Time–Frequency Non-Stationary Modeling for Multivariate Time Series Forecasting

Haoyi Zhao[1], Yishan Jiang[2], Ji Chang[1,2,*], Jiqian Yang[3], Dawei Ma[1], Wenjun Lv[1,2]

[1]Institute of Advanced Technology, University of Science and Technology of China
[2]Department of Automation, University of Science and Technology of China
[3]Shenzhen International Graduate School, Tsinghua University

{haoyizhao,jiangyishan,dwma}@mail.ustc.edu.cn
{changji,wlv}@ustc.edu.cn, jq-yang26@mails.tsinghua.edu.cn

Abstract

Multivariate time series forecasting is important in real-world applications, but practical data often show both temporal and spectral non-stationarity. Existing methods may treat severe spectral drift as informative evolution, allow unreliable spectral components to contaminate temporal representations through credibility-agnostic time–frequency interaction, and infer cross-variable dependencies from noisy local correlations, which may lead to spurious long-term dependencies. To address these issues, we propose TFNS, a dual-stream time–frequency framework for suppressing unreliable spectral evidence under non-stationarity. Specifically, we introduce a Spectral Credibility Estimator to quantify patch-level spectral stability and softly suppress low-credibility frequency components for more reliable frequency modeling; a Credibility-Gated Cross-Interaction module to enable bidirectional time–frequency enhancement while adaptively constraining cross-domain information injection according to spectral reliability; and a Spectral Cointegration Graph to capture stable long-term cross-variable structure in the spectral domain while mitigating spurious dependencies under non-stationary drift. Together, these modules form a progressive pipeline from local spectral purification to reliability-aware cross-domain interaction and robust long-term structure modeling. Experiments on 11 benchmarks show that TFNS achieves state-of-the-art performance, ranking first in 44/55 MSE and 43/55 MAE comparisons. The code and datasets are available at https://github.com/Yoon1230/TFNS.

1 Introduction

Time series forecasting is fundamental to many real-world applications, where accurate long-term prediction supports planning, control, and decision-making. However, real-world multivariate time series (MTS) often exhibit non-stationarity, evolving cross-variable dependencies, and nonlinear temporal dynamics, making robust long-term forecasting challenging [Liu et al., 2022, 2023b, 2024a].

A major difficulty in real-world forecasting lies in dual non-stationarity [Liu et al., 2022]. On the one hand, temporal distributions may shift significantly due to changing regimes, seasonal variations, interventions, or environmental changes. On the other hand, spectral structures may also gradually evolve, as the frequency composition of local patterns changes over time. Such spectral drift does not always indicate meaningful temporal evolution, but may instead reflect unstable or noisy local patterns [Zhou et al., 2022a, Yi et al., 2023b, Wu et al., 2022]. Directly exploiting unreliable spectral

*Ji Chang is the corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

components can severely distort representation learning, contaminate time–frequency interaction, and weaken generalization under future distribution shifts.

Recent deep forecasting models, including recurrent, convolutional, attention-based, and Transformer-style architectures, have improved long-range temporal modeling [Wu et al., 2021, Lin et al., 2024, Zhang et al., 2024, Han et al., 2024, Hu et al., 2024, Chen et al., 2024, Wang et al., 2024b, Yi et al., 2023a]. Frequency-domain and time–frequency methods further capture periodicity and latent temporal structures, while graph-based models enhance cross-variable dependency modeling. Despite these advances, existing methods still face three key limitations. Frequency-aware models often treat local spectral patterns as equally reliable, causing unstable frequency components to enter downstream modeling. Time–frequency interaction mechanisms usually lack reliability control, allowing unreliable spectral evidence to contaminate temporal representations. Cross-variable dependencies are often inferred from noisy temporal correlations or local feature similarity, which may induce spurious long-term relations under non-stationarity.

To address these issues, we propose **TFNS**, a **T**ime–**F**requency forecasting framework for **N**on-**S**tationary multivariate time series. TFNS adopts a dual-stream architecture and uses spectral credibility to regulate information flow. It first employs a Spectral Credibility Estimator (SCE) to quantify patch-level spectral stability and suppress unreliable frequency components. Based on the purified spectral representations, a Credibility-Gated Cross-Interaction (CGI) module performs reliability-aware bidirectional interaction between temporal and frequency streams. Finally, a Spectral Cointegration Graph (SCG) captures stable long-term cross-variable structures in the spectral domain and calibrates local relations with global spectral priors to mitigate spurious dependencies.

Extensive experiments on 11 real-world benchmark datasets demonstrate the effectiveness of TFNS. The proposed framework achieves strong forecasting performance across different horizons, especially under challenging non-stationary conditions, ranking first in 44/55 MSE and 43/55 MAE comparisons.

The main contributions of this work are summarized as follows:

- We propose **TFNS**, a unified time–frequency forecasting framework that explicitly incorporates spectral credibility for robust non-stationary multivariate time series forecasting.
- We develop a progressive credibility-aware modeling pipeline consisting of SCE for local spectral purification, CGI for reliability-controlled time–frequency interaction, and SCG for robust cross-variable structure modeling with conservative global spectral calibration.
- Experiments on 11 real-world benchmark datasets show that **TFNS** achieves state-of-the-art performance across various horizons and non-stationary conditions.

2 Related Work

Multivariate Time Series Forecasting. Multivariate time series forecasting aims to predict future trajectories of multiple variables from historical observations. The key challenge lies in modeling long-range temporal dependencies and cross-variable interactions. Existing methods address this problem through recurrent networks, convolutional architectures, attention mechanisms, Transformer-style models, graph-based structures, and MLP-based designs [Wu et al., 2021, 2022, Nie et al., 2022, Liu et al., 2024b, Luo and Wang, 2024]. In particular, graph learning and relational aggregation methods explicitly model cross-variable dependencies [Yi et al., 2023a, Xue et al., 2023, Lu et al., 2024, Wang et al., 2024c, Qiu et al., 2025], while attention-based, patch-based, multi-scale, and decomposition-based methods capture long-range temporal patterns efficiently [Wu et al., 2021, Chen et al., 2024]. Despite their effectiveness, most methods do not explicitly assess the reliability of local temporal or spectral evidence, limiting robustness under strong non-stationarity.

Non-Stationary Modeling in Time Series. Real-world time series often exhibit non-stationarity, where temporal statistics and spectral structures may vary over time [Liu et al., 2023b]. Existing approaches can be broadly categorized into time-domain adaptation, frequency-domain analysis, and joint time–frequency modeling. Time-domain methods alleviate distribution shift through normalization, distribution alignment, decomposition, or stationarization-oriented transformations [Liu et al., 2022, 2023c, Ye et al., 2024, Dai et al., 2024, Fan et al., 2023, Liu et al., 2024a]. Some studies further consider concept drift and robust forecasting under evolving distributions [Wen et al., 2023, Wang et al., 2024a]. Frequency-domain methods exploit periodicity and seasonality by transforming sequences into the spectral domain or applying decomposition techniques [Zhou et al., 2022b,a, Woo

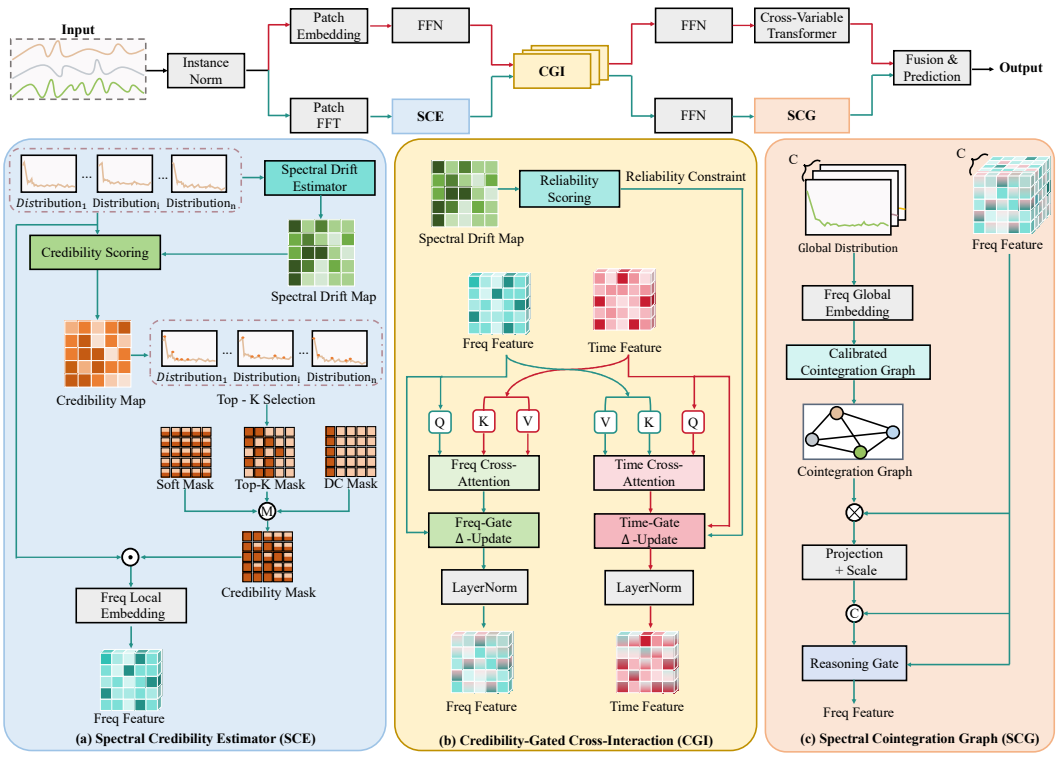


Figure 1: Illustration of the proposed TFNS framework. TFNS adopts a dual-stream time–frequency architecture, where SCE estimates local spectral credibility and purifies frequency representations, CGI performs reliability-aware bidirectional interaction between temporal and frequency streams, and SCG models stable cross-variable spectral relations before final fusion and prediction.

et al., 2022, Yang et al., 2023, Yi et al., 2023b, Chen et al., 2022]. Joint time–frequency methods integrate temporal and spectral representations through dual-branch architectures, fusion modules, or cross-domain interaction mechanisms. However, these methods often assume that local spectral evidence is equally reliable, and their cross-domain interaction usually lacks explicit reliability control. As a result, unstable spectral components may contaminate temporal representations and induce spurious cross-variable dependencies under non-stationary drift.

Overall, existing methods alleviate non-stationarity to some extent, but unified reliability-aware time–frequency modeling for multivariate long-horizon forecasting remains under-explored. In contrast, TFNS explicitly characterizes local spectral credibility, regulates time–frequency interaction accordingly, and models stable long-term cross-variable structures under dual non-stationarity.

3 Methodology

3.1 Problem Formulation

Given historical observations $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_I] \in \mathbb{R}^{C \times I}$ with C variables and input length I, the goal of multivariate time series forecasting is to predict $\hat{\mathbf{Y}} \in \mathbb{R}^{C \times O}$ that closely matches the ground-truth future sequence $\mathbf{Y} = [\mathbf{x}_{I+1}, \ldots, \mathbf{x}_{I+O}] \in \mathbb{R}^{C \times O}$, where O is the prediction length.

3.2 Overall Framework

As shown in Figure 1, TFNS adopts a dual-stream time–frequency architecture for $\mathbf{X} \in \mathbb{R}^{C \times I}$. After instance normalization, each variable is split into N patches of length P. The temporal stream embeds the patches and refines them with an FFN to obtain $\mathbf{H}^t \in \mathbb{R}^{C \times N \times d}$, where d is the embedding

dimension. In parallel, the frequency stream applies patch-wise Fourier analysis to obtain spectra $\mathbf{Z} \in \mathbb{C}^{C\times N\times F}$ and magnitudes $\mathbf{A} \in \mathbb{R}^{C\times N\times F}$, where $F = \lfloor P/2 \rfloor + 1$ is the number of frequency bins. For notational simplicity, we omit the variable index in SCE and CGI, and model cross-variable dependencies later using the Cross-Variable Transformer and SCG.

The local spectra are first processed by SCE, enabling spectral drift estimation, credibility scoring, and credibility-aware suppression of unreliable components to produce purified frequency representations $\mathbf{H}^f \in \mathbb{R}^{C\times N\times d}$. The purified frequency features are then sent to CGI together with the temporal features, enabling credibility-gated bidirectional interaction between the two streams. After CGI, the temporal branch is further refined by an FFN and a Cross-Variable Transformer (see Appendix B.1 for details) to enhance temporal and cross-variable modeling, while the frequency branch is processed by SCG to capture stable cross-variable spectral structures. Finally, the refined temporal and frequency representations are adaptively fused and fed into the prediction head to generate the forecasts.

3.3 Spectral Credibility Estimator

Given the patch-wise spectra $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ and magnitude spectra $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$ from the frequency stream, where $\mathbf{z}_n \in \mathbb{C}^F$ is the complex spectrum of the n-th patch and $\mathbf{a}_n = |\mathbf{z}_n| \in \mathbb{R}^F$ is its magnitude spectrum, the key challenge is to identify reliable frequency components under evolving spectral structures. Existing frequency-domain methods often treat local spectral evidence as equally reliable across patches. However, in non-stationary time series, frequency stability may vary over time, and directly using all local spectra can introduce noisy or unstable evidence into downstream modeling. To address this issue, we propose the Spectral Credibility Estimator (SCE), which evaluates local spectral credibility and purifies frequency representations before cross-domain interaction. As shown in Figure 1(a), SCE estimates spectral drift, computes credibility-aware scores, and applies a credibility mask before frequency local embedding.

Spectral Drift Estimation. To measure local spectral stability, we normalize each patch magnitude spectrum and estimate frequency-wise drift between adjacent normalized spectra:

$$\bar{\mathbf{a}}_n = \frac{\mathbf{a}_n}{\sum_{f=1}^{F} a_{n,f} + \epsilon}, \qquad \mathbf{d}_n = \phi(|\bar{\mathbf{a}}_n - \bar{\mathbf{a}}_{n-1}|)\,, \tag{1}$$

where ϵ is a small constant, and $\mathbf{d}_n \in \mathbb{R}^F$ is the frequency-wise drift vector of the n-th patch. Collecting all patch-level vectors gives the spectral drift map $\mathbf{D} = [\mathbf{d}_1; \ldots; \mathbf{d}_N] \in \mathbb{R}^{N\times F}$. The mapping $\phi(\cdot)$ standardizes the raw drift values over the patch–frequency plane within each variable instance and then applies a sigmoid transformation. Boundary replication is used for the first patch; details of this handling are provided in Appendix B.2.

Credibility Scoring. We compute a credibility-aware score for each frequency bin:

$$s_{n,f} = a_{n,f}(1 - \alpha d_{n,f}) + \beta\, b_f\, \mu_{d,n}\, \mu_{a,n}, \tag{2}$$

where α and β control drift penalization and low-frequency compensation, respectively; $\mu_{d,n} = \frac{1}{F}\sum_{f=1}^{F} d_{n,f}$ and $\mu_{a,n} = \frac{1}{F}\sum_{f=1}^{F} a_{n,f}$ denote the average drift and energy of the n-th patch; and $b_f = 1 - \frac{f-1}{F}$ is a low-frequency bias. The first term penalizes high-drift frequencies, while the second provides conservative low-frequency compensation to prevent low-frequency components from being collectively over-penalized under strong overall spectral variation. This yields a credibility map indicating reliable components for downstream modeling.

Credibility-Aware Spectral Purification. Based on the credibility scores, we select the Top-K reliable frequencies Ω_n and construct the credibility mask from three sources: a Top-K mask, a DC-preservation mask, and a soft-retention mask. Specifically, $m^{\text{top}}_{n,f} = 1$ if $f \in \Omega_n$ and 0 otherwise, $m^{\text{dc}}_{n,f} = 1$ for the zero-frequency component and 0 otherwise, and $\lambda \in [0, 1)$ is the soft-retention weight. The final mask is obtained by element-wise maximum and applied to the complex spectrum:

$$m_{n,f} = \max\Big(m^{\text{top}}_{n,f}, m^{\text{dc}}_{n,f}, \lambda\Big)\,, \qquad \hat{\mathbf{z}}_n = \mathbf{m}_n \odot \mathbf{z}_n. \tag{3}$$

Thus, reliable frequencies and the zero-frequency component are fully preserved, while unselected frequencies are softly retained rather than removed. Here, $\odot$ denotes element-wise multiplication.

This suppresses unreliable spectral components while preserving weak background information. Finally, the filtered spectrum is mapped into the latent space through frequency local embedding:

$$\mathbf{h}_n^f = \mathrm{Emb}_f^{\mathrm{loc}}([\mathrm{Re}(\hat{\mathbf{z}}_n)\,;\mathrm{Im}(\hat{\mathbf{z}}_n)])\,, \qquad \mathbf{H}^f = [\mathbf{h}_1^f, \ldots, \mathbf{h}_N^f] \in \mathbb{R}^{N\times d}. \tag{4}$$

Here, $\mathrm{Emb}_f^{\mathrm{loc}}(\cdot)$ maps the filtered spectrum into a d-dimensional latent representation. The purified frequency representations are fed into CGI for reliability-aware time–frequency interaction.

3.4 Credibility-Gated Cross-Interaction

While SCE purifies frequency representations at the source, effective forecasting still requires interaction between temporal and frequency streams. However, direct fusion may inject unstable spectral evidence into temporal representations and cause cross-domain contamination. To address this issue, we propose the Credibility-Gated Cross-Interaction (CGI) module, which enables reliability-aware bidirectional interaction between the two streams.

As shown in Figure 1(b), CGI takes the temporal representations $\mathbf{H}^t = [\mathbf{h}_1^t, \ldots, \mathbf{h}_N^t] \in \mathbb{R}^{N\times d}$ and the purified frequency representations $\mathbf{H}^f = [\mathbf{h}_1^f, \ldots, \mathbf{h}_N^f] \in \mathbb{R}^{N\times d}$ as input. It first scores patch-level reliability from spectral drift, then performs bidirectional cross-domain attention, and finally updates both streams through credibility-gated Δ-updates.

Patch-level Reliability Scoring. To introduce the spectral reliability estimated by SCE into cross-domain interaction, we aggregate frequency-wise stability into a patch-level reliability score:

$$\rho_n = \left(\frac{1}{F}\sum_{f=1}^{F}(1-d_{n,f})\right)^{\gamma}, \qquad n = 1, \ldots, N, \tag{5}$$

where $d_{n,f}$ is the spectral drift of frequency f in patch n, and $\gamma > 1$ sharpens reliability differences. Thus, ρ_n measures patch-level spectral reliability.

Bidirectional Cross-Domain Attention. CGI performs bidirectional cross-attention to exchange complementary information between the two streams:

$$\tilde{\mathbf{H}}^t = \mathrm{MHA}\big(\mathbf{H}^t, \mathbf{H}^f, \mathbf{H}^f\big)\,, \qquad \tilde{\mathbf{H}}^f = \mathrm{MHA}\big(\mathbf{H}^f, \mathbf{H}^t, \mathbf{H}^t\big)\,. \tag{6}$$

Here, $\tilde{\mathbf{H}}^t$ denotes the temporal representations enhanced by the frequency stream, while $\tilde{\mathbf{H}}^f$ denotes the frequency representations enhanced by the temporal stream.

Credibility-Gated Update. To avoid indiscriminate cross-domain injection, CGI computes adaptive gates from the original and cross-enhanced representations:

$$\mathbf{G}^t = \sigma\Big(\mathbf{W}_t[\mathbf{H}^t; \tilde{\mathbf{H}}^t] + \mathbf{b}_t\Big)\,, \qquad \mathbf{G}^f = \sigma\Big(\mathbf{W}_f[\mathbf{H}^f; \tilde{\mathbf{H}}^f] + \mathbf{b}_f\Big)\,. \tag{7}$$

Here, $\mathbf{G}^t$ and $\mathbf{G}^f$ control the injection strength from frequency to time and from time to frequency, respectively. Since unreliable spectral evidence may contaminate temporal modeling, the temporal gate is constrained by patch-level reliability during the update:

$$\mathbf{H}^{t'} = \mathrm{LN}\Big(\mathbf{H}^t + \min(\mathbf{G}^t, \boldsymbol{\rho}) \odot (\tilde{\mathbf{H}}^t - \mathbf{H}^t)\Big)\,, \qquad \mathbf{H}^{f'} = \mathrm{LN}\Big(\mathbf{H}^f + \mathbf{G}^f \odot (\tilde{\mathbf{H}}^f - \mathbf{H}^f)\Big)\,. \tag{8}$$

Here, $\mathrm{LN}(\cdot)$ denotes layer normalization, $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_N]$ is broadcast along the feature dimension, and $\min(\mathbf{G}^t, \boldsymbol{\rho})$ limits temporal absorption of unreliable spectral evidence. The term $\tilde{\mathbf{H}} - \mathbf{H}$ represents cross-domain incremental information, enabling gated residual updates rather than direct feature addition. For stacked CGI layers, outputs are fed into the next layer.

3.5 Spectral Cointegration Graph

Although CGI enables reliability-aware time–frequency interaction, cross-variable dependencies estimated from local frequency features may still be affected by non-stationary drift. To mitigate spurious relations, we propose the *Spectral Cointegration Graph* (SCG), which uses global spectral priors to calibrate cross-variable relations and refine frequency features through graph reasoning. As shown in Figure 1(c), SCG extracts global frequency embeddings, constructs a calibrated cointegration graph, and refines frequency features through graph propagation and reasoning gates.

Global Frequency Embedding. For each normalized sequence $\tilde{\mathbf{x}}_c \in \mathbb{R}^I$, we apply RFFT over the full historical window and embed the concatenated real and imaginary parts:

$$\mathbf{h}_c^{f,\mathrm{glo}} = \mathrm{Emb}_f^{\mathrm{glo}}([\mathrm{Re}(\mathrm{RFFT}(\tilde{\mathbf{x}}_c)) ; \mathrm{Im}(\mathrm{RFFT}(\tilde{\mathbf{x}}_c))]) , \qquad \bar{\mathbf{h}}^{f,\mathrm{glo}} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{h}_c^{f,\mathrm{glo}}. \tag{9}$$

Here, $\mathbf{h}_c^{f,\mathrm{glo}} \in \mathbb{R}^d$ captures the global spectrum of variable c, $\mathrm{Emb}_f^{\mathrm{glo}}(\cdot)$ is the frequency global embedding layer, and $\bar{\mathbf{h}}^{f,\mathrm{glo}} \in \mathbb{R}^d$ is the shared global context. Compared with patch-level spectra, this full-window context provides a more stable prior for graph calibration.

Calibrated Cointegration Graph. Given the CGI-enhanced frequency representation $\mathbf{H}_c^{f'} = [\mathbf{h}_{1,c}^{f'}, \ldots, \mathbf{h}_{N,c}^{f'}] \in \mathbb{R}^{N\times d}$, we first aggregate patch-level features into a variable-level spectral summary and estimate a local cointegration graph:

$$\mathbf{u}_c = \frac{1}{N}\sum_{n=1}^{N} \mathbf{W}_u \mathbf{h}_{n,c}^{f'}, \qquad A_{c,c'}^{\mathrm{loc}} = \sigma(\mathrm{MLP}_{\mathrm{coint}}([\mathbf{u}_c; \mathbf{u}_{c'}])) . \tag{10}$$

Here, $\mathbf{u}_c \in \mathbb{R}^d$ is the variable-level spectral summary, $\mathbf{W}_u \in \mathbb{R}^{d\times d}$ is a learnable projection, $A_{c,c'}^{\mathrm{loc}}$ denotes the local relation strength between c and c', and $\sigma(\cdot)$ denotes the sigmoid function.

Since short-term drift may affect local relations, SCG estimates pairwise corrections from the global spectral context and applies them with a conservative calibration strength:

$$\Delta_{c,c'} = \tanh\big(\mathrm{MLP}_{\Delta}\left([\mathbf{u}_c; \mathbf{u}_{c'}; \bar{\mathbf{h}}^{f,\mathrm{glo}}]\right)\big) , \qquad \mathbf{A} = \mathrm{Clip}\big(\mathbf{A}^{\mathrm{loc}} + \eta\boldsymbol{\Delta}, 0, 1\big) . \tag{11}$$

Here, $\boldsymbol{\Delta} \in \mathbb{R}^{C\times C}$ collects pairwise correction terms, η is an adaptive calibration strength, and $\mathrm{Clip}(\cdot)$ bounds graph weights into $[0, 1]$. Thus, the global spectral prior provides bounded correction rather than replacing local relations; details are given in Appendix B.3. We then symmetrize $\mathbf{A}$, add self-loops, and apply degree normalization to obtain the normalized propagation graph $\tilde{\mathbf{A}} \in \mathbb{R}^{C\times C}$; details are provided in Appendix B.4.

Graph-Based Frequency Refinement. For each patch n, we stack frequency features of all variables as $\mathbf{X}_n = [\mathbf{h}_{n,1}^{f'}, \ldots, \mathbf{h}_{n,C}^{f'}]^\top \in \mathbb{R}^{C\times d}$. SCG propagates cross-variable spectral information over the normalized cointegration graph and projects the propagated messages:

$$\mathbf{M}_n = s_m \, \tilde{\mathbf{A}}\mathbf{X}_n\mathbf{W}_m, \tag{12}$$

where $\mathbf{M}_n \in \mathbb{R}^{C\times d}$ is the propagated message, $\mathbf{W}_m \in \mathbb{R}^{d\times d}$ is the message projection matrix, and s_m is a learnable message scale used to control propagation strength and reduce over-smoothing.

To avoid overwriting the original frequency features, SCG uses a reasoning gate to adaptively combine original features and graph-propagated messages:

$$\boldsymbol{\Gamma}_n = \sigma(\mathbf{W}_g[\mathbf{X}_n; \mathbf{M}_n] + \mathbf{b}_g) , \qquad \mathbf{X}'_n = \mathrm{LN}(\boldsymbol{\Gamma}_n \odot \mathbf{X}_n + (1 - \boldsymbol{\Gamma}_n) \odot \mathbf{M}_n) . \tag{13}$$

Here, $\boldsymbol{\Gamma}_n \in \mathbb{R}^{C\times d}$ is the reasoning gate, $\mathbf{W}_g$ and $\mathbf{b}_g$ are learnable gate parameters, and $\mathrm{LN}(\cdot)$ denotes layer normalization. The gate controls how much graph-propagated information is absorbed while preserving the original frequency features. Collecting the refined features over all patches and variables yields the SCG-enhanced frequency representations $\mathbf{H}^{f''} \in \mathbb{R}^{C\times N\times d}$, which are used for final time–frequency fusion.

3.6 Time–Frequency Fusion and Prediction

After the preceding modules refine the temporal and frequency branches, TFNS obtains $\mathbf{H}^{t''}, \mathbf{H}^{f''} \in \mathbb{R}^{C\times N\times d}$. For each variable c, we concatenate its temporal and frequency representations and compute a nonlinear fusion feature together with adaptive branch weights:

$$\tilde{\mathbf{H}}_c = \mathrm{MLP}_{\mathrm{fus}}\Big([\mathbf{H}_c^{t''}; \mathbf{H}_c^{f''}]\Big), \qquad [w_c^t, w_c^f] = \mathrm{Softmax}\Big(\mathbf{W}_{\mathrm{fus}}[\mathbf{H}_c^{t''}; \mathbf{H}_c^{f''}] + \mathbf{b}_{\mathrm{fus}}\Big), \tag{14}$$

where $\mathbf{W}_{\mathrm{fus}} \in \mathbb{R}^{2d\times 2}$ and $\mathbf{b}_{\mathrm{fus}} \in \mathbb{R}^2$ are learnable fusion parameters (distinct from $\mathbf{W}_g$ and $\mathbf{b}_g$ in SCG), and $[w_c^t, w_c^f]$ are the adaptive weights for the temporal and frequency branches, respectively.

The fused representation combines nonlinear fusion and adaptive branch weighting:

$$\mathbf{H}_c^{\text{fus}} = \text{LN}\Big(\tilde{\mathbf{H}}_c + w_c^t \mathbf{H}_c^{t''} + w_c^f \mathbf{H}_c^{f''}\Big), \tag{15}$$

where layer normalization $\text{LN}(\cdot)$ is applied along the feature dimension.

Finally, collecting all variables gives $\mathbf{H}^{\text{fus}} \in \mathbb{R}^{C\times N\times d}$, and the prediction head with inverse instance normalization produces the forecast:

$$\hat{\mathbf{Y}} = \text{IN}^{-1}\Big(\text{Head}(\mathbf{H}^{\text{fus}})\Big). \tag{16}$$

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate the proposed TFNS on 11 widely used time-series forecasting datasets spanning multiple domains, including electricity (ETTh1, ETTh2, ETTm1, ETTm2, and Electricity), healthcare (ILI and Covid-19), transportation (Traffic), web traffic (Wike2000), environment (Weather), and banking (NN5). Detailed dataset statistics are provided in Appendix A.1.

Baselines. We select several state-of-the-art forecasting models as baselines, including DTAF Lu et al. [2026], Amplifier Fei et al. [2025], FITS Xu et al. [2023], iTransformer Liu et al. [2023a], Pathformer Chen et al. [2024], PatchTST Nie et al. [2022], ConvTimeNet Cheng et al. [2025], TimeMixer Wang et al. [2024b], TimesNet Wu et al. [2022], and DLinear Zeng et al. [2023]. These baselines cover diverse forecasting paradigms, including frequency-aware, time–frequency, Transformer-based, convolution-based, temporal-variation, MLP-based, and linear models. Detailed descriptions of the baselines are provided in Appendix A.2.

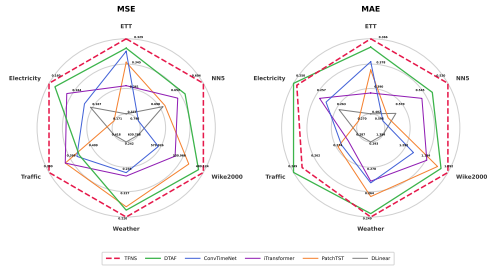


Figure 2: Model performance comparison in the time series forecasting task.

Settings. For the forecasting horizon, we use prediction lengths of 24, 36, 48, and 60 for the ILI, Covid-19, NN5, and Wike2000 datasets, and 96, 192, 336, and 720 for the remaining datasets. TFNS and all baselines use exactly the same search space and tuning budget for input-length selection. The candidate input lengths are $\{36, 104\}$ for ILI, Covid-19, NN5, and Wike2000, and $\{96, 336, 512\}$ for the remaining datasets. For each dataset and prediction length, we independently select the best input length for each method according to validation MSE. TFNS is trained with the Adam [Kingma and Ba, 2014] optimizer, and mean absolute error (MAE) and mean squared error (MSE) are used as evaluation metrics. All experiments are implemented on the TFB benchmark [Qiu et al., 2024] with PyTorch and conducted on a single NVIDIA RTX 4090 GPU. The reported results are averaged over three independent runs. Further analyses of computational efficiency and hyperparameter sensitivity are provided in Appendices A.3 and A.4.

4.2 Experimental Results

Figure 2 and Table 1 present a comprehensive comparison between TFNS and representative forecasting models. The radar charts in Figure 2 provide an intuitive overview of model performance across multiple datasets, with MSE shown on the left and MAE on the right. Different colored lines represent different models, and curves closer to the outer boundary indicate better relative performance. As shown in the figure, TFNS occupies the outermost region on both metrics across most datasets, suggesting strong overall forecasting accuracy and robustness.

The detailed results in Table 1 further support these observations. TFNS ranks first in 44 out of 55 MSE entries and 43 out of 55 MAE entries, demonstrating that its advantage is not limited to a few isolated cases but is consistently maintained across diverse datasets and prediction horizons. The improvement is particularly evident on challenging non-stationary datasets. Compared with DTAF,

Table 1: Full and average results of the multivariate forecasting task. "Avg." is the mean over all prediction lengths. The best results are highlighted in bold, and the second-best results are underlined.

Model		**TFNS (Ours)**		DTAF (2026)		Amplifier (2025)		ConvTimeNet (2025)		iTransformer (2024)		Pathformer (2024)		FITS (2024)		TimeMixer (2024)		PatchTST (2023)		TimesNet (2023)		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
ILI	24	1.704	**0.779**	1.733	0.784	1.947	0.877	**1.469**	0.800	1.783	0.846	2.086	0.922	2.182	1.002	1.804	0.820	1.932	0.872	2.131	0.958	2.208	1.031
	36	1.493	**0.773**	1.693	0.807	1.739	0.863	**1.450**	0.845	1.746	0.860	1.912	0.882	2.241	1.029	1.891	0.926	1.869	0.866	2.612	0.974	2.032	0.981
	48	1.671	0.809	1.614	**0.782**	1.806	0.907	**1.572**	0.875	1.716	0.898	1.985	0.905	2.272	1.036	1.752	0.866	1.891	0.883	1.916	0.897	2.209	1.063
	60	**1.632**	**0.816**	1.714	0.832	1.786	0.907	1.741	0.920	2.183	0.963	1.999	0.929	2.642	1.142	1.831	0.930	1.914	0.896	1.995	0.905	2.292	1.086
	Avg.	1.625	**0.794**	1.688	0.801	1.820	0.889	**1.558**	0.860	1.857	0.892	1.995	0.910	2.334	1.052	1.820	0.886	1.902	0.879	2.164	0.934	2.185	1.040
Covid-19	24	0.894	**0.030**	**0.885**	0.030	3.691	0.088	2.381	0.079	1.001	0.038	1.424	0.058	4.934	0.104	2.781	0.087	1.045	0.042	2.215	0.064	9.587	0.240
	36	**1.156**	**0.035**	1.159	0.037	4.149	0.100	2.873	0.092	1.236	0.042	3.109	0.085	4.935	0.113	2.877	0.093	1.397	0.051	2.531	0.081	5.333	0.173
	48	**1.462**	**0.041**	1.479	0.045	6.155	0.118	4.923	0.116	1.710	0.056	4.543	0.096	5.927	0.131	9.916	0.138	1.769	0.062	2.672	0.079	9.598	0.246
	60	**1.852**	0.055	1.883	**0.051**	8.317	0.144	5.495	0.131	2.005	0.062	4.227	0.105	6.880	0.145	36.191	0.215	2.216	0.068	3.018	0.085	7.778	0.297
	Avg.	**1.341**	**0.040**	1.352	0.041	5.578	0.112	3.918	0.104	1.488	0.050	3.326	0.086	5.669	0.123	12.941	0.133	1.607	0.056	2.609	0.077	8.074	0.239
NN5	24	**0.667**	**0.539**	0.716	0.559	0.918	0.685	0.744	0.585	0.727	0.568	0.769	0.602	0.870	0.663	0.723	0.723	0.740	0.596	0.739	0.579	0.752	0.594
	36	**0.608**	**0.522**	0.644	0.537	1.392	0.897	0.823	0.625	0.664	0.552	0.701	0.583	0.814	0.655	0.657	0.557	0.694	0.595	0.717	0.585	0.690	0.578
	48	**0.578**	**0.513**	0.621	0.535	2.196	1.170	0.734	0.594	0.633	0.543	0.668	0.573	0.780	0.644	0.630	0.550	0.667	0.585	0.647	0.558	0.665	0.573
	60	**0.564**	**0.507**	0.593	0.522	2.671	1.322	0.684	0.577	0.615	0.537	0.655	0.570	0.781	0.650	0.612	0.543	0.653	0.582	0.633	0.547	0.659	0.575
	Avg.	**0.604**	**0.520**	0.644	0.538	1.794	1.018	0.746	0.595	0.660	0.550	0.698	0.582	0.811	0.653	0.656	0.593	0.688	0.589	0.684	0.567	0.692	0.580
Wike2000	24	**420.258**	**0.960**	428.048	0.982	565.072	1.210	478.811	1.067	453.475	1.011	664.769	1.357	669.429	1.351	447.039	1.079	457.185	1.023	521.367	1.215	556.014	1.274
	36	**473.694**	**1.068**	483.089	1.094	612.685	1.294	584.305	1.228	515.830	1.132	699.269	1.431	711.479	1.440	510.073	1.221	511.943	1.115	506.339	1.201	613.963	1.355
	48	**500.069**	**1.136**	509.093	1.161	631.723	1.362	663.078	1.364	578.335	1.214	847.933	1.604	757.815	1.537	542.797	1.303	531.901	1.179	523.611	1.251	659.957	1.437
	60	**526.475**	**1.204**	545.186	1.257	857.566	1.602	611.973	1.350	634.947	1.402	942.510	1.717	791.442	1.615	574.808	1.402	554.829	1.244	559.841	1.329	693.140	1.510
	Avg.	**480.124**	**1.092**	491.354	1.124	666.762	1.367	584.542	1.252	545.647	1.190	788.620	1.527	732.541	1.486	518.679	1.251	513.965	1.140	527.790	1.249	630.768	1.394
ETTh1	96	**0.354**	**0.383**	0.359	0.384	0.371	0.392	0.372	0.397	0.386	0.405	0.372	0.392	0.376	0.396	0.372	0.401	0.377	0.397	0.389	0.412	0.379	0.403
	192	**0.390**	**0.411**	0.400	0.414	0.409	0.424	0.408	0.414	0.424	0.440	0.408	0.415	0.400	0.418	0.413	0.430	0.409	0.425	0.440	0.443	0.408	0.419
	336	**0.412**	**0.420**	0.415	0.424	0.443	0.448	0.412	0.430	0.449	0.460	0.438	0.434	0.419	0.435	0.438	0.450	0.431	0.444	0.523	0.487	0.440	0.440
	720	**0.418**	**0.446**	0.421	0.449	0.447	0.461	0.442	0.457	0.495	0.487	0.450	0.463	0.435	0.458	0.486	0.484	0.457	0.477	0.521	0.495	0.471	0.493
	Avg.	**0.393**	**0.415**	0.399	0.418	0.418	0.431	0.408	0.424	0.438	0.448	0.417	0.426	0.408	0.427	0.427	0.441	0.419	0.436	0.468	0.459	0.424	0.439
ETTh2	96	**0.270**	**0.325**	0.272	0.329	0.292	0.354	0.276	0.342	0.297	0.348	0.279	0.336	0.277	0.345	0.281	0.351	0.274	0.337	0.334	0.370	0.300	0.364
	192	**0.328**	**0.362**	0.330	0.367	0.346	0.390	0.329	0.380	0.372	0.403	0.345	0.380	0.331	0.379	0.349	0.387	0.348	0.384	0.404	0.413	0.387	0.423
	336	0.362	0.392	0.368	0.398	0.378	0.425	**0.324**	**0.377**	0.388	0.417	0.378	0.408	0.350	0.396	0.366	0.413	0.377	0.416	0.389	0.435	0.490	0.487
	720	0.389	0.427	0.406	0.434	0.402	0.440	0.392	0.432	0.424	0.444	0.437	0.455	**0.382**	**0.425**	0.401	0.436	0.406	0.441	0.434	0.448	0.704	0.597
	Avg.	0.337	**0.377**	0.344	0.382	0.355	0.402	**0.330**	0.383	0.370	0.403	0.360	0.395	0.335	0.386	0.349	0.397	0.351	0.394	0.390	0.416	0.470	0.468
ETTm1	96	**0.274**	**0.329**	0.282	0.330	0.313	0.359	0.292	0.345	0.300	0.353	0.290	0.335	0.303	0.345	0.293	0.345	0.289	0.343	0.340	0.378	0.300	0.345
	192	**0.324**	**0.356**	0.325	0.358	0.350	0.378	0.329	0.368	0.341	0.380	0.337	0.363	0.337	0.365	0.335	0.372	0.329	0.368	0.392	0.404	0.336	0.366
	336	**0.354**	**0.372**	0.357	0.376	0.376	0.392	0.363	0.390	0.374	0.396	0.374	0.384	0.368	0.384	0.368	0.386	0.362	0.390	0.423	0.426	0.367	0.386
	720	**0.405**	**0.411**	0.407	0.415	0.459	0.439	0.427	0.428	0.429	0.430	0.428	0.416	0.420	0.413	0.426	0.417	0.416	0.423	0.475	0.453	0.419	0.416
	Avg.	**0.339**	**0.367**	0.343	0.370	0.375	0.392	0.353	0.383	0.361	0.390	0.357	0.374	0.357	0.377	0.356	0.380	0.349	0.381	0.408	0.415	0.356	0.378
ETTm2	96	**0.158**	**0.246**	0.161	0.246	0.184	0.271	0.167	0.257	0.175	0.266	0.164	0.250	0.165	0.254	0.165	0.256	0.165	0.255	0.189	0.265	0.164	0.255
	192	**0.215**	0.287	0.215	**0.284**	0.248	0.312	0.222	0.295	0.242	0.312	0.219	0.288	0.219	0.291	0.225	0.298	0.221	0.293	0.254	0.310	0.224	0.304
	336	0.269	**0.317**	0.269	0.318	0.321	0.362	0.276	0.329	0.282	0.337	**0.267**	0.319	0.272	0.326	0.277	0.332	0.276	0.327	0.313	0.345	0.277	0.337
	720	**0.352**	**0.375**	0.367	0.384	0.374	0.400	0.358	0.381	0.375	0.394	0.361	0.377	0.359	0.381	0.360	0.387	0.362	0.381	0.413	0.402	0.371	0.401
	Avg.	**0.248**	**0.306**	0.253	0.308	0.282	0.336	0.256	0.316	0.268	0.327	0.253	0.308	0.254	0.313	0.257	0.318	0.256	0.314	0.292	0.330	0.259	0.324
Weather	96	**0.144**	**0.182**	0.145	0.185	0.146	0.201	0.154	0.204	0.157	0.207	0.148	0.195	0.172	0.225	0.147	0.198	0.149	0.196	0.168	0.214	0.170	0.230
	192	**0.188**	**0.227**	0.190	0.228	0.190	0.240	0.200	0.248	0.200	0.248	0.191	0.235	0.215	0.261	0.192	0.243	0.191	0.239	0.219	0.262	0.216	0.273
	336	**0.238**	**0.268**	0.241	0.269	0.241	0.281	0.252	0.288	0.252	0.287	0.243	0.274	0.261	0.295	0.247	0.284	0.242	0.279	0.278	0.302	0.258	0.307
	720	**0.309**	**0.318**	0.313	0.321	0.313	0.330	0.328	0.337	0.320	0.336	0.318	0.326	0.326	0.341	0.318	0.330	0.312	0.330	0.353	0.351	0.323	0.362
	Avg.	**0.220**	**0.249**	0.222	0.251	0.222	0.263	0.233	0.269	0.232	0.270	0.225	0.258	0.244	0.280	0.226	0.264	0.223	0.261	0.254	0.282	0.242	0.293
Electricity	96	**0.129**	0.230	0.131	0.225	0.141	0.238	0.134	0.232	0.134	0.230	0.135	**0.222**	0.139	0.237	0.153	0.256	0.143	0.247	0.169	0.271	0.140	0.237
	192	0.152	0.246	**0.147**	**0.235**	0.164	0.257	0.158	0.254	0.154	0.250	0.157	0.253	0.154	0.250	0.168	0.269	0.158	0.260	0.180	0.280	0.154	0.251
	336	**0.164**	**0.251**	0.165	0.255	0.188	0.280	0.165	0.259	0.169	0.265	0.170	0.267	0.170	0.268	0.189	0.291	0.168	0.267	0.204	0.304	0.169	0.268
	720	0.196	**0.278**	0.200	0.284	0.204	0.293	0.205	0.293	**0.194**	0.288	0.211	0.302	0.212	0.304	0.228	0.320	0.214	0.307	0.205	0.304	0.204	0.301
	Avg.	**0.160**	0.251	0.161	**0.250**	0.174	0.267	0.166	0.260	0.163	0.258	0.168	0.261	0.169	0.265	0.184	0.284	0.171	0.270	0.189	0.290	0.167	0.264
Traffic	96	**0.361**	0.242	0.370	**0.232**	0.399	0.283	0.376	0.265	0.363	0.265	0.384	0.250	0.400	0.280	0.369	0.257	0.370	0.262	0.595	0.312	0.395	0.275
	192	**0.379**	0.250	0.392	**0.243**	0.413	0.287	0.392	0.271	0.384	0.273	0.405	0.257	0.412	0.288	0.400	0.272	0.386	0.269	0.613	0.322	0.407	0.280
	336	**0.391**	0.256	0.408	**0.252**	0.424	0.295	0.405	0.277	0.396	0.277	0.424	0.265	0.426	0.301	0.407	0.272	0.396	0.275	0.626	0.332	0.417	0.286
	720	**0.428**	**0.267**	0.441	0.269	0.458	0.311	0.436	0.294	0.445	0.308	0.452	0.283	0.478	0.339	0.461	0.316	0.435	0.295	0.635	0.340	0.454	0.308
	Avg.	**0.390**	0.254	0.403	**0.249**	0.424	0.294	0.402	0.277	0.397	0.281	0.416	0.264	0.429	0.302	0.409	0.279	0.397	0.275	0.617	0.326	0.418	0.287
1st Count		**44**	**43**	2	9	0	0	6	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0

Table 2: Ablation study of TFNS components. Results are averaged over all horizons.

Model	**Full TFNS**		Module						SCE						CGI								SCG					
			w/o SCE		w/o CGI		w/o SCG		w/o Drift		w/o Soft		w/o DC		w/o T→F		w/o F→T		w/o Rel.		w/o Gate		w/o Glob.		w/o RG		w/o Coint.	
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	MSE	MAE
ETTh1	**0.393**	**0.415**	0.403	0.424	0.401	0.422	0.405	0.426	0.400	0.422	0.398	0.420	0.396	0.418	0.398	0.420	0.397	0.419	0.400	0.421	0.399	0.420	0.401	0.422	0.398	0.420	0.403	0.424
Weather	**0.220**	**0.249**	0.227	0.255	0.228	0.257	0.226	0.254	0.225	0.253	0.224	0.252	0.222	0.251	0.225	0.254	0.224	0.253	0.227	0.256	0.226	0.255	0.224	0.252	0.223	0.251	0.225	0.253
Traffic	**0.390**	**0.254**	0.412	0.271	0.406	0.267	0.418	0.275	0.407	0.267	0.402	0.264	0.397	0.259	0.401	0.263	0.399	0.261	0.404	0.266	0.402	0.264	0.410	0.269	0.404	0.264	0.414	0.272
NN5	**0.604**	**0.520**	0.615	0.530	0.612	0.528	0.617	0.533	0.612	0.528	0.610	0.526	0.607	0.523	0.610	0.526	0.609	0.525	0.611	0.527	0.610	0.526	0.613	0.529	0.609	0.525	0.615	0.531

TFNS reduces the average MSE/MAE by 6.2%/3.3% on NN5 and 2.3%/2.8% on Wike2000. Since DTAF is a strong baseline designed for non-stationary forecasting, these gains verify the effectiveness of TFNS in capturing evolving temporal and spectral dynamics. Overall, these results show that TFNS delivers consistently strong performance across diverse forecasting scenarios.

4.3 Ablation Studies

Overall module ablation. We first remove one core module at a time from the full model, resulting in three degraded variants: w/o SCE, w/o CGI, and w/o SCG. The corresponding results are reported in Table 2. Removing any one of the three modules consistently degrades the performance, demonstrating that all of them contribute positively to the final forecasting results. Specifically, w/o SCE removes spectral credibility estimation and frequency-domain purification, w/o CGI removes credibility-aware bidirectional time–frequency interaction, and w/o SCG removes frequency-domain cointegration graph reasoning. Overall, these results indicate that the advantage of TFNS does not come from any single design alone, but from the synergy of spectral purification, credibility-aware cross-domain interaction, and spectral cointegration graph reasoning with global calibration.

Effect of Spectral Credibility Estimator. We analyze the internal design of SCE by considering three degraded variants: w/o Drift, w/o Soft, and w/o DC in Table 2. All variants underperform the full model. Removing spectral drift modeling weakens the model's ability to distinguish stable and unstable frequency components, showing explicitly modeling local spectral variation is necessary for frequency-domain purification. Removing the soft-retention mask degrades performance, indicating softly retaining unselected frequencies is more effective than hard filtering for preserving useful information. Removing zero-frequency preservation harms performance, suggesting the zero-frequency trend is important for maintaining robust frequency representations under drastic local spectral varia-

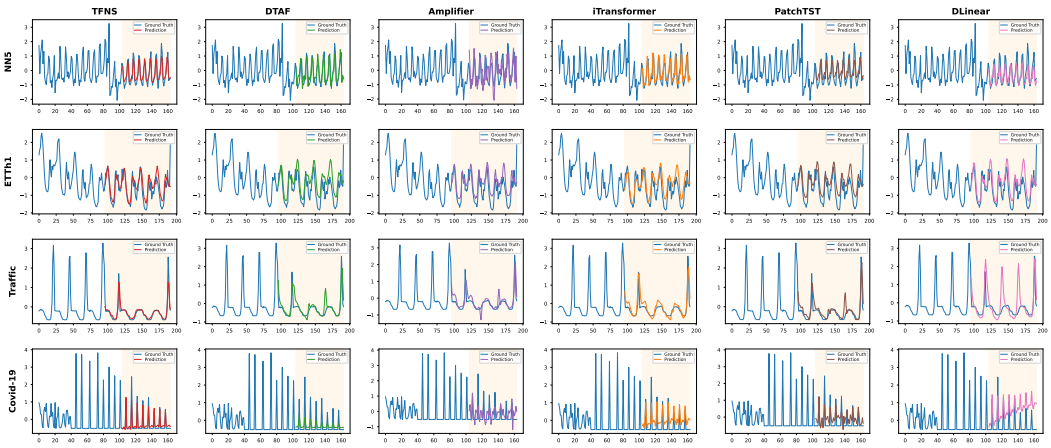


Figure 3: Prediction visualization on four representative datasets. Covid-19 and NN5 use input-104-prediction-60, while ETTh1 and Traffic use input-96-prediction-96.

tion. These results show SCE's effectiveness comes from the joint effect of spectral drift modeling, soft retention, and zero-frequency preservation, rather than simple frequency selection alone.

Impact of Credibility-Gated Interaction. For CGI, we consider four degraded variants: w/o T→F, w/o F→T, w/o Rel., and w/o Gate in Table 2. All variants perform worse than the full model. Removing either interaction direction causes clear drops, indicating one-way interaction is insufficient to exploit complementarity between temporal and frequency domains. Removing the reliability constraint further degrades performance, showing temporal updates should be bounded by spectral credibility. Removing the gating mechanism also weakens performance, suggesting gated updates are necessary to control cross-domain information. Overall, CGI's advantage comes from the coordinated effect of bidirectional interaction, reliability-aware constraint, and gated updates.

Effect of Spectral Cointegration Graph. We study three degraded SCG variants: w/o Glob., w/o RG, and w/o Coint. in Table 2. Removing any of these degrades performance. Removing global spectral calibration increases vulnerability to local noise and non-stationary drift, showing that the global context guides local relation calibration. Removing the reasoning gate also hurts performance, as graph-propagated messages should not overwrite original frequency features and adaptive gating balances original representations with graph reasoning. Replacing the calibrated cointegration graph with a simpler relation graph also harms performance, showing TFNS's gain comes not only from graph propagation but also from modeling spectral cointegration relations. Overall, SCG's effectiveness arises from the joint contribution of global calibration, reasoning gate, and spectral cointegration modeling.

4.4 Visualization of Prediction Results

To provide an intuitive comparison, we visualize the predictions of TFNS and representative baselines on four datasets, namely NN5, ETTh1, Traffic, and Covid-19, as shown in Fig. 3. Overall, TFNS produces forecasts that better align with the ground truth under different sequence patterns, including periodic fluctuations, sharp peaks, and local trend changes.

Specifically, on NN5 and ETTh1, TFNS better preserves the phase and amplitude of periodic variations, while several baselines show phase shifts, amplitude bias, or over-smoothing. On Traffic and Covid-19, TFNS more accurately tracks sharp peaks and local variations, whereas competing methods often underestimate peaks or produce misaligned forecasts. These results further verify the effectiveness of TFNS in modeling both periodic structures and non-stationary variations.

4.5 Limitations

TFNS involves several hyperparameters, including the input length, patch length, and number of CGI blocks. Although the selected settings are effective, the optimal configuration may vary across datasets with different sampling frequencies and non-stationary patterns. Future work could explore adaptive hyperparameter selection to improve robustness and generalization.

5 Conclusion

In this paper, we propose TFNS, a time–frequency framework for non-stationary multivariate time series forecasting. TFNS progressively improves forecasting robustness through three components: SCE evaluates patch-level spectral credibility and suppresses unreliable frequency components; CGI enables reliability-aware bidirectional interaction between temporal and frequency representations; and SCG models stable spectral cross-variable relations with conservative global calibration. Together, these designs reduce the influence of unstable spectral evidence and spurious dependencies under non-stationary drift. Experiments on 11 benchmark datasets demonstrate that TFNS achieves robust and accurate forecasting across diverse real-world scenarios.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 62503448, in part by the Centrally Guided Local Science and Technology Development Special Fund of Anhui Province under Grant 202407a12020009, in part by the Fundamental Research Funds for the Central Universities under Grant WK2100250063, and in part by the National Key R&D Program of China under Grant 2022YFB3206904. The authors declare no competing interests.

References

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. *arXiv preprint arXiv:2402.05956*, 2024.

Weiqi Chen, Wenwei Wang, Bingqing Peng, Qingsong Wen, Tian Zhou, and Liang Sun. Learning to rotate: Quaternion transformer for complicated periodical time series forecasting. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 146–156, 2022.

Mingyue Cheng, Jiqian Yang, Tingyue Pan, Qi Liu, Zhi Li, and Shijin Wang. Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 171–180, 2025.

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.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7522–7529, 2023.

Jingru Fei, Kun Yi, Wei Fan, Qi Zhang, and Zhendong Niu. Amplifier: Bringing attention to neglected low-energy components in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 11645–11653, 2025.

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.

Jiaxi Hu, Yuehong Hu, Wei Chen, Ming Jin, Shirui Pan, Qingsong Wen, and Yuxuan Liang. Attractor memory for long-term time series forecasting: A chaos perspective. *Advances in Neural Information Processing Systems*, 37:20786–20818, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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, 2024.

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*, 2024a.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022.

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*, 2023a.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36: 12271–12290, 2023b.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. *arXiv preprint arXiv:2402.02368*, 2024b.

Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Advances in Neural Information Processing Systems*, 36:14273–14292, 2023c.

Jiecheng Lu, Xu Han, Yan Sun, and Shihao Yang. Cats: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. *arXiv preprint arXiv:2403.01673*, 2024.

Junkai Lu, Peng Chen, Chenjuan Guo, Yang Shu, Meng Wang, and Bin Yang. Towards non-stationary time series forecasting with temporal stabilization and frequency differencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24070–24078, 2026.

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

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.*, 17(9):2363–2377, 2024.

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*, pages 1185–1196, 2025.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, and Jianxin Liao. Rethinking the power of timestamps for robust time series forecasting: A global-local fusion perspective. *Advances in Neural Information Processing Systems*, 37:22206–22232, 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.

Qingsong Wen, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, Tieniu Tan, et al. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. *Advances in Neural Information Processing Systems*, 36:69949–69980, 2023.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting (2022). *URL https://arxiv. org/abs/2202.01381*, 2022.

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. *arXiv preprint arXiv:2307.03756*, 2023.

Wang Xue, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. Card: Channel aligned robust blend transformer for time series forecasting. *arXiv preprint arXiv:2305.12095*, 2023.

Fuhao Yang, Xin Li, Min Wang, Hongyu Zang, Wei Pang, and Mingzhong Wang. Waveform: Graph enhanced wavelet learning for long sequence forecasting of multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 10754–10761, 2023.

Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. *Advances in Neural Information Processing Systems*, 37: 31350–31379, 2024.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in neural information processing systems*, 36:69638–69660, 2023a.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36:76656–76679, 2023b.

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, pages 11121–11128, 2023.

Jiawen Zhang, Shun Zheng, Xumeng Wen, Xiaofang Zhou, Jiang Bian, and Jia Li. Elastst: Towards robust varied-horizon forecasting with elastic time-series transformer. *Advances in Neural Information Processing Systems*, 37:119174–119197, 2024.

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*, pages 27268–27286. PMLR, 2022b.

A Experimental Details

A.1 Dataset

We evaluate TFNS on 11 widely used multivariate time series forecasting datasets from diverse domains, including electricity and energy (ETTh1, ETTh2, ETTm1, ETTm2, Electricity), health (ILI and Covid-19), transportation (Traffic), web traffic (Wike2000), environment (Weather), and banking (NN5). These datasets cover different sampling frequencies, temporal patterns, numbers of variables, and degrees of non-stationarity, providing a comprehensive test bed for evaluating the robustness and generalization ability of TFNS. Detailed dataset statistics are reported in Table 3.

Table 3: Statistics of the 11 multivariate time-series forecasting datasets used in our experiments.

Dataset	Domain	Frequency	Timestamps	Channels	Split	Description
ILI	Health	1 week	966	7	7:1:2	Weekly influenza-like illness indicators from the CDC.
Covid-19	Health	1 day	1,392	948	7:1:2	Daily COVID-19 related records for epidemic dynamics analysis.
NN5	Banking	1 day	791	111	7:1:2	Daily cash withdrawals from ATMs in the UK.
Wike2000	Web	1 day	792	2,000	7:1:2	Daily page views of 2,000 Wikipedia pages.
ETTh1	Electricity	1 hour	14,400	7	6:2:2	Electricity transformer dataset with seven indicators, such as oil temperature and useful load.
ETTh2	Electricity	1 hour	14,400	7	6:2:2	Electricity transformer dataset with seven indicators such as oil temperature and useful load.
ETTm1	Electricity	15 mins	57,600	7	6:2:2	Electricity transformer dataset sampled every 15 minutes.
ETTm2	Electricity	15 mins	57,600	7	6:2:2	Electricity transformer dataset sampled every 15 minutes.
Weather	Environment	10 mins	52,696	21	7:1:2	Weather dataset containing 21 meteorological indicators.
Electricity	Electricity	1 hour	26,304	321	7:1:2	Hourly electricity consumption records in kWh from 2012 to 2014.
Traffic	Traffic	1 hour	17,544	862	7:1:2	Road occupancy rates collected by 862 sensors on San Francisco Bay area freeways.

A.2 Baselines

We compare TFNS with several representative and state-of-the-art forecasting models under the same experimental protocol. The selected baselines cover different modeling paradigms, including frequency-aware modeling, time–frequency modeling, Transformer-based modeling, convolution-based modeling, temporal-variation modeling, MLP-based modeling, and linear modeling.

DTAF Lu et al. [2026] is a recent time–frequency forecasting model designed to enhance forecasting performance by jointly exploiting temporal and frequency-domain information.

Amplifier Fei et al. [2025] improves forecasting performance by strengthening the learning of low-frequency components, which are often underrepresented in standard spectral modeling.

FITS Xu et al. [2023] is a frequency-domain forecasting model that uses spectral interpolation to capture periodic patterns and long-term trends.

iTransformer Liu et al. [2023a] applies attention over inverted tokens, treating variables as tokens to better capture cross-variable dependencies in multivariate time series.

Pathformer Chen et al. [2024] employs a multi-scale Transformer architecture with adaptive pathways to integrate information from different temporal resolutions.

PatchTST Nie et al. [2022] segments time series into patches and models patch-level temporal dependencies with a Transformer backbone, reducing the effective sequence length while preserving local temporal patterns and improving the efficiency of long-sequence forecasting.

ConvTimeNet Cheng et al. [2025] is a convolution-based forecasting model that captures temporal dependencies through hierarchical convolutional structures, providing an efficient alternative to attention-based sequence modeling.

TimeMixer Wang et al. [2024b] is an MLP-based forecasting model that performs multi-scale temporal mixing without relying on self-attention, making it an efficient baseline for long-term forecasting.

TimesNet Wu et al. [2022] models multi-periodic temporal variations by transforming a one-dimensional time series into a two-dimensional representations and applying temporal 2D-variation modeling to capture complex intra-period and inter-period patterns.

DLinear Zeng et al. [2023] is a simple yet strong linear baseline that decomposes the time series into trend and seasonal components for forecasting, offering a competitive reference for assessing the benefits of more expressive nonlinear architectures.

A.3 Efficiency Analysis

Table 4 compares four time series forecasting models on the Weather and ETTh1 datasets across four prediction horizons (96, 192, 336, and 720) in terms of inference time and parameter count. It can be observed that our model maintains strong parameter efficiency overall: among the eight settings, it achieves the smallest parameter count in five cases, the second smallest in two cases, and the third smallest in one case. Meanwhile, on ETTh1, our model consistently ranks second in inference speed across all prediction horizons, indicating that it preserves competitive inference efficiency while keeping the model size well controlled. In comparison, Pathformer generally achieves faster inference but usually at the cost of a larger parameter count. TimesNet attains smaller parameter counts in several cases, yet its inference speed is overall less competitive, and its parameter count increases substantially on Weather. Crossformer, although showing relatively stable inference time, consistently requires a much larger number of parameters. Overall, these results demonstrate that our model achieves a favorable trade-off between inference time and parameter cost.

Table 4: Efficiency analysis. TIME denotes inference time in seconds, and SIZE denotes the number of parameters in millions.

Models		**TFNS(Ours)**		**Pathformer**		**TimesNet**		**Crossformer**	
Metrics		TIME	SIZE	TIME	SIZE	TIME	SIZE	TIME	SIZE
Weather	96	3.533	**0.856M**	**2.244**	5.396M	3.557	37.521M	2.452	2.287M
	192	3.548	1.260M	2.152	5.655M	**1.843**	**1.203M**	2.433	11.364M
	336	3.606	**1.699M**	**2.307**	6.042M	3.501	9.421M	2.565	2.208M
	720	3.653	**3.423M**	**2.900**	7.074M	31.943	9.459M	4.236	11.787M
ETTh1	96	1.527	**0.291M**	**1.078**	0.933M	1.994	0.605M	1.603	2.686M
	192	1.496	**0.529M**	**1.219**	1.000M	2.432	0.614M	2.059	11.336M
	336	1.467	0.818M	**1.062**	1.097M	2.598	**0.628M**	2.187	11.408M
	720	1.478	1.534M	**1.375**	1.353M	2.813	**0.666M**	4.075	11.600M

Table 5 further reports per-epoch training time, training FLOPs, and inference FLOPs, together with average inference time and parameter count. The results are averaged over four prediction horizons (96, 192, 336, and 720), with FLOPs reported in TFLOPs. TFNS achieves the lowest parameter count and training and inference FLOPs on Weather, and ranks second on these metrics on ETTh1. Although Pathformer achieves shorter training and inference times on both datasets, TFNS maintains competitive computational efficiency alongside its forecasting performance.

Table 5: Average computational cost on Weather and ETTh1 over four prediction horizons (96, 192, 336, and 720). Training time is measured per epoch. Parameters are reported in millions, and FLOPs in TFLOPs. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	**Metric** ↓	**TFNS**	**Pathformer**	**TimesNet**	**Crossformer**
Weather	Training time (s/epoch)	47.238	**28.844**	130.117	34.818
	Inference time (s)	3.585	**2.401**	10.211	2.922
	Parameters (M)	**1.810**	6.042	14.401	6.912
	Training FLOPs (T)	**121.909**	125.421	5367.988	505.353
	Inference FLOPs (T)	**11.013**	11.600	499.840	47.123
ETTh1	Training time (s/epoch)	14.543	**8.621**	26.837	20.971
	Inference time (s)	1.492	**1.184**	2.459	2.481
	Parameters (M)	0.793	1.096	**0.628**	9.258
	Training FLOPs (T)	2.920	**0.775**	76.025	58.941
	Inference FLOPs (T)	0.307	**0.084**	7.740	6.246

A.4 Hyperparameter Analysis

We analyze the sensitivity of TFNS to the input length, patch length, number of CGI blocks, and the drift-penalty and low-frequency-compensation coefficients. Unless otherwise specified, the two

coefficients are set to $\alpha = 0.5$ and $\beta = 1.0$. The complete dataset- and horizon-specific configurations are provided in the released scripts.

Effect of Input Length. The input length determines how much historical information is available for forecasting. Intuitively, longer input sequences provide richer temporal context and help capture long-term dependencies. We evaluate TFNS with input lengths ranging from 48 to 512. As shown in Fig. 4, the prediction errors generally decrease as the input length increases, indicating that incorporating longer historical information improves forecasting performance. Moreover, TFNS consistently achieves the lowest error across almost all input lengths, demonstrating its strong ability to effectively exploit long-term temporal dependencies.

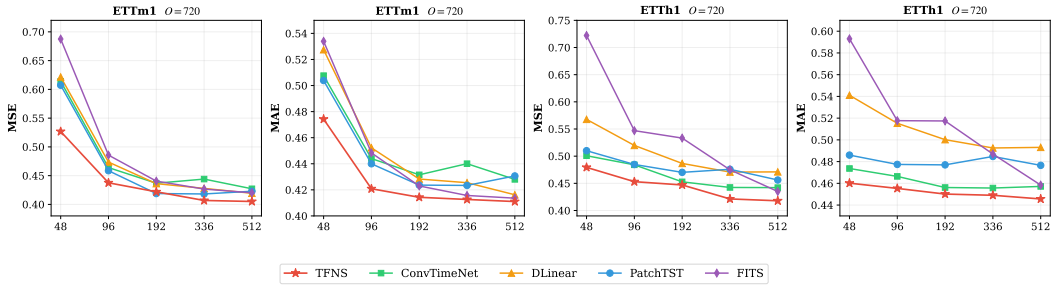


Figure 4: Input length analysis on ETTh1 and ETTm1 with forecasting horizon $O = 720$. The prediction errors decrease as the input length increases. TFNS consistently achieves the lowest MSE and MAE across different input lengths, demonstrating its effectiveness in exploiting long-term temporal dependencies.

Effect of Patch Length. The patch length controls the granularity of local temporal and spectral representations. Smaller patches capture fine-grained variations but may yield unstable spectral estimates, while larger patches provide broader context but can over-smooth local non-stationary patterns. We evaluate patch lengths of 8, 16, 32, and 48 on ETTh1 and Weather. As shown in Figure 5, the optimal patch length varies across datasets and prediction horizons. ETTh1 shows horizon-dependent preferences, while Weather generally favors a moderate patch length, with $P = 16$ performing best in most horizons. These results suggest that proper patch segmentation is important for balancing local sensitivity and spectral stability in TFNS.

Effect of the Number of CGI Blocks. The number of CGI blocks determines how many rounds of time–frequency interaction are performed in TFNS. As shown in Table 6, the optimal number of CGI blocks is both dataset- and horizon-dependent. For ETTh1, short-term horizons (96) perform best with 1 block, medium horizons (192, 336) benefit from 4 blocks, and the long horizon (720) performs best with 2 blocks. In contrast, for Weather, short-term (96) and long-term (720) horizons benefit from 4 blocks, but medium horizons show different behavior: horizon 192 is optimal with 4 blocks, while horizon 336 achieves the lowest MSE with only 1 block. These results indicate that shallow interaction may suffice for some datasets and horizons, whereas deeper cross-domain refinement is necessary for others.

Effect of α and β. We evaluate the drift-penalty coefficient α and the low-frequency-compensation coefficient β on ETTh1 and Weather. We vary α over $\{0.25, 0.50, 0.75, 1.00\}$ with $\beta = 1.0$, and vary β over $\{0.50, 1.00, 1.50, 2.00\}$ with $\alpha = 0.5$, keeping all other configurations unchanged. Table 7 reports the average MSE and MAE over four prediction horizons (96, 192, 336, and 720). The default setting achieves the lowest MSE and MAE on both datasets, while neighboring values yield comparable results, suggesting that TFNS is not highly sensitive to small changes in these coefficients within the tested ranges.

A.5 Ablation of SCE Scoring Factors

To examine the contributions of the SCE scoring factors, we compare Full SCE with Energy Only, Stability Only, w/o Low-frequency Compensation, and w/o Drift Penalty on ETTh1, Weather, Traffic,

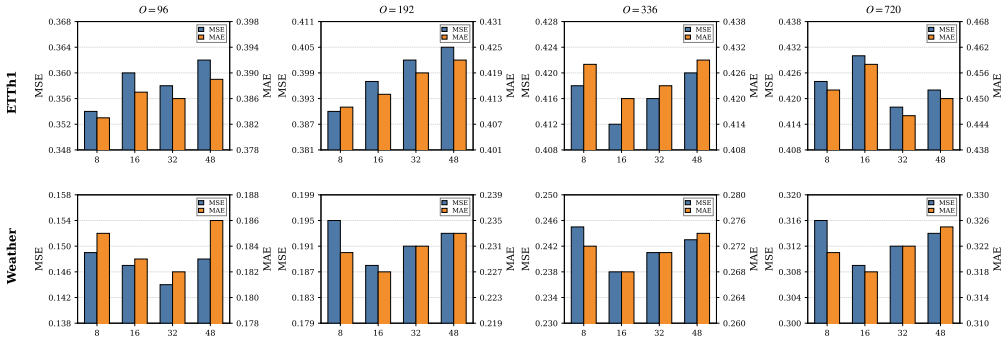


Figure 5: Sensitivity analysis of patch length on ETTh1 and Weather under different prediction horizons. We report MSE and MAE for patch lengths of 8, 16, 32, and 48.

Table 6: Effect of the number of CGI blocks on ETTh1 and Weather. Here, B_{CGI} denotes the number of stacked CGI blocks. The best results for each prediction length are highlighted in bold.

Number		$B_{\mathrm{CGI}}=1$		$B_{\mathrm{CGI}}=2$		$B_{\mathrm{CGI}}=3$		$B_{\mathrm{CGI}}=4$	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.354**	**0.383**	0.357	0.385	0.359	0.387	0.360	0.388
	192	0.396	0.414	0.394	0.413	0.393	0.412	**0.390**	**0.411**
	336	0.418	0.425	0.416	0.423	0.415	0.422	**0.412**	**0.420**
	720	0.421	0.447	**0.418**	**0.445**	0.419	0.446	0.420	0.446
Weather	96	0.148	0.186	0.147	0.185	0.145	0.183	**0.144**	**0.182**
	192	0.193	0.232	0.191	0.230	0.190	0.229	**0.188**	**0.227**
	336	**0.238**	**0.268**	0.241	0.271	0.243	0.273	0.240	0.270
	720	0.314	0.323	0.312	0.321	0.311	0.320	**0.309**	**0.318**

Table 7: Sensitivity to the drift-penalty coefficient α and low-frequency-compensation coefficient β on ETTh1 and Weather. Results are averaged over four prediction horizons. The best results within each parameter sweep are highlighted in bold.

Setting		α			β		
Metrics		Value	MSE	MAE	Value	MSE	MAE
ETTh1		0.25	0.395	0.416	0.50	0.395	0.416
		0.50	**0.393**	**0.415**	1.00	**0.393**	**0.415**
		0.75	0.394	0.417	1.50	0.394	0.417
		1.00	0.397	0.423	2.00	0.399	0.425
Weather		0.25	0.221	0.250	0.50	0.221	0.250
		0.50	**0.220**	**0.249**	1.00	**0.220**	**0.249**
		0.75	0.221	0.251	1.50	0.223	0.250
		1.00	0.222	0.251	2.00	0.225	0.254

and NN5, keeping all other configurations unchanged. Table 8 reports the average MSE and MAE over the four forecasting horizons of each dataset. Full SCE achieves the lowest MSE and MAE on all four datasets. These results support the complementary roles of the three scoring factors.

B Model Implementation Details

B.1 Cross-Variable Transformer

After the temporal branch outputs the intermediate representation $\mathbf{H}^{t'}$, we refine the temporal features to capture inter-variable dependencies using a Cross-Variable Transformer.

Table 8: Ablation of SCE scoring factors on ETTh1, Weather, Traffic, and NN5. Results are averaged over the four forecasting horizons of each dataset. The best results are highlighted in bold.

Variant	ETTh1		Weather		Traffic		NN5	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Stability Only	0.402	0.422	0.223	0.250	0.408	0.268	0.613	0.528
Energy Only	0.399	0.418	0.225	0.253	0.404	0.265	0.611	0.529
w/o Low-frequency Compensation	0.394	0.416	0.223	0.252	0.400	0.261	0.609	0.524
w/o Drift Penalty	0.397	0.418	0.224	0.251	0.392	0.256	0.607	0.522
Full SCE	**0.393**	**0.415**	**0.220**	**0.249**	**0.390**	**0.254**	**0.604**	**0.520**

For each patch n, let $\bar{\mathbf{X}}_n^t = [\bar{\mathbf{h}}_{n,1}^t, \ldots, \bar{\mathbf{h}}_{n,C}^t]^\top \in \mathbb{R}^{C\times d}$ denote the temporal features of all variables. The cross-variable refinement is applied as:

$$\mathbf{X}_n^{t''} = \mathrm{LN}\Big(\bar{\mathbf{X}}_n^t + \mathrm{MHA}_{\mathrm{var}}(\bar{\mathbf{X}}_n^t, \bar{\mathbf{X}}_n^t, \bar{\mathbf{X}}_n^t)\Big), \tag{17}$$

followed by a feed-forward network and layer normalization:

$$\mathbf{X}_n^{t''} = \mathrm{LN}\Big(\mathbf{X}_n^{t''} + \mathrm{FFN}_{\mathrm{var}}(\mathbf{X}_n^{t''})\Big). \tag{18}$$

Collecting all patches yields the refined temporal representation $\mathbf{H}^{t''} \in \mathbb{R}^{C\times N\times d}$. This design ensures that inter-variable dependencies are explicitly modeled at each patch level, before proceeding to subsequent modules such as CGI and SCG.

B.2 First Patch Handling in Spectral Drift Estimation

In the Spectral Credibility Estimator (SCE), the spectral drift for each patch is computed as the absolute difference between adjacent normalized spectra:

$$\mathbf{d}_n = \phi\big(|\bar{\mathbf{a}}_n - \bar{\mathbf{a}}_{n-1}|\big), \quad n = 1, \ldots, N, \tag{19}$$

where $\bar{\mathbf{a}}_n$ is the normalized magnitude spectrum of the n-th patch.

For the first patch, there is no previous patch to compute the difference. To handle this, we use boundary replication by setting the first patch drift equal to that of the second patch, i.e., $\mathbf{d}_1 = \mathbf{d}_2$. This ensures that all patches have valid drift vectors and that the subsequent standardization in $\phi(\cdot)$ over the patch–frequency drift map is well-defined, avoiding any zero-variance or division-by-zero issues. Consequently, the first patch is aligned with the local spectral context, and the spectral drift estimation remains consistent across the entire sequence.

B.3 Adaptive Calibration Strength

In SCG, the calibration strength controls how much the global spectral prior adjusts the local cointegration graph. Given the local variable-level spectral summaries $\{\mathbf{u}_c\}_{c=1}^C$ and the global frequency embeddings $\{\mathbf{h}_c^{f,\mathrm{glo}}\}_{c=1}^C$, we first compute the averages:

$$\mathbf{u}_{\mathrm{avg}} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{u}_c, \qquad \bar{\mathbf{h}}^{f,\mathrm{glo}} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{h}_c^{f,\mathrm{glo}}. \tag{20}$$

A local–global matching weight is computed as

$$r_{\mathrm{match}} = \sigma\Big(\mathrm{MLP}_\eta([\mathbf{u}_{\mathrm{avg}}; \bar{\mathbf{h}}^{f,\mathrm{glo}}])\Big), \tag{21}$$

where $r_{\mathrm{match}} \in [0, 1]$ measures consistency between local summaries and the global spectral prior.

We further estimate the informativeness of the global spectral prior:

$$v_{\mathrm{g}} = \frac{1}{d}\sum_{j=1}^{d}\mathrm{Var}_c(h_{c,j}^{f,\mathrm{glo}}), \qquad r_{\mathrm{info}} = \sigma(a v_{\mathrm{g}} - b), \tag{22}$$

where a and b scale the informativeness score to $[0, 1]$.

Finally, the effective adaptive calibration strength is computed as

$$\eta_{\text{eff}} = \text{Clamp}\Big(\eta_0 \cdot r_{\text{match}} \cdot r_{\text{info}}, 0, \eta_{\max}\Big), \tag{23}$$

where η_0 is a learnable base strength, $r_{\text{match}} \cdot r_{\text{info}}$ acts as an adaptive weight, and $\eta_{\max}$ is a hard upper bound preventing excessive correction.

B.4 Graph Normalization and Self-Loop

After calibration, the adjacency matrix is symmetrized:

$$\mathbf{A}^{\text{sym}} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^{\top}), \tag{24}$$

and self-loops are added:

$$\mathbf{A}^{\text{loop}} = \mathbf{A}^{\text{sym}} + \text{ReLU}(\omega)\mathbf{I}, \tag{25}$$

where ω is a learnable self-loop weight and $\mathbf{I}$ is the identity matrix.

Finally, degree normalization is applied:

$$\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}^{\text{loop}}\mathbf{D}^{-1/2}, \qquad D_{c,c} = \sum_{c'=1}^{C} A^{\text{loop}}_{c,c'}. \tag{26}$$

Here, $\mathbf{D}$ is the degree matrix, and $\tilde{\mathbf{A}}$ is used for graph-based frequency propagation in SCG.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the proposed TFNS framework, its three main components, and the scope of the empirical evaluation. The claims are supported by the methodology and experimental results presented in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations associated with hyperparameter sensitivity, including the input sequence length, patch length, and the number of CGI modules. It also points out that adaptive hyperparameter selection remains a promising direction for future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions.
   - The authors should reflect on the scope of the claims made.
   - The authors should reflect on the factors that influence the performance of the approach.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present theoretical theorems or formal proofs. The main contributions are methodological and empirical.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material.

- Any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper?

   Answer: [Yes]

   Justification: The paper provides the model architecture, benchmark datasets, forecasting horizons, evaluation metrics, implementation environment, and key hyperparameter settings. Additional dataset and baseline details are provided in the appendix and reproducibility materials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Reproducibility can be accomplished in various ways, including architecture details, instructions, code, data, or checkpoints.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments are conducted on publicly available benchmark datasets, and the source code, data preparation instructions, configuration files, and reproduction scripts are provided in the supplemental material. These materials include sufficient instructions to reproduce the main experimental results reported in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer.
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - The authors should provide instructions on data access and preparation.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines.
   - At submission time, to preserve anonymity, the authors should release anonymized versions if applicable.
   - Providing as much information as possible in supplemental material is recommended.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the benchmark datasets, data splits, prediction horizons, input-length selection, evaluation metrics, optimizer, implementation framework, and hardware environment used in the experiments. These details are sufficient to understand the reported results.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports results averaged over three independent runs, but does not include error bars, confidence intervals, or statistical significance tests. We omit them due to the high computational cost of repeatedly training all baselines across all datasets and forecasting horizons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests.
   - The factors of variability that the error bars are capturing should be clearly stated.
   - The method for calculating the error bars should be explained.
   - The assumptions made should be given.
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it.
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the main computational environment used for the experiments. All experiments are implemented with PyTorch on the TFB benchmark and conducted on a single NVIDIA RTX 4090 GPU with 48 GB memory, which provides a practical reference for reproducing the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The research uses public time-series benchmark datasets and does not involve human subjects, private personal data, or high-risk deployment. We have reviewed the NeurIPS Code of Ethics and do not identify any deviation.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the potential benefits of improved time-series forecasting in domains such as energy, transportation, finance, and public health. It also acknowledges that inaccurate forecasts in high-stakes applications may lead to suboptimal decisions and should be used with appropriate domain validation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses, fairness considerations, privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, when it gives incorrect results, and from misuse.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, pre-trained language models, image generators, or scraped datasets with substantial misuse potential. Therefore, special release safeguards are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards.
    - Datasets that have been scraped from the Internet could pose safety risks.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets used in the paper properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper uses publicly available benchmark datasets and baseline methods, and credits the corresponding sources through citations. Dataset sources and usage information are provided in the experimental section, appendix, or supplemental reproducibility materials.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license should be included for each asset.
- For scraped data from a particular source, the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper releases source code, configuration files, data preparation instructions, and reproduction scripts as supplemental materials. These materials are documented to support reproduction of the main experimental results while preserving anonymity at submission time.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets if applicable.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects. All experiments are conducted on existing time-series benchmark datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board approvals were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, crowdsourcing, or user studies. Therefore, IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval may be required for human subjects research.

- We recognize that the procedures for this may vary significantly between institutions and locations.
- For initial submissions, do not include any information that would break anonymity.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as an important, original, or non-standard component. Any use of language tools for writing or editing, if any, does not affect the methodology, experiments, scientific rigor, or originality of the work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to the LLM policy in the NeurIPS handbook for what should or should not be described.