preprint arXiv:2505.20774*, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Deep adaptive input normalization for time series forecasting. *IEEE transactions on neural networks and learning systems*, 31(9):3760–3765, 2019.

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. Dbloss: Decomposition-based loss function for time series forecasting. *arXiv preprint arXiv:2510.23672*, 2025a.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1185–1196, 2025b.

Said E Said and David A Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 1984.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yuting Sun, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. Tinyad: Memory-efficient anomaly detection for time-series data in industrial iot. *IEEE Transactions on Industrial Informatics*, 20(1):824–834, 2023.

Shiyu Wang, Zhixuan Chu, Yinbo Sun, Yu Liu, Yuliang Guo, Yang Chen, Huiyang Jian, Lintao Ma, Xingyu Lu, and Jun Zhou. Multiscale representation enhanced temporal flow fusion model for long-term workload forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 4948–4956, 2024a.

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. *arXiv preprint arXiv:2405.14616*, 2024b.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024c.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, volume 2024, pp. 26295–26318, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Boya Zhang, Shuaijie Yin, Huiwen Zhu, and Xing He. Freqcycle: A multi-scale time-frequency analysis method for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28159–28167, 2026a.

Zhiwei Zhang, Xinyi Du, Xuanchi Guo, Weihao Wang, and Wenjuan Han. Emaformer: Enhancing transformer through embedding armor for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28636–28644, 2026b.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022a.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022b.

A Implementation Details

A.1 Dataset Details

We conduct experiments on 12 widely used time series datasets, including benchmarks for long-term forecasting and PEMS datasets for short-term forecasting. Dataset statistics are summarized in Table 6. Detailed descriptions are provided below.

- **ETT (Electricity Transformer Temperature)** (Zhou et al., 2021) records oil temperature and load data from electricity transformers in two regions of China, spanning from 2016 to 2018. It includes two temporal resolutions: hourly-level (ETTh1, ETTh2) and 15-minute-level (ETTm1, ETTm2).
- **Weather** (Wu et al., 2022) contains 21 meteorological indicators collected in Germany throughout 2020 at a 10-minute interval, including key variables such as temperature and visibility.
- **Electricity** (Wu et al., 2022) consists of hourly electricity consumption data (in kWh) from 321 clients, spanning from 2012 to 2014.
- **Traffic** (Wu et al., 2022) includes hourly road occupancy rates collected from 862 sensors on highways in the San Francisco Bay Area between 2015 and 2016.
- **Solar-Energy** (Lin et al., 2024a) contains solar power generation data from 137 photovoltaic plants in Alabama, USA, recorded at 10-minute intervals throughout 2006.
- **PEMS** (Liu et al., 2022) comprises four public traffic datasets (PEMS03, PEMS04, PEMS07, PEMS08) collected from the Caltrans Performance Measurement System. The data are aggregated every 5 minutes, resulting in 12 observations per hour and 288 per day.

Table 6: Detailed statistics of the datasets. Channels denote the number of variables, and Period denotes the length of a complete cycle. Data Split indicates the ratio of training/validation/test sets.

Task	Prediction Length	Dataset	Channels	Period	Data Split	Frequency
Long-term Forecasting	{96, 192, 336, 720}	ETTh1, ETTh2	7	24	6:2:2	15 min
		ETTm1, ETTm2	7	96	6:2:2	15 min
		Weather	21	144	7:1:2	10 min
		Electricity	321	168	7:1:2	1 hour
		Traffic	862	168	7:1:2	1 hour
		Solar-Energy	137	144	7:1:2	10 min
Short-term Forecasting	{12, 24, 48, 96}	PEMS03	358	288	3:1:1	5 min
		PEMS04	307	288	3:1:1	5 min
		PEMS07	883	288	3:1:1	5 min
		PEMS08	170	288	3:1:1	5 min

A.2 Evaluation Metrics

To evaluate the performance of our model on both long-term and short-term forecasting tasks, we employ two standard metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). These metrics are defined as follows:

$$\text{MSE} = \frac{1}{H \times C} \sum_{i=1}^{H} \sum_{j=1}^{C} (Y_{i,j} - \hat{Y}_{i,j})^2, \quad \text{MAE} = \frac{1}{H \times C} \sum_{i=1}^{H} \sum_{j=1}^{C} |Y_{i,j} - \hat{Y}_{i,j}| \tag{9}$$

where $Y_{i,j}$ and $\hat{Y}_{i,j}$ denote the ground truth and the predicted values for the j-th channel at the i-th time step, respectively. Here, H represents the number of future time steps (forecasting horizon) and C denotes the number of channels (variables). For both metrics, a lower value indicates superior forecasting performance.

A.3 Algorithm Details

The detailed procedure of the proposed FSR framework is summarized in Algorithm 1.

Algorithm 1: Fast-to-Slow Correction (FSC)

Require: $\mathbf{x} \in \mathbb{R}^{C \times L}$, period length P, start phase t
Ensure: prediction $\hat{\mathbf{Y}} \in \mathbb{R}^{C \times H}$, orthogonal loss $\mathcal{L}_{\text{ortho}}$

Initialize learnable periodic embeddings $\mathbf{Q} \in \mathbb{R}^{C \times P}$
Initialize learnable frequency scaling vector $\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_{L/2-1}]$
// Construct global periodic prior $\boldsymbol{\pi}$
for $\tau = 0$ to $L - 1$ **do**
  $i_\tau \leftarrow (t + \tau) \bmod P$
end for
$\mathbf{I} \leftarrow [i_0, i_1, \ldots, i_{L-1}]$
$\boldsymbol{\pi} \leftarrow \text{Lookup}(\mathbf{Q}, \mathbf{I}) \in \mathbb{R}^{C \times L}$
// Construct Adaptive Channel Encoding (ACE)
for $k = 0$ to $\frac{L}{2} - 1$ **do**
  $\omega_k \leftarrow \exp\left(-\frac{2k \ln(10000)}{L}\right)$
  $\tilde{\omega}_k \leftarrow \alpha_k \cdot \omega_k$
  for $v = 0$ to $C - 1$ **do**
    $\mathbf{ACE}[v, 2k] \leftarrow \sin(v \cdot \tilde{\omega}_k)$
    $\mathbf{ACE}[v, 2k+1] \leftarrow \cos(v \cdot \tilde{\omega}_k)$
  end for
end for
$\mathbf{ACE} \in \mathbb{R}^{C \times L}$
// Fast-Slow Decoupling
$\mathbf{h}_{\text{slow}} \leftarrow \text{Linear}_{\text{slow}}(\boldsymbol{\pi}) \in \mathbb{R}^{C \times D}$
$\mathbf{h}_{\text{fast}} \leftarrow \text{Linear}_{\text{fast}}(\mathbf{x} + \mathbf{ACE}) \in \mathbb{R}^{C \times D}$
// FSC-Refiner: Fast-to-Slow Correction
$\mathbf{h}_{\text{lag}} \leftarrow \mathbf{h}_{\text{fast}} - \mathbf{h}_{\text{slow}} \in \mathbb{R}^{C \times D}$
$\tilde{\mathbf{h}}_{\text{lag}} \leftarrow \text{MLP}(\mathbf{h}_{\text{lag}}) + \mathbf{h}_{\text{lag}} \in \mathbb{R}^{C \times D}$
$\mathbf{h}_{\text{align}} \leftarrow \text{Linear}(\tilde{\mathbf{h}}_{\text{lag}}) \in \mathbb{R}^{C \times D}$
$\mathbf{h}_{\text{fsc}} \leftarrow \mathbf{h}_{\text{slow}} + \mathbf{h}_{\text{align}} \in \mathbb{R}^{C \times D}$
// Output projection & Loss Function
$\hat{\mathbf{Y}} \leftarrow \text{Linear}(\mathbf{h}_{\text{fsc}}) \in \mathbb{R}^{C \times H}$
$\mathcal{L}_{\text{forecast}} \leftarrow \text{MSE}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{C \times H} \|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2$
$\mathcal{L}_{\text{ortho}} \leftarrow \frac{1}{C} \sum_{c=1}^{C} \left(\mathbf{h}_{\text{slow}}^{(c)} \cdot \mathbf{h}_{\text{align}}^{(c)}\right)^2$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{forecast}} + \lambda \mathcal{L}_{\text{ortho}}$
return $\hat{\mathbf{Y}}, \mathcal{L}$

A.4 Pearson Correlation

The *pearson.corrs* metric quantifies the alignment between historical periodic priors and real-time dynamics. To accurately reflect the local distribution shifts encountered during actual inference, the calculation is performed at the sliding-window level. Specifically, for each testing window m, the multivariate real-time input matrix and the corresponding periodic prior matrix ($\in \mathbb{R}^{C \times L}$, where C is the number of variables and L is the sequence length) are flattened into 1D vectors, denoted as $\mathbf{x}^{(m)}$ and $\mathbf{p}^{(m)}$ respectively, of length $N = C \times L$. The Pearson correlation coefficient $r^{(m)}$ is computed independently for each window as:

$$r^{(m)} = \frac{\sum_{i=1}^{N}(x_i^{(m)} - \bar{x}^{(m)})(p_i^{(m)} - \bar{p}^{(m)})}{\sqrt{\sum_{i=1}^{N}(x_i^{(m)} - \bar{x}^{(m)})^2}\sqrt{\sum_{i=1}^{N}(p_i^{(m)} - \bar{p}^{(m)})^2}}$$

where $\bar{x}^{(m)}$ and $\bar{p}^{(m)}$ denote the means of vectors $\mathbf{x}^{(m)}$ and $\mathbf{p}^{(m)}$. The final reported metric is the arithmetic mean of these coefficients across all M sliding windows in the test set, calculated as pearson.corrs $= 1/M \sum_{m=1}^{M} r^{(m)}$.

Table 7: Pearson Correlation Results are averaged over $H \in \{12, 24, 48, 96\}$ for PEMS and $H \in \{96, 192, 336, 720\}$ for other datasets.

Horizon	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Solar	Traffic	PEMS03	PEMS04	PEMS07	PEMS08
h0	0.084	0.045	-0.251	-0.111	0.076	0.317	0.042	0.255	0.083	0.105	0.021	-0.162
h1	-0.241	0.216	0.124	0.400	0.285	-0.342	-0.149	-0.111	-0.258	-0.236	0.087	-0.016
h2	0.080	0.027	0.244	0.273	0.247	0.142	-0.192	-0.093	0.253	0.143	-0.125	-0.038
h3	0.435	0.002	0.348	0.318	0.0	0.404	0.076	0.270	-0.332	-0.318	0.241	0.320
avg	0.090	0.073	0.116	0.220	0.152	0.130	-0.056	0.080	-0.064	-0.077	0.056	0.026

A.5 Hyperparameter Settings for FSC

Our FSC framework involves two key hyperparameters: the scaling factor α in the ACE module and the balancing weight λ in the loss function. We summarize their initialization settings across different datasets in Table 8. The results show that both α and λ vary across datasets, indicating that their optimal settings are dataset-dependent and require tailored configurations for the best forecasting performance.

Table 8: Key hyperparameters (α for the ACE module and λ for the loss function) across datasets.

Dataset	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Traffic	Solar	PEMS03	PEMS04	PEMS07	PEMS08
α (ACE module)	0.8	1.0	0.8	0.0	1.0	0.0	1.0	0.5	0.0	1.0	0.0	0.0
λ (Loss function)	0.001	0.001	0.001	0.0005	0.01	0.0001	0.0001	0.0001	0.001	0.001	0.001	0.001

B Adaptive Channel Encoding (ACE)

To provide explicit structural information for inter-channel modeling, we introduce Adaptive Channel Encoding (ACE), motivated by the benefits of explicitly modeling channel order (Wang et al., 2025). ACE assigns a sinusoidal representation to each channel and adaptively modulates its frequency components through learnable scaling factors.

B.1 Analysis of ACE Injection Positions

To investigate the effect of ACE placement, we construct several variants by injecting ACE at different stages of the fast-slow decoupling framework. All variants keep other components unchanged and only modify the injection position, as shown in Table 9. We denote $\mathbf{ACE}$ as the adaptive channel encoding generated following the formulation in Section 3.2. Its output dimension is determined by the representation at the injection stage, e.g., $\mathbf{ACE} \in \mathbb{R}^{C \times L}$ for the input space and $\mathbf{ACE} \in \mathbb{R}^{C \times D}$ for latent representations. Specifically, the original fast and slow representations are defined as $\mathbf{h}_{\text{fast}} = \text{Linear}_{\text{fast}}(\mathbf{x})$ and $\mathbf{h}_{\text{slow}} = \text{Linear}_{\text{slow}}(\boldsymbol{\pi})$, respectively. The injection positions are defined as follows. **Fast_latent** injects ACE into the latent representation of the fast branch, defined as $\mathbf{h}_{\text{fast}} = \mathbf{h}_{\text{fast}} + \mathbf{ACE}$. **Refined_lag** injects ACE after lag feature refinement, formulated as $\tilde{\mathbf{h}}_{\text{lag}} = \mathbf{h}_{\text{lag}} + \text{MLP}(\mathbf{h}_{\text{lag}}) + \mathbf{ACE}$. **Align_latent**

Table 9: Comparison of CPE Injection Positions. Input length is 96. Best results are in **bold**.

Model		Fast_lat		Ref_lag		Align_lat		Slow_feat		FSC	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.370	0.398	0.372	0.399	0.364	0.396	0.369	0.398	0.365	0.395
	192	0.410	0.424	0.412	0.425	0.411	0.425	0.409	0.422	0.407	0.423
	336	0.454	0.444	0.457	0.445	0.456	0.448	0.447	0.444	0.445	0.443
	720	0.447	0.462	0.452	0.468	0.476	0.475	0.462	0.475	0.453	0.469
	avg	0.420	0.432	0.423	0.434	0.427	0.436	0.422	0.435	**0.417**	**0.432**
Solar	96	0.176	0.232	0.176	0.232	0.176	0.231	0.174	0.230	0.174	0.229
	192	0.195	0.247	0.194	0.247	0.192	0.246	0.193	0.246	0.193	0.245
	336	0.200	0.244	0.201	0.245	0.200	0.245	0.208	0.256	0.200	0.244
	720	0.210	0.248	0.209	0.248	0.211	0.248	0.209	0.247	0.210	0.248
	avg	0.195	0.243	0.195	0.243	0.195	0.242	0.196	0.245	**0.194**	**0.241**
weather	96	0.152	0.198	0.154	0.201	0.155	0.201	0.153	0.199	0.152	0.198
	192	0.201	0.243	0.202	0.244	0.203	0.245	0.202	0.244	0.202	0.244
	336	0.258	0.287	0.259	0.287	0.260	0.288	0.258	0.286	0.258	0.286
	720	0.340	0.341	0.340	0.341	0.341	0.341	0.340	0.341	0.339	0.340
	avg	**0.238**	**0.267**	0.239	0.268	0.240	0.269	0.238	0.268	**0.238**	**0.267**
PEMS08	12	0.067	0.163	0.074	0.180	0.066	0.164	0.066	0.163	0.066	0.164
	24	0.085	0.181	0.091	0.193	0.087	0.189	0.090	0.193	0.082	0.177
	48	0.114	0.205	0.116	0.207	0.110	0.200	0.114	0.207	0.112	0.200
	96	0.153	0.231	0.155	0.233	0.152	0.225	0.157	0.228	0.152	0.222
	avg	0.105	0.195	0.109	0.203	0.104	0.195	0.107	0.198	**0.103**	**0.191**

injects ACE into the final alignment signal, computed as $\mathbf{h}_{\text{align}} = \text{Linear}_{\text{fs}}\big(\text{Dropout}(\tilde{\mathbf{h}}_{\text{lag}})\big) + \mathbf{ACE}$. **Slow_feat** injects ACE into the slow branch, expressed as $\mathbf{h}_{\text{slow}} = \text{Linear}_{\text{slow}}(\boldsymbol{\pi} + \mathbf{ACE})$.

B.2 Hyperparameter Sensitivity Analysis

To evaluate the sensitivity of ACE to the initialization of the learnable scaling factor α, we conduct experiments with α initialized to $\{0.0, 0.5, 0.8, 1.0\}$, while keeping all other settings unchanged. The full results are reported in Table 10.

The results show that ACE achieves stable performance under different initialization values of α, demonstrating its robustness to initialization. Different datasets exhibit different preferences for the initial scaling strength, where larger or smaller initialization values may be beneficial depending on their temporal characteristics. These results indicate that the learnable scaling factor can be effectively adapted during training, rather than relying on a fixed initialization.

Table 10: Sensitivity analysis of the initialization scale of the learnable frequency scaling factors $\boldsymbol{\alpha}$. The input length is fixed to 96. The best average results are highlighted in bold.

Initialization Scale of $\boldsymbol{\alpha}$		0.0		0.5		0.8		1.0		without α	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.366	0.396	0.366	0.396	0.365	0.395	0.365	0.395	0.365	0.395
	192	0.408	0.423	0.408	0.423	0.407	0.423	0.408	0.423	0.408	0.423
	336	0.449	0.443	0.446	0.443	0.445	0.443	0.446	0.443	0.446	0.443
	720	0.469	0.480	0.455	0.471	0.453	0.469	0.454	0.470	0.451	0.469
	avg	0.423	0.435	0.419	0.433	**0.417**	**0.432**	0.418	0.433	**0.417**	**0.432**
Solar	96	0.174	0.229	0.174	0.229	0.175	0.229	0.175	0.229	0.175	0.229
	192	0.192	0.247	0.193	0.245	0.193	0.245	0.194	0.246	0.194	0.245
	336	0.200	0.244	0.200	0.244	0.201	0.244	0.201	0.244	0.201	0.244
	720	0.209	0.248	0.210	0.248	0.210	0.248	0.210	0.248	0.214	0.257
	avg	**0.194**	0.242	**0.194**	**0.241**	0.195	0.242	0.195	0.242	0.196	0.244
weather	96	0.154	0.200	0.153	0.199	0.152	0.198	0.152	0.198	0.152	0.198
	192	0.203	0.245	0.202	0.244	0.202	0.243	0.202	0.244	0.202	0.244
	336	0.262	0.289	0.259	0.287	0.259	0.286	0.258	0.286	0.258	0.286
	720	0.341	0.341	0.339	0.340	0.340	0.341	0.339	0.340	0.340	0.340
	avg	0.240	0.269	0.238	0.268	**0.238**	**0.267**	**0.238**	**0.267**	**0.238**	**0.267**
PEMS08	12	0.066	0.164	0.066	0.162	0.065	0.163	0.066	0.162	0.066	0.162
	24	0.082	0.177	0.091	0.194	0.087	0.189	0.090	0.189	0.086	0.190
	48	0.112	0.200	0.111	0.196	0.111	0.197	0.111	0.198	0.119	0.213
	96	0.152	0.222	0.149	0.222	0.149	0.221	0.145	0.218	0.147	0.220
	avg	**0.103**	**0.191**	0.104	0.193	**0.103**	0.193	**0.103**	0.192	0.104	0.196

C More Experimental Details

C.1 Comparison of Full Results

Table 11 reports the full multivariate forecasting results on 12 benchmark datasets across all prediction horizons. FSC achieves 28 best results on long-term forecasting tasks and 22 best results on short-term forecasting tasks, demonstrating consistent performance across datasets and temporal scales. Note that FreqCycle reports results on only 7 datasets, while DUET reports results on only 8 datasets; thus, comparisons involving these two methods are conducted within their respective subsets.

Table 11: Full results on 12 datasets with input length 96. *Avg* denotes the average over four forecasting horizons: 96, 192, 336, 720 for long-term and 12, 24, 48, 96 for short-term forecasting. Best and second-best results are shown in **bold** and underlined, respectively. Lower is better.

Model		FSC (Ours)		GTR (2026)		FreqCycle (2026)		DUET (2025)		TQNet (2025)		S-Mamba (2025)		CycleNet (2024)		SOFTS (2024)		FITS (2024)		SparseTSF (2024)		iTransformer (2024)		DLinear (2023)	
Metric		MSE↓	MAE↓	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.365**	0.395	0.367	0.391	0.369	0.390	0.377	0.393	0.371	0.393	0.386	0.405	0.375	0.395	0.381	0.399	0.412	0.416	0.391	**0.388**	0.386	0.405	0.386	0.400
	192	**0.407**	0.423	0.420	0.421	0.417	**0.418**	0.429	0.425	0.428	0.426	0.443	0.437	0.436	0.428	0.435	0.431	0.493	0.463	0.440	**0.418**	0.441	0.436	0.437	0.432
	336	**0.445**	0.443	0.476	0.447	0.459	0.439	0.471	0.446	0.476	0.446	0.489	0.468	0.496	0.455	0.480	0.452	0.493	0.463	0.482	**0.438**	0.487	0.458	0.481	0.459
	720	**0.453**	0.469	0.493	0.476	0.456	**0.454**	0.496	0.480	0.487	0.470	0.502	0.489	0.520	0.484	0.499	0.488	0.534	0.513	0.489	0.460	0.503	0.491	0.519	0.516
	Avg	**0.417**	0.432	0.439	0.434	0.425	**0.425**	0.443	0.436	0.441	0.434	0.455	0.450	0.457	0.441	0.449	0.443	0.483	0.463	0.450	0.426	0.454	0.448	0.456	0.452
ETTh2	96	0.283	**0.335**	0.290	0.342	**0.282**	**0.336**	0.296	0.345	0.295	0.343	0.296	0.348	0.298	0.344	0.297	0.347	0.298	0.345	0.306	0.343	0.297	0.349	0.333	0.387
	192	**0.358**	**0.388**	0.362	0.389	0.361	**0.388**	0.368	0.389	0.367	0.393	0.376	0.396	0.372	0.396	0.373	0.394	0.388	0.400	0.383	0.390	0.380	0.400	0.477	0.476
	336	**0.406**	0.423	0.414	0.429	0.415	0.428	0.411	**0.422**	0.417	0.427	0.424	0.431	0.431	0.439	0.410	0.426	0.423	0.430	0.425	0.424	0.428	0.432	0.594	0.541
	720	0.426	0.450	0.423	0.442	0.425	0.443	0.412	0.434	0.433	0.446	0.426	0.444	0.450	0.458	**0.411**	**0.433**	0.432	0.447	0.426	0.437	0.427	0.445	0.831	0.657
	Avg	**0.368**	0.399	0.372	0.400	0.371	0.399	0.372	**0.397**	0.378	0.402	0.381	0.405	0.388	0.409	0.373	0.400	0.385	0.405	0.385	0.398	0.383	0.407	0.559	0.515
ETTm1	96	**0.304**	**0.349**	0.305	**0.349**	0.310	0.353	0.324	0.354	0.311	0.353	0.333	0.368	0.319	0.360	0.325	0.361	0.338	0.354	0.339	0.356	0.334	0.368	0.345	0.372
	192	**0.343**	**0.371**	0.349	0.375	0.351	0.374	0.369	0.379	0.356	0.378	0.376	0.390	0.360	0.381	0.375	0.389	0.385	0.377	0.387	0.379	0.377	0.391	0.380	0.389
	336	**0.370**	**0.394**	0.380	0.398	0.383	0.396	0.404	0.402	0.390	0.401	0.408	0.413	0.389	0.403	0.405	0.412	0.418	0.398	0.418	0.400	0.426	0.420	0.413	0.413
	720	**0.424**	**0.428**	0.435	0.436	0.446	0.434	0.463	0.437	0.452	0.440	0.475	0.448	0.447	0.441	0.466	0.447	0.485	0.436	0.488	0.437	0.491	0.459	0.474	0.453
	Avg	**0.359**	**0.385**	0.367	0.389	0.372	0.389	0.390	0.393	0.377	0.393	0.398	0.405	0.379	0.396	0.393	0.402	0.406	0.391	0.408	0.393	0.407	0.410	0.403	0.407
ETTm2	96	0.166	0.247	0.168	0.249	**0.162**	**0.244**	0.174	0.255	0.173	0.256	0.179	0.263	0.163	0.246	0.180	0.261	0.183	0.259	0.180	0.258	0.180	0.264	0.193	0.292
	192	0.230	0.288	0.232	0.292	**0.226**	**0.286**	0.243	0.302	0.238	0.298	0.250	0.309	0.229	0.290	0.246	0.306	0.248	0.300	0.244	0.298	0.250	0.309	0.284	0.362
	336	0.288	0.328	0.287	0.330	**0.283**	**0.326**	0.304	0.341	0.301	0.340	0.312	0.349	0.284	0.327	0.319	0.352	0.308	0.350	0.303	0.335	0.311	0.348	0.369	0.427
	720	0.390	0.390	0.386	0.390	**0.383**	**0.387**	0.399	0.397	0.397	0.396	0.411	0.406	0.389	0.391	0.405	0.401	0.409	0.417	0.400	0.392	0.412	0.407	0.554	0.522
	Avg	0.268	0.313	0.268	0.315	**0.263**	**0.311**	0.280	0.324	0.277	0.323	0.288	0.332	0.266	0.314	0.287	0.330	0.287	0.331	0.281	0.320	0.288	0.332	0.350	0.401
Weather	96	**0.152**	**0.198**	0.154	0.200	0.159	0.203	0.163	0.202	0.157	0.200	0.165	0.210	0.158	0.203	0.166	0.208	0.173	0.213	0.182	0.217	0.174	0.214	0.196	0.255
	192	**0.202**	**0.244**	**0.202**	**0.244**	0.208	0.248	0.218	0.252	0.206	0.245	0.214	0.252	0.207	0.247	0.217	0.253	0.221	0.255	0.227	0.256	0.221	0.254	0.237	0.296
	336	**0.258**	**0.286**	0.259	0.287	0.264	0.289	0.274	0.294	0.262	0.287	0.274	0.297	0.262	0.289	0.282	0.300	0.279	0.295	0.281	0.295	0.278	0.296	0.283	0.335
	720	**0.339**	**0.340**	0.341	0.342	0.343	0.342	0.349	0.343	0.344	0.342	0.350	0.345	0.344	0.344	0.356	0.351	0.359	0.345	0.356	0.344	0.358	0.349	0.345	0.381
	Avg	**0.238**	**0.267**	0.239	0.268	0.243	0.270	0.251	0.273	0.242	0.269	0.251	0.276	0.243	0.271	0.255	0.278	0.258	0.277	0.261	0.278	0.258	0.278	0.265	0.317
Electricity	96	**0.131**	**0.228**	0.134	0.229	0.140	0.234	0.145	0.233	0.134	0.229	0.139	0.235	0.136	0.229	0.143	0.233	0.206	0.281	0.201	0.263	0.148	0.240	0.197	0.282
	192	**0.150**	0.247	0.152	0.245	0.154	0.246	0.163	0.248	0.154	0.247	0.159	0.255	0.152	**0.244**	0.158	0.248	0.203	0.281	0.199	0.266	0.162	0.253	0.196	0.285
	336	**0.167**	0.265	0.171	0.264	0.170	0.264	0.175	**0.262**	0.169	0.264	0.176	0.272	0.170	0.264	0.178	0.269	0.220	0.302	0.212	0.281	0.178	0.269	0.209	0.301
	720	**0.199**	0.297	0.208	0.300	0.210	0.297	0.204	**0.291**	0.201	0.294	0.204	0.298	0.212	0.299	0.218	0.305	0.275	0.354	0.252	0.314	0.225	0.317	0.245	0.333
	Avg	**0.162**	0.259	0.166	0.260	0.168	0.260	0.172	**0.258**	0.164	0.259	0.170	0.265	0.168	0.259	0.174	0.264	0.226	0.304	0.216	0.281	0.178	0.270	0.212	0.300
Traffic	96	0.436	0.262	0.440	0.263	0.438	0.262	0.407	0.252	0.413	0.261	0.382	0.261	0.458	0.296	**0.376**	**0.251**	0.674	0.397	0.597	0.329	0.395	0.268	0.650	0.396
	192	0.443	0.272	0.454	0.274	0.434	**0.258**	0.431	0.262	0.432	0.271	**0.396**	0.267	0.457	0.294	0.398	0.261	0.626	0.389	0.567	0.312.	0.417	0.276	0.598	0.370
	336	0.462	0.277	0.472	0.282	0.446	**0.258**	0.456	0.269	0.450	0.277	0.417	0.276	0.470	0.299	**0.415**	0.269	0.679	0.436	0.577	0.315	0.433	0.283	0.605	0.373
	720	0.508	0.296	0.514	0.301	0.476	**0.265**	0.509	0.292	0.486	0.295	0.460	0.300	0.502	0.314	**0.447**	0.287	0.788	0.496	0.615	0.335	0.467	0.302	0.645	0.394
	Avg	0.462	0.277	0.470	0.280	0.448	**0.261**	0.451	0.269	0.445	0.276	0.414	0.276	0.472	0.301	**0.409**	0.267	0.691	0.427	0.589	0.322	0.428	0.282	0.625	0.383
Solar	96	0.174	0.229	0.176	0.235	-	-	0.200	**0.207**	**0.173**	0.233	0.205	0.244	0.190	0.247	0.200	0.230	-	-	0.267	0.264	0.203	0.237	0.290	0.378
	192	**0.193**	0.245	**0.193**	0.244	-	-	0.228	**0.233**	0.199	0.257	0.237	0.270	0.210	0.266	0.229	0.253	-	-	0.301	0.283	0.233	0.261	0.320	0.398
	336	**0.200**	**0.244**	0.201	0.250	-	-	0.262	**0.244**	0.211	0.263	0.258	0.288	0.217	0.266	0.243	0.269	-	-	0.326	0.295	0.248	0.273	0.353	0.415
	720	0.210	**0.248**	**0.205**	0.251	-	-	0.258	0.249	0.209	0.270	0.260	0.288	0.223	0.266	0.245	0.272	-	-	0.320	0.289	0.249	0.275	0.356	0.413
	Avg	**0.194**	0.241	**0.194**	0.245	-	-	0.237	**0.233**	0.198	0.256	0.240	0.273	0.210	0.261	0.229	0.256	-	-	0.303	0.282	0.233	0.262	0.330	0.401
PEMS03	12	**0.057**	**0.155**	**0.057**	0.156	-	-	-	-	0.060	0.161	0.065	0.169	0.066	0.172	0.064	0.165	-	-	-	-	0.071	0.174	0.122	0.243
	24	**0.070**	**0.171**	**0.070**	0.172	-	-	-	-	0.077	0.182	0.087	0.196	0.089	0.201	0.083	0.188	-	-	-	-	0.093	0.201	0.201	0.317
	48	**0.094**	**0.195**	**0.094**	0.198	-	-	-	-	0.104	0.215	0.133	0.243	0.136	0.247	0.114	0.223	-	-	-	-	0.125	0.236	0.333	0.425
	96	0.128	**0.223**	**0.127**	0.228	-	-	-	-	0.148	0.253	0.201	0.305	0.182	0.282	0.156	0.264	-	-	-	-	0.164	0.275	0.457	0.515
	Avg	**0.087**	**0.186**	**0.087**	0.189	-	-	-	-	0.097	0.203	0.122	0.228	0.118	0.226	0.104	0.210	-	-	-	-	0.113	0.222	0.278	0.375
PEMS04	12	**0.065**	**0.162**	**0.065**	0.164	-	-	-	-	0.067	0.166	0.076	0.180	0.078	0.186	0.074	0.176	-	-	-	-	0.078	0.183	0.148	0.272
	24	**0.074**	**0.175**	0.075	0.177	-	-	-	-	0.077	0.181	0.084	0.193	0.099	0.212	0.088	0.194	-	-	-	-	0.095	0.205	0.224	0.340
	48	**0.089**	**0.191**	0.093	0.197	-	-	-	-	0.097	0.206	0.115	0.224	0.133	0.248	0.110	0.219	-	-	-	-	0.120	0.233	0.355	0.437
	96	**0.108**	**0.210**	0.114	0.219	-	-	-	-	0.123	0.233	0.137	0.248	0.167	0.281	0.135	0.244	-	-	-	-	0.150	0.262	0.452	0.504
	Avg	**0.084**	**0.184**	0.087	0.189	-	-	-	-	0.091	0.197	0.103	0.211	0.119	0.232	0.102	0.208	-	-	-	-	0.111	0.221	0.295	0.388
PEMS07	12	0.052	**0.142**	**0.051**	**0.142**	-	-	-	-	**0.051**	0.143	0.063	0.159	0.062	0.162	0.057	0.152	-	-	-	-	0.067	0.165	0.115	0.242
	24	**0.063**	**0.155**	**0.063**	0.156	-	-	-	-	**0.063**	0.159	0.081	0.183	0.086	0.192	0.073	0.173	-	-	-	-	0.088	0.190	0.210	0.329
	48	**0.081**	**0.175**	0.082	0.178	-	-	-	-	**0.081**	0.179	0.093	0.192	0.128	0.234	0.096	0.195	-	-	-	-	0.110	0.215	0.398	0.458
	96	0.107	**0.200**	0.110	0.202	-	-	-	-	**0.103**	0.203	0.117	0.217	0.176	0.268	0.120	0.218	-	-	-	-	0.139	0.245	0.594	0.553
	Avg	0.076	**0.168**	0.076	0.169	-	-	-	-	**0.075**	0.171	0.089	0.188	0.113	0.214	0.086	0.184	-	-	-	-	0.101	0.204	0.329	0.396
PEMS08	12	**0.066**	**0.164**	0.071	0.168	-	-	-	-	0.071	0.170	0.076	0.178	0.082	0.185	0.074	0.171	-	-	-	-	0.079	0.182	0.154	0.276
	24	**0.082**	**0.177**	0.095	0.192	-	-	-	-	0.096	0.196	0.104	0.209	0.117	0.226	0.104	0.201	-	-	-	-	0.115	0.219	0.248	0.353
	48	**0.112**	**0.200**	0.148	0.234	-	-	-	-	0.149	0.244	0.167	0.228	0.169	0.268	0.164	0.253	-	-	-	-	0.186	0.235	0.440	0.470
	96	**0.152**	**0.222**	0.256	0.293	-	-	-	-	0.253	0.309	0.245	0.280	0.233	0.306	0.211	0.253	-	-	-	-	0.221	0.267	0.674	0.565
	Avg	**0.103**	**0.191**	0.142	0.222	-	-	-	-	0.142	0.230	0.148	0.224	0.150	0.246	0.138	0.220	-	-	-	-	0.150	0.226	0.379	0.416
long 1st Count		**33**		5		18		7		1		1		1		5		0		0		0		0	
short 1st Count		**29**		8		-		-		4		0		0		0		-		-		0		0	
all 1st Count		**62**		11		-		-		5		1		1		5		-		-		0		0	

C.2 Comparison of Ablation Results

We perform ablation studies to examine the design of FSC and the robustness of its components, focusing on the ACE module and the FSC-Refiner module. Full results are provided in Table 12.

We build several variants by removing or replacing key components:

- **w/o FSC**: eliminates the FSC-Refiner and directly fuses the two streams, i.e., $h_{\text{fsr}} = h_{\text{fast}} + h_{\text{slow}}$, to evaluate the overall effectiveness of the FSC module.
- **w/o ACE**: removes the entire ACE module to evaluate the overall contribution of channel encoding.
- **w/o RM**: bypasses the MLP refinement and directly projects lag features, i.e., $\tilde{h}_{\text{lag}} = h_{\text{lag}}, h_{\text{align}} = \text{Linear}(h_{\text{lag}})$, to assess the effect of nonlinear refinement.
- **w/o orth**: removes the orthogonal loss to examine its impact on feature decoupling.
- **Re-Time** utilizes sinusoidal functions to capture sequential dependencies along the temporal dimension.
- **ACE-mean**: averages ACE across channels and broadcasts the shared encoding to evaluate the importance of channel-specific representations.
- **Re-C1**: replaces sinusoidal encoding with rotary positional encoding (RoPE) (Su et al., 2024) to compare different encoding schemes.
- **lag-fast**: directly uses fast features as lag representations, i.e., $h_{\text{lag}} = h_{\text{fast}}$, to evaluate the role of slow features as a reference anchor.
- **lag-add**: replaces subtraction with addition, i.e., $h_{\text{lag}} = h_{\text{fast}} + h_{\text{slow}}$, to examine the necessity of explicit difference modeling.
- **lag-cat**: derives the lag feature via linear projection of concatenated representations, i.e., $h_{\text{lag}} = \text{Linear}(\text{Cat}(h_{\text{fast}}, h_{\text{slow}}))$, to compare implicit fusion with explicit subtraction.
- **only-S**: uses only slow features for prediction, i.e., $h_{\text{fsr}} = h_{\text{slow}}$, to evaluate the contribution of fast dynamic correction.
- **only-A**: uses only aligned features for prediction, i.e., $h_{\text{fsr}} = h_{\text{align}}$, to evaluate the necessity of preserving slow features as a stable reference.

Table 12: Ablation Results. The input length is fixed to 96. The best results are in **bold**.

Model		w/o FSC		w/o ACE		w/o RM		w/o orth		Re-Time		CPE-mean		Re-C1		lag-fast		lag-add		lag-cat		only-S		only-A		FSC	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.367	0.395	**0.360**	0.393	0.363	**0.392**	0.366	0.396	0.366	0.397	0.365	0.396	0.361	0.393	0.366	0.395	0.365	0.396	0.361	0.393	0.466	0.463	0.366	0.396	0.365	0.395
	192	0.411	**0.420**	0.407	0.423	0.414	0.422	0.409	0.422	0.412	0.424	**0.406**	0.422	0.411	0.425	0.409	0.424	0.407	0.423	0.404	0.423	0.485	0.476	0.412	0.424	0.407	0.423
	336	0.464	0.446	0.455	0.446	0.460	0.444	0.449	**0.443**	0.446	**0.443**	0.450	0.444	0.460	0.448	0.449	0.444	0.446	**0.443**	0.450	0.449	0.498	0.488	0.447	0.444	**0.445**	**0.443**
	720	0.467	0.468	0.481	0.482	0.525	0.498	0.460	0.474	**0.442**	**0.462**	0.470	0.479	0.476	0.481	0.464	0.476	0.454	0.470	0.478	0.482	0.491	0.499	0.452	0.470	0.453	0.469
	avg	0.427	0.432	0.426	0.436	0.441	0.439	0.421	0.434	**0.416**	**0.431**	0.423	0.435	0.427	0.437	0.422	0.435	0.418	0.433	0.423	0.437	0.485	0.482	0.419	0.434	0.417	0.432
Solar	96	0.183	0.238	0.174	**0.229**	0.176	0.233	0.174	**0.229**	0.176	0.232	0.174	**0.229**	0.175	0.231	0.174	**0.229**	0.175	0.230	0.176	0.231	0.291	0.373	**0.173**	0.231	0.174	**0.229**
	192	0.202	0.249	0.192	0.247	0.200	0.251	0.193	0.246	0.200	0.248	0.193	0.246	0.194	0.246	0.193	0.247	0.195	0.246	0.198	**0.245**	0.302	0.379	**0.190**	0.249	0.193	**0.245**
	336	0.213	0.257	0.204	0.248	0.202	0.246	**0.200**	**0.244**	0.202	0.247	**0.200**	0.245	0.201	0.244	0.203	0.250	0.201	0.245	0.229	0.280	0.317	0.386	0.207	0.255	**0.200**	**0.244**
	720	0.216	0.252	0.209	0.248	0.213	0.251	0.209	**0.247**	0.215	0.255	0.209	**0.247**	0.210	0.248	**0.208**	0.251	0.210	0.248	0.209	0.251	0.314	0.383	0.209	0.249	0.210	0.248
	avg	0.204	0.249	0.195	0.243	0.198	0.245	**0.194**	**0.241**	0.198	0.245	**0.194**	0.242	0.195	0.242	0.195	0.244	0.195	0.242	0.203	0.252	0.306	0.380	0.195	0.246	**0.194**	**0.241**
Weather	96	0.156	0.202	0.155	0.201	0.154	0.200	0.153	**0.198**	0.154	0.201	0.155	0.201	0.154	0.199	0.153	0.199	0.153	0.199	0.153	**0.198**	0.189	0.236	0.154	0.200	**0.152**	**0.198**
	192	0.205	0.245	0.203	0.245	0.204	0.245	**0.202**	0.244	0.203	0.246	0.203	0.245	0.203	0.244	0.202	0.244	0.203	0.244	0.203	**0.243**	0.238	0.273	**0.202**	0.245	**0.202**	0.244
	336	0.262	0.287	0.261	0.288	0.259	0.286	0.258	0.286	0.260	0.288	0.261	0.288	0.260	0.287	0.258	0.286	**0.257**	**0.285**	0.259	0.286	0.287	0.306	0.259	0.287	0.258	0.286
	720	0.344	0.342	0.342	0.341	0.340	**0.340**	0.340	**0.340**	0.341	0.342	0.341	0.341	0.341	0.342	0.340	**0.340**	**0.339**	**0.340**	0.340	0.341	0.357	0.350	0.340	**0.340**	**0.339**	**0.340**
	avg	0.242	0.269	0.240	0.269	0.239	0.268	**0.238**	**0.267**	0.239	0.269	0.240	0.269	0.239	0.268	**0.238**	**0.267**	**0.238**	**0.267**	0.239	**0.267**	0.268	0.291	0.239	0.268	**0.238**	**0.267**
PEMS08	12	0.072	0.172	0.067	0.163	0.070	0.169	0.067	0.164	0.069	0.170	0.068	0.167	**0.066**	**0.162**	**0.066**	0.164	**0.066**	0.163	0.071	0.171	0.409	0.384	0.067	0.164	**0.066**	0.164
	24	0.094	0.192	0.090	0.191	0.091	0.188	0.086	0.187	0.084	0.181	0.087	0.188	0.089	0.191	0.084	0.181	0.084	0.182	0.092	0.193	0.409	0.392	0.084	0.179	**0.082**	**0.177**
	48	0.129	0.223	0.114	0.200	0.119	0.215	0.111	0.200	0.112	0.205	0.116	0.208	0.116	0.202	0.116	0.207	**0.109**	**0.199**	0.111	0.201	0.408	0.389	0.110	0.202	0.112	0.200
	96	0.172	0.255	0.153	0.224	0.171	0.254	**0.148**	0.223	0.155	0.228	0.155	0.227	0.151	0.224	0.156	0.233	0.154	0.228	0.150	0.224	0.397	0.385	0.151	0.228	0.152	**0.222**
	avg	0.132	0.223	0.119	0.205	0.127	0.219	**0.103**	0.193	0.105	0.196	0.119	0.208	0.105	0.195	0.119	0.207	**0.103**	0.193	0.106	0.197	0.405	0.389	0.115	0.203	**0.103**	**0.191**
1st Count		1		2		2		9		3		**4**		**2**		**3**		**7**		**3**		**0**		**4**		**15**	

C.3 Effectiveness of FSC.

To assess the effectiveness and generalization of FSC, we conduct experiments on three benchmark datasets with strong periodic patterns. the detailed results are provided in Table 13.

C.4 Robustness Analysis

Table 14 reports the mean and standard deviation over four random seeds (2024–2027) on three datasets. FSC achieves the lowest averaged MSE and MAE on all datasets. In terms of stability, it attains the lowest or jointly lowest variance in 5 out of 6 averaged metrics: lowest on ETTh1, jointly lowest with CycleNet on Weather, and lowest MSE variance on PEMS04 (its MAE variance is

Table 13: Effectiveness and generalizability of FSC. The table compares the original models (Original) with their FSC-augmented counterparts (+FSC), along with the relative improvements (Imp.). Memory (Mem.) denotes the peak GPU memory usage during training.

Model		iTransformer									PatchTST									DLinear								
Setup		Original			+ FSC Tech.			Imp. ↑			Original			+ FSC Tech.			Imp. ↑			Original			+ FSC Tech.			Imp. ↑		
Metric		MSE	MAE	Mem.	MSE	MAE	Mem.	MSE	MAE	Mem.	MSE	MAE	Mem.	MSE	MAE	Mem.	MSE	MAE	Mem.	MSE	MAE	Mem.	MSE	MAE	Mem.	MSE	MAE	Mem.
Electricity	96	0.150	0.241	1175	**0.134**	**0.230**	1225	10.7%	4.6%	4.2%	0.164	0.254	6092	**0.134**	**0.231**	6540	22.4%	10.0%	7.3%	0.195	0.277	421	**0.136**	**0.234**	443	30.2%	15.5%	5.2%
	192	0.164	0.256	1165	**0.152**	**0.248**	1249	7.3%	3.1%	7.2%	0.174	0.267	6092	**0.152**	**0.249**	6560	14.5%	7.2%	7.7%	0.194	0.279	441	**0.152**	**0.250**	501	21.6%	10.4%	13.6%
	336	0.180	0.273	1227	**0.162**	**0.262**	1289	10.0%	4.0%	5.0%	0.193	0.285	6112	**0.168**	**0.267**	6682	14.9%	6.7%	9.3%	0.207	0.279	495	**0.167**	**0.268**	613	19.3%	3.9%	23.8%
	720	0.210	0.300	1261	**0.208**	**0.297**	1387	0.9%	1.0%	9.0%	0.232	0.321	6122	**0.207**	**0.300**	6712	12.1%	7.0%	9.6%	0.244	0.330	593	**0.210**	**0.304**	775	13.9%	7.9%	30.7%
	Avg	0.176	0.267	1207	**0.164**	**0.259**	1287	6.8%	3.0%	6.6%	0.191	0.282	6104	**0.165**	**0.262**	6624	15.8%	7.6%	8.5%	0.210	0.295	487	**0.166**	**0.264**	583	20.9%	10.5%	19.7%
PEMS03	12	0.106	0.219	1993	**0.060**	**0.162**	2121	43.4%	26.1%	6.4%	0.072	0.179	11738	**0.061**	**0.164**	12050	18.0%	9.1%	2.6%	0.105	0.221	405	**0.063**	**0.168**	411	40.0%	24.0%	1.5%
	24	0.091	0.199	1989	**0.076**	**0.182**	2119	16.5%	8.5%	6.5%	0.103	0.213	11736	**0.078**	**0.184**	12050	32.1%	15.8%	2.7%	0.183	0.299	401	**0.079**	**0.186**	425	56.8%	37.8%	5.9%
	48	0.192	0.306	2009	**0.102**	**0.210**	2141	46.9%	31.4%	6.5%	0.155	0.263	12004	**0.106**	**0.215**	12046	46.2%	22.3%	3.5%	0.315	0.407	401	**0.109**	**0.217**	427	65.4%	46.7%	6.5%
	96	0.236	0.344	2011	**0.132**	**0.239**	2141	44.1%	39.3%	6.5%	0.205	0.305	12024	**0.142**	**0.245**	12046	44.4%	24.5%	0.2%	0.455	0.508	441	**0.145**	**0.247**	463	68.1%	51.4%	5.0%
	Avg	0.156	0.267	2000	**0.092**	**0.198**	2130	41.0%	34.8%	6.5%	0.134	0.240	11876	**0.097**	**0.202**	12048	38.1%	18.8%	1.4%	0.265	0.359	412	**0.099**	**0.204**	431	62.6%	43.2%	4.6%
PEMS04	12	0.080	0.187	1695	**0.070**	**0.172**	1809	12.5%	8.0%	6.7%	0.087	0.195	10252	**0.073**	**0.178**	10508	19.2%	9.6%	2.5%	0.114	0.228	385	**0.077**	**0.185**	409	32.5%	18.9%	6.2%
	24	0.096	0.206	1699	**0.082**	**0.189**	1813	14.6%	8.3%	6.7%	0.119	0.231	10256	**0.087**	**0.197**	10510	36.8%	17.3%	2.5%	0.187	0.298	389	**0.096**	**0.207**	413	48.7%	30.5%	6.2%
	48	0.129	0.242	1713	**0.97**	**0.207**	1807	24.8%	14.5%	5.5%	0.172	0.279	10482	**0.109**	**0.220**	10522	57.8%	26.8%	0.4%	0.319	0.402	401	**0.122**	**0.237**	423	61.8%	41.0%	5.2%
	96	0.178	0.286	1715	**0.126**	**0.238**	1809	29.2%	16.8%	5.5%	0.219	0.321	10482	**0.137**	**0.248**	10524	59.9%	29.4%	0.4%	0.424	0.481	421	**0.150**	**0.263**	453	64.6%	45.3%	7.6%
	Avg	0.121	0.230	1705	**0.094**	**0.201**	1809	22.3%	12.6%	6.1%	0.149	0.257	10368	**0.101**	**0.211**	10516	47.5%	21.8%	1.4%	0.261	0.352	399	**0.111**	**0.223**	424	57.5%	36.6%	6.3%

slightly higher than TQNet but still low). Overall, FSC demonstrates strong robustness to random initialization, with its advantage becoming more evident as the forecasting horizon increases.

Table 14: Robustness across random seeds. For each method, results under each seed are first averaged over four forecasting horizons, and we then report the mean ± standard deviation across four seeds (2024–2027). Error bars denote standard deviation. FSC achieves lower errors with reduced variability on most datasets, indicating strong stability.

Model		CycleNet(2024)		TQNet(2025)		GTR(2026)		FSC(ours)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.3756 ± 0.0018	0.3954 ± 0.0013	0.3713 ± 0.0006	0.3933 ± 0.0004	0.3695 ± 0.0010	**0.3920** ± 0.0005	**0.3625** ± 0.0021	0.3945 ± 0.0006
	192	0.4369 ± 0.0032	0.4284 ± 0.0024	0.4295 ± 0.0011	0.4241 ± 0.0014	0.4253 ± 0.0016	0.4231 ± 0.0011	**0.4069** ± 0.0013	**0.4225** ± 0.0004
	336	0.4965 ± 0.0017	0.4559 ± 0.0036	0.4782 ± 0.0027	0.4489 ± 0.0034	0.4753 ± 0.0076	0.4467 ± 0.0030	**0.4448** ± 0.0051	**0.4440** ± 0.0024
	720	0.5181 ± 0.0266	0.4836 ± 0.0157	0.5005 ± 0.0125	0.4799 ± 0.0099	0.5079 ± 0.0173	0.4829 ± 0.0092	**0.4605** ± 0.0060	**0.4737** ± 0.0039
	avg	0.4568 ± 0.0062	0.4408 ± 0.0035	0.4449 ± 0.0038	0.4366 ± 0.0032	0.4445 ± 0.0063	0.4362 ± 0.0033	**0.4188** ± 0.0025	**0.4337** ± 0.0015
Weather	96	0.1591 ± 0.0005	0.2037 ± 0.0004	0.1569 ± 0.0010	0.2002 ± 0.0010	0.1553 ± 0.0009	0.2003 ± 0.0013	**0.1530** ± 0.0011	**0.1984** ± 0.0009
	192	0.2067 ± 0.0003	0.2476 ± 0.0001	0.2054 ± 0.0002	0.2450 ± 0.0004	0.2038 ± 0.0007	0.2448 ± 0.0008	**0.2018** ± 0.0003	**0.2439** ± 0.0002
	336	0.2629 ± 0.0007	0.2898 ± 0.0011	0.2630 ± 0.0009	0.2879 ± 0.0007	0.2611 ± 0.0005	0.2874 ± 0.0008	**0.2582** ± 0.0005	**0.2862** ± 0.0005
	720	0.3445 ± 0.0008	0.3445 ± 0.0004	0.3431 ± 0.0003	0.3416 ± 0.0001	0.3415 ± 0.0006	0.3417 ± 0.0005	**0.3397** ± 0.0004	**0.3403** ± 0.0002
	avg	0.2433 ± 0.0003	0.2714 ± 0.0002	0.2422 ± 0.0005	0.2687 ± 0.0004	0.2404 ± 0.0004	0.2686 ± 0.0004	**0.2382** ± 0.0003	**0.2672** ± 0.0002
PEMS04	12	0.0782 ± 0.0002	0.1861 ± 0.0003	0.0665 ± 0.0001	0.1665 ± 0.0004	0.0657 ± 0.0001	0.1645 ± 0.0003	**0.0652** ± 0.0001	**0.1626** ± 0.0002
	24	0.0988 ± 0.0001	0.2124 ± 0.0003	0.0772 ± 0.0005	0.1815 ± 0.0005	0.0758 ± 0.0005	0.1775 ± 0.0014	**0.0741** ± 0.0002	**0.1741** ± 0.0007
	48	0.1321 ± 0.0003	0.2480 ± 0.0003	0.0964 ± 0.0011	0.2057 ± 0.0009	0.0945 ± 0.0011	0.2006 ± 0.0019	**0.0883** ± 0.0005	**0.1905** ± 0.0009
	96	0.1679 ± 0.0022	0.2814 ± 0.0025	0.1234 ± 0.0012	0.2333 ± 0.0012	0.1167 ± 0.0009	0.2235 ± 0.0018	**0.1086** ± 0.0017	**0.2124** ± 0.0034
	avg	0.1192 ± 0.0005	0.2320 ± 0.0006	0.0909 ± 0.0005	0.1968 ± 0.0003	0.0882 ± 0.0005	0.1917 ± 0.0010	**0.0841** ± 0.0004	**0.1849** ± 0.0007

C.5 Periodic Channel Modeling

In traffic forecasting and other scenarios with strong cross-channel dependencies, variables often share similar periodic structures while exhibiting significant inter-channel correlations. In FSC, CPE models local dynamics and cross-channel relations, while FSC-Refiner aligns slow periodic features with local patterns. However, neither explicitly enforces channel-wise consistency of periodic representations. As periodic features are learned from inputs containing local perturbations, strong cross-channel dependencies may introduce channel-wise drift or redundancy, limiting model performance.

To address this issue, GTR (Cao et al., 2026) introduces channel-wise aggregation to integrate features and capture shared global information. Inspired by this idea, we incorporate channel aggregation into FSC at the periodic feature level to mitigate channel-wise inconsistency.

Specifically, we consider two settings for constructing periodic representations, and we denote C as the number of channels, L as the input sequence length, and D as the hidden feature dimensionality. In the **Prior Periodic Stage (PPS)**, the representation is built solely from the periodic embedding: $Z^{(PPS)} = \boldsymbol{\pi} \in \mathbb{R}^{C \times L}$. In the **Fusion-enhanced Periodic Stage (FPS)**, the input is first combined

with periodic information and then projected via a linear mapping: $Z^{(FPS)} = \mathrm{Linear}(X + \boldsymbol{\pi}) \in \mathbb{R}^{C \times D}$.

Given a periodic representation $Z \in \mathbb{R}^{C \times d}$, where $d = L$ for PPS and $d = D$ for FPS, we apply a unified channel-wise aggregation. We first compute the mean over the channel dimension:

$$z_{\mathrm{agg}} = \frac{1}{C} \sum_{c=1}^{C} Z_{c,:} \in \mathbb{R}^{1 \times d}.$$

and then broadcast it back to the original channel space:

$$\tilde{Z} = \mathrm{Broadcast}(z_{\mathrm{agg}}) \in \mathbb{R}^{C \times d},$$

Table 15: Comparison results of periodic channel modeling experiments on the Traffic dataset.

Model	96		192		336		720		*Avg*	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
FSC	0.436	0.262	0.443	0.272	0.462	0.277	0.508	0.296	0.462	0.277
+PPS	0.405	0.270	0.419	0.277	0.434	0.284	0.466	0.301	0.432	0.283
+FPS	0.401	0.270	0.419	0.277	0.433	0.283	0.464	0.301	0.429	0.283

From the results in Table 15, both variants lead to consistent improvements in MSE, validating the effectiveness of introducing cross-channel aggregation at the periodic feature level. The FPS variant performs better overall, likely because it incorporates both periodic structure and input-dependent dynamics, providing a richer context for channel-wise aggregation.

D Theoretical Analysis

A Bias-Noise View of Fast-to-Slow Correction. FSC is motivated by the observation that the periodic prior provides stable historical information but can become mismatched with current dynamics under distribution shifts, whereas the current input contains more recent information but is more susceptible to noise. We formalize this behavior with a latent representation $\mathbf{h}^*$ corresponding to the current underlying state in the shared feature space:

$$\mathbf{h}_{\mathrm{slow}} = \mathbf{h}^* + \mathbf{b} + \boldsymbol{\epsilon}, \qquad \mathbf{h}_{\mathrm{fast}} = \mathbf{h}^* + \boldsymbol{\delta}, \tag{10}$$

where $\mathbf{b}$ denotes the systematic lag of the slow representation, while $\boldsymbol{\epsilon}$ and $\boldsymbol{\delta}$ denote residual errors in the slow and fast representations. We assume

$$\mathbb{E}[\boldsymbol{\epsilon}] = \mathbb{E}[\boldsymbol{\delta}] = \mathbf{0}, \qquad \mathbb{E}|\boldsymbol{\epsilon}|^2 = \sigma_\epsilon^2, \qquad \mathbb{E}|\boldsymbol{\delta}|^2 = \sigma_\delta^2, \tag{11}$$

and approximate the residual errors as weakly correlated:

$$\mathbb{E}\left[\boldsymbol{\epsilon}^\top \boldsymbol{\delta}\right] \approx 0. \tag{12}$$

The lag $\mathbf{b}$ is treated as fixed within the current forecasting window. This assumption is consistent with the proposed fast-slow decoupling. In the mechanism analysis, the Pearson correlation between the resulting $\mathbf{h}_{\mathrm{slow}}$ and $\mathbf{h}_{\mathrm{fast}}$ is substantially closer to zero than that between the original periodic prior $\boldsymbol{\pi}$ and input $\mathbf{x}$, indicating weaker linear dependence after projection. This provides empirical support for the weak-correlation approximation used in the analysis.

The Fast-Slow Difference Reveals the Lag. Define the fast-slow discrepancy as

$$\mathbf{d} = \mathbf{h}_{\mathrm{fast}} - \mathbf{h}_{\mathrm{slow}}. \tag{13}$$

Substituting the above representation model yields

$$\mathbf{d} = -\mathbf{b} + \boldsymbol{\delta} - \boldsymbol{\epsilon}. \tag{14}$$

Therefore,

$$\mathbb{E}[\mathbf{d}] = -\mathbf{b}. \tag{15}$$

Hence, the discrepancy provides a direct signal of the slow-feature lag: when the periodic representation becomes more mismatched with the current dynamics, the magnitude of $\mathbf{d}$ increases accordingly. This explains why FSC uses the fast-slow difference as a correction signal rather than directly replacing the stable slow representation with the fast one.

Optimal Linear Correction. To characterize the effect of correction strength, consider a scalar linear correction

$$\widehat{\mathbf{h}}(w) = \mathbf{h}_{\text{slow}} + w\mathbf{d}, \tag{16}$$

where w controls how strongly the discrepancy is used to correct the slow representation. Its estimation error is

$$\widehat{\mathbf{h}}(w) - \mathbf{h}^* = (1-w)\mathbf{b} - (1-w)\boldsymbol{\epsilon} + w\boldsymbol{\delta}. \tag{17}$$

Under the above assumptions, the expected squared error becomes

$$\begin{aligned} \mathcal{E}(w) &= \mathbb{E}\left[\left|\widehat{\mathbf{h}}(w) - \mathbf{h}^*\right|^2\right] \\ &= (1-w)^2\left(|\mathbf{b}|^2 + \sigma_\epsilon^2\right) + w^2\sigma_\delta^2. \end{aligned} \tag{18}$$

Theorem 1 (Optimal Correction Strength)**.** *Under the above bias-noise model, the optimal scalar correction coefficient is*

$$w^* = \frac{|\mathbf{b}|^2 + \sigma_\epsilon^2}{|\mathbf{b}|^2 + \sigma_\epsilon^2 + \sigma_\delta^2}. \tag{19}$$

Proof. Differentiating $\mathcal{E}(w)$ with respect to w gives

$$\frac{\partial \mathcal{E}(w)}{\partial w} = -2(1-w)\left(|\mathbf{b}|^2 + \sigma_\epsilon^2\right) + 2w\sigma_\delta^2. \tag{20}$$

Setting the derivative to zero directly gives the stated solution. Since

$$\frac{\partial^2 \mathcal{E}(w)}{\partial w^2} = 2\left(|\mathbf{b}|^2 + \sigma_\epsilon^2 + \sigma_\delta^2\right) > 0, \tag{21}$$

the solution is the unique minimizer. □

Interpretation. The optimal coefficient reveals two properties of fast-to-slow correction. First,

$$\frac{\partial w^*}{\partial |\mathbf{b}|^2} > 0, \tag{22}$$

so stronger slow-feature lag favors stronger correction. Second,

$$\frac{\partial w^*}{\partial \sigma_\delta^2} < 0, \tag{23}$$

so noisier fast features favor weaker correction. Thus, effective correction should not simply replace slow features with fast features; instead, it should adapt the contribution of the fast-slow discrepancy according to the severity of the lag and the noise level of the fast representation. Moreover, the optimal correction achieves

$$\mathcal{E}(w^*) = \frac{\left(|\mathbf{b}|^2 + \sigma_\epsilon^2\right)\sigma_\delta^2}{|\mathbf{b}|^2 + \sigma_\epsilon^2 + \sigma_\delta^2}, \tag{24}$$

which is no larger than the error of the uncorrected slow representation,

$$\mathcal{E}(0) = |\mathbf{b}|^2 + \sigma_\epsilon^2. \tag{25}$$

This analytically supports the use of the fast representation as a correction source when the slow representation suffers from lag.

Connection to FSC-Refiner. The optimal coefficient w^* depends on the unknown and input-dependent lag and noise levels, making a fixed correction strength inadequate in practice. FSC therefore replaces the scalar correction with the learned FSC-Refiner:

$$\mathbf{h}_{\text{fsc}} = \mathbf{h}_{\text{slow}} + f_\theta\left(\mathbf{h}_{\text{fast}} - \mathbf{h}_{\text{slow}}\right), \tag{26}$$

where f_θ is implemented by the lightweight residual MLP and projection layers. Thus, w^* provides an analytical reference for the input-dependent correction performed by FSC-Refiner.

E More Visualization Results

E.1 Mechanism Analysis of FSC

We visualize the channel-wise self-Pearson correlation coefficient matrices of individual features, computed respectively for the original periodic information, the FSC-corrected output features, and the ground truth (input length 96, forecasting length 96) on the ETTh1, ETTm2, PEMS07, and PEMS08 datasets, as shown in Figure 7.

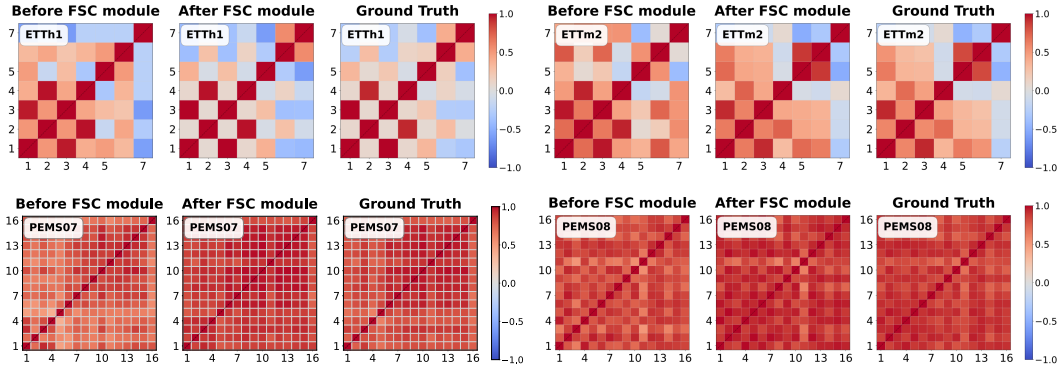

Figure 7: Channel-wise self-Pearson correlation matrices of original periodic information, FSC-corrected output, and ground truth (input/forecasting length 96) on ETTh1, ETTm2, PEMS07, and PEMS08.

We also visualize the channel-wise Pearson correlation coefficient matrices between the original periodic information and the original input before entering FSC, between the Slow and Fast features extracted by FSC, and between the aligned lag and slow features, on the ETTh1, ETTm2, PEMS07, and PEMS08 datasets (input length 96, forecasting length 96), as shown in Figure 8.

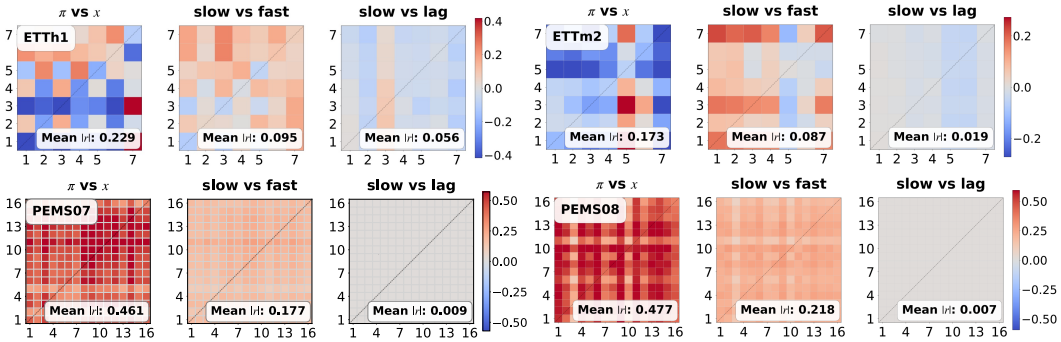

Figure 8: Channel-wise Pearson correlation matrices of original periodic information vs. original input before FSC, Slow vs. Fast features extracted by FSC, and aligned lag vs. slow features (input/forecasting length 96) on ETTh1, ETTm2, PEMS07, and PEMS08.

E.2 Visualization of Forecasting Results

We visualize predicted and ground-truth sequences on representative datasets to compare model behaviors under distribution shifts. As shown in Figure 9, when ETTh1 becomes highly irregular, periodic methods such as CycleNet and GTR exhibit clear phase delays. While they capture stable frequency patterns (i.e., slow features h_{slow}), their reliance on fixed periodic structures limits adaptation to rapidly changing dynamics. In contrast, FSC uses fast-varying features to correct the slow representation, enabling predictions to better track local fluctuations and preserve sharp peaks and drops. Similarly, on PEMS04 (Figure 12), FSC adapts more promptly to a strong upward shift than periodic methods. These results demonstrate that FSC effectively corrects slow-feature bias and improves adaptation to evolving dynamics.

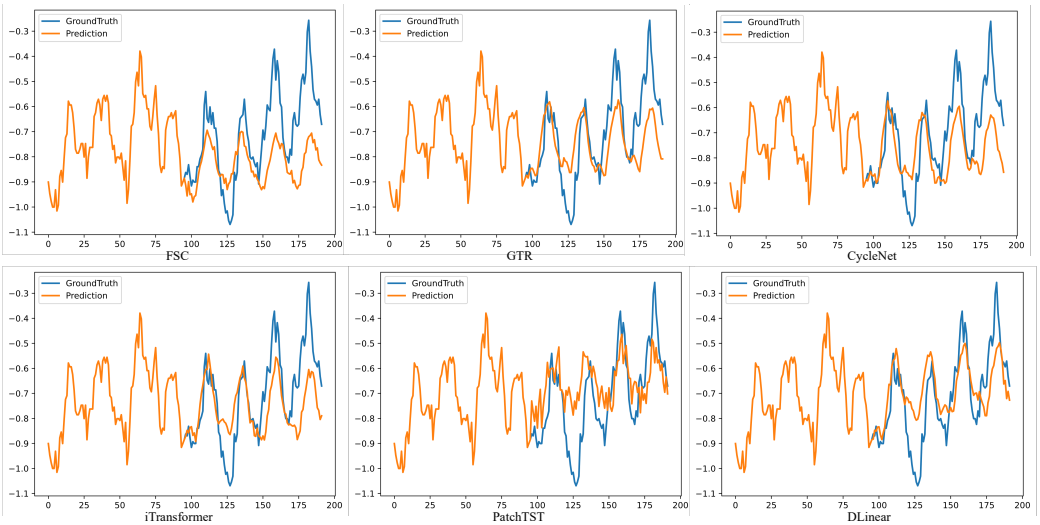

Figure 9: Visualization with input length 96 and forecasting length 96 on the ETTh1 dataset.

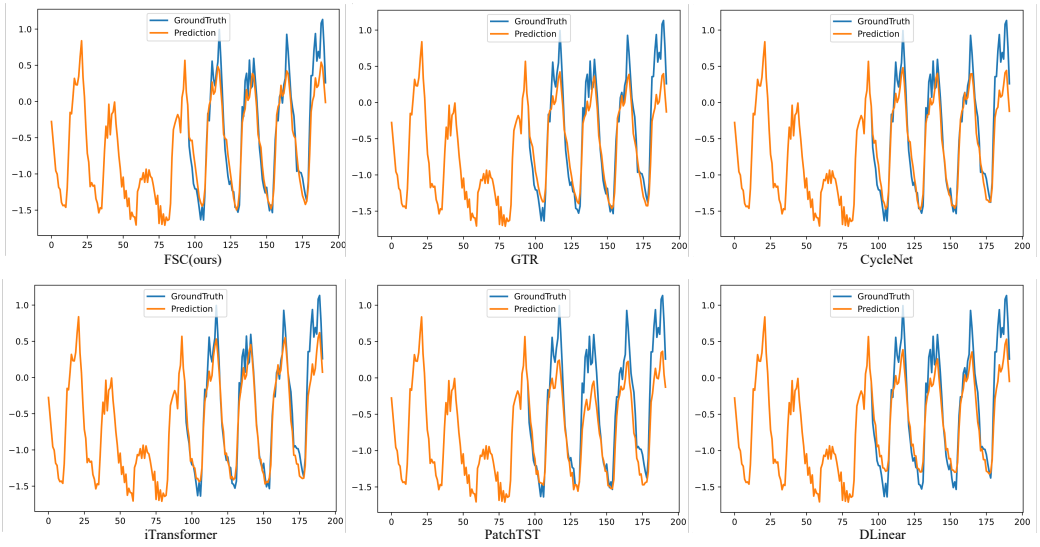

Figure 10: Visualization with input length 96 and forecasting length 96 on the Electricity dataset.

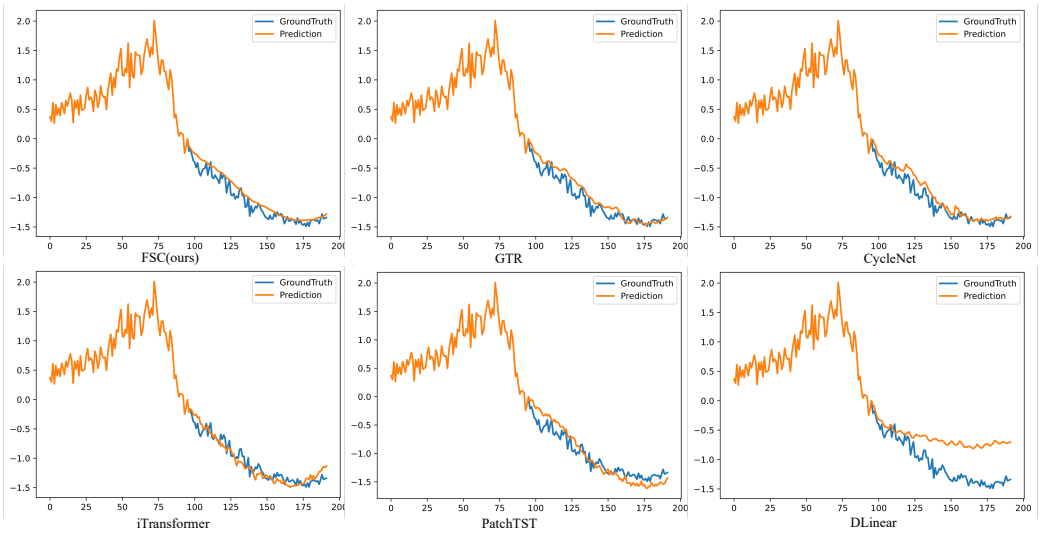

Figure 11: Visualization with input length 96 and forecasting length 96 on the PEMS03 dataset.

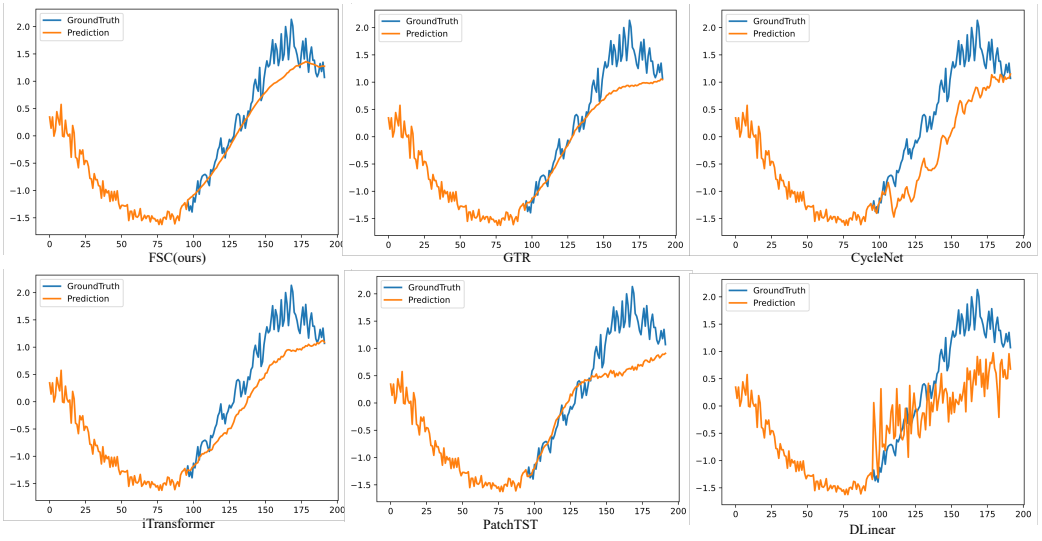

Figure 12: Visualization with input length 96 and forecasting length 96 on the PEMS04 dataset.