GLNET: HIERARCHICAL GLOBAL-LOCAL MODELING FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term time series forecasting requires models that can capture long-range dependencies while maintaining computational efficiency. Although Transformer-based approaches have achieved strong performance, their quadratic complexity and potential overfitting issues limit scalability, while lightweight linear models often struggle to capture complex temporal dynamics. To address this gap, we propose Global-Local Network (GLNet), a unified yet branch-specialized architecture that integrates global sequence modeling with local temporal refinement. The core of GLNet is the Global-Local Block (GLB), a dimension-agnostic operator that combines pointwise linear projections for global temporal mixing with convolutions for local pattern extraction. GLNet adopts a decomposition-based dual-flow design: the Seasonal Flow applies hierarchical patch-wise GLB modeling to capture local waveform variations and long-range periodic evolution, while the Trend Flow employs an activation-free global trend operator to model smooth long-term dynamics. Extensive experiments on eight real-world datasets demonstrate that GLNet achieves leading mean forecasting accuracy while maintaining faster inference speed and lower memory consumption compared with existing approaches.

1 INTRODUCTION

Long-term time series forecasting (LTSF) plays a pivotal role in numerous real-world applications, ranging from energy management and weather forecasting to traffic flow analysis and economic indicator prediction (Shao et al., 2022; Castán-Lascorz et al., 2022; Qiu et al., 2025b; 2024). However, capturing the intricate temporal dependencies and handling the inherent non-stationarity of real-world data remains a formidable challenge.

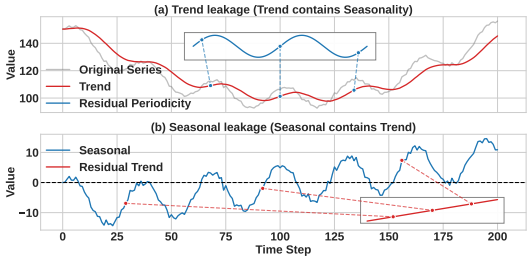


Figure 1: Component leakage in time series decomposition: Trend component contains seasonal residuals, and seasonal component contains trend residuals. The decomposition is performed using Exponential Moving Average (EMA).

In recent years, Transformer-based architectures have achieved remarkable success in LTSF, primarily due to the attention mechanism's ability to model long-range dependencies (Vaswani et al., 2017). Advanced models such as Pathformer (Chen et al., 2024) and TimeXer (Wang et al., 2024c) further demonstrate this efficacy. However, these methods typically suffer from quadratic computational complexity and high memory consumption with respect to the lookback window. Furthermore, the data-hungry nature of Transformers often leads to overfitting, particularly on datasets with limited samples or significant noise.

To mitigate these computational bottlenecks, there has been a resurgence of interest in simpler, more efficient architectures. Linear models and Convolutional Neural Networks (CNNs) have emerged as strong competitors, offering compelling performance with lower latency. For instance, DUET (Qiu et al., 2025a) and TimeMixer++ (Wang et al., 2025) demonstrate that properly designed lightweight

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.

structures can rival complex Transformers. However, these approaches face a fundamental dilemma: Pure Linear models generally capture only global statistical properties (e.g., trend and periodicity) while lacking the capacity to perceive local fine-grained variations. Conversely, standard CNNs are constrained by limited receptive fields, and expanding them via deep stacking that often leads to optimization difficulties. Consequently, a critical gap remains: **How can we achieve both global semantic expansion and fine-grained local modeling simultaneously while maintaining high computational efficiency?**

To address the challenge of non-stationarity, recent studies have adopted sequence decomposition as a core strategy. For instance, trend-seasonal decomposition is demonstrated in xPatch (Stitsyuk & Choi, 2025) and ETSformer (Woo et al., 2022b), while frequency-domain and modal decomposition approaches are explored in works such as Times2D (Nematirad et al., 2025) and CoST (Woo et al., 2022a). Although effective, such decomposition-based methods often overlook a subtle but critical issue: Decomposition is inherently imperfect. As illustrated in Figure 1, the extracted trend component often retains residual high-frequency fluctuations, while the seasonal component may contain traces of underlying trends. Rigidly assigning separate architectures to each component fails to account for this information leakage. Pure Multilayer Perceptrons (MLPs) may miss the residual fluctuations in the trend, while pure CNNs may struggle with the residual global drift in the seasonal part. Therefore, a robust framework requires a unified architecture capable of adaptively handling both global and local features, regardless of the decomposition branch.

In this paper, we propose the Global-Local Network (**GLNet**), a unified dual-stream architecture constructed upon a dimension-agnostic building block, the Global-Local Block (GLB). Unlike standard operators that rely solely on dense connections or fixed kernels, the GLB decouples feature processing by sequentially applying a linear projection for global semantic expansion and a depth-wise convolution for local context refinement.

Building on this foundational unit, we devise a customized dual-stream strategy. For the Seasonal Flow, we implement a hierarchical patching mechanism that transforms 1D temporal signals into 2D grids, enabling the GLB to capture multi-scale periodic variations. Conversely, the Trend Flow adopts a streamlined variation of the GLB operating on whole-series embeddings, prioritizing the preservation of global evolutionary trajectories. Finally, the processed representations are integrated via a linear fusion layer to generate the forecast. In summary, the key contributions of our work comprise:

- We identify the decomposition leakage problem in existing dual-stream methods and address it through a unified backbone design. Our approach employs specialized variations of the GLB to robustly handle the imperfect separation of trend and seasonal components.
- We propose GLNet, a lightweight framework that achieves leading mean forecasting accuracy in LTSF. We enable the simultaneous modeling of global dependencies and local details with high computational efficiency, significantly reducing the computational overhead compared to Transformer-based architectures.
- Extensive experiments on widely used real-world benchmarks demonstrate that GLNet consistently outperforms existing Transformer-based and MLP/CNN-based baselines in both accuracy and computational efficiency.

2 Related Work

Transformer-based Models. Recent Transformer-based approaches for time series forecasting mainly focus on improving attention inductive bias and modeling efficiency. MTST (Zhang et al., 2024) adopts multi-branch structures to capture temporal patterns at different resolutions, while Fredformer (Piao et al., 2024) mitigates frequency-domain bias by reweighting spectral components to avoid low-frequency dominance. Meanwhile, several studies revisit the role of self-attention in temporal modeling. CATS (Kim et al., 2024) demonstrates that cross-attention alone can model historical-to-future dependencies with reduced complexity, whereas SAMformer (Ilbert et al., 2024) improves generalization by addressing optimization issues in deep self-attention through a lightweight design and sharpness-aware minimization.

Other lines of work aim to enhance Transformer representations via auxiliary structures or alternative tokenizations. Sequence Complementor (Chen et al., 2025b) introduces learnable complemen-

tary sequences to enrich input representations, while HDT (Feng et al., 2025) employs hierarchical discrete token modeling for efficient high-dimensional forecasting. More recently, large-scale pretraining paradigms have been explored. Timer (Liu et al., 2024b) formulates forecasting as a generative pretraining task on heterogeneous data, and Time-MoE (Shi et al., 2025) further improves scalability via a mixture-of-experts mechanism. Together, these works indicate a shift toward scalable and representation-aware Transformer designs for time series forecasting.

Patching-based hybrid architectures. Recent studies explore patching as an alternative paradigm for LTSF by segmenting input sequences into local temporal patches, thereby reducing computing length and facilitating long-range dependency modeling. PatchTST (Nie et al., 2023) demonstrates that fixed-length patches can effectively serve as Transformer tokens, while Crossformer (Zhang & Yan, 2023) further enhances this idea by explicitly modeling cross-dimension dependencies in multivariate settings. Departing from attention-centric designs, CARD (Wang et al., 2024b) partitions input time series into fixed-length patches, mapping each patch to a token via MLP with positional encoding, thus converting the sequence into a format suitable for Transformers. U-Mixer (Ma et al., 2024) divides the sequence into patches and employs linear mappings with positional encoding to effectively capture local temporal patterns. More recently, xpatch (Stitsyuk & Choi, 2025) utilizes patching technology within a Non-linear Stream, primarily for processing the seasonal components of the data.

While Transformer-based models excel in capturing long-range dependencies, their quadratic complexity incurs high computational costs and optimization difficulties. Recent patching and mixer-based architectures improve efficiency, yet they often rely on fixed patch interactions or simplified MLP structures. So we propose GLNet, which seamlessly integrates linear projections for global semantic expansion and depth-wise convolutions for local feature extraction. This design enables multi-scale temporal modeling without the heavy computational burden of self-attention, achieving an optimal balance between effectiveness and efficiency.

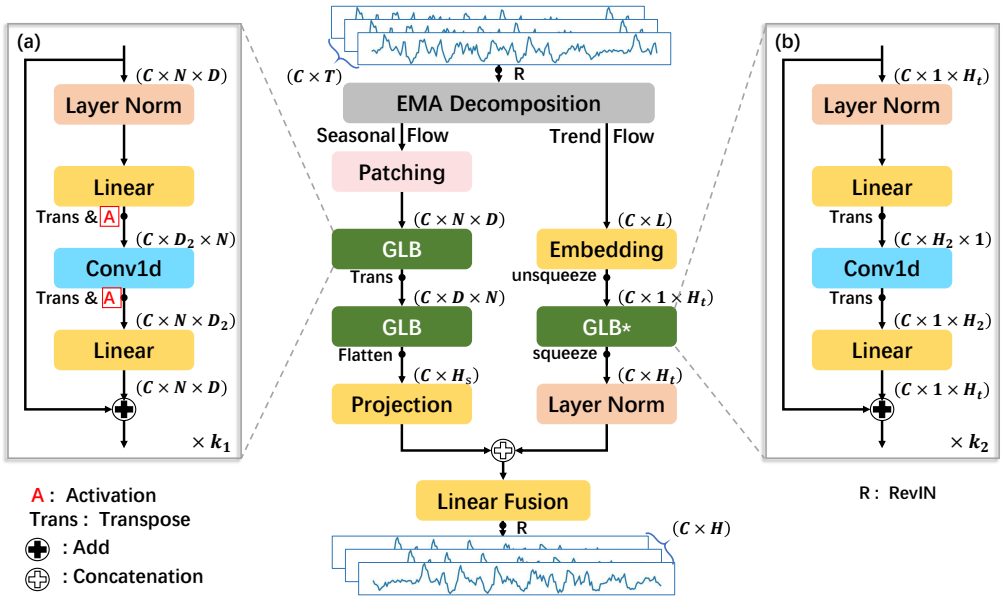


Figure 2: GLNet Model Overview. (a) GLB block. (b) GLB* block.

3 METHOD

In both univariate and multivariate LTSF tasks, given historical observations $X = \{x_1, ..., x_T\} \in \mathbb{R}^{T \times C}$ with T time steps and C variates, where the i-th variate is denoted as $x^{(i)}$, we aim to predict the future H time steps $Y = \{x_{T+1}, ..., x_{T+H}\} \in \mathbb{R}^{H \times C}$.

3.1 STRUCTURE OVERVIEW

The overall framework of our model is illustrated in Figure 2. To address the inherent non-stationarity of real-world time series, we utilize Reversible Instance Normalization (RevIN) (Kim et al., 2022) as a preprocessing step to alleviate distribution shifts. After normalizing the sequence, we decompose the input sequence into trend and seasonal components via EMA decomposition. The trend component is processed by the Trend Flow and the seasonal component by the Seasonal Flow. Their outputs are fused, and finally, the inverse RevIN transformation is applied to generate the final predictions.

EMA Decomposition. Most models utilize Simple Moving Average (SMA) to decompose the original series into trend and seasonal components. However, SMA requires boundary padding and may distort the sequence at the head and tail (Stitsyuk & Choi, 2025). We employ EMA to avoid these issues. See Appendix E for details.

While decomposition is a powerful strategy, it is practically imperfect: high-frequency noise often contaminates the trend component, while global drift permeates the seasonal part. To robustly handle this inevitable information leakage, we propose a unified dual-stream framework. Both streams share a common foundation GLB to ensure structural harmony during feature fusion, yet they are specialized to target distinct frequency characteristics. Specifically, we implement a Seasonal Flow with hierarchical patching for periodic micro-dynamics and a Trend Flow with global serialization for low-frequency evolution.

3.2 GLB

The GLB serves as the unified and dimension-agnostic computational unit in our model. Regardless of the semantic context, the GLB consistently treats the last dimension as the sequence dimension and the second-to-last dimension as the feature dimension.

Structurally, the GLB adopts a residual, dimension-preserving design:

$$\mathbf{X}^{(l)} = \mathbf{X}^{(l-1)} + \mathcal{F}\left(\text{LayerNorm}(\mathbf{X}^{(l-1)})\right) \tag{1}$$

where the mapping function $\mathcal{F}$ decouples the processing of dimensions to capture complementary patterns, which includes the following two operators.

- **Linear for Global Sequence Mixing:** Two pointwise linear layers impose global interactions on the sequence dimension. Both layers mix information along the sequence dimension, thereby modeling long-range dependencies across the entire sequence index.
- **Conv1d for Local Feature Refinement:** A depthwise 1D convolution with kernel size k is applied to the feature dimension N. This captures local correlations between adjacent features.

The entire operation of $\mathcal{F}$ can be summarized as:

$$\mathcal{F}(\mathbf{X}) = \text{Dropout}\left(\mathcal{T}^{-1}\left(\phi(\mathcal{T}(\sigma(\mathbf{X}\mathbf{W}_1)))\right)\mathbf{W}_2\right) \tag{2}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are linear weights acting on dimension D, σ is the activation, $\mathcal{T}$ represents the transpose operation to swap N and D, and ϕ denotes the depthwise convolution sliding over N. This design ensures that the model simultaneously learns global structural evolution (via Linear) and local feature continuity (via Conv).

3.3 SEASONAL FLOW

The Seasonal Flow is designed to capture high-frequency periodic patterns and their evolution over time. It adopts a channel-independent strategy, treating each variate of the input series $\mathbf{X}_{sea} \in \mathbb{R}^{B\times T\times C}$ as a separate sample. The data flow undergoes a hierarchical transformation process utilizing k_1 layers of GLB to extract features from orthogonal dimensions.

Patching. First, we reshape the input to $\mathbb{R}^{(B\times C)\times T}$ to merge the batch and variate dimensions. Adopting the patching mechanism (Nie et al., 2023), we unfold the temporal dimension T into a 2D

grid of shape $\mathbb{R}^{(B\times C)\times N\times P}$, where P is the patch length (representing intra-patch high-frequency details) and N is the number of patches (representing inter-patch temporal evolution). This 2D representation allows us to decouple local waveform distinctness from global periodic recurrence.

Intra-Patch Modeling. This stage aims to capture the shape and details of local waveforms. We feed the tensor $\mathbf{Z}_{intra} \in \mathbb{R}^{(B\times C)\times N\times P}$ into stacked GLB layers. In this configuration, the last dimension is P. Consequently, the GLB's linear projection globally mixes the information within each patch, learning a latent representation of the local waveform shape. Simultaneously, the GLB's convolution slides along the N dimension, smoothing the connection between adjacent patches to ensure temporal continuity. This process effectively encodes the micro-structure of the time series.

Inter-Patch Modeling. To model the long-term evolution of seasonality, we transpose the feature tensor to $\mathbf{Z}_{inter} \in \mathbb{R}^{(B\times C)\times P\times N}$. Now, the last dimension corresponds to the patch index N. We apply another stack of GLB layers to this transformed view. Here, the GLB's linear layer acts on the full sequence history N, capturing global dependencies and long-range periodic trends across the entire lookback window. The convolution operation now slides along the P dimension, refining the features by aggregating information from adjacent time steps within a patch. This stage captures the macro-dynamics of the series.

Projection. After hierarchical processing, the feature maps are flattened and projected via a linear layer to generate the seasonal forecast component $\hat{\mathbf{Y}}_{sea} \in \mathbb{R}^{(B\times C)\times H_s}$.

3.4 Trend Flow

Complementary to the Seasonal Flow, the Trend Flow focuses on the low-frequency, slowly evolving dynamics of the time series. Since trend components are smooth and globally correlated, localized patching methods may fragment the holistic context. Therefore, we design this stream to operate on the entire historical sequence simultaneously, utilizing k_2 layers of a activation-free variant of the GLB, termed GLB*.

Embedding. We first reshape the channel-independent trend input to $\mathbb{R}^{(B\times C)\times T}$. Instead of slicing the sequence, we employ a linear embedding to project the entire time dimension T into a latent trend representation H_t. To align this 2D tensor with the 3D input requirement of our unified GLB*, we perform an `unsqueeze` operation, reshaping the data to $\mathbb{R}^{(B\times C)\times 1\times H_t}$. Here, the dimension 1 serves as a pseudo-spatial dimension, while H_t contains the encoded global history.

GLB*: Degenerate Processing. The transformed tensor is processed by stacked GLB* layers. Unlike the Seasonal Flow, we intentionally remove the activation functions (σ) within these blocks to prevent aggressive non-linear transformations, aligning with the smooth nature of trend components. The **Linear Layer** operates on the last dimension H_t. Since this dimension represents the compressed information of the entire lookback window, the linear projection performs global context mixing, allowing the model to analyze the holistic trend direction. The **Conv1d** operates on the second-to-last dimension (which is 1). In this degenerate case ($kernel = 1, stride = 1$), the convolution ceases to extract neighbor dependencies and effectively functions as a pointwise constraint. This structural design enforces the model to rely solely on the global embedding for forecasting, filtering out any high-frequency noise that might be interpreted as local patterns.

Layer Norm. We apply Layer Normalization to the trend representations $\mathbb{R}^{(B\times C)\times H_t}$. By standardizing the feature distribution, we prevent the trend stream from overshadowing the seasonal details, ensuring numerically stable and balanced feature integration in the subsequent fusion stage. While we remove activation functions to avoid complex non-linear deformations, LayerNorm inherently introduces a mild non-linearity via mean and variance normalization. This design is theoretically crucial: it prevents consecutive GLB* layers from mathematically collapsing into a single affine transformation, thereby justifying the depth of the Trend Flow ($k_2 \geq 1$) and providing sufficient expressiveness to model complex but smooth global drifts without overfitting.

3.5 Linear Fusion

Finally, to synthesize the decoupled representations, we employ a static learned weighting strategy. We concatenate the outputs from the Seasonal Flow ($\hat{\mathbf{Y}}_{sea}$) and the Trend Flow ($\hat{\mathbf{Y}}_{trend}$) along the temporal dimension, resulting in a $\mathbf{Z}_{fuse} \in \mathbb{R}^{(B\times C)\times(H_t+H_s)}$. This concatenated feature is

Table 1: Long-term forecasting results. The lookback window is searched to the best for a fairer comparison. All results are averaged across four different forecasting horizons: $T \in \{96, 192, 336, 720\}$. The lower value of MSE (or MAE) is the better performance. The best performance is highlighted in **red and bold**, and the second best is marked with blue and underline. See Table 6 in Appendix for full results.

Models	GLNet (Ours)		xPatch (2025)		SimpleTM (2025a)		CARD (2024b)		U-Mixer (2024)		PDF (2024)		FITS (2024)		iTransformer (2024a)		TimeMixer (2024a)		PatchTST (2023)		MICN (2023)		DLinear (2023)		TimesNet (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	MSE	MAE
Weather	**0.211**	**0.248**	0.215	0.248	0.231	0.268	0.224	0.253	0.224	0.253	0.226	0.297	0.231	0.270	0.235	0.271	0.227	0.266	0.225	0.262	0.238	0.288	0.243	0.298	0.257	0.284
Traffic	0.389	0.264	0.396	**0.251**	0.387	0.272	0.385	0.252	0.416	0.268	0.393	0.268	0.417	0.287	**0.384**	0.268	0.402	0.276	0.394	0.270	0.496	0.312	0.413	0.288	0.619	0.331
Electricity	**0.153**	**0.247**	0.154	0.247	0.162	0.259	0.159	0.252	0.169	0.268	0.158	0.252	0.166	0.261	0.164	0.259	0.175	0.271	0.165	0.262	0.182	0.290	0.170	0.267	0.201	0.302
ETTh1	**0.386**	**0.408**	0.395	0.412	0.422	0.437	0.409	0.419	0.425	0.429	0.410	0.429	0.407	0.424	0.452	0.445	0.432	0.442	0.413	0.427	0.419	0.447	0.430	0.441	0.487	0.470
ETTh2	**0.300**	**0.356**	0.308	0.360	0.345	0.391	0.323	0.369	0.368	0.394	0.343	0.388	0.334	0.384	0.379	0.409	0.362	0.405	0.337	0.385	0.473	0.470	0.456	0.459	0.411	0.431
ETTm1	**0.343**	**0.369**	0.345	0.370	0.373	0.394	0.358	0.373	0.356	0.379	0.343	0.375	0.356	0.377	0.364	0.392	0.364	0.387	0.350	0.381	0.355	0.383	0.357	0.379	0.413	0.419
ETTm2	**0.236**	**0.300**	0.242	0.302	0.264	0.326	0.245	0.303	0.276	0.321	0.249	0.312	0.249	0.312	0.272	0.329	0.263	0.318	0.249	0.314	0.280	0.346	0.271	0.337	0.302	0.339
Solar	**0.185**	**0.213**	0.199	0.224	0.190	0.252	0.207	0.227	0.211	0.250	0.200	0.263	0.230	0.267	0.214	0.265	0.203	0.269	0.200	0.284	0.253	0.301	0.239	0.300	0.211	0.281
1st count	7	7	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

then passed through a learnable linear projection layer $\mathcal{W}_{fuse} \in \mathbb{R}^{(H_t+H_s)\times H}$ to generate the final prediction $\hat{\mathbf{Y}}$. Unlike simple additive reconstruction, this mechanism allows the model to learn static weighting contributions of high-frequency seasonal patterns and low-frequency trend evolution for each future time step H, optimizing the global forecasting accuracy.

The clear structural distinction between the two flows effectively mitigates the leakage problem highlighted in the Introduction. Handling leakage requires opposing strategies based on the physical properties of the signals. The Trend Flow must reject high-frequency seasonal noise to prevent severe overfitting; thus, simplifying GLB* acts as a strict low-pass filter forcing the network to solely extract global drift. Removing activation functions and employing 1×1 convolutions provide this strong smoothing constraint. Conversely, the Seasonal Flow must actively model leaked low-frequency trend drifts alongside periodic cycles, thus retaining the full non-linear capacity of the standard GLB. Both streams share a unified backbone via the identical dimension-decoupling paradigm. Degrading one to an activation-free filter perfectly complements the other, forming a complete defense against leakage.

4 Experiments

Setups. We conduct extensive experiments on eight widely-used real-world benchmark datasets: ETT (4 subsets) (Zhou et al., 2021), Electricity, Traffic, Weather (Wu et al., 2021), and Solar-energy (Lai et al., 2018). We re-evaluate all baselines across various lookback windows and report their best results. All models are configured to predict four different horizons: 96, 192, 336, 720. The lower MSE or MAE indicates a more accurate prediction result.

Baselines. We include a comprehensive set of recent and SOTA time series baselines, including: TimesNet (Wu et al., 2023), DLinear (Zeng et al., 2023), MICN (Wang et al., 2023), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024a), TimeMixer (Wang et al., 2024a), FITS (Xu et al., 2024), PDF (Dai et al., 2024), U-Mixer (Ma et al., 2024), CARD (Wang et al., 2024b), SimpleTM (Chen et al., 2025a) and xPatch (Stitsyuk & Choi, 2025).

4.1 Main Results

Table 1 presents the comprehensive evaluation of GLNet against 12 SOTA baselines in eight widely-used datasets. GLNet achieves leading mean forecasting accuracy, securing the lowest mean MSE on 7 out of 8 datasets and the lowest mean MAE on 7 out of 8 datasets. Even in scenarios where GLNet does not rank first, such as on the Traffic dataset, it remains highly competitive with marginal gaps to the top performer. This broad success across diverse domains, ranging from the smooth periodicity of energy loads to the rapid fluctuations of weather conditions, demonstrates the model's robustness and strong generalization capabilities.

In direct comparison with the most competitive efficient baselines, particularly xPatch (Stitsyuk & Choi, 2025), GLNet achieves the strongest mean performance. While xPatch serves as a strong runner-up, our model attains lower mean MSE and MAE on seven of the eight datasets. Notably, on the challenging Solar dataset, GLNet reduces the MSE by approximately 7.0% compared to xPatch (0.185 $vs.$ 0.199). This performance gap validates the effectiveness of the proposed Global-Local architecture. Unlike xPatch, which processes patches with simple linear mappings, GLNet employs a hierarchical mechanism to simultaneously capture intra-patch micro-dynamics and inter-patch macro-evolution. This capability allows our model to extract richer temporal representations and handle complex variations more robustly than purely linear baselines.

Furthermore, GLNet consistently outperforms models like PatchTST, iTransformer, and TimeMixer, which are Transformer-based architectures. For instance, on the Electricity dataset, GLNet achieves an MSE of 0.153, lower than PatchTST (0.165) and iTransformer (0.164). These results challenge the necessity of heavy self-attention mechanisms for LTSF. They confirm that our dual-stream architecture can effectively model long-range dependencies while avoiding the risks of overfitting and high computational overhead.

4.2 Ablation Study

To further verify the effectiveness of our proposed components, we conduct a comprehensive ablation study on the GLNet architecture, with results summarized in Table 2. We investigate the contribution of the dual-stream decomposition strategy by independently removing (w/o) the Trend Flow and the Seasonal Flow. The results show a clear performance degradation when either branch is removed, with the average MSE increasing from 0.265 to 0.277 in Sea. cases. Notably, on datasets with distinct periodicity and shifting baselines like Traffic, removing the Trend Flow leads to a significant error increase (MSE rises from 0.389 to 0.408). This confirms that a single stream is insufficient to model complex temporal variations, and the explicit disentanglement of low-frequency evolution and high-frequency seasonality is essential for accurate forecasting.

Table 2: Ablation study of GLNet. The best results are highlighted in **bold**. Avg denotes the average performance across the five datasets.

Model	GLNet		w/o GLB/GLB*		Replace GLB/GLB*		w/o Trend Flow		w/o Sea. Flow	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	**0.211**	**0.248**	0.219	0.255	0.217	0.252	0.216	0.254	0.214	0.251
Traffic	**0.389**	**0.264**	0.417	0.276	0.408	0.274	0.408	0.275	0.411	0.273
Electricity	**0.153**	**0.247**	0.158	0.251	0.161	0.254	0.158	0.250	0.155	0.248
ETTh1	**0.386**	**0.408**	**0.386**	0.409	0.392	0.419	0.394	0.411	0.395	0.420
Solar	**0.185**	**0.213**	0.223	0.231	0.191	0.221	0.212	0.225	0.212	0.222
Avg	**0.265**	**0.276**	0.281	0.284	0.274	0.284	0.278	0.283	0.277	0.283

We also scrutinize the design of our core encoders by modifying the fundamental building blocks across both streams. We entirely remove the GLB from the Seasonal Flow and the GLB* from the Trend Flow (w/o GLB/GLB*). This configuration yields the poorest performance among all variants (Average MSE 0.281), indicating that without these deep feature extractors, the model loses its non-linear modeling capacity and effectively degenerates into a shallow baseline that underfits the data. Furthermore, we substitute both the GLB and GLB* with standard Multi-Head Self-Attention layers (Replace GLB/GLB*). Although introducing attention mechanisms recovers some performance compared to removing the blocks entirely, it still lags behind the proposed GLNet (0.274 $vs.$ 0.265). This comparison demonstrates that replacing our convolution-based blocks with self-attention not only sacrifices the strong inductive bias for local context, but crucially destroys the activation-free smoothing constraint of the Trend Flow. By losing this essential low-pass filter property, the attention-based Trend Flow becomes highly susceptible to high-frequency seasonal leakage. The model subsequently overfits to these transient fluctuations, which directly drives the observed performance degradation.

Overall, the full GLNet architecture achieves the best performance across all datasets except for minor fluctuations on ETTh1. The ablation results consistently demonstrate that both the macro-level dual-stream structure and the micro-level GLB operator contribute significantly to the model's success, further validating that GLNet's hierarchical approach effectively synthesizes global semantic expansion and local temporal contextualization.

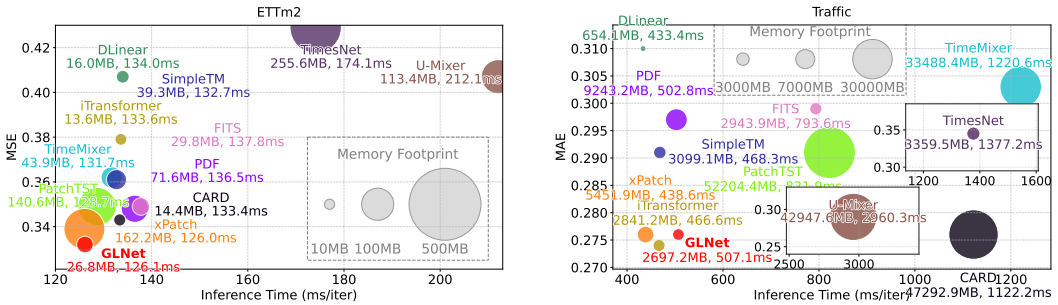


Figure 3: Model efficiency comparison. The running efficiency of 11 models on the ETTm2 and Traffic datasets with the prediction length $H = 720$ and the batch size $B = 32$. The x-axis represents inference time (the left is the better), the y-axis represents MSE or MAE (the lower is the better), and the bubble size represents memory usage (the smaller is the better).

4.3 Model Analysis

Efficient Analysis. To validate the computational efficiency and deployment feasibility of GLNet, we visualize the trade-off between inference speed, memory footprint, and forecasting accuracy in Figure 3. On the ETTm2 dataset, GLNet is optimally positioned in the bottom-left corner, achieving the lowest MSE (0.332) while maintaining a rapid inference time that rivals the fastest baseline, xPatch. A critical advantage of GLNet lies in its memory efficiency: It consumes only 26.8 MB of memory, which is an 83% reduction compared to xPatch (162.2 MB) and an 80% reduction compared to PatchTST (140.6 MB). This stark contrast highlights that while high-performing baselines often rely on memory-intensive operations (e.g., attention matrices or redundancy in patching), GLNet's GLB design effectively minimizes resource overhead. Furthermore, while lightweight models like iTransformer and DLinear demonstrate similar memory footprints, they suffer from significantly higher error rates (MSE > 0.378). Consequently, GLNet represents the most Pareto-efficient solution, successfully breaking the common trade-off between model complexity and forecasting precision. On the Traffic dataset, GLNet stands out among channel-independent models by achieving the lowest MAE while maintaining fast inference time and high memory efficiency.

The Influence of Patch Length. We further investigate the sensitivity of GLNet to the patch length P, a key hyperparameter in patch-based architectures, by varying $P \in \{16, 32, 48, 64\}$ across different prediction horizons. Figure 4 illustrates the MSE performance on the ETTm2 and Traffic datasets. A striking observation is the flat trajectory of the performance curves, indicating that GLNet is remarkably robust to changes in patch size. For instance, on the ETTm2 dataset with a prediction length of 96, the MSE fluctuates negligibly between 0.148 and 0.150 as P quadruples from 16 to 64. Similarly, for the long-term forecast of 720 steps, the error remains stable around 0.332. This insensitivity can be attributed to our hierarchical dual-stream design: While the patch length defines the granularity of local feature extraction in the Seasonal Flow, the Trend Flow operates on global embeddings, and the Linear Fusion block dynamically balances these components. Consequently, even if the patch size is suboptimal for capturing specific local periodicities, the model compensates through its global trend modeling and inter-patch processing, ensuring stable performance without the need for exhaustive hyperparameter tuning.

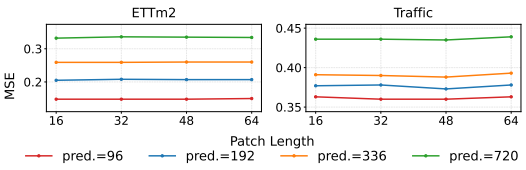


Figure 4: The MSE results with different patch lengths in ETTm2 dataset (left) and Traffic dataset (right).

The Influence of Network Depth. We examine the impact of model capacity on forecasting performance by varying the number of stacked GLB layers in the Seasonal Flow (k_1) and GLB* layers in the Trend Flow (k_2). Figure 5 displays the impact of network depth on model performance. Observations from the Traffic dataset reveal a clear dependency on the depth of the Seasonal Flow. As k_1 increases from 1 to 4, the MSE steadily decreases from 0.398 to approximately 0.388. This suggests that for datasets with complex, high-frequency traffic patterns, a deeper seasonal encoder

is crucial to disentangle intricate periodic dependencies. Conversely, the performance remains relatively stable regardless of the Trend Flow depth (k_2), indicating that the low-frequency evolution in Traffic data is sufficiently smooth to be modeled by a shallow projection. On the ETTm2 dataset, the model exhibits improved performance as the Seasonal Flow depth increases from 1 to 3 (MSE reduces from 0.262 to 0.259), after which the performance saturates. Similar to Traffic, the Trend Flow depth (k_2) shows marginal impact, with optimal results achievable even at $k_2 = 1$. The Seasonal Flow generally requires depth ($k_1 \geq 2$) to capture micro-dynamic details effectively, whereas the Trend Flow is robust with a shallower architecture ($k_2 \in \{1, 2\}$). This aligns with our design philosophy: the Seasonal Flow handles complex local variations requiring deep non-linear extraction, while the Trend Flow handles global drifts, where the mild non-linearity from LayerNorm provides sufficient expressiveness without requiring deep stacks.

4.4 The Influence of Lookback Window.

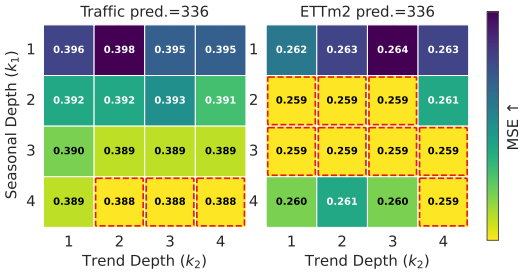


Figure 5: Analysis of network depth. The figure displays MSE values for varying trend flow depths (N_l = 1-4) and seasonal flow depths (N_s = 1-4). Lighter colors indicate lower errors.

For models with strong capabilities in capturing long-term temporal dependencies, longer lookback window generally lead to improved performance (Nie et al., 2023). To validate the effectiveness of our model in this regard, we conduct experiments with varying lookback window $L \in \{96, 192, 336, 720\}$ under a fixed prediction horizon of 720. As illustrated in Figure 6, GLNet exhibits a consistent monotonic decrease in MAE on both datasets as the lookback window expands, validating its ability to effectively leverage extended historical context without suffering from overfitting. Specifically, on the ETTm2 dataset, GLNet achieves its best performance at $L = 720$ (0.366), surpassing the previous SOTA xPatch (0.367) and significantly outperforming PatchTST (0.379). Crucially, on the Traffic dataset, while methods like PatchTST begin to deteriorate at $L = 720$ (MAE rising from 0.291 to 0.305) and DLinear shows performance saturation, GLNet maintains a steady downward trajectory (MAE dropping from 0.330 to 0.276). This suggests that our dual-stream design, which combines the Trend Flow's global receptive field with the Seasonal Flow's hierarchical patching, successfully mitigates the noise accumulation often observed in long-sequence modeling, allowing the model to extract richer temporal signals from longer histories.

5 Conclusion

In this paper, we introduce GLNet, a unified dual-stream architecture. By leveraging the proposed GLB, the model adeptly captures both global semantic evolution and local temporal micro-dynamics. Crucially, this design avoids the heavy computational burden of self-attention mechanisms, offering a highly efficient alternative for long-sequence modeling without compromising feature expressiveness. Extensive experiments on eight real-world datasets demonstrate that GLNet achieves leading mean forecasting accuracy while maintaining faster inference speeds and lower memory usage. Our ablation studies and sensitivity analyses further corroborate the robustness of GLNet regarding key hyperparameters, including patch length, lookback window, and network depth. Future work will focus on extending GLNet to cross-variate dependencies.

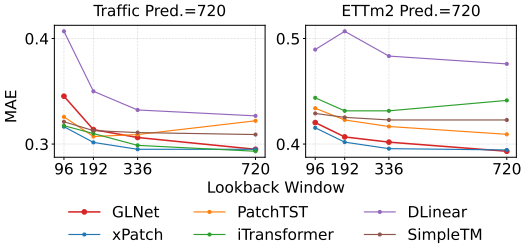


Figure 6: The performance of GLNet with respect to lookback window.

AI USE STATEMENT

In this work, we used generative AI tools for searching for and identifying relevant literature, assisting in the summarization and analysis of existing literature, editing already-written passages of the manuscript to improve readability and consistency, and formatting the reference list. We have reviewed all AI-assisted work. Every AI-suggested reference and literature summary was checked by the authors against the original publication, every suggested wording change was checked against the authors' intended meaning, and all technical claims, citations, and reported numbers were verified by the authors against the original sources and experimental records. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure that our results are reproducible. The GLNet architecture is fully specified in Section 3 and illustrated in Figure 2. All eight datasets used in our experiments are publicly available, and their sources, statistics, and the chronological train/validation/test splits we adopt are described in Appendix C and summarized in Table 3. The full training protocol, covering the optimizer, learning-rate schedule, number of epochs, early-stopping criteria, hardware, and the dataset- and horizon-specific network depths, is documented in Appendix D and Table 4. Our source code and the scripts required to reproduce all tables and figures are available as supplementary material.

REFERENCES

M.A. Castán-Lascorz, P. Jiménez-Herrera, A. Troncoso, and G. Asencio-Cortés. A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting. *Inf. Sci.*, 586(C):611–627, mar 2022.

Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. SimpleTM: A simple baseline for multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=oANkBaVci5.

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. In *International Conference on Learning Representations (ICLR)*, 2024.

Xiwen Chen, Peijie Qiu, Wenhui Zhu, Huayu Li, Hao Wang, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. Sequence complementor: Complementing transformers for time series forecasting with learnable sequences, 2025b. URL https://arxiv.org/abs/2501.02735.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. *International Conference on Learning Representations*, 2024.

Shibo Feng, Peilin Zhao, Liu Liu, Pengcheng Wu, and Zhiqi Shen. Hdt: Hierarchical discrete transformer for multivariate time series forecasting, 2025. URL https://arxiv.org/abs/2502.08302.

Romain Ilbert, Ambroise Odonnat, Vasilii Feofanov, Aladin Virmaux, Giuseppe Paolo, Themis Palpanas, and Ievgen Redko. Samformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention, 2024. URL https://arxiv.org/abs/2402.10198.

Dongbin Kim, Jinseong Park, Jaewook Lee, and Hoki Kim. Are self-attentions effective for time series forecasting?, 2024. URL https://arxiv.org/abs/2405.16877.

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*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

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

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

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=JePfAI8fah.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models, 2024b. URL https://arxiv.org/abs/2402.02368.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Xiang Ma, Xuemei Li, Lexin Fang, Tianlong Zhao, and Caiming Zhang. U-mixer: An unet-mixer architecture with stationarity correction for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14255–14262, 2024.

Reza Nematirad, Anil Pahwa, and Balasubramaniam Natarajan. Times2d: Multi-period decomposition and derivative mapping for general time series forecasting, 2025. URL https://arxiv.org/abs/2504.00118.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 2400–2410. ACM, August 2024. doi: 10.1145/3637528.3671928. URL http://dx.doi.org/10.1145/3637528.3671928.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S Jensen, Zhenli Sheng, et al. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *arXiv preprint arXiv:2403.20150*, 2024.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting, 2025a. URL https://arxiv.org/abs/2412.10859.

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.

Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and ChristianS. Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment*, 15(11):2733–2746, Jul 2022. doi: 10.14778/3551793.3551827.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts, 2025. URL https://arxiv.org/abs/2409.16040.

Artyom Stitsyuk and Jaesik Choi. xpatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20601–20609, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2024a.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis, 2025. URL https://arxiv.org/abs/2410.16032.

Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. CARD: Channel aligned robust blend transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=MJksrOhurE.

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, 2024c. URL https://arxiv.org/abs/2402.19072.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting, 2022a. URL https://arxiv.org/abs/2202.01575.

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

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. In *International Conference on Learning Representations*, 2023.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters, 2024. URL https://arxiv.org/abs/2307.03756.

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.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures, 2022. URL https://arxiv.org/abs/2207.01186.

Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Multi-resolution time-series transformer for long-term forecasting. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4222–4230. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/zhang24l.html.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

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, 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, 2022.

A APPENDIX

B PROOF OF EFFICIENT COMPUTATION

In this section, we provide a rigorous mathematical derivation of the computational complexity of the proposed GLNet. Let the input time series be $\mathbf{X} \in \mathbb{R}^{C \times L}$, where C is the number of channels and L is the lookback window length.

The Seasonal Stream begins with a hierarchical patching operation. Given a patch size P and stride S, the temporal sequence of length L is transformed into N patches. The number of patches is formally defined as:

$$N = \left\lfloor \frac{L - P}{S} \right\rfloor + 1 \tag{3}$$

The patched input tensor is denoted as $\mathbf{X}^{(p)} \in \mathbb{R}^{C \times N \times P}$. The Seasonal Flow processes this tensor through a series of GLB, which operate in two stages: Intra-patch and Inter-patch.

Intra-patch stage. The intra-patch stage models the internal dynamics of each patch. The linear projection can be expressed using explicit Einstein summation or standard matrix multiplication:

$$(\mathbf{X}_{intra})_{i,k,j} = \sum_{m=1}^{P} \mathbf{X}^{(p)}_{i,k,m} \mathbf{W}^{m,j}_{intra} \tag{4}$$

where $\mathbf{W}_{intra} \in \mathbb{R}^{P \times P}$ is the weight matrix for the intra-patch projection. For each of the C channels and N patches, performing this projection requires $O(P^2)$ operations. Therefore, the computational complexity of the Intra-patch stage is:

$$\mathcal{C}_{intra} = \mathcal{O}(NP^2C) \tag{5}$$

Inter-patch stage. The inter-patch stage captures the global evolution across patches. The intermediate tensor $\mathbf{X}_{inter}$ is obtained by transposing the dimensions, and the subsequent linear projection is defined as:

$$\mathbf{H}_{i,m,j} = \sum_{k=1}^{N} (\mathbf{X}_{inter})_{i,k,j} \mathbf{W}^{k,m}_{inter} \tag{6}$$

where $\mathbf{W}_{inter} \in \mathbb{R}^{N \times N}$ is the weight matrix. This operation takes $O(N^2)$ for each of the C channels and P elements within a patch. Thus, the complexity of the Inter-patch stage is:

$$\mathcal{C}_{inter} = \mathcal{O}(PN^2C) \tag{7}$$

Trend Stream. The Trend Stream processes the whole-series embedding without patching. The initial linear embedding maps the sequence of length L to a hidden dimension H_t, incurring a one-time projection complexity of $\mathcal{O}(L \cdot C \cdot H_t)$. Subsequently, the stacked GLB* layers operate on the compressed H_t dimension, introducing a recurrent per-layer complexity of $\mathcal{O}(C \cdot H_t^2)$. The total computational cost for the Trend Stream with k_2 layers is:

$$\mathcal{C}_{trend} = \mathcal{O}(L \cdot C \cdot H_t + k_2 \cdot C \cdot H_t^2) \tag{8}$$

Total Complexity. The overall computational complexity of the GLNet model per layer is the sum of the complexities of the Seasonal and Trend streams:

$$\mathcal{C}(\text{GLNet}) = \mathcal{O}(NP^2C + PN^2C + L \cdot C \cdot H_t + C \cdot H_t^2) \tag{9}$$

Substituting $N \approx L/S$ into the equation, we obtain:

$$\mathcal{C}(\text{GLNet}) = \mathcal{O}\left(C \cdot \left[\frac{L^2P}{S^2} + \frac{LP^2}{S} + L \cdot H_t + H_t^2\right]\right) \tag{10}$$

Although the term $\frac{L^2P}{S^2}$ introduces a quadratic dependence on L, it achieves significant practical efficiency by minimizing the quadratic constant factor. Because the stride $S > 1$ (typically $S = P/2$ or $S = P$), the denominator S^2 drastically attenuates the L^2 term. This patching mechanism effectively reduces the length of the sequence processed by the inter-patch mixing layers, making GLNet significantly more computationally efficient than standard Transformer models.

C DATASETS

To evaluate long-term forecasting performance, we conduct experiments on eight widely-used real-world datasets, including Weather, Traffic, Electricity, Solar-energy, and four Electric Transformer Temperature (ETT) datasets (ETTh1, ETTh2, ETTm1, ETTm2). These datasets have been extensively adopted as benchmarks and span multiple domains such as energy, transportation, and meteorology.

As summarized in Table 3, we report key characteristics of each dataset, including the total number of time steps, the number of variables, and the sampling frequency. Following the standard protocol (Zhou et al., 2021), we split all datasets chronologically into training, validation, and test sets. For the ETT datasets, a split ratio of 6:2:2 is applied, while a ratio of 7:1:2 is used for the remaining datasets. All sets are normalized to zero mean and unit variance using the mean and standard deviation derived from the training set. Each dataset consists of a single continuous long-term time series, from which samples are extracted via a sliding window approach.

Further details regarding the datasets are as follows:

- **Weather** [1]: Contains 21 meteorological indicators (e.g., air temperature and humidity) recorded at 10-minute intervals throughout 2020 in Germany.
- **Electricity** [2]: Consists of hourly data on electricity consumption from 321 clients between 2012 and 2014.
- **Traffic** [3]: Comprises road occupancy rates collected from 862 sensors on highways in the San Francisco Bay Area, provided by the California Department of Transportation.
- **ETT** [4]: (Electricity Transformer Temperature) This dataset includes measurements from seven sensors in two Chinese counties, covering indicators such as load and oil temperature. It comprises four subsets: ETTh1 and ETTh2 for hourly data, and ETTm1 and ETTm2 for 15-minute intervals.
- **Solar-energy** [5]: This dataset includes solar power production records from 137 PV plants in 2006.

Table 3: The detail statistics of datasets

Dataset	Dim	Prediction Length	Dataset Size	Split	Frequency	domain
ETTh1, ETTh2	7	{96, 192, 336, 720}	17420	6:2:2	Hourly	Electricity
ETTm1, ETTm2	7	{96, 192, 336, 720}	69680	6:2:2	15 min	Electricity
Weather	21	{96, 192, 336, 720}	52696	7:1:2	10 min	Weather
Electricity	321	{96, 192, 336, 720}	26304	7:1:2	Hourly	Electricity
Traffic	862	{96, 192, 336, 720}	17544	7:1:2	Hourly	Traffic
Solar-energy	137	{96, 192, 336, 720}	52063	7:1:2	10 min	Solar-energy

D EXPERIMENT DETAILS

Experimental Settings. All experiments are implemented in PyTorch (Paszke et al., 2019) using Python 3.11, and conducted on a single NVIDIA H100 GPU. The default training procedure runs for 100 epochs, with early stopping applied on the validation set. The patience threshold is 10 epochs for ETT, Electricity, and Weather datasets, and 13 epochs for Solar and Traffic datasets. Our proposed method is optimized using the ADAMW optimizer (Loshchilov & Hutter, 2019), with a sigmoid-based learning rate scheduler, following the exact same training scheme as xPatch (Stitsyuk & Choi, 2025).

[1] https://www.bgc-jena.mpg.de/wetter/
[2] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
[3] https://pems.dot.ca.gov/
[4] https://github.com/zhouhaoyi/ETDataset
[5] https://www.nrel.gov/grid/solar-power-data.html

Model parameter. Within the GLB and GLB* blocks, the linear layers are dimension-preserving across all datasets. The convolution dilation rate is fixed at 1. To accommodate varying dataset scales and complex dynamics, the network depths for the Seasonal Flow (k_1) and Trend Flow (k_2) are treated as tunable hyperparameters, typically searched within $\{1, 2, 3, 4\}$. While $k_1 = 2$ and $k_2 = 2$ serve as a common baseline, the optimal configurations are dataset-specific and prediction-horizon-dependent. Detailed hyperparameter configurations for full reproducibility are provided in Table 4.

Table 4: Detailed network hyperparameters configurations across varying prediction horizons for all datasets.

Dataset	Prediction Length	Seasonal Flow Depth	Trend Flow Depth	Horizon Window
ETTh1	96	2	1	720
	192	2	1	720
	336	2	2	336
	720	2	2	336
ETTh2	96	2	2	336
	192	2	2	336
	336	2	2	336
	720	1	2	720
ETTm1	96	1	2	720
	192	2	1	336
	336	2	1	720
	720	2	2	720
ETTm2	96	1	1	720
	192	4	1	720
	336	3	2	720
	720	2	1	720
Solar	96	1	2	720
	192	1	4	720
	336	1	2	720
	720	2	1	720
Electricity	96	4	2	720
	192	4	3	720
	336	1	1	720
	720	4	4	720
Traffic	96	2	4	720
	192	2	2	720
	336	2	2	720
	720	2	2	720
Weather	96	1	4	720
	192	1	2	720
	336	1	1	720
	720	3	1	720

Metric. We adopt the mean square error (MSE) and mean absolute error (MAE) for long-term forecasting.

$$MSE = \frac{1}{H}\sum_{i=1}^{H}(\widehat{x}_i - x_i)^2 \tag{11}$$

$$MAE = \frac{1}{H}\sum_{i=1}^{H}|\widehat{x}_i - x_i| \tag{12}$$

where H is the number of observations in the forecast horizon, $\widehat{x}_i$ is the predicted value, and x_i is the actual value for observation i.

Table 5: Performance comparison with different smoothing factors.

α	Metric	Weather					ETTh1					ETTh2					ETTm1					ETTm2				
		96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg
0.1	MSE	0.148	0.190	0.220	0.294	0.213	0.353	0.374	0.392	0.437	0.389	0.227	0.277	0.314	0.394	0.303	0.289	0.322	0.356	0.414	0.345	0.147	0.206	0.259	0.333	0.237
	MAE	0.189	0.230	0.262	0.318	0.250	0.382	0.395	0.411	0.452	0.410	0.301	0.337	0.367	0.428	0.358	0.339	0.356	0.377	0.407	0.369	0.239	0.279	0.316	0.365	0.300
0.3	MSE	0.144	0.186	0.220	0.292	**0.211**	0.348	0.373	0.391	0.430	0.386	0.228	0.277	0.314	0.379	**0.299**	0.283	0.320	0.358	0.412	**0.343**	0.148	0.205	0.259	0.332	**0.236**
	MAE	0.185	0.228	0.263	0.314	0.248	0.380	0.394	0.411	0.448	**0.408**	0.302	0.336	0.366	0.419	**0.356**	0.337	0.355	0.378	0.406	**0.369**	0.239	0.279	0.317	0.366	**0.300**
0.5	MSE	0.144	0.187	0.221	0.292	**0.211**	0.348	0.374	0.387	0.428	**0.384**	0.227	0.276	0.315	0.376	**0.299**	0.287	0.320	0.366	0.421	0.349	0.147	0.205	0.260	0.331	**0.236**
	MAE	0.184	0.228	0.262	0.313	**0.247**	0.383	0.394	0.409	0.447	**0.408**	0.301	0.336	0.367	0.420	**0.356**	0.337	0.355	0.380	0.409	0.370	0.239	0.281	0.317	0.366	0.301
0.7	MSE	0.146	0.187	0.221	0.292	0.211	0.350	0.374	0.387	0.429	0.385	0.227	0.276	0.313	0.380	0.299	0.291	0.320	0.363	0.417	0.348	0.148	0.209	0.261	0.334	0.238
	MAE	0.188	0.228	0.263	0.314	0.248	0.383	0.393	0.409	0.447	0.408	0.302	0.336	0.366	0.420	0.356	0.338	0.355	0.380	0.409	0.371	0.239	0.282	0.318	0.367	0.302

E DECOMPOSITION

We employ EMA decomposition, which calculates the trend recursively. For the normalized input $\hat{x}^{(i)}$ of the i-th variate, the trend component $X_T^{(i)}$ is computed as:

$$\begin{aligned} s_1^{(i)} &= \hat{x}_1^{(i)} \\ s_t^{(i)} &= \alpha \cdot \hat{x}_t^{(i)} + (1-\alpha) \cdot s_{t-1}^{(i)}, \quad t > 1 \\ X_T^{(i)} &= [s_1^{(i)}, s_2^{(i)}, ..., s_T^{(i)}] \end{aligned} \tag{13}$$

where α ($0 < \alpha < 1$) is a smoothing factor.

By using EMA's recursive formulation, boundary distortion is effectively avoided and the integrity of the original sequence is preserved.

Sensitivity Analysis of Smoothing Factor. To assess the model's sensitivity to the hyperparameter α, we conduct extensive experiments with different values of α, as detailed in Table 5. The empirical results demonstrate that GLNet exhibits significant robustness to changes in the smoothing factor within the moderate range of $[0.3, 0.5]$. For instance, on the Weather and ETTh2 datasets, the performance differences between $\alpha = 0.3$ and $\alpha = 0.5$ are negligible, with MSE fluctuations typically less than 0.002. While extremely low or high values (e.g., 0.1 or 0.7) may lead to suboptimal decomposition due to over-smoothing or under-smoothing of the trend component, the model's accuracy does not decline significantly, thereby obviating the need for laborious parameter tuning. For the sake of simplicity, we adopt $\alpha = 0.3$ as the default configuration.

F FULL RESULTS

Due to the space limitation of the main text, we place the full results of long-term forecasting in Table 6.

G LIMITATIONS

While GLNet achieves leading mean forecasting accuracy across various benchmarks, we acknowledge that the current architecture adopts a channel-independent design, treating each variate independently. This design choice is deliberate, as it prioritizes robustness against distribution shifts, minimizes overfitting, and significantly reduces computational overhead, factors that contribute to its strong performance on standard benchmarks. However, we recognize that in certain real-world forecasting scenarios where cross-variate correlations are highly critical, this assumption might restrict the model's ability to capture complex inter-variable dynamics. We consider this a minor trade-off given the substantial gains in efficiency and stability.

H MORE EXTRA EXPERIMENTS STUDIES

We have conducted additional experiments and analyses.

Table 6: The complete results for **long-term forecasting** are detailed. We compare extensive competitive models under four prediction lengths. The lookback window is searched to the best for a fairer comparison. Avg is averaged from all four prediction lengths. The best performance is highlighted in **red and bold**, and the second best is marked with blue and underline.

Models		**GLNet** (Ours)		xPatch (2025)		SimpleTM (2025a)		CARD (2024b)		U-Mixer (2024)		PDF (2024)		FITS (2024)		iTransformer (2024a)		TimeMixer (2024a)		PatchTST (2023)		MICN (2023)		DLinear (2023)		TimesNet (2023)	
Metrics		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
Weather	96	**0.144**	**0.185**	0.147	0.186	0.157	0.203	0.149	0.189	0.154	0.192	0.148	0.196	0.158	0.210	0.160	0.210	0.149	0.202	0.149	0.197	0.173	0.233	0.166	0.234	0.170	0.217
	192	**0.186**	**0.228**	0.189	0.228	0.198	0.247	0.192	0.232	0.188	0.228	0.192	0.390	0.201	0.249	0.202	0.248	0.194	0.247	0.193	0.240	0.213	0.270	0.218	0.278	0.219	0.262
	336	**0.220**	0.263	0.224	**0.261**	0.248	0.285	0.241	0.268	0.242	0.269	0.243	0.270	0.249	0.286	0.254	0.288	0.247	0.283	0.244	0.281	0.256	0.307	0.262	0.313	0.279	0.304
	720	**0.292**	**0.314**	0.298	0.316	0.320	0.335	0.313	0.324	0.314	0.321	0.321	0.333	0.317	0.335	0.323	0.337	0.320	0.333	0.313	0.332	0.308	0.342	0.326	0.366	0.359	0.355
	Avg	**0.211**	**0.248**	0.215	0.248	0.231	0.268	0.224	0.253	0.224	0.253	0.226	0.297	0.231	0.270	0.235	0.271	0.227	0.266	0.225	0.262	0.238	0.288	0.243	0.298	0.257	0.284
Traffic	96	0.360	0.255	0.364	0.236	0.382	0.260	**0.353**	**0.235**	0.385	0.251	0.360	0.246	0.393	0.274	0.361	0.262	0.369	0.257	0.365	0.256	0.459	0.298	0.388	0.275	0.594	0.317
	192	0.373	0.257	0.381	0.245	**0.365**	0.264	0.369	**0.243**	0.401	0.260	0.382	0.261	0.405	0.280	0.375	0.267	0.400	0.272	0.383	0.263	0.473	0.311	0.399	0.280	0.615	0.329
	336	0.388	0.268	0.391	**0.246**	**0.380**	0.272	0.388	0.253	0.418	0.269	0.393	0.268	0.422	0.293	0.385	0.269	0.407	0.272	0.394	0.270	0.469	0.301	0.413	0.288	0.628	0.334
	720	0.435	0.276	0.446	0.276	0.420	0.291	0.430	0.276	0.458	0.290	0.438	0.297	0.449	0.299	**0.416**	**0.274**	0.432	0.303	0.433	0.291	0.581	0.339	0.452	0.310	0.638	0.345
	Avg	0.389	0.264	0.396	**0.251**	0.387	0.272	0.385	0.252	0.416	0.268	0.393	0.268	0.417	0.287	**0.384**	0.268	0.402	0.276	0.394	0.270	0.496	0.312	0.413	0.288	0.619	0.331
Electricity	96	**0.120**	**0.215**	0.126	0.218	0.143	0.237	0.129	0.224	0.148	0.255	0.127	0.221	0.137	0.234	0.132	0.228	0.143	0.241	0.136	0.235	0.157	0.265	0.144	0.237	0.175	0.279
	192	**0.140**	**0.233**	0.141	0.233	0.144	0.243	0.150	0.241	0.163	0.264	0.146	0.241	0.152	0.247	0.155	0.250	0.160	0.256	0.153	0.250	0.176	0.287	0.153	0.250	0.188	0.292
	336	**0.158**	**0.251**	0.158	0.252	0.167	0.264	0.164	0.258	0.168	0.263	0.163	0.258	0.168	0.264	0.169	0.265	0.178	0.276	0.166	0.263	0.182	0.295	0.169	0.267	0.210	0.310
	720	0.195	0.289	**0.192**	**0.283**	0.195	0.291	0.193	0.284	0.197	0.288	0.197	0.288	0.208	0.299	0.201	0.294	0.219	0.312	0.206	0.299	0.213	0.312	0.213	0.315	0.233	0.327
	Avg	**0.153**	**0.247**	0.154	0.247	0.162	0.259	0.159	0.252	0.169	0.268	0.158	0.252	0.166	0.261	0.164	0.259	0.175	0.271	0.165	0.262	0.182	0.290	0.170	0.267	0.201	0.302
ETTh1	96	**0.348**	**0.380**	0.355	0.382	0.369	0.395	0.366	0.389	0.370	0.389	0.360	0.392	0.374	0.395	0.387	0.405	0.374	0.401	0.370	0.395	0.377	0.413	0.378	0.401	0.404	0.423
	192	**0.373**	**0.394**	0.381	0.396	0.422	0.422	0.404	0.410	0.418	0.417	0.391	0.413	0.402	0.416	0.441	0.436	0.426	0.437	0.410	0.418	0.399	0.429	0.415	0.425	0.484	0.449
	336	**0.391**	**0.411**	0.399	0.414	0.438	0.454	0.427	0.424	0.431	0.430	0.423	0.438	0.423	0.431	0.471	0.446	0.442	0.448	0.429	0.433	0.428	0.446	0.448	0.448	0.503	0.483
	720	**0.430**	**0.448**	0.447	0.455	0.458	0.478	0.439	0.452	0.481	0.481	0.466	0.474	0.430	0.453	0.509	0.494	0.485	0.483	0.444	0.460	0.472	0.498	0.480	0.489	0.558	0.524
	Avg	**0.386**	**0.408**	0.395	0.412	0.422	0.437	0.409	0.419	0.425	0.429	0.410	0.429	0.407	0.424	0.452	0.445	0.432	0.442	0.413	0.427	0.419	0.447	0.430	0.441	0.487	0.470
ETTh2	96	**0.228**	**0.302**	0.238	0.305	0.287	0.346	0.256	0.318	0.290	0.335	0.274	0.338	0.274	0.340	0.301	0.354	0.284	0.349	0.274	0.337	0.314	0.373	0.293	0.357	0.374	0.400
	192	**0.277**	**0.336**	0.286	0.340	0.342	0.384	0.319	0.361	0.362	0.388	0.337	0.378	0.331	0.377	0.382	0.408	0.350	0.391	0.341	0.381	0.417	0.438	0.386	0.421	0.407	0.421
	336	**0.314**	**0.366**	0.323	0.371	0.348	0.396	0.337	0.380	0.385	0.410	0.367	0.403	0.352	0.396	0.407	0.428	0.397	0.435	0.345	0.393	0.471	0.473	0.465	0.474	0.417	0.443
	720	**0.379**	0.419	0.385	0.421	0.403	0.436	0.379	**0.418**	0.435	0.441	0.393	0.431	0.380	0.424	0.426	0.446	0.418	0.446	0.388	0.428	0.688	0.594	0.680	0.584	0.446	0.459
	Avg	**0.300**	**0.356**	0.308	0.360	0.345	0.391	0.323	0.369	0.368	0.394	0.343	0.388	0.334	0.384	0.379	0.409	0.362	0.405	0.337	0.385	0.473	0.470	0.456	0.459	0.411	0.431
ETTm1	96	0.283	0.337	**0.281**	**0.333**	0.325	0.363	0.296	0.338	0.309	0.350	0.285	0.340	0.303	0.345	0.301	0.355	0.297	0.349	0.289	0.342	0.302	0.348	0.300	0.344	0.346	0.383
	192	0.320	**0.355**	**0.319**	0.356	0.356	0.381	0.339	0.361	0.329	0.364	0.321	0.364	0.337	0.365	0.343	0.381	0.339	0.375	0.331	0.369	0.334	0.368	0.336	0.366	0.407	0.409
	336	0.358	**0.378**	0.358	0.379	0.375	0.399	0.371	0.381	0.363	0.385	**0.354**	0.383	0.367	0.384	0.376	0.400	0.373	0.392	0.364	0.392	0.372	0.392	0.369	0.387	0.420	0.424
	720	0.412	**0.406**	0.423	0.414	0.435	0.432	0.428	0.414	0.422	0.418	**0.411**	0.414	0.418	0.412	0.434	0.434	0.446	0.429	0.416	0.422	0.412	0.423	0.423	0.420	0.478	0.460
	Avg	**0.343**	**0.369**	0.345	0.370	0.373	0.394	0.358	0.373	0.356	0.379	0.343	0.375	0.356	0.377	0.364	0.392	0.364	0.387	0.350	0.381	0.355	0.383	0.357	0.379	0.413	0.419
ETTm2	96	**0.148**	**0.239**	0.152	0.241	0.175	0.262	0.159	0.245	0.167	0.251	0.161	0.251	0.163	0.253	0.175	0.266	0.171	0.253	0.162	0.254	0.170	0.269	0.163	0.257	0.193	0.275
	192	**0.205**	**0.279**	0.212	0.281	0.233	0.308	0.213	0.282	0.234	0.296	0.218	0.291	0.216	0.291	0.242	0.313	0.236	0.299	0.216	0.293	0.233	0.319	0.223	0.304	0.263	0.320
	336	**0.259**	0.317	0.265	0.318	0.289	0.344	0.264	**0.316**	0.297	0.338	0.268	0.327	0.268	0.326	0.290	0.341	0.281	0.331	0.270	0.329	0.295	0.359	0.291	0.355	0.324	0.358
	720	**0.332**	**0.366**	0.339	0.367	0.361	0.390	0.343	0.367	0.407	0.400	0.348	0.379	0.349	0.378	0.379	0.397	0.362	0.388	0.349	0.379	0.420	0.435	0.407	0.433	0.429	0.403
	Avg	**0.236**	**0.300**	0.242	0.302	0.264	0.326	0.245	0.303	0.276	0.321	0.249	0.312	0.249	0.312	0.272	0.329	0.263	0.318	0.249	0.314	0.280	0.346	0.271	0.337	0.302	0.339
Solar	96	**0.162**	**0.189**	0.172	0.204	0.170	0.235	0.182	0.208	0.195	0.246	0.181	0.247	0.207	0.255	0.196	0.249	0.182	0.249	0.170	0.234	0.192	0.251	0.211	0.279	0.198	0.270
	192	**0.186**	**0.213**	0.196	0.224	0.189	0.254	0.204	0.226	0.214	0.249	0.200	0.259	0.227	0.266	0.224	0.265	0.208	0.276	0.204	0.302	0.224	0.282	0.235	0.296	0.206	0.276
	336	**0.191**	**0.214**	0.209	0.229	0.201	0.260	0.215	0.231	0.218	0.250	0.208	0.269	0.240	0.272	0.212	0.267	0.208	0.278	0.212	0.293	0.257	0.307	0.252	0.310	0.208	0.284
	720	0.203	0.238	0.221	**0.237**	**0.199**	0.258	0.226	0.241	0.216	0.254	0.212	0.275	0.246	0.276	0.226	0.280	0.211	0.271	0.215	0.307	0.339	0.363	0.257	0.315	0.232	0.294
	Avg	**0.185**	**0.213**	0.199	0.224	0.190	0.252	0.207	0.227	0.211	0.250	0.200	0.263	0.230	0.267	0.214	0.265	0.203	0.269	0.200	0.284	0.253	0.301	0.239	0.300	0.211	0.281
1st count		**22**	**22**	3	5	3	0	1	4	0	0	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0

H.1 Sensitivity Analysis of Kernel Size

We further investigate the impact of the receptive field size within the GLB by varying the kernel size K of the depthwise convolution mechanism. As shown in Table 7, we evaluate the model's performance with $K \in \{3, 7, 15, 31\}$. The results indicate that GLNet is remarkably robust to variations in kernel size, with performance metrics remaining stable across different configurations. For instance, on the Electricity and Weather datasets, the average MSE shows minimal fluctuation (e.g., holding steady at 0.153 for Electricity), suggesting that the specific range of local context aggregation is not a bottleneck for our architecture. While larger kernels (e.g., $K = 15$) provide a marginal benefit on ETTh1, smaller kernels (e.g., $K = 3$) achieve the best or comparable performance on ETTh2 and Weather. Given that increasing the kernel size does not yield a statistically significant gain while potentially increasing computational overhead, we adopt $K = 3$ as the default configuration for all experiments. This decision prioritizes computational efficiency and enforces a strong inductive bias for fine-grained local feature extraction, leaving the modeling of long-range dependencies to the hierarchical patching structure.

H.2 Stability Analysis

Pooled standard deviation. To rigorously evaluate the robustness of our model against random weight initialization and data partitioning, we employ the pooled standard deviation as a metric of stability. Since we evaluate performance across four distinct prediction horizons ($H \in$

Table 7: Performance comparison with different kernel sizes in GLB.

K	Metric	ETTh1					ETTh2					Weather					Electricity				
		96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg
3	MSE	0.348	0.373	0.391	0.430	0.386	0.228	0.277	0.314	0.379	**0.299**	0.144	0.186	0.220	0.292	**0.211**	0.120	0.140	0.158	0.195	**0.153**
	MAE	0.380	0.394	0.411	0.448	0.408	0.302	0.336	0.366	0.419	0.356	0.185	0.228	0.263	0.314	0.248	0.215	0.233	0.251	0.289	**0.247**
7	MSE	0.351	0.371	0.388	0.426	0.384	0.226	0.274	0.315	0.380	**0.299**	0.145	0.187	0.222	0.292	0.212	0.121	0.141	0.156	0.195	**0.153**
	MAE	0.382	0.392	0.411	0.446	0.408	0.299	0.333	0.366	0.416	**0.354**	0.187	0.227	0.265	0.315	0.249	0.218	0.235	0.252	0.286	0.248
15	MSE	0.350	0.371	0.387	0.425	**0.383**	0.225	0.276	0.311	0.384	0.300	0.145	0.188	0.222	0.294	0.212	0.120	0.142	0.158	0.198	0.155
	MAE	0.379	0.393	0.408	0.447	**0.406**	0.299	0.335	0.364	0.419	**0.354**	0.186	0.227	0.264	0.317	0.248	0.218	0.236	0.251	0.288	0.248
31	MSE	0.350	0.370	0.386	0.426	**0.383**	0.226	0.274	0.317	0.383	0.299	0.146	0.187	0.220	0.291	**0.211**	0.121	0.142	0.157	0.196	0.154
	MAE	0.380	0.392	0.409	0.448	0.407	0.299	0.333	0.367	0.418	**0.354**	0.186	0.226	0.262	0.314	**0.247**	0.218	0.236	0.252	0.289	0.249

$\{96, 192, 336, 720\}$), simply averaging the standard deviations from each horizon may not accurately reflect the overall variance. Instead, we utilize the pooled standard deviation, which aggregates the variance across all horizons to provide a unified measure of uncertainty. The calculation is defined as:

$$SD_{pooled} = \sqrt{\frac{\sum_{i=1}^{4} \sum_{j=1}^{n} (x_{ij} - \bar{x}_i)^2}{4(n-1)}} \tag{14}$$

where n represents the number of independent experimental runs ($n = 3$ in our experiments), x_{ij} denotes the metric value of the j-th run at the i-th horizon, and $\bar{x}_i$ is the mean metric for that horizon. A lower pooled SD indicates that the model consistently converges to a stable solution regardless of random seed variations.

Significance test. To further validate that the observed performance improvements are statistically significant and not merely artifacts of random noise, we conduct a significance test using Type II Analysis of Variance (ANOVA). In this experimental design, the model architecture is treated as the primary factor, while the prediction horizon serves as a blocking factor. This blocking strategy is crucial to eliminate the variance introduced by the varying difficulty of different forecasting lengths, allowing us to focus purely on the performance differences between models. The null hypothesis (H_0) posits that there is no significant difference between the mean performance of the proposed GLNet and the baseline models. We compute the p-value for the model factor with a significance level of $\alpha = 0.05$. A p-value below this threshold allows us to reject the null hypothesis, confirming that the performance gap is statistically significant.

Result. We compare the proposed GLNet against xPatch (the second-best baseline) and SimpleTM (the third-best baseline) on long-term forecasting tasks across eight datasets, with statistical results summarized in Table 8. The table reports the aggregated means, pooled standard deviations, and the corresponding p-values from the ANOVA test. GLNet demonstrates comparable stability to the other baselines, achieving a competitive pooled standard deviation in the majority of datasets. Furthermore, the significance testing results indicate statistically significant improvements for GLNet in many scenarios. In several comparisons, the p-values are below the 0.05 threshold against SimpleTM and xPatch, providing statistical evidence that GLNet often outperforms both baselines on specific datasets. Even in competitive scenarios like the Weather dataset, GLNet maintains a significant lead over SimpleTM ($p = 0.000$), validating the effectiveness and robustness of the proposed hierarchical global-local architecture.

I Instance Normalization

In this work, we employ RevIN (Kim et al., 2022) as an architecture-free technique to address distribution shift (non-stationarity). To assess the impact of these shifts in time series data, we present a comprehensive comparison between models trained with and without RevIN across multiple benchmark datasets. Table 9 illustrates the quantitative results of this ablation study, where models without RevIN are denoted with an asterisk (*). The table clearly demonstrates that standard models incorporating RevIN consistently outperform their counterparts lacking this normalization module in the majority of cases. This suggesting that RevIN plays a pivotal role in stabilizing the training process and enhancing the predictive performance of various architectures, including GLNet, xPatch, CARD, and iTransformer.

Table 8: Performance comparison of models on long-term forecasting tasks. The table reports the averaged MSE and MAE across four prediction lengths, along with their pooled standard deviations (SD). Lower values indicate better model performance.

Dataset	Model	MSE (Pooled SD)	MAE (Pooled SD)	MSE p-value	MAE p-value
Weather	GLNet	0.211 (0.001)	0.248 (0.00012)	-	-
	xPatch	0.215 (0.0030)	0.248 (0.0015)	0.001	0.543
	SimpleTM	0.231 (0.0113)	0.267 (0.0038)	0.000	0.000
Traffic	GLNet	0.389 (0.0010)	0.264 (0.0010)	-	-
	xPatch	0.396 (0.0024)	0.251 (0.0012)	0.000	0.000
	SimpleTM	0.387 (0.0171)	0.272 (0.0030)	0.619	0.000
Electricity	GLNet	0.153 (0.0010)	0.247 (0.0019)	-	-
	xPatch	0.154 (0.0006)	0.247 (0.0008)	0.319	0.917
	SimpleTM	0.162 (0.0069)	0.259 (0.0058)	0.004	0.000
ETTh1	GLNet	0.385 (0.0015)	0.408 (0.0024)	-	-
	xPatch	0.396 (0.0008)	0.412 (0.0010)	0.000	0.000
	SimpleTM	0.422 (0.0086)	0.437 (0.0098)	0.000	0.000
ETTh2	GLNet	0.299 (0.0016)	0.356 (0.0014)	-	-
	xPatch	0.308 (0.0128)	0.359 (0.0088)	0.021	0.131
	SimpleTM	0.345 (0.0098)	0.391 (0.0047)	0.000	0.000
ETTm1	GLNet	0.343 (0.0020)	0.369 (0.0015)	-	-
	xPatch	0.345 (0.0025)	0.370 (0.0010)	0.172	0.151
	SimpleTM	0.373 (0.0107)	0.394 (0.0034)	0.000	0.000
ETTm2	GLNet	0.236 (0.0008)	0.300 (0.0008)	-	-
	xPatch	0.242 (0.0023)	0.302 (0.0010)	0.000	0.000
	SimpleTM	0.264 (0.0034)	0.326 (0.0034)	0.000	0.000
Solar	GLNet	0.185 (0.0031)	0.213 (0.0070)	-	-
	xPatch	0.199 (0.0045)	0.224 (0.0052)	0.000	0.002
	SimpleTM	0.190 (0.0047)	0.252 (0.0053)	0.037	0.000

A detailed examination of the results reveals that the performance gap between models with and without RevIN is most pronounced in datasets known for significant non-stationarity, such as ETTh2. For example, the average MSE of iTransformer on ETTh2 increases drastically from 0.379 to 0.797 when RevIN is removed, indicating a severe degradation in handling temporal distribution shifts. Similarly, GLNet shows a substantial improvement with RevIN on the same dataset (0.299 $vs.$ 0.629 MSE). While there are isolated instances where the non-RevIN variants achieve comparable or slightly better results, such as xPatch on the Weather dataset, the overall aggregation of metrics highlights the necessity of instance normalization for robust long-term forecasting.

It is noteworthy that GLNet exhibits intrinsic robustness against non-stationarity. GLNet maintains highly competitive performance even when RevIN is removed, distinguishing it from other baselines. For instance, on the ETTm1 dataset, the performance gap is negligible (0.344 *vs.* 0.343), and on the Weather dataset, GLNet* even marginally outperforms the standard GLNet (0.210 *vs.* 0.211). Furthermore, compared to baselines like xPatch, whose performance deteriorates sharply without RevIN on certain datasets (e.g., xPatch on ETTh1 rises significantly from 0.395 to 0.551), GLNet* remains remarkably stable (0.392). This resilience suggests that our hierarchical dual-stream architecture, specifically the Trend Flow's capacity to explicitly model global evolution, effectively internalizes the handling of mild distribution shifts, reducing the model's over-reliance on pre-processing techniques.

J Short Term Forecasting

We evaluate the short-term time series forecasting (STSF) ability of GLNet. We use the M4 dataset, which comprises 100,000 heterogeneous time series collected from diverse domains including economics, finance, industry, and demographics. Unlike the LTSF datasets, where all samples are de-

Table 9: Comparison of forecasting errors between the baselines and the models with and without RevIN instance normalization. The asterisk (*) denotes models trained without RevIN.

Models		**GLNet**		GLNet*		xPatch		xPatch*		CARD		CARD*		iTransformer		iTransformer*	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.348**	**0.380**	0.356	0.395	**0.355**	**0.382**	0.466	0.465	**0.366**	**0.389**	0.372	0.393	**0.387**	**0.405**	0.423	0.443
	192	**0.373**	**0.394**	0.379	0.407	**0.381**	**0.396**	0.497	0.484	**0.404**	**0.410**	0.415	0.416	**0.441**	**0.436**	0.452	0.462
	336	0.391	**0.411**	**0.387**	0.413	**0.399**	**0.414**	0.497	0.507	0.427	0.424	**0.422**	**0.421**	**0.471**	**0.446**	0.500	0.495
	720	**0.430**	**0.448**	0.446	0.470	**0.447**	**0.455**	0.745	0.601	0.439	0.452	**0.431**	**0.449**	**0.509**	**0.494**	0.588	0.575
	Avg	**0.386**	**0.408**	0.392	0.421	**0.395**	**0.412**	0.551	0.515	**0.409**	**0.419**	0.410	0.420	**0.452**	**0.445**	0.491	0.493
ETTh2	96	**0.228**	**0.302**	0.382	0.417	**0.238**	**0.305**	0.337	0.386	**0.256**	**0.318**	0.291	0.339	**0.301**	**0.354**	0.499	0.499
	192	**0.277**	**0.336**	0.435	0.449	**0.286**	**0.340**	0.362	0.396	**0.319**	**0.361**	0.348	0.378	**0.382**	**0.408**	0.740	0.649
	336	**0.314**	**0.366**	0.432	0.445	**0.323**	**0.371**	0.413	0.433	**0.337**	**0.380**	0.355	0.389	**0.407**	**0.428**	0.781	0.693
	720	**0.379**	**0.419**	1.269	0.813	**0.385**	**0.421**	0.720	0.588	0.379	0.418	**0.377**	**0.412**	**0.426**	**0.446**	1.167	0.814
	Avg	**0.299**	**0.356**	0.629	0.531	**0.308**	**0.360**	0.458	0.451	**0.323**	**0.369**	0.343	0.380	**0.379**	**0.409**	0.797	0.664
ETTm1	96	**0.283**	**0.337**	0.289	0.345	**0.281**	**0.333**	0.299	0.348	0.296	0.338	**0.283**	**0.327**	**0.301**	**0.355**	0.324	0.380
	192	0.320	**0.355**	**0.319**	0.360	**0.319**	**0.356**	0.333	0.369	0.339	0.361	**0.329**	**0.355**	**0.343**	**0.381**	0.364	0.407
	336	**0.358**	**0.378**	**0.358**	0.386	**0.358**	**0.379**	0.375	0.391	0.371	0.381	**0.368**	**0.378**	**0.376**	**0.400**	0.410	0.438
	720	0.412	**0.406**	**0.410**	0.411	**0.423**	**0.414**	0.442	0.424	**0.428**	**0.414**	0.438	0.418	**0.434**	**0.434**	0.469	0.471
	Avg	**0.343**	**0.369**	0.344	0.375	**0.345**	**0.370**	0.362	0.383	0.358	0.373	**0.354**	**0.369**	**0.364**	**0.392**	0.392	0.424
ETTm2	96	**0.148**	**0.239**	0.164	0.256	**0.152**	**0.241**	0.193	0.277	**0.159**	**0.245**	0.161	0.246	**0.175**	**0.266**	0.199	0.298
	192	**0.205**	**0.279**	0.230	0.301	**0.212**	**0.281**	0.256	0.317	**0.213**	**0.282**	0.214	**0.282**	**0.242**	**0.313**	0.415	0.456
	336	**0.259**	**0.317**	0.292	0.341	**0.265**	**0.318**	0.308	0.347	**0.264**	**0.316**	0.268	0.319	**0.290**	**0.341**	0.610	0.559
	720	**0.332**	**0.366**	0.396	0.404	**0.339**	**0.367**	0.411	0.410	**0.343**	**0.367**	0.349	0.370	**0.379**	**0.397**	0.771	0.654
	Avg	**0.236**	**0.300**	0.270	0.326	**0.242**	**0.302**	0.292	0.338	**0.245**	**0.303**	0.248	0.304	**0.272**	**0.329**	0.499	0.492
Weather	96	**0.144**	**0.185**	0.148	0.192	**0.147**	**0.186**	**0.147**	0.189	0.149	**0.189**	0.149	0.190	**0.160**	**0.210**	0.172	0.234
	192	**0.186**	**0.228**	0.193	0.239	**0.189**	**0.228**	0.190	0.234	**0.192**	**0.232**	0.195	0.234	**0.202**	**0.248**	0.218	0.278
	336	0.220	**0.263**	**0.218**	0.269	0.224	0.261	**0.215**	**0.260**	**0.241**	**0.268**	0.245	0.274	**0.254**	**0.288**	0.278	0.323
	720	0.292	**0.314**	**0.281**	0.316	0.298	0.316	**0.283**	**0.311**	**0.313**	**0.324**	0.318	0.326	**0.323**	**0.337**	0.372	0.396
	Avg	0.211	**0.248**	**0.210**	0.254	0.215	**0.248**	**0.208**	0.249	**0.224**	**0.253**	0.227	0.256	**0.235**	**0.271**	0.260	0.308

Table 10: Datasets and mapping details of M4 dataset.

Dataset	Sample Numbers (train set, test set)	variable Number	Prediction Length
M4 Yearly	(23000, 23000)	1	6
M4 Quarterly	(24000, 24000)	1	8
M4 Monthly	(48000, 48000)	1	18
M4 Weekly	(359, 359)	1	13
M4 Daily	(4227, 4227)	1	14
M4 Hourly	(414, 414)	1	48

rived from a single continuous sequence and thus likely to share similar temporal characteristics, the M4 dataset aggregates time series from varied sources. This results in significantly divergent temporal dynamics, making forecasting on M4 particularly challenging. Table 10 provides a statistical overview of the M4 dataset.

J.1 Implementation Details

Training Details. Our method is trained using the SMAPE loss function and optimized with the ADAM optimizer (Kingma & Ba, 2014), and a sigmoid learning rate scheduling strategy (Stitsyuk & Choi, 2025). The evaluation metrics employed are symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MASE), and overall weighted average (OWA). Following the approach of (Wu et al., 2023), we fix the lookback window of all models to twice the forecast horizon.

Table 11: Full results for the **short-term forecasting** task in the M4 dataset.

Models		**GLNet** (Ours)	xPatch (2025)	SimpleTM (2025a)	CARD (2024b)	U-Mixer (2024)	iTransformer (2024a)	TimeMixer (2024a)	PatchTST (2023)	MICN (2023)	DLinear (2023)	TimesNet (2023)	FEDformer (2022)	LightTS (2022)
Yearly	SMAPE	**13.292**	13.498	13.567	13.343	13.370	13.975	13.350	13.304	14.515	14.358	13.449	13.676	13.488
	MASE	**2.974**	3.039	3.048	3.023	3.011	3.202	3.027	2.996	3.358	3.128	2.989	3.104	3.061
	OWA	**0.781**	0.795	0.799	0.788	0.788	0.830	0.789	0.784	0.867	0.833	0.788	0.809	0.798
Quarterly	SMAPE	10.181	10.271	10.121	10.145	**10.012**	10.703	10.175	10.274	11.400	10.498	10.086	10.642	10.327
	MASE	1.188	1.209	1.179	1.192	1.165	1.279	1.198	**0.866**	1.381	1.239	1.181	1.251	1.196
	OWA	0.895	0.907	0.890	0.895	0.879	0.952	0.899	**0.824**	1.021	0.928	0.888	0.939	0.905
Monthly	SMAPE	**12.614**	12.815	12.995	12.839	13.027	16.554	12.757	12.814	13.822	13.395	12.783	13.920	12.738
	MASE	**0.934**	0.962	0.963	0.958	1.044	1.391	0.939	0.955	1.080	1.007	0.946	1.062	0.941
	OWA	**0.876**	0.896	0.903	0.896	0.959	1.228	0.884	0.894	0.987	0.938	0.888	0.982	0.884
Others	SMAPE	**4.498**	4.849	5.051	5.421	4.953	5.223	5.046	5.085	6.147	5.113	5.014	4.874	5.411
	MASE	**3.060**	3.270	3.448	3.347	3.291	3.449	3.423	3.105	4.238	3.667	3.364	3.206	3.471
	OWA	**0.956**	1.026	1.075	1.098	1.040	1.093	1.071	1.071	1.315	1.116	1.058	1.018	1.117
WA	SMAPE	**11.780**	11.963	12.040	11.938	11.812	13.990	11.888	11.931	13.016	12.507	11.901	12.625	11.965
	MASE	**1.570**	1.614	1.619	1.609	1.588	1.884	1.606	1.611	1.834	1.684	1.593	1.684	1.617
	OWA	**0.845**	0.863	0.867	0.860	0.851	1.008	0.858	0.861	0.959	0.901	0.855	0.906	0.864

Table 12: Performance comparison of models on short-term forecasting. The table reports the average values and pooled standard deviations of SMAPE, MASE, and OWA across Yearly, Quarterly, Monthly and other frequencies with three random seeds. The p-values from Type II ANOVA (with frequency as a blocking factor). Lower values indicate better performance.

Model	SMAPE (Pooled SD)	MASE (Pooled SD)	OWA (Pooled SD)	SMAPE p-value	MASE p-value	OWA p-value
xPatch	10.358 (0.052)	2.120 (0.020)	0.906 (00.006)	-	-	-
GLNet	10.146 (0.063)	2.039 (0.024)	0.877 (0.0006)	0.0000	0.0005	0.00000

Model Parameters. By default, the model depth is set to $k_1 = 4$, $k_2 = 3$ for short-term forecasting. In the patch embedding layer, we adopt a patch size of $P = 6$ and a stride of $S = 3$. For the final patch, padding is applied by replicating the last data point. The hidden dimension of the patch embedding is set to P. Within the GLB* blocks, the convolution dilation rate is fixed at 1, while the hidden dimension is set equal to the forecast length. The smoothing factor α for the EMA is set to 0.3.

Metric. For the evaluation of short-term forecasting performance, we adopt the SMAPE, MASE, and OWA as the evaluation metrics. The formulations of these metrics are given as follows:

$$\text{SMAPE} = \frac{200}{H} \sum_{i=1}^{H} \frac{|x_i - \widehat{x}_i|}{|x_i| + |\widehat{x}_i|} \tag{15}$$

$$\text{MASE} = \frac{1}{H} \sum_{i=1}^{H} \frac{|x_i - \widehat{x}_i|}{\frac{1}{L-p} \sum_{j=p+1}^{L} |x_j - x_{j-p}|} \tag{16}$$

$$\text{OWA} = \frac{1}{2} \left[\frac{\text{SMAPE}}{\text{SMAPE}_{\text{Naive2}}} + \frac{\text{MASE}}{\text{MASE}_{\text{Naive2}}} \right] \tag{17}$$

where p is the periodicity of data, H is the forecast horizon, L is the historical lookback window length, $\widehat{x}_i$ is the predicted value, and x_i is the actual value.

J.2 Main Results for Short Forecasting

Table 11 presents the comprehensive evaluation results on the M4 dataset across varied sampling frequencies. GLNet demonstrates superior adaptability to short-term forecasting, achieving the best performance in terms of the Weighted Average (WA) across all three evaluation metrics (SMAPE: 11.780, MASE: 1.570, OWA: 0.845). In a direct comparison with the most recent efficient baselines, GLNet surpasses xPatch, SimpleTM, and CARD in the overall weighted standing, validating

the effectiveness of our hierarchical Global-Local architecture even on shorter sequences. Furthermore, GLNet consistently outperforms complex Transformer-based models, such as iTransformer and TimeMixer, by a significant margin (e.g., reducing the WA OWA from 1.008 to 0.845). This suggests that the heavy self-attention mechanisms, often beneficial for capturing long-range dependencies, may not be necessary or optimal for the diverse and shorter series found in the M4 collection.

Breaking down the performance by granularity, GLNet excels particularly in datasets with distinct low-frequency trends and high-frequency seasonality. It secures the top rank in the Yearly and Monthly subsets, which constitute a large portion of the dataset, achieving an OWA of 0.781 and 0.876, respectively. Notably, in the other category (comprising Weekly, Daily, and Hourly data), GLNet exhibits remarkable robustness, outperforming the runner-up xPatch in SMAPE (4.498 $vs.$ 4.849) and achieving the lowest OWA of 0.956. While PatchTST shows competitive results in the Quarterly subset, GLNet remains the most balanced and consistent performer overall. The ability to dominate across such heterogeneous domains, from the slow-evolving yearly economics data to the rapid fluctuations in hourly metrics, confirms that our decoupled Trend and Seasonal Flow Networks can effectively generalize to diverse temporal dynamics without requiring task-specific architectural adjustments.

J.3 Stability Analysis for Short Term

To rigorously assess the robustness of the model under different random initializations, we adopt the statistical methodology described in Appendix H.2. Specifically, we conducte three independent experimental runs with different random seeds across the Yearly, Quarterly, Monthly, and other frequencies of the M4 dataset. We calculate the pooled standard deviation to quantify the performance fluctuation of the models. Furthermore, to determine whether the performance differences between GLNet and the comparative baselines are statistically significant, we employed ANOVA.

Table 12 presents the summary statistics and hypothesis testing results. Compared with xPatch, which also adopts a decomposition-based architecture, GLNet demonstrates superior average accuracy. Moreover, the p-values for all metrics are significantly below the 0.05 threshold. These near-zero p-values provide strong statistical evidence that the improvements achieved by GLNet are not due to chance in random seed selection but are statistically significant, thereby confirming the model's reliability and effectiveness in diverse STSF scenarios.

K Show Case

Figure 7 provide qualitative visualizations comparing the proposed GLNet model with recent SOTA models.

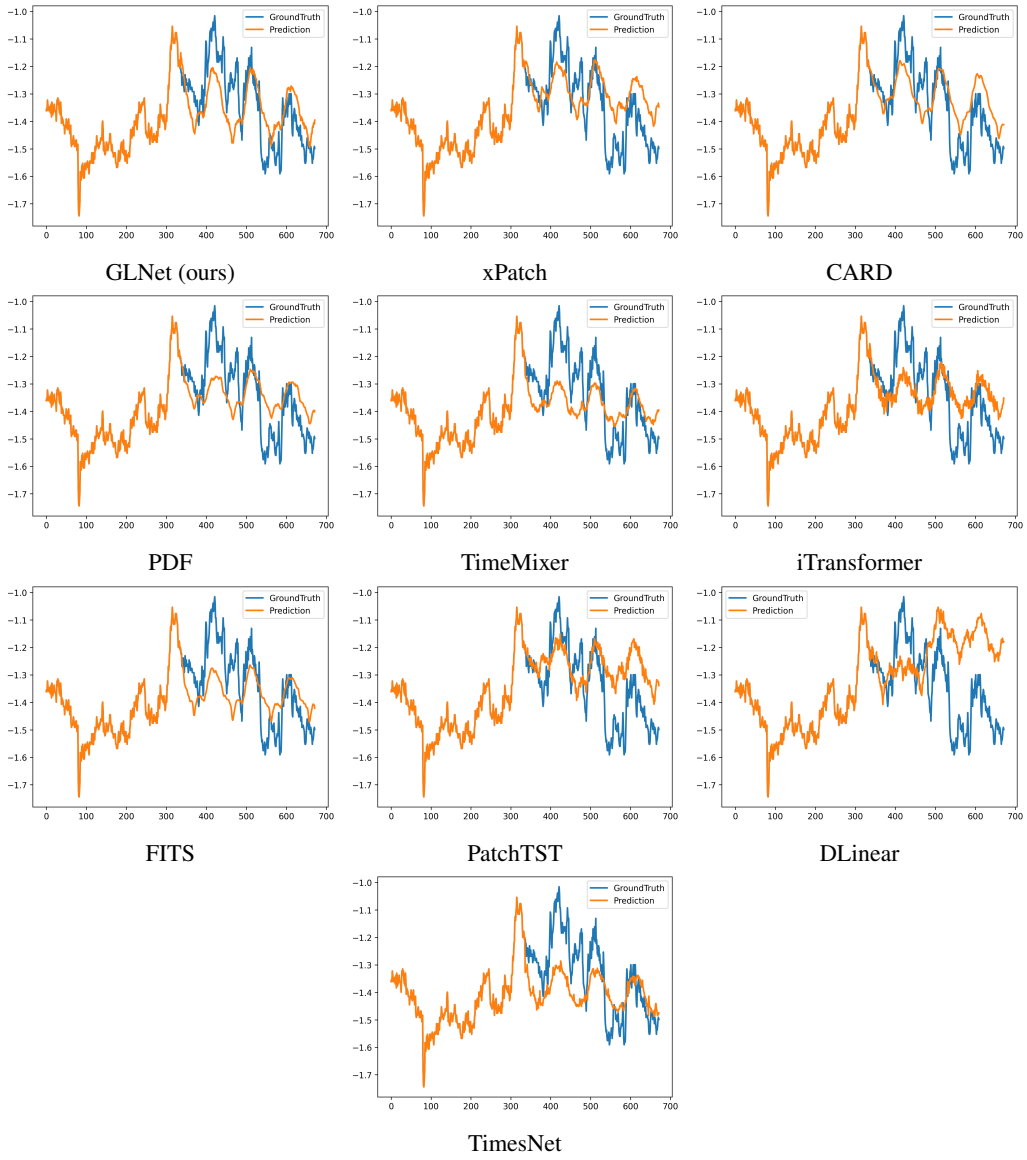


Figure 7: Visualization of ETTm1 predictions by different models under the input-336-predict-336 setting. The blue lines stand for the ground truth and the orange lines stand for predicted values.