NOT ALL HISTORY MATTERS: QUERY-CONDITIONED EVIDENCE RETRIEVAL FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models have shown promise for transferable time-series forecasting, yet most LLM-based forecasters still process the entire historical context, making long-horizon prediction vulnerable to redundant tokens, irrelevant temporal noise, and high computational cost. We aim to address the core gap of how an LLM forecaster can decide which parts of a long history are truly useful for a specific forecasting query and horizon. We propose **Forecast-Conditioned Evidence Seeking** (**ForeSeek**), a query-conditioned evidence retrieval framework that reformulates LLM-based forecasting as a retrieve-then-predict process. ForeSeek introduces Horizon-Aware Evidence Routing to select forecast-relevant historical segments, an Attention Conservation objective to preserve the predictive information flow of full-context reasoning, and a Dual-Path Forecasting design that combines evidence-based LLM reasoning with a lightweight temporal path for stable prediction. Experiments on seven benchmarks show that ForeSeek achieves strong long-term, few-shot, and zero-shot forecasting performance, while reducing LLM backbone FLOPs by 70.99% and total FLOPs by 52.56% on Time-LLM under the default Top-K ratio of 0.3. These results suggest that forecast-conditioned evidence selection offers an effective and efficient alternative to uniform full-context inference for LLM-based time-series forecasting.

1 INTRODUCTION

Time series forecasting is central to decision-making in dynamic systems, including energy, transportation, finance, weather, and industrial monitoring. The key challenge is to derive reliable long-horizon predictions from noisy historical observations, especially when in-domain supervision is limited or the target domain shifts (Deng et al., 2024).

Large Language Models (LLMs) offer a promising path toward more transferable forecasting systems. Their strong in-context learning and generalization abilities (Brown et al., 2020) have motivated recent studies that adapt pretrained language models to time series through prompt construction, temporal reprogramming, and semantic alignment (Zhou et al., 2023; Jin et al., 2023; Pan et al., 2024; Hu et al., 2025b;a). These works have shown that LLMs can be repurposed for time series forecasting, moving beyond purely task-specific forecasting architectures.

However, many representative LLM-based forecasters still rely on a *full-input* paradigm: after prompt construction, patch tokenization, or time-series reprogramming, the complete historical context is passed to the pretrained language backbone for prediction (Zhou et al., 2023; Jin et al., 2023; Pan et al., 2024; Hu et al., 2025b). This design implicitly assumes that the model can reliably identify useful forecasting evidence from a long mixed sequence containing prompts, temporal patches, redundant patterns, and irrelevant noise. This assumption is particularly fragile for time series, where predictive cues are often implicit, sparse, and horizon-dependent, appearing through lagged correlations, repeated local motifs, regime shifts, or multi-scale temporal structures rather than explicit semantic units (Zhou et al., 2022; Wu et al., 2022; Fons et al., 2024; Wu et al., 2025).

A natural alternative is to shorten the context, but simple truncation or generic prompt/token compression does not directly resolve this forecasting-specific issue. Recent-history truncation can discard delayed or periodic evidence, while generic compression methods are primarily designed to

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.

reduce token cost or preserve linguistic salience, rather than to model horizon-conditioned temporal relevance (Jiang et al., 2023). Moreover, long-context LLM studies show that simply providing more context does not guarantee reliable use of relevant information, since model performance can be sensitive to the position, density, and formatting of evidence (Liu et al., 2024; Fons et al., 2024). Thus, the key unresolved problem is not merely how to reduce input length, but how to determine *which historical evidence should be retained for a specific forecasting objective*.

This paper addresses the following question: *How can LLM-based forecasters reason over long and noisy time-series histories without processing all historical tokens uniformly?* As illustrated in Figure 1, an effective solution should improve the accuracy–efficiency trade-off rather than merely compressing the input. To this end, we propose **Forecast-Conditioned Evidence Seeking** (**ForeSeek**), a query-conditioned evidence retrieval framework that reformulates LLM-based forecasting from full-context reasoning into a retrieve-then-predict process. At a high level, ForeSeek identifies horizon-relevant historical evidence conditioned on the recent query window and the target forecasting horizon, then lets the LLM focus on the selected evidence instead of the entire history. To maintain stable prediction under context compression, ForeSeek further preserves the core information flow of full-context reasoning during training and combines evidence-based LLM reasoning with a lightweight temporal path.

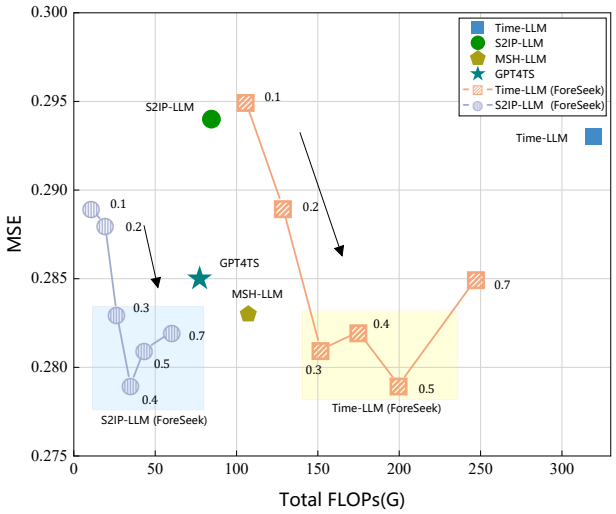


Figure 1: Pareto frontier of MSE versus FLOPs (G). Lower is better on both axes.

Our main contributions are summarized as follows:

- We identify query-conditioned historical redundancy as a key bottleneck in LLM-based long-context forecasting. Rather than asking only how to make LLMs understand time series, we highlight a complementary question: which parts of a long historical context should an LLM actually rely on for a given forecast?
- We introduce ForeSeek, a forecast-conditioned evidence retrieval framework that enables LLM forecasters to selectively reason over horizon-relevant historical segments. The framework shifts LLM-based forecasting from uniform full-context processing to objective-aware evidence seeking, while preserving prediction stability under compressed contexts.
- Extensive experiments on seven real-world benchmarks show that ForeSeek achieves strong forecasting accuracy in long-term, few-shot, and zero-shot settings. Under the default Top-K ratio of 0.3, ForeSeek reduces LLM backbone FLOPs by 70.99% and total FLOPs by 52.56% when applied to Time-LLM, demonstrating a favorable accuracy–efficiency trade-off.

2 Related Work

Time-Series Forecasting and Temporal Foundation Models. Time-series forecasting has progressed from statistical methods such as ARIMA (Box et al., 2015) and Prophet (Taylor & Letham, 2018) to neural architectures that model temporal dependencies with stronger inductive biases. Representative examples include recurrent models such as LSTM (Hochreiter & Schmidhuber, 1997), convolutional sequence models such as TCN (Bai et al., 2018), efficient Transformer variants such as Informer (Zhou et al., 2021), decomposition- and frequency-based models such as FEDformer (Zhou et al., 2022), 2D temporal-variation modeling in TimesNet (Wu et al., 2022), patch-based modeling

in PatchTST (Nie et al., 2022), and variate-wise attention in iTransformer (Liu et al., 2023). More recently, temporal foundation models (Ansari et al., 2024; Goswami et al., 2024) have explored large-scale pretraining for transferable time-series representation. These studies provide strong forecasting backbones, but mainly focus on representation learning or architectural design. In contrast, our work studies how an LLM-based forecaster should selectively use a long mixed prompt–patch context once such a context has already been constructed.

LLM Adaptation for Time-Series Forecasting. The success of pretrained language models has motivated a growing line of work that adapts LMs and LLMs to time-series forecasting. GPT4TS (Zhou et al., 2023) shows that frozen pretrained Transformer blocks can be repurposed for general time-series analysis, while LLMTime (Gruver et al., 2023) formulates numerical sequences as textual continuations for zero-shot forecasting. TEMPO (Cao et al., 2023) introduces prompt-based generative pretraining for temporal dynamics, and Time-LLM (Jin et al., 2023) reprograms time-series patches into the LLM embedding space with prompt-as-prefix guidance. Other methods improve temporal–linguistic alignment through text-prototype matching in TEST (Sun et al., 2023), semantic-space informed prompting in S2IP-LLM (Pan et al., 2024), data-efficient LLM alignment in LLM4TS (Chang et al., 2025), context-level structural alignment in Context-Alignment (Hu et al., 2025b), distribution-aware past–future alignment in Bridging Past and Future (Hu et al., 2025a), and multi-scale hypergraph alignment in MSH-LLM (Shang et al., 2026). These methods substantially advance the question of how to make LLMs understand time series. Nevertheless, most of them still feed the constructed historical context into the LLM in a largely uniform manner, implicitly relying on the backbone to identify useful evidence from redundant, noisy, and horizon-dependent temporal patterns. ForeSeek addresses this remaining gap by asking which historical segments should be retained for a specific forecasting query before LLM reasoning.

Long-Context Compression and Evidence Retrieval. Long-context efficiency has been studied in both time-series modeling and language modeling. For time series, Informer (Zhou et al., 2021), FEDformer (Zhou et al., 2022), and PatchTST (Nie et al., 2022) reduce long-sequence cost through efficient attention, frequency-domain modeling, or patch tokenization, while rank-based analyses suggest that temporal Transformers contain compressible structures (Yu et al., 2025). For LLMs, long contexts can be costly and unreliable; Lost in the Middle (Liu et al., 2024) shows that evidence usage depends strongly on context position, and LLMLingua (Jiang et al., 2023) reduces token cost by preserving salient prompt content. For forecasting, Global Temporal Retrieval (Cao et al., 2026) retrieves long-cycle temporal evidence beyond the look-back window. However, these methods do not address horizon-conditioned evidence selection over mixed prompt–patch LLM contexts. ForeSeek fills this gap by retrieving forecast-relevant evidence within the LLM forecasting pipeline while preserving the predictive information flow of full-context reasoning.

3 METHODOLOGY

We introduce **Forecast-Conditioned Evidence Seeking** (**ForeSeek**), a retrieval-based framework for efficient and stable LLM-based time-series forecasting. Given a historical series, ForeSeek first constructs a mixed token sequence by combining textual prompt tokens and temporal patch tokens. It then partitions the mixed context into segment-level evidence candidates and scores them using a horizon-aware forecasting query. The selected evidence segments are processed by an LLM-based evidence path, while a lightweight direct temporal path preserves coarse temporal information. During training, a full-context teacher provides attention and prediction supervision; during inference, the teacher is removed. The framework of our method is shown in Fig.2.

3.1 PROBLEM SETUP AND MIXED TOKEN REPRESENTATION

Given a multivariate historical time series $X \in \mathbb{R}^{T\times C}$, where T is the look-back length and C is the number of variables, the goal is to predict the future sequence $Y \in \mathbb{R}^{H\times C}$ over a horizon of length H. We denote by X_q the recent query window immediately preceding the prediction horizon. Following LLM-based forecasting pipelines, we convert X into temporal patch tokens and combine them with textual prompt tokens constructed from input statistics, such as trend, extrema, and lag information.

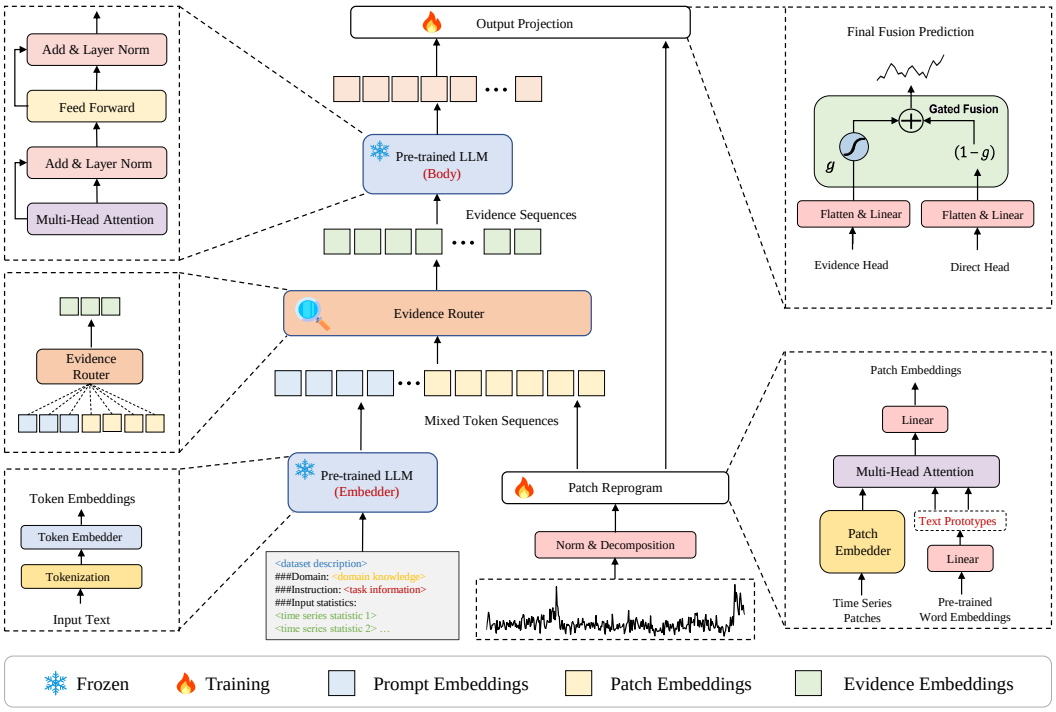


Figure 2: Overview of the proposed ForeSeek framework. ForeSeek introduces an Evidence Router that performs Horizon-Aware Evidence Routing to select forecasting-relevant evidence segments from the mixed prompt-patch context. The selected evidence is fed into the LLM-based evidence path, while a direct temporal path provides a global trend estimate. The two paths are adaptively fused to generate the final forecast. During training, a full-context teacher provides attention and prediction supervision, while during inference it is discarded.

Formally, let $P \in \mathbb{R}^{N_p \times d}$ denote the prompt-token embeddings and $E \in \mathbb{R}^{N_x \times d}$ denote the reprogrammed temporal patch tokens in the LLM embedding space. The mixed token sequence is

$$S = [P; E] \in \mathbb{R}^{N \times d}, \qquad N = N_p + N_x, \tag{1}$$

where $[\,;\,]$ denotes concatenation along the token dimension. Although this mixed context enables the LLM to jointly use textual guidance and temporal evidence, it also introduces substantial redundancy when the historical context is long. ForeSeek therefore aims to select a compact evidence sequence from S before LLM reasoning.

3.2 Evidence Router with Horizon-Aware Evidence Routing

ForeSeek uses an **Evidence Router** to identify forecast-relevant segments from the mixed context. We use $S^{(r)} = S$ as the retrievable sequence and partition it into N_s fixed-length contiguous segments with length ℓ and stride Δ:

$$G_i = S^{(r)}_{a_i:a_i+\ell-1} \in \mathbb{R}^{\ell \times d}, \qquad a_i = 1 + (i-1)\Delta, \qquad i = 1, \ldots, N_s. \tag{2}$$

Each segment is summarized into a compact index token through attention-weighted pooling. For the t-th token in segment G_i, we compute

$$\alpha_{i,t} = \frac{\exp(\mathbf{w}_a^\top G_{i,t})}{\sum_{u=1}^{\ell} \exp(\mathbf{w}_a^\top G_{i,u})}, \qquad k_i = \sum_{t=1}^{\ell} \alpha_{i,t} G_{i,t} \in \mathbb{R}^d, \tag{3}$$

where $\mathbf{w}_a \in \mathbb{R}^d$ is a learnable attention vector. This produces one index token k_i for each candidate evidence segment.

To make retrieval forecast-conditioned, ForeSeek constructs a horizon-aware query from the recent state and the target horizon. Let $E_q \subseteq E$ denote the patch tokens corresponding to X_q. We obtain

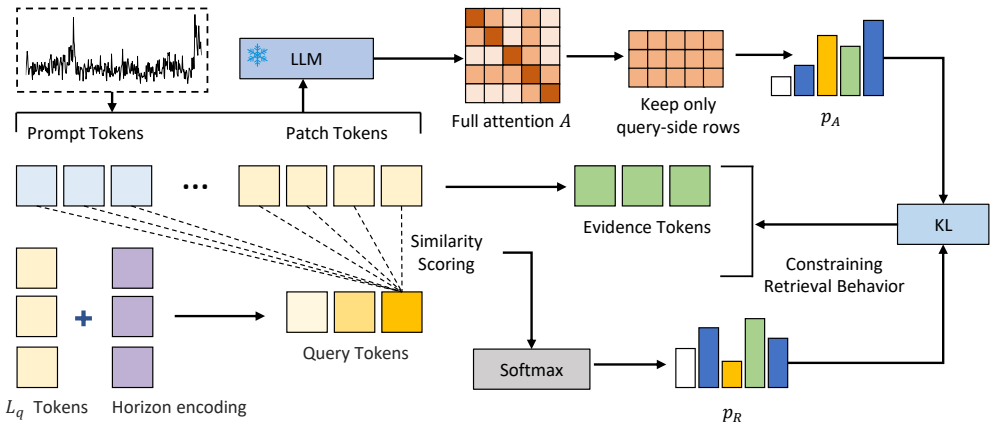


Figure 3: **Evidence Router with Horizon-Aware Evidence Routing (HAER)**. The mixed token sequence is partitioned into historical segments, and each segment is summarized into a lightweight index token. Given a horizon-aware forecasting query, HAER selects the most relevant segments to form the evidence sequence. Attention Conservation further aligns the router distribution with the full-context teacher attention, encouraging the selected evidence to preserve the core information flow of the original context.

a recent-state summary $e_q = \mathrm{MeanPool}(E_q)$ and a horizon representation $r_H = \psi(M_y)$, where M_y denotes the temporal markers and horizon descriptors of the prediction interval. The query and routing distribution are defined as

$$q = W_q(e_q + r_H), \qquad s_i = \frac{q^\top W_k k_i}{\sqrt{d}}, \qquad p_R(i) = \frac{\exp(s_i)}{\sum_{j=1}^{N_s} \exp(s_j)}. \tag{4}$$

Here, $W_q, W_k \in \mathbb{R}^{d\times d}$ are learnable projections, and $p_R \in \Delta^{N_s-1}$ is the router distribution over evidence segments.

Given a retention ratio ρ, ForeSeek selects $K = \lceil \rho N_s \rceil$ segments with the largest routing scores. The selected segments are restored to their original order and concatenated as the evidence sequence:

$$\mathcal{I}_K = \mathrm{TopK}(\{s_i\}_{i=1}^{N_s}, K), \qquad S_{\mathrm{ev}} = \mathrm{Concat}\big(\{G_i\}_{i\in \mathrm{sort}(\mathcal{I}_K)}\big). \tag{5}$$

The Top-K operation is used to construct the discrete evidence sequence, while the differentiable router distribution p_R is directly optimized by the attention-conservation objective introduced below. Thus, ForeSeek does not rely on back-propagating through the discrete Top-K selection.

3.3 Attention Conservation with a Full-Context Teacher

Hard evidence selection reduces input length but may disrupt the information flow of the original mixed context. To stabilize retrieval-based forecasting, we introduce **Attention Conservation**, which transfers full-context attention signals to the Evidence Router during training.

The Full-Context Teacher processes the complete mixed sequence S and produces a teacher prediction $\hat{Y}_T$ together with the final-layer attention maps. Let $A^{(h)} \in \mathbb{R}^{N\times N}$ denote the attention matrix of head h, and let $\mathcal{Q}$ denote the token indices corresponding to the recent query region. For each segment G_i, let $\mathcal{T}_i$ be its token-index set in $S^{(r)}$. We aggregate the attention mass from query tokens to each historical segment as

$$\tilde{a}_i = \sum_{t\in\mathcal{T}_i} \frac{1}{|\mathcal{H}|\,|\mathcal{Q}|} \sum_{h\in\mathcal{H}} \sum_{u\in\mathcal{Q}} A^{(h)}_{u,t}, \qquad p_A(i) = \frac{\tilde{a}_i}{\sum_{j=1}^{N_s} \tilde{a}_j + \varepsilon}, \tag{6}$$

where $\mathcal{H}$ is the set of attention heads and ε is a small constant for numerical stability. The resulting $p_A \in \Delta^{N_s-1}$ provides a full-context estimate of which segments are attended to by the forecasting query.

We train the Evidence Router to match this teacher distribution:

$$\mathcal{L}_{\text{align}} = \text{KL}(\text{sg}(p_A) \,\|\, p_R)\,, \tag{7}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. This objective encourages the compressed evidence sequence to preserve the main query-side information flow of the full context, rather than selecting segments solely by local saliency.

3.4 Dual-Path Forecasting and Joint Optimization

ForeSeek uses two complementary prediction paths. The **evidence path** feeds the selected evidence sequence S_{ev} into the LLM-based forecasting backbone and produces $\hat{Y}_{\text{ev}}$. The **direct temporal path** bypasses the LLM and predicts from temporal patch representations E, producing $\hat{Y}_{\text{dir}}$. The final forecast is obtained through a learnable gated fusion:

$$\hat{Y} = g\hat{Y}_{\text{ev}} + (1-g)\hat{Y}_{\text{dir}}, \qquad g = \sigma(\gamma), \qquad \gamma \in \mathbb{R}. \tag{8}$$

The evidence path focuses on horizon-relevant local evidence, while the direct path provides a lightweight temporal estimate that helps preserve global trends and low-frequency patterns.

The training objective contains three terms. The first is the forecasting loss on the fused prediction. The second is the attention-conservation loss for training the Evidence Router. The third is a prediction consistency loss that distills the full-context teacher prediction into the evidence path:

$$\mathcal{L}_{\text{task}} = \frac{1}{HC}\|\hat{Y} - Y\|_F^2, \qquad \mathcal{L}_{\text{cons}} = \text{SmoothL1}\big(\hat{Y}_{\text{ev}}, \text{sg}(\hat{Y}_T)\big), \tag{9}$$

and the overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}}, \tag{10}$$

where λ_{align} and λ_{cons} balance the auxiliary objectives. During inference, the Full-Context Teacher is removed; ForeSeek forecasts using only the lightweight Evidence Router and the dual-path network, thereby reducing redundancy in mixed-token LLM inputs.

4 Experiments

We evaluate ForeSeek on long-term forecasting, few-shot forecasting, zero-shot transfer, ablation studies, difficulty-stratified diagnostics, and inference efficiency. The goal is to verify whether the Evidence Router with HAER can improve both the accuracy and efficiency of LLM-based time-series forecasters.To ensure a fair comparison, we use GPT-2 as the default backbone for all compared LLM-based methods. In addition, we apply our framework to both Time-LLM and S2IP-LLM.

4.1 Long-Term Forecasting

Setup. We evaluate ForeSeek on seven standard benchmarks, including ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, and Traffic, following standard long-term forecasting protocols (Zhou et al., 2021; Wu et al., 2022). The input length is 512, with prediction horizons in $\{96, 192, 336, 720\}$, and performance is measured by MSE and MAE. We compare with LLM-based forecasters, including Time-LLM (Jin et al., 2023), S2IP-LLM (Pan et al., 2024), and GPT4TS (Zhou et al., 2023), and non-LLM baselines, including DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2022), FEDformer (Zhou et al., 2022), Stationary (Liu et al., 2022), and iTransformer (Liu et al., 2023). GPT-2 (Radford et al., 2019) is the default LLM backbone. ForeSeek is instantiated on Time-LLM and S2IP-LLM, with HAER and a Top-K ratio of 0.3 by default. The full-context teacher is discarded at inference.

Results. As shown in Table 1, ForeSeek consistently improves Time-LLM on several benchmarks, achieving the best or tied-best MSE on ETTh1, ETTh2, ETTm2, and Electricity, while remaining competitive on Weather and Traffic. It also improves S2IP-LLM on most datasets, indicating that the proposed retrieval mechanism generalizes across different LLM-based forecasting pipelines. Overall, these results support our hypothesis that query-relevant evidence selection can reduce redundant historical context without sacrificing predictive accuracy.

Table 1: Full long-term forecasting results. We set the forecasting horizons $H \in \{96, 192, 336, 720\}$. A lower value indicates better performance. Red: the best, Blue with underline: the second best. Our full results are in Appendix C.1.

Dataset	ForeSeek (Ours)				LLM-Based						Others									
	Time-LLM		S2IP-LLM		S2IP-LLM		Time-LLM		GPT4TS		iTransformer		DLinear		TimesNet		FEDformer		Stationary	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.421	0.439	0.430	0.442	0.439	0.450	0.426	0.435	0.427	0.426	0.451	0.462	0.422	0.437	0.458	0.450	0.440	0.460	0.570	0.537
ETTh2	0.343	0.392	0.345	0.392	0.364	0.408	0.361	0.398	0.354	0.394	0.382	0.414	0.431	0.446	0.414	0.427	0.437	0.449	0.526	0.516
ETTm1	0.366	0.399	0.359	0.382	0.346	0.380	0.354	0.384	0.352	0.383	0.370	0.399	0.357	0.378	0.400	0.406	0.448	0.452	0.481	0.456
ETTm2	0.253	0.315	0.259	0.319	0.266	0.328	0.275	0.334	0.266	0.326	0.272	0.331	0.267	0.333	0.291	0.333	0.305	0.349	0.306	0.347
Weather	0.233	0.268	0.240	0.276	0.229	0.267	0.237	0.269	0.237	0.270	0.304	0.335	0.248	0.300	0.259	0.287	0.309	0.360	0.288	0.314
Electricity	0.166	0.266	0.167	0.264	0.169	0.267	0.167	0.264	0.167	0.263	0.203	0.298	0.166	0.263	0.192	0.295	0.214	0.327	0.193	0.296
Traffic	0.412	0.291	0.403	0.288	0.404	0.287	0.407	0.289	0.414	0.294	0.389	0.295	0.433	0.295	0.620	0.336	0.610	0.376	0.624	0.340

4.2 Few-Shot Forecasting

Setups. A growing body of recent work has shown that LLMs exhibit notable advantages in few-shot learning (Jin et al., 2023). In this section, we evaluate ForeSeek under limited-data training scenarios and examine whether query-conditioned evidence retrieval can improve forecasting accuracy with fewer effective input tokens.

Results. Tables 2 and 3 show that ForeSeek improves Time-LLM and S2IP-LLM under 10% training data, indicating that query-conditioned retrieval remains effective with limited supervision. Under the more challenging 5% setting, the gains are smaller but ForeSeek remains competitive, particularly on ETTh2 and ETTm2, suggesting that router learning becomes harder when supervision is extremely scarce.

Table 2: Few-shot learning on 10% training data. All results are averaged from four different forecasting horizons: $\mathbf{H} \in \{96, 192, 336, 720\}$. A lower value indicates better performance. Red: the best, Blue with underline: the second best. Our full results are in Appendix C.2.

Dataset	ForeSeek (Ours)				LLM-Based						Others									
	Time-LLM		S2IP-LLM		S2IP-LLM		Time-LLM		GPT4TS		iTransformer		DLinear		TimesNet		FEDformer		Stationary	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.587	0.525	0.632	0.539	0.660	0.551	0.785	0.553	0.590	0.525	0.910	0.645	0.691	0.600	0.869	0.628	0.639	0.561	0.915	0.639
ETTh2	0.382	0.416	0.416	0.439	0.434	0.448	0.424	0.441	0.397	0.421	0.489	0.483	0.605	0.538	0.479	0.465	0.466	0.475	0.462	0.455
ETTm1	0.432	0.424	0.435	0.433	0.461	0.447	0.487	0.461	0.464	0.441	0.728	0.565	0.411	0.429	0.677	0.537	0.722	0.605	0.797	0.578
ETTm2	0.281	0.331	0.285	0.333	0.297	0.342	0.305	0.344	0.293	0.335	0.336	0.373	0.316	0.368	0.320	0.353	0.463	0.488	0.332	0.366

Table 3: Few-shot learning on 5% training data. All results are averaged over the available forecasting horizons. A lower value indicates better performance. Red: the best, Blue with underline: the second best. Our full results are in Appendix C.2.

Dataset	ForeSeek (Ours)				LLM-Based						Others									
	Time-LLM		S2IP-LLM		S2IP-LLM		Time-LLM		GPT4TS		iTransformer		DLinear		TimesNet		FEDformer		Stationary	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.697	0.550	0.866	0.617	0.799	0.596	0.648	0.549	0.681	0.560	1.070	0.710	0.750	0.611	0.925	0.647	0.658	0.562	0.943	0.646
ETTh2	0.391	0.419	0.473	0.470	0.488	0.478	0.398	0.426	0.400	0.433	0.488	0.475	0.827	0.615	0.463	0.454	0.463	0.454	0.470	0.489
ETTm1	0.487	0.461	0.583	0.520	0.511	0.478	0.477	0.451	0.472	0.450	0.784	0.596	0.400	0.417	0.717	0.561	0.730	0.592	0.857	0.598
ETTm2	0.298	0.345	0.331	0.371	0.311	0.353	0.307	0.348	0.308	0.346	0.356	0.388	0.399	0.426	0.344	0.372	0.381	0.404	0.341	0.372

4.3 Zero-Shot Forecasting

Setups. We evaluate zero-shot transfer on eight ETT cross-domain tasks by training on one dataset and directly testing on another without fine-tuning. The tasks cover transfers among ETTh1, ETTh2, ETTm1, and ETTm2, and results are reported in Table 4.

Results. As shown in Table 4, ForeSeek achieves strong zero-shot generalization across all transfer settings. The ForeSeek variants obtain the best MSE on all eight tasks, with S2IP-LLM+ForeSeek ranking first on six of them. Compared with the original backbones, ForeSeek consistently improves

Table 4: Zero-shot learning results on ETT datasets. Red: the best, Blue with underline: the second best. Appendix C.3 shows our detailed results.

Methods	ForeSeek (Ours)				LLM-Based								Others					
	Time-LLM		S2IP-LLM		S2IP-LLM		Time-LLM		LLMTime		GPT4TS		iTransformer		DLinear		TimesNet	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1 → *ETTh2*	0.351	0.392	0.344	0.391	0.363	0.406	0.384	0.409	0.992	0.708	0.406	0.422	0.457	0.445	0.493	0.488	0.421	0.431
ETTh1 → *ETTm2*	0.311	0.365	0.256	0.316	0.321	0.376	0.317	0.370	1.867	0.869	0.325	0.363	0.360	0.390	0.415	0.452	0.327	0.361
ETTh2 → *ETTh1*	0.567	0.514	0.492	0.488	0.586	0.542	0.663	0.540	1.961	0.981	0.757	0.578	0.868	0.625	0.703	0.574	0.865	0.621
ETTh2 → *ETTm2*	0.305	0.361	0.294	0.352	0.306	0.362	0.339	0.371	1.867	0.869	0.335	0.370	0.335	0.382	0.328	0.386	0.342	0.376
ETTm1 → *ETTh2*	0.384	0.417	0.365	0.402	0.410	0.426	0.440	0.449	0.992	0.708	0.433	0.439	0.455	0.458	0.464	0.475	0.457	0.454
ETTm1 → *ETTm2*	0.268	0.324	0.262	0.318	0.274	0.327	0.311	0.343	1.867	0.869	0.313	0.348	0.319	0.363	0.335	0.389	0.322	0.354
ETTm2 → *ETTh2*	0.373	0.412	0.379	0.410	0.389	0.420	0.429	0.448	0.992	0.708	0.435	0.443	0.432	0.447	0.455	0.471	0.435	0.443
ETTm2 → *ETTm1*	0.470	0.439	0.479	0.456	0.638	0.530	0.588	0.503	1.933	0.984	0.769	0.567	0.706	0.572	0.649	0.537	0.769	0.567

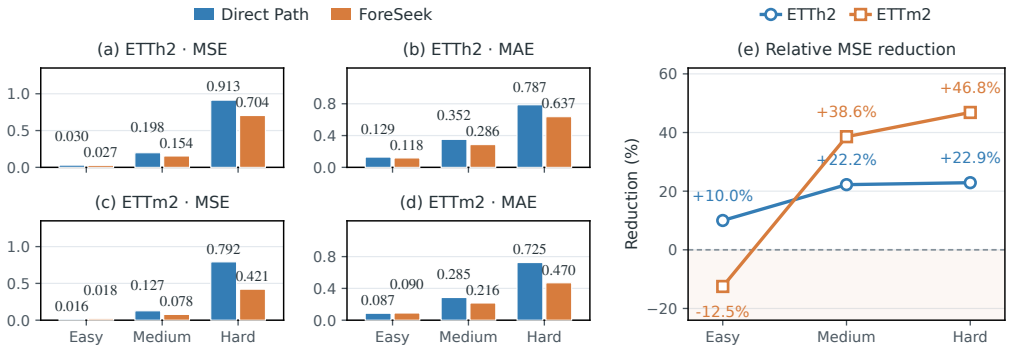


Figure 4: Difficulty-stratified diagnostic analysis on ETTh2 and ETTm2. Test samples are divided into Easy, Medium, and Hard subsets according to the baseline error distribution.

performance, indicating that query-conditioned evidence retrieval enhances cross-domain transferability and robustness under unseen target domains.

4.4 Ablation Studies

We conduct ablations on ETTh2 and ETTm2 to evaluate ForeSeek's key components, including the backbone, evidence source, training objectives, and dual-path design. All variants are implemented on Time-LLM, and results are shown in Table 5.

Components. Under the same ForeSeek framework, GPT-2 outperforms Llama and Qwen2.5, indicating that the gains are mainly due to query-conditioned retrieval rather than larger model scale. Replacing HAER-selected evidence with random or recent evidence consistently degrades performance, especially on ETTm2, showing that forecast-relevant evidence selection is more effective than naive context reduction.

Objectives and architecture. Removing attention alignment or consistency loss worsens performance, confirming the benefit of full-context supervision for stable sparse retrieval. The dual-path design is also crucial: without the direct path, MSE increases to 0.332 on ETTh2 and 0.200 on ETTm2, while the direct-only variant performs worse. This suggests that the direct path captures coarse trends, whereas the evidence path contributes fine-grained query-relevant information.

Difficulty analysis. As shown in Figure 4, ForeSeek brings larger improvements on harder samples. On ETTh2, the relative MSE reduction increases from 10.0% on Easy samples to over 22% on Medium and Hard samples. On ETTm2, the gains reach 38.6% and 46.8% on Medium and Hard samples, respectively, indicating that evidence retrieval is particularly helpful when useful historical cues are harder to identify.

These results verify that ForeSeek benefits from query-conditioned evidence retrieval, full-context supervision, and dual-path forecasting, rather than model scaling or simple context shortening.

Table 5: Ablation study of ForeSeek on ETTh2 and ETTm2. All variants are implemented on top of Time-LLM. The default GPT-2 setting corresponds to the full ForeSeek model. For evidence-source variants, all models are evaluated under the prediction horizon of 96. Light-blue subscripts indicate relative error increases (%) over the default GPT-2 model.

Category	Variant	Path Used		ETTh2		ETTm2	
		Direct	Evidence	MSE	MAE	MSE	MAE
Backbone	GPT-2 (Default; 12)	✓	ForeSeek	**0.281**	**0.341**	**0.163**	**0.254**
	Llama (12)	✓	ForeSeek	0.338$_{+20.3\%}$	0.387$_{+13.5\%}$	0.197$_{+20.9\%}$	0.286$_{+12.6\%}$
	Qwen2.5 (12)	✓	ForeSeek	0.343$_{+22.1\%}$	0.387$_{+13.5\%}$	0.186$_{+14.1\%}$	0.273$_{+7.5\%}$
Evidence source	Direct-only	✓	×	0.362$_{+28.8\%}$	0.416$_{+22.0\%}$	0.293$_{+79.8\%}$	0.358$_{+40.9\%}$
	Random Evidence	✓	Random	0.290$_{+3.2\%}$	0.358$_{+5.0\%}$	0.228$_{+39.9\%}$	0.311$_{+22.4\%}$
	Recent Evidence	✓	Recent	0.288$_{+2.5\%}$	0.355$_{+4.1\%}$	0.246$_{+50.9\%}$	0.328$_{+29.1\%}$
Loss	w/o attention alignment	✓	ForeSeek	0.288$_{+2.5\%}$	0.352$_{+3.2\%}$	0.179$_{+9.8\%}$	0.276$_{+8.7\%}$
	w/o consistency loss	✓	ForeSeek	0.293$_{+4.3\%}$	0.358$_{+5.0\%}$	0.172$_{+5.5\%}$	0.266$_{+4.7\%}$
Architecture	w/o direct path	×	ForeSeek	0.332$_{+18.1\%}$	0.384$_{+12.6\%}$	0.200$_{+22.7\%}$	0.301$_{+18.5\%}$
	w/o evidence path	✓	×	0.362$_{+28.8\%}$	0.416$_{+22.0\%}$	0.293$_{+79.8\%}$	0.358$_{+40.9\%}$

4.5 Efficiency

ForeSeek improves efficiency by replacing full-context LLM inference with query-conditioned evidence compression. Instead of passing all prompt and patch tokens to the frozen LLM, it routes only a compact set of relevant evidence segments, so the LLM cost scales with the Top-K ratio.

Figure 5 shows that ForeSeek significantly reduces token usage and computation. At the default Top-K ratio of 0.3, it reduces LLM FLOPs by about 71% for both backbones, with total FLOPs savings of 52.56% on Time-LLM and 68.77% on S2IP-LLM.

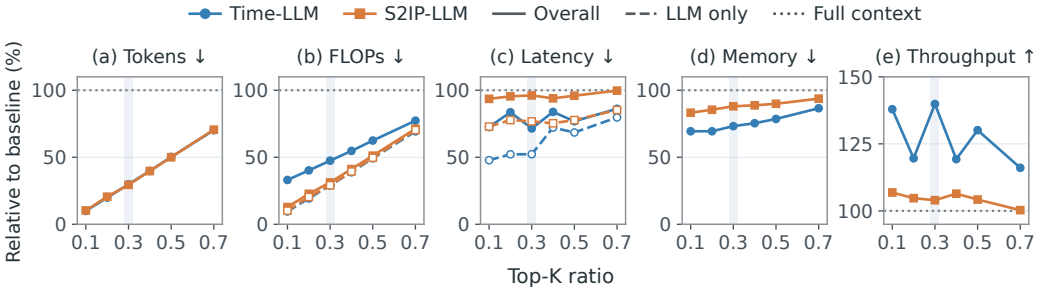


Figure 5: Efficiency of ForeSeek across Top-K ratios, normalized to each backbone's full-context baseline (100%, dotted line). Lower is better except for throughput; shading marks the default ratio 0.3. Solid/dashed lines denote overall/LLM-only FLOPs and latency. At 0.3, total FLOPs are 151.68G for Time-LLM and 26.41G for S2IP-LLM, versus 319.71G and 84.56G at full context. Raw values are in Table 14.

5 Conclusion

We presented ForeSeek, a query-conditioned evidence retrieval framework for efficient LLM-based long-horizon forecasting. Its retrieve-then-predict design combines Attention Conservation and Dual-Path Forecasting to stabilize prediction with compressed contexts. Across seven benchmarks, ForeSeek achieves competitive accuracy, particularly in few-shot and zero-shot settings, while reducing token usage and computational cost. Difficulty-stratified analysis further shows larger gains on more challenging samples, suggesting that selective historical evidence retrieval is particularly useful when forecasting cues are harder to identify. Future work will explore ultra-long contexts, adaptive evidence budgets, and multi-scale routing.

AI USE STATEMENT

Generative AI tools were used for language polishing and translation of the manuscript. OpenAI Codex was additionally used to assist with conversion to the ICLR 2027 LaTeX format, including template integration, bibliography formatting, and layout checks.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

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

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

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

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, 16(3):1–20, 2025.

Songgaojun Deng, Olivier Sprangers, Ming Li, Sebastian Schelter, and Maarten De Rijke. Domain generalization in time series forecasting. *ACM Transactions on Knowledge Discovery from Data*, 18(5):1–24, 2024.

Elizabeth Fons, Rachneet Kaur, Soham Palande, Zhen Zeng, Tucker Balch, Manuela Veloso, and Svitlana Vyetrenko. Evaluating large language models on time series feature understanding: A comprehensive taxonomy and benchmark. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21598–21634, 2024.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, pp. 16115–16152. PMLR, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in neural information processing systems*, 36:19622–19635, 2023.

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

Yifan Hu, Jie Yang, Tian Zhou, Peiyuan Liu, Yujin Tang, Rong Jin, and Liang Sun. Bridging past and future: Distribution-aware alignment for time series forecasting. *arXiv preprint arXiv:2509.14181*, 2025a.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-alignment: Activating and enhancing llm capabilities in time series. *arXiv preprint arXiv:2501.03747*, 2025b.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

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

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

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*, 2023.

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.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. s^2 IP-LLM: Semantic space informed prompt learning with LLM for time series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Zongjiang Shang, Dongliang Cui, Binqing Wu, and Ling Chen. Multi-scale hypergraph meets llms: Aligning large language models for time series analysis. *arXiv preprint arXiv:2602.04369*, 2026.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*, 2023.

Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

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.

Wen Wu, Ziyang Zhang, Liwei Liu, Xuenan Xu, Jimin Zhuang, Ke Fan, Qitan Lv, Junlin Liu, Chen Zhang, Zheqi Yuan, et al. Scits: Scientific time series understanding and generation with llms. *arXiv preprint arXiv:2510.03255*, 2025.

Annan Yu, Danielle C Maddix, Boran Han, Xiyuan Zhang, Abdul Fatir Ansari, Oleksandr Shchur, Christos Faloutsos, Andrew Gordon Wilson, Michael W Mahoney, and Yuyang Wang. Understanding transformers for time series: Rank structure, flow-of-ranks, and compressibility. *arXiv preprint arXiv:2510.03358*, 2025.

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.

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.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Appendix contents

CONTENTS PAGE

A Method Details **14**
- A.1 Shared Formulation . . . 14
- A.2 Time-LLM Instantiation . . . 15
- A.3 S2IP-LLM Instantiation . . . 15
- A.4 Comparison of Backbone Instantiations . . . 16
- A.5 Training and Inference Algorithm . . . 17

B Experimental Setup **18**
- B.1 Datasets and Splits . . . 18
- B.2 Baseline Models . . . 18
- B.3 Implementation and Training Settings . . . 19
- B.4 Forecasting Metrics . . . 19

C Complete Forecasting Results **20**
- C.1 Long-Term Forecasting . . . 20
- C.2 Few-Shot Forecasting . . . 21
- C.3 Zero-Shot Transfer . . . 22
- C.4 Target-Trained Evaluation on GIFT-Eval . . . 23

D Efficiency and Computational Cost **24**
- D.1 Metric Definitions and Measurement Protocols . . . 24
- D.2 Inference Efficiency across Retention Ratios . . . 25
- D.3 Training Cost and Teacher Overhead . . . 25

E Model Analysis **26**
- E.1 Matched-Budget Selector Comparison . . . 26
- E.2 Selected-Evidence Categories . . . 26
- E.3 Teacher Supervision and Component Ablations . . . 27
- E.4 Stability and Sensitivity . . . 27
  - Random-Seed Stability . . . 27
  - Evidence-Retention Ratio . . . 28
  - Segment Length and Stride . . . 28

F Qualitative Visualizations **29**
- F.1 Forecasts on ETT and Electricity . . . 29
- F.2 Forecasts on Traffic . . . 30
- F.3 Evidence-Routing Case Studies . . . 31

A METHOD DETAILS

A.1 SHARED FORMULATION

Although ForeSeek follows the same index-then-expand retrieval principle across different LLM-based forecasters, its concrete realization depends on how each backbone constructs contextual tokens and parameterizes the full-context prediction branch. In the unified formulation of the main text, let $\mathbf{S}$ denote the backbone-specific contextual token sequence and let $\mathbf{S}^{(r)} \subseteq \mathbf{S}$ denote the retrievable part. ForeSeek partitions $\mathbf{S}^{(r)}$ into overlapping segments

$$\mathbf{G}_i = \mathbf{S}^{(r)}_{a_i:a_i+\ell-1}, \qquad a_i = 1 + (i-1)\Delta \tag{11}$$

where ℓ and Δ denote the segment length and stride, respectively. Each segment is summarized into an index vector

$$\mathbf{z}_i = \mathrm{Pool}(\mathbf{G}_i) \tag{12}$$

and scored by a horizon-conditioned query $\mathbf{q}$:

$$s_i = \phi(\mathbf{q}, \mathbf{z}_i) \tag{13}$$

The Top-K evidence segments are then selected as

$$\mathcal{I}_K = \mathrm{TopK}(\{s_i\}_{i=1}^{N_s}, K) \tag{14}$$

which form the evidence sequence

$$\mathbf{S}^{(e)} = \mathrm{Concat}\big(\{\mathbf{G}_i\}_{i\in\mathcal{I}_K}\big) \tag{15}$$

Importantly, both Time-LLM and S2IP-LLM instantiate ForeSeek with the same Evidence Router and HAER mechanism. The forecasting network follows a dual-path design, consisting of an evidence path and a direct temporal path. A full-context teacher is used only during training to supervise the Evidence Router and the evidence path, and is removed during inference.

Specifically, the training-only full-context teacher processes the complete contextual sequence and produces a teacher prediction:

$$\hat{\mathbf{Y}}^{(T)} = f_T(\mathbf{S}) \tag{16}$$

The evidence path operates on the retrieved evidence sequence:

$$\hat{\mathbf{Y}}^{(e)} = f_e(\mathbf{S}^{(e)}) \tag{17}$$

while the direct temporal path bypasses the LLM backbone and predicts from the temporal representations:

$$\hat{\mathbf{Y}}^{(d)} = f_d(\mathbf{X}) \tag{18}$$

The final forecast is obtained by gated fusion between the evidence path and the direct temporal path:

$$g = \sigma(\gamma), \qquad \hat{\mathbf{Y}} = g\hat{\mathbf{Y}}^{(e)} + (1-g)\hat{\mathbf{Y}}^{(d)} \tag{19}$$

Here, $\hat{\mathbf{Y}}^{(T)}$ is used only for training-time supervision and is not involved in inference.

Accordingly, the overall training objective can be written as

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_{align}\mathcal{L}_{align} + \lambda_{cons}\mathcal{L}_{cons} + \lambda_{aux}\mathcal{L}_{aux} \tag{20}$$

with

$$\mathcal{L}_{task} = \left\|\hat{\mathbf{Y}} - \mathbf{Y}\right\|_2^2, \qquad \mathcal{L}_{aux} = \left\|\hat{\mathbf{Y}}^{(T)} - \mathbf{Y}\right\|_2^2 \tag{21}$$

The consistency loss distills the prediction behavior of the full-context teacher into the evidence path:

$$\mathcal{L}_{cons} = \mathrm{SmoothL1}\left(\hat{\mathbf{Y}}^{(e)}, \hat{\mathbf{Y}}^{(T)}\right) \tag{22}$$

The attention alignment loss $\mathcal{L}_{align}$ follows the same definition as in the main text.

A.2 TIME-LLM INSTANTIATION

For Time-LLM, the contextual sequence is instantiated as the concatenation of textual prompt tokens and reprogrammed patch tokens,

$$\mathbf{S}_{\text{Time}} = [\mathbf{P}_{\text{text}}; \mathbf{E}_{\text{patch}}], \tag{23}$$

where $\mathbf{P}_{\text{text}}$ is obtained from prompt templates constructed from input statistics, and $\mathbf{E}_{\text{patch}}$ is produced by patch embedding followed by reprogramming into the LLM embedding space. Given future temporal markers $\mathbf{M}_y$, the horizon-conditioned query is formed as

$$\mathbf{h}_y = \mathbf{W}_h \left(\frac{1}{H} \sum_{t=1}^{H} \mathbf{M}_y(t) \right), \qquad \mathbf{q}_{\text{Time}} = \mathbf{W}_q \mathbf{h}_y. \tag{24}$$

In our implementation, the segment index is instantiated by learnable pooling, for which attention pooling is used in the reported models:

$$\alpha_{i,j} = \frac{\exp(\mathbf{w}^\top \mathbf{g}_{i,j})}{\sum_{u=1}^{\ell} \exp(\mathbf{w}^\top \mathbf{g}_{i,u})}, \qquad \mathbf{z}_i = \sum_{j=1}^{\ell} \alpha_{i,j} \mathbf{g}_{i,j}, \tag{25}$$

and the relevance score is evaluated by

$$s_i = \frac{\mathbf{q}_{\text{Time}}^\top \mathbf{W}_k \mathbf{z}_i}{\sqrt{d}}. \tag{26}$$

Under this instantiation, the training-only full-context teacher f_T operates on the complete mixed token sequence $\mathbf{S}_{\text{Time}}$, while the evidence branch f_e operates on the retrieved evidence sequence $\mathbf{S}_{\text{Time}}^{(e)}$. The direct temporal path f_d bypasses the LLM backbone and predicts from temporal patch representations.

A.3 S2IP-LLM INSTANTIATION

For S2IP-LLM, the same dual-path formulation is retained, but the contextual sequence is decomposition-aware and prompt-enhanced. Specifically, the input series is first decomposed into trend, seasonal, and residual components,

$$\mathbf{X} \mapsto \left(\mathbf{X}^{(tr)}, \mathbf{X}^{(se)}, \mathbf{X}^{(re)} \right), \tag{27}$$

which are jointly encoded into decomposition-aware patch tokens. Combined with prompt-pool outputs, this yields

$$\mathbf{S}_{\text{S2IP}} = [\mathbf{P}_{\text{S2IP}}; \mathbf{E}_{\text{decomp}}]. \tag{28}$$

The horizon-conditioned query is similarly defined as

$$\mathbf{h}_y = \mathbf{W}_h \left(\frac{1}{H} \sum_{t=1}^{H} \mathbf{M}_y(t) \right), \qquad \mathbf{q}_{\text{S2IP}} = \mathbf{W}_q \mathbf{h}_y. \tag{29}$$

The retrievable scope can be either the mixed sequence $\mathbf{S}_{\text{S2IP}}$ or its patch-only part $\mathbf{E}_{\text{decomp}}$, but the segment indexing and Top-K selection equations remain unchanged. The key model-specific difference lies in the parameterization of the training-only full-context teacher and decoding heads: in S2IP-LLM, f_T is instantiated as a decoder over the complete prompt-enhanced contextual sequence, whereas f_e decodes from the retrieved evidence sequence. The direct temporal path f_d remains a lightweight non-LLM forecasting head. Therefore, both Time-LLM and S2IP-LLM share the same dual-path ForeSeek formulation, while differing only in the construction of contextual tokens and in the parameterization of the full-context branch.

In other words, the retrieval equations are backbone-invariant, whereas the contextual tokenization and branch-specific decoding heads are backbone-specific.

A.4 Comparison of Backbone Instantiations

Table 6: Shared and backbone-specific components of ForeSeek when instantiated on Time-LLM and S2IP-LLM. ForeSeek uses the same retrieval, training, and fusion principles across backbones, while differing mainly in how contextual tokens are constructed and decoded.

Component	**Type**	**Time-LLM (ForeSeek)**	**S2IP-LLM (ForeSeek)**
Contextual tokens	Backbone-specific	Textual prompt tokens and reprogrammed patch tokens, denoted as $[\mathbf{P}_{\text{text}}; \mathbf{E}_{\text{patch}}]$.	Prompt-pool tokens and decomposition-aware patch tokens, denoted as $[\mathbf{P}_{\text{S2IP}}; \mathbf{E}_{\text{decomp}}]$.
Retrievable scope	Backbone-specific	Mixed prompt–patch context constructed by Time-LLM.	Prompt-enhanced decomposition-aware context constructed by S2IP-LLM.
Horizon-aware query	Shared	Constructed from the recent query window and temporal markers of the target forecasting horizon.	Same as Time-LLM + ForeSeek.
Segment index	Shared	Historical context is partitioned into segments, and each segment is summarized by attention-weighted pooling.	Same as Time-LLM + ForeSeek.
Evidence routing	Shared	HAER scores segment indices with the horizon-aware query and selects Top-K evidence segments.	Same as Time-LLM + ForeSeek.
Evidence path	Shared	The LLM processes the retrieved evidence sequence to produce an evidence-based forecast.	Same as Time-LLM + ForeSeek.
Direct temporal path	Shared	A lightweight temporal head bypasses the LLM and predicts a global trend estimate from temporal patch representations.	Same as Time-LLM + ForeSeek.
Training-only full-context teacher	Shared objective, backbone-specific input	The full Time-LLM context is used during training to provide attention and prediction supervision.	The full S2IP-LLM context is used during training to provide attention and prediction supervision.
Prediction fusion	Shared	Evidence-path and direct-path predictions are combined by gated fusion.	Same as Time-LLM + ForeSeek.

A.5 Training and Inference Algorithm

This appendix summarizes the end-to-end training and inference procedures of ForeSeek. Algorithm 1 covers mixed-token construction, horizon-aware evidence routing, evidence selection, dual-path forecasting, and joint optimization. The full-context teacher is used only during training to provide attention-alignment and prediction-consistency supervision. It is discarded at inference, so the inference-time model processes only the routed evidence through the LLM backbone.

Algorithm 1 ForeSeek: Forecast-Conditioned Evidence Seeking

Require: Historical series $X \in \mathbb{R}^{T\times C}$, future markers M_y, horizon H, retention ratio ρ, segment length ℓ, stride Δ

Ensure: Forecast prediction $\hat{Y}$; training loss $\mathcal{L}$ if training

Construct prompt embeddings P from task description and input statistics
Patchify X and reprogram temporal patches into the LLM space to obtain E
Form the mixed token sequence $S \leftarrow [P; E] \in \mathbb{R}^{N\times d}$
Set the retrievable sequence $S^{(r)} \leftarrow S$ or the patch-only part E
Partition $S^{(r)}$ into segments $\{G_i\}_{i=1}^{N_s}$ with length ℓ and stride Δ
for $i = 1, \ldots, N_s$ **do**
    Compute segment attention weights $\alpha_{i,t} \leftarrow \mathrm{softmax}_t(w_a^\top G_{i,t})$
    Summarize the segment as $k_i \leftarrow \sum_{t=1}^{\ell} \alpha_{i,t} G_{i,t}$
end for
Extract recent query-window patch tokens E_q from E
Compute recent-state summary $e_q \leftarrow \mathrm{MeanPool}(E_q)$
Encode the forecasting horizon $r_H \leftarrow \psi(M_y)$
Build the horizon-aware query $q \leftarrow W_q(e_q + r_H)$
for $i = 1, \ldots, N_s$ **do**
    Compute HAER score $s_i \leftarrow q^\top W_k k_i / \sqrt{d}$
end for
Obtain router distribution $p_R \leftarrow \mathrm{softmax}(\{s_i\}_{i=1}^{N_s})$
Select $K \leftarrow \lceil \rho N_s \rceil$ segments with the largest routing scores
Restore selected segments to their original temporal order
Concatenate selected segments as evidence sequence S_{ev}
Obtain evidence-path prediction $\hat{Y}_{\mathrm{ev}} \leftarrow \mathrm{LLM}(S_{\mathrm{ev}})$
Obtain direct temporal prediction $\hat{Y}_{\mathrm{dir}} \leftarrow \mathrm{DirectPath}(E)$
Fuse two paths by $\hat{Y} \leftarrow g\hat{Y}_{\mathrm{ev}} + (1-g)\hat{Y}_{\mathrm{dir}}$
if training **then**
    Run the full-context teacher on S to obtain $\hat{Y}_{\mathrm{full}}$ and attention A
    Aggregate query-side attention over segments to obtain p_A
    Compute $\mathcal{L}_{\mathrm{align}} \leftarrow \mathrm{KL}(\mathrm{sg}(p_A) \,\|\, p_R)$
    Compute $\mathcal{L}_{\mathrm{cons}} \leftarrow \mathrm{SmoothL1}(\hat{Y}_{\mathrm{ev}}, \mathrm{sg}(\hat{Y}_{\mathrm{full}}))$
    Compute $\mathcal{L}_{\mathrm{task}} \leftarrow \mathrm{MSE}(\hat{Y}, Y)$
    return $\hat{Y},\ \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{cons}}\mathcal{L}_{\mathrm{cons}}$
end if
return $\hat{Y}$

B EXPERIMENTAL SETUP

B.1 DATASETS AND SPLITS

Dataset statistics are summarized in Table 7. We evaluate the long-term forecasting performance on the well-established seven different benchmarks, including four ETT datasets Zhou et al. (2021) (i.e., ETTh1, ETTh2, ETTm1, and ETTm2), Weather, Electricity and Traffic from Wu et al. (2022).

Table 7: Following the statistical summaries provided by Wu et al. (2022), we define the dimension as the total number of variates (or channels). Furthermore, the dataset volume is partitioned sequentially into (training, validation, testing) subsets.

Tasks	Dataset	Dim.	Series Length	Dataset Size	Frequency	Domain
Long-term Forecasting	ETTm1	7	{96, 192, 336, 720}	(34465, 11521, 11521)	15 min	Temperature
	ETTm2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	15 min	Temperature
	ETTh1	7	{96, 192, 336, 720}	(8545, 2881, 2881)	1 hour	Temperature
	ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	1 hour	Temperature
	Electricity	321	{96, 192, 336, 720}	(18317, 2633, 5261)	1 hour	Electricity
	Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	1 hour	Transportation
	Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	10 min	Weather

B.2 BASELINE MODELS

iTransformer Liu et al. (2023): Inverts the traditional temporal attention paradigm by treating each univariate series as an independent token. This variate-centric architecture effectively captures multivariate correlations and establishes robust generalization across diverse forecasting benchmarks.

FEDformer Zhou et al. (2022): Integrates seasonal-trend decomposition with a frequency-enhanced Transformer block. By operating in the frequency domain, it efficiently models both macroscopic global trends and fine-grained temporal structures.

Stationary Liu et al. (2022): Addresses the over-stationarization dilemma in temporal modeling via a Non-stationary Transformer. It employs series stationarization for input stability, alongside a destationary attention mechanism to re-inject intrinsic non-stationary dynamics, significantly reducing forecasting errors.

ETSFormer Woo et al. (2022): Synergizes the classical principles of exponential smoothing with modern Transformer architectures. It replaces standard self-attention with an exponential smoothing attention mechanism to explicitly model seasonal and trend components.

TimesNet Wu et al. (2022): Reconceptualizes 1D time series as 2D tensors based on multiple periodicities. By employing 2D convolutional inception blocks, it inherently captures complex intra-period and inter-period variations.

DLinear Zeng et al. (2023): Serves as a highly efficient linear baseline that questions the necessity of complex self-attention for long-horizon forecasting. By decomposing the series and applying simple linear layers, it mitigates the permutation-invariant flaws of Transformers while offering high interpretability and minimal computational overhead.

GPT4TS Zhou et al. (2023): GPT4TS Zhou et al. (2023): Capitalizes on the reasoning capabilities of pre-trained LLMs for temporal analysis. By keeping the core self-attention and feed-forward blocks completely frozen, it effectively circumvents data scarcity issues and achieves robust cross-domain generalization.

Time-LLM Jin et al. (2023): Pioneers a time series reprogramming framework that bridges the modality gap between continuous temporal patches and discrete text. By enriching the context with natural language prompts, it activates the forecasting potential of foundational LLMs. In our implementation, we utilize GPT-2 Radford et al. (2019) as the backbone.

S2IP-LLM Pan et al. (2024): S2IP-LLM introduces a semantic space informed prompting mechanism that aligns time series embeddings with semantic anchors derived from pre-trained LLMs to enhance forecasting representations.

MSH-LLM Shang et al. (2026): MSH-LLM addresses the semantic sparsity of time series by proposing a Multi-Scale Hypergraph mechanism that captures group-wise interactions as hyperedges, enabling cross-modality alignment with LLMs at multiple granularities.

B.3 Implementation and Training Settings

All deep learning models in this study are implemented in PyTorch and trained on a hardware setup comprising NVIDIA RTX 6000 Ada Generation (96GB) and GeForce RTX 5090 GPUs. We leverage the pre-trained language models provided by Radford et al. (2019) as our foundational backbone. Our overall framework is built upon the architecture of Time-LLM Jin et al. (2023), which we significantly extend by introducing the Horizon-Aware Evidence Routing (HAER) mechanism for evidence segment retrieval. The experimental configurations strictly adhere to the standard evaluation protocols widely adopted in long-horizon time series forecasting Wu et al. (2022); Nie et al. (2022). During the training phase, models are optimized using the Adam optimizer Kingma & Ba (2014) with a default learning rate of 10^{-4}. To prevent overfitting and ensure optimal generalization, an Early Stopping strategy is consistently applied across all experiments.

B.4 Forecasting Metrics

We use Mean Squared Error (MSE) and Mean Absolute Error (MAE) to evaluate the performance of long-term, few-shot, and zero-shot forecasting.

$$\text{MSE} = \frac{1}{H}\sum_{h=1}^{H}(Y_h - \hat{Y}_h)^2, \quad \text{MAE} = \frac{1}{H}\sum_{h=1}^{H}|Y_h - \hat{Y}_h| \tag{30}$$

Where H denotes the number of data samples, corresponding to the forecasting horizon in the experiment. Y_h and $\hat{Y}_h$ refer to the h-th ground truth and its corresponding prediction, respectively.

C Complete Forecasting Results

C.1 Long-Term Forecasting

Table 8 presents the full long-term forecasting results across all prediction horizons. The detailed breakdown shows that the advantage of ForeSeek is not limited to a single horizon, but remains consistent across different forecasting lengths.

Table 8: Full long-term forecasting results. All results are averaged from four different forecasting horizons: $\mathbf{H} \in \{96, 192, 336, 720\}$. Red: the best, Blue with underline: the second best.

Methods		ForeSeek (Ours)				LLM-BASED						Others									
		Time-LLM		S2IP-LLM		S2IP-LLM		TIME-LLM		GPT4TS		iTransformer		DLinear		TimesNet		FEDformer		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.388	0.412	0.369	0.397	0.373	0.400	0.383	0.404	0.376	0.397	0.395	0.420	0.375	0.399	0.384	0.402	0.376	0.419	0.513	0.491
	192	0.422	0.436	0.413	0.430	0.414	0.430	0.427	0.431	0.416	0.418	0.427	0.441	0.405	0.416	0.436	0.429	0.420	0.448	0.534	0.504
	336	0.436	0.443	0.427	0.438	0.460	0.469	0.430	0.436	0.442	0.433	0.445	0.457	0.439	0.443	0.491	0.469	0.459	0.465	0.588	0.535
	720	0.439	0.465	0.512	0.503	0.510	0.499	0.465	0.469	0.477	0.456	0.537	0.530	0.472	0.490	0.521	0.500	0.506	0.507	0.643	0.616
	Avg	0.421	0.439	0.430	0.442	0.439	0.450	0.426	0.435	0.427	0.426	0.451	0.462	0.422	0.437	0.458	0.450	0.440	0.460	0.570	0.537
ETTh2	96	0.281	0.341	0.279	0.344	0.294	0.357	0.293	0.348	0.285	0.342	0.304	0.360	0.289	0.353	0.340	0.374	0.358	0.397	0.476	0.458
	192	0.341	0.389	0.334	0.376	0.349	0.389	0.356	0.391	0.354	0.389	0.377	0.403	0.383	0.418	0.402	0.414	0.429	0.439	0.512	0.493
	336	0.359	0.404	0.368	0.411	0.364	0.411	0.372	0.408	0.373	0.407	0.405	0.429	0.448	0.465	0.452	0.452	0.496	0.487	0.552	0.551
	720	0.392	0.432	0.397	0.438	0.447	0.474	0.421	0.446	0.406	0.441	0.443	0.464	0.605	0.551	0.462	0.468	0.463	0.474	0.562	0.560
	Avg	0.343	0.392	0.345	0.392	0.364	0.408	0.361	0.398	0.354	0.394	0.382	0.414	0.431	0.446	0.414	0.427	0.437	0.449	0.526	0.516
ETTm1	96	0.309	0.365	0.301	0.351	0.293	0.348	0.294	0.345	0.292	0.346	0.312	0.366	0.299	0.343	0.338	0.375	0.379	0.419	0.386	0.398
	192	0.344	0.381	0.342	0.372	0.325	0.367	0.330	0.368	0.332	0.372	0.347	0.385	0.335	0.365	0.374	0.387	0.426	0.441	0.459	0.444
	336	0.380	0.408	0.370	0.389	0.359	0.389	0.365	0.392	0.366	0.394	0.379	0.404	0.369	0.386	0.410	0.411	0.445	0.459	0.495	0.464
	720	0.432	0.442	0.423	0.416	0.406	0.417	0.427	0.431	0.417	0.421	0.441	0.442	0.425	0.421	0.478	0.450	0.543	0.490	0.585	0.516
	Avg	0.366	0.399	0.359	0.382	0.346	0.380	0.354	0.384	0.352	0.383	0.370	0.399	0.357	0.378	0.400	0.406	0.448	0.452	0.481	0.456
ETTm2	96	0.163	0.254	0.169	0.257	0.168	0.259	0.175	0.265	0.173	0.262	0.179	0.271	0.167	0.269	0.187	0.267	0.203	0.287	0.192	0.274
	192	0.216	0.291	0.224	0.296	0.229	0.306	0.243	0.316	0.229	0.301	0.242	0.313	0.224	0.303	0.249	0.309	0.269	0.328	0.280	0.339
	336	0.273	0.329	0.277	0.330	0.285	0.345	0.294	0.343	0.286	0.341	0.288	0.344	0.281	0.342	0.321	0.351	0.325	0.366	0.334	0.361
	720	0.358	0.388	0.365	0.392	0.371	0.400	0.389	0.410	0.378	0.401	0.378	0.397	0.397	0.421	0.408	0.403	0.421	0.415	0.417	0.413
	Avg	0.253	0.315	0.259	0.319	0.266	0.328	0.275	0.334	0.266	0.326	0.272	0.331	0.267	0.333	0.291	0.333	0.305	0.349	0.306	0.347
Weather	96	0.158	0.208	0.164	0.215	0.150	0.200	0.163	0.210	0.162	0.212	0.253	0.304	0.176	0.237	0.172	0.220	0.217	0.296	0.173	0.223
	192	0.200	0.247	0.209	0.258	0.193	0.242	0.205	0.245	0.204	0.248	0.280	0.319	0.220	0.282	0.219	0.261	0.276	0.336	0.245	0.285
	336	0.256	0.285	0.257	0.290	0.250	0.286	0.257	0.287	0.254	0.286	0.321	0.344	0.265	0.319	0.280	0.306	0.339	0.380	0.321	0.338
	720	0.319	0.330	0.328	0.342	0.324	0.338	0.323	0.332	0.326	0.337	0.364	0.374	0.333	0.362	0.365	0.359	0.403	0.428	0.414	0.410
	Avg	0.233	0.268	0.240	0.276	0.229	0.267	0.237	0.269	0.237	0.270	0.304	0.335	0.248	0.300	0.259	0.287	0.309	0.360	0.288	0.314
Electricity	96	0.137	0.235	0.141	0.239	0.139	0.240	0.140	0.236	0.139	0.238	0.147	0.248	0.140	0.237	0.168	0.272	0.193	0.308	0.169	0.273
	192	0.140	0.242	0.149	0.247	0.155	0.253	0.150	0.249	0.153	0.251	0.165	0.267	0.153	0.249	0.184	0.289	0.201	0.315	0.182	0.286
	336	0.173	0.274	0.168	0.269	0.171	0.271	0.168	0.267	0.169	0.266	0.178	0.279	0.169	0.267	0.198	0.300	0.214	0.329	0.200	0.304
	720	0.215	0.311	0.208	0.302	0.210	0.302	0.209	0.302	0.206	0.297	0.322	0.398	0.203	0.301	0.220	0.320	0.246	0.355	0.222	0.321
	Avg	0.166	0.266	0.167	0.264	0.169	0.267	0.167	0.264	0.167	0.263	0.203	0.298	0.166	0.263	0.192	0.295	0.214	0.327	0.193	0.296
Traffic	96	0.389	0.283	0.375	0.275	0.378	0.276	0.384	0.278	0.388	0.282	0.367	0.288	0.410	0.282	0.593	0.321	0.587	0.366	0.612	0.338
	192	0.401	0.284	0.399	0.287	0.397	0.283	0.398	0.286	0.407	0.290	0.378	0.293	0.423	0.287	0.617	0.336	0.604	0.373	0.613	0.340
	336	0.410	0.287	0.408	0.290	0.406	0.287	0.408	0.289	0.412	0.294	0.389	0.294	0.436	0.296	0.629	0.336	0.621	0.383	0.618	0.328
	720	0.448	0.309	0.430	0.300	0.434	0.301	0.436	0.303	0.450	0.312	0.401	0.304	0.466	0.315	0.640	0.350	0.626	0.382	0.653	0.355
	Avg	0.412	0.291	0.403	0.288	0.404	0.287	0.407	0.289	0.414	0.294	0.389	0.295	0.433	0.295	0.620	0.336	0.610	0.376	0.624	0.340

C.2 FEW-SHOT FORECASTING

The following tables report detailed few-shot forecasting results under limited supervision. These results further show that the proposed retrieval mechanism remains effective when training data are scarce, where selecting relevant evidence becomes especially important.

Table 9: Few-shot learning on 10% training data. All results are averaged from four different forecasting horizons: $\mathbf{H} \in \{96, 192, 336, 720\}$. Red: the best, Blue with underline: the second best.

Methods		ForeSeek (Ours)				LLM-BASED						Others									
		TIME-LLM		S2IP-LLM		S2IP-LLM		TIME-LLM		GPT4TS		iTransformer		DLinear		TimesNet		FEDformer		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.461	0.453	0.450	0.449	0.450	0.481	0.720	0.533	0.458	0.456	0.790	0.586	0.492	0.495	0.861	0.628	0.512	0.499	0.918	0.639
	192	0.555	0.508	0.538	0.508	0.586	0.507	0.747	0.545	0.570	0.516	0.837	0.609	0.565	0.538	0.797	0.593	0.624	0.555	0.915	0.629
	336	0.622	0.550	0.709	0.555	0.603	0.524	0.793	0.551	0.608	0.535	0.780	0.575	0.721	0.622	0.941	0.648	0.691	0.574	0.939	0.644
	720	0.710	0.590	0.829	0.644	0.999	0.692	0.880	0.584	0.725	0.591	1.234	0.811	0.986	0.743	0.877	0.641	0.728	0.614	0.887	0.645
	Avg	0.587	0.525	0.632	0.539	0.660	0.551	0.785	0.553	0.590	0.525	0.910	0.645	0.691	0.600	0.869	0.628	0.639	0.561	0.915	0.639
ETTh2	96	0.311	0.367	0.308	0.365	0.339	0.383	0.334	0.381	0.331	0.374	0.404	0.435	0.357	0.411	0.378	0.409	0.382	0.416	0.389	0.411
	192	0.378	0.410	0.381	0.414	0.404	0.427	0.430	0.438	0.402	0.411	0.470	0.474	0.569	0.519	0.490	0.467	0.478	0.474	0.473	0.455
	336	0.387	0.426	0.444	0.459	0.451	0.462	0.449	0.458	0.406	0.433	0.489	0.485	0.671	0.572	0.537	0.494	0.504	0.501	0.507	0.480
	720	0.453	0.463	0.532	0.519	0.540	0.519	0.485	0.490	0.449	0.464	0.593	0.538	0.824	0.648	0.510	0.491	0.499	0.509	0.477	0.472
	Avg	0.382	0.416	0.416	0.439	0.434	0.448	0.424	0.441	0.397	0.421	0.489	0.483	0.605	0.538	0.479	0.465	0.466	0.475	0.462	0.455
ETTm1	96	0.360	0.383	0.388	0.406	0.391	0.407	0.412	0.422	0.390	0.404	0.709	0.556	0.352	0.392	0.583	0.501	0.578	0.518	0.761	0.568
	192	0.417	0.421	0.403	0.416	0.434	0.432	0.447	0.438	0.429	0.423	0.717	0.548	0.382	0.412	0.630	0.528	0.617	0.546	0.781	0.574
	336	0.428	0.419	0.436	0.434	0.450	0.443	0.497	0.465	0.469	0.439	0.735	0.575	0.419	0.434	0.725	0.568	0.998	0.775	0.803	0.587
	720	0.523	0.474	0.512	0.477	0.567	0.506	0.594	0.521	0.569	0.498	0.752	0.584	0.490	0.477	0.769	0.549	0.693	0.579	0.844	0.581
	Avg	0.432	0.424	0.435	0.433	0.461	0.447	0.487	0.461	0.464	0.441	0.728	0.565	0.411	0.429	0.677	0.537	0.722	0.605	0.797	0.578
ETTm2	96	0.189	0.267	0.191	0.273	0.204	0.287	0.224	0.296	0.188	0.269	0.245	0.322	0.213	0.303	0.212	0.285	0.291	0.399	0.229	0.308
	192	0.249	0.308	0.247	0.308	0.257	0.317	0.260	0.317	0.251	0.309	0.274	0.338	0.278	0.345	0.270	0.323	0.307	0.379	0.291	0.343
	336	0.303	0.345	0.307	0.352	0.329	0.361	0.312	0.349	0.307	0.346	0.361	0.394	0.338	0.385	0.323	0.353	0.543	0.559	0.348	0.376
	720	0.383	0.403	0.393	0.400	0.398	0.404	0.424	0.416	0.426	0.417	0.467	0.442	0.436	0.440	0.474	0.449	0.712	0.614	0.461	0.438
	Avg	0.281	0.331	0.285	0.333	0.297	0.342	0.305	0.344	0.293	0.335	0.336	0.373	0.316	0.368	0.320	0.353	0.463	0.488	0.332	0.366

Table 10: Few-shot learning on 5% training data. All results are averaged from four different forecasting horizons: $\mathbf{H} \in \{96, 192, 336, 720\}$. Red: the best, Blue with underline: the second best.

Methods		ForeSeek (Ours)				LLM-BASED						Others									
		TIME-LLM		S2IP-LLM		S2IP-LLM		TIME-LLM		GPT4TS		iTransformer		DLinear		TimesNet		FEDformer		Stationary	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.592	0.505	0.668	0.532	0.620	0.526	0.518	0.498	0.543	0.506	0.808	0.610	0.547	0.503	0.892	0.625	0.593	0.529	0.952	0.650
	192	0.733	0.562	0.871	0.624	0.765	0.585	0.702	0.547	0.748	0.580	0.928	0.658	0.720	0.604	0.940	0.665	0.652	0.563	0.943	0.645
	336	0.765	0.583	1.058	0.695	1.012	0.678	0.725	0.603	0.754	0.595	1.475	0.861	0.984	0.727	0.945	0.653	0.731	0.594	0.935	0.644
	720	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Avg	0.697	0.550	0.866	0.617	0.799	0.596	0.648	0.549	0.681	0.560	1.070	0.710	0.750	0.611	0.925	0.647	0.658	0.562	0.943	0.646
ETTh2	96	0.344	0.394	0.409	0.435	0.362	0.407	0.384	0.420	0.376	0.421	0.397	0.427	0.442	0.456	0.409	0.420	0.390	0.424	0.408	0.423
	192	0.402	0.422	0.531	0.501	0.547	0.511	0.394	0.424	0.418	0.441	0.438	0.445	0.617	0.542	0.483	0.464	0.457	0.465	0.497	0.468
	336	0.427	0.440	0.478	0.473	0.556	0.517	0.416	0.433	0.408	0.439	0.631	0.553	1.424	0.849	0.499	0.479	0.477	0.483	0.507	0.481
	720	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Avg	0.391	0.419	0.473	0.470	0.488	0.478	0.398	0.426	0.400	0.433	0.488	0.475	0.827	0.615	0.463	0.454	0.463	0.454	0.470	0.489
ETTm1	96	0.371	0.394	0.418	0.434	0.396	0.413	0.422	0.424	0.386	0.405	0.589	0.510	0.332	0.374	0.606	0.518	0.628	0.544	0.823	0.587
	192	0.447	0.445	0.505	0.477	0.458	0.449	0.448	0.440	0.440	0.438	0.703	0.565	0.358	0.390	0.681	0.539	0.666	0.566	0.844	0.591
	336	0.487	0.463	0.606	0.537	0.516	0.482	0.452	0.447	0.485	0.459	0.898	0.641	0.402	0.416	0.786	0.597	0.807	0.628	0.870	0.603
	720	0.641	0.542	0.801	0.631	0.672	0.567	0.585	0.491	0.577	0.499	0.948	0.671	0.511	0.489	0.796	0.593	0.822	0.633	0.893	0.611
	Avg	0.487	0.461	0.583	0.520	0.511	0.478	0.477	0.451	0.472	0.450	0.784	0.596	0.400	0.417	0.717	0.561	0.730	0.592	0.857	0.598
ETTm2	96	0.204	0.284	0.228	0.308	0.206	0.286	0.205	0.277	0.199	0.280	0.265	0.339	0.236	0.326	0.220	0.299	0.229	0.320	0.238	0.316
	192	0.264	0.323	0.267	0.332	0.278	0.336	0.267	0.336	0.256	0.316	0.310	0.362	0.306	0.373	0.311	0.361	0.394	0.361	0.298	0.349
	336	0.310	0.352	0.360	0.388	0.325	0.364	0.309	0.347	0.318	0.353	0.373	0.399	0.380	0.423	0.338	0.366	0.378	0.427	0.353	0.380
	720	0.413	0.421	0.468	0.454	0.434	0.427	0.448	0.432	0.460	0.436	0.478	0.454	0.674	0.583	0.509	0.465	0.523	0.510	0.475	0.445
	Avg	0.298	0.345	0.331	0.371	0.311	0.353	0.307	0.348	0.308	0.346	0.356	0.388	0.399	0.426	0.344	0.372	0.381	0.404	0.341	0.372

C.3 ZERO-SHOT TRANSFER

We further provide the complete zero-shot forecasting results on the ETT benchmarks. This setting highlights the transferability of the proposed framework, since the model must rely on retrieved contextual evidence without task-specific adaptation.

Table 11: Full zero-shot learning results on ETT datasets across all forecasting horizons. A lower value indicates better performance. Red: the best, Blue with underline: the second best.

Methods		ForeSeek (Ours)				LLM-Based								Others					
		TIME-LLM		S2IP-LLM		S2IP-LLM		Time-LLM		LLMTime		GPT4TS		iTransformer		DLinear		TimesNet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1 ↓ ETTh2	96	0.312	0.353	0.275	0.338	0.282	0.341	0.324	0.368	0.510	0.576	0.335	0.374	0.353	0.394	0.347	0.400	0.358	0.387
	192	0.335	0.379	0.339	0.381	0.356	0.396	0.398	0.396	0.523	0.586	0.412	0.417	0.437	0.445	0.447	0.460	0.427	0.429
	336	0.358	0.402	0.359	0.402	0.376	0.418	0.410	0.423	0.640	0.637	0.441	0.444	0.482	0.476	0.515	0.505	0.449	0.451
	720	0.397	0.435	0.403	0.441	0.437	0.467	0.403	0.449	2.296	1.034	0.438	0.452	0.556	0.506	0.665	0.589	0.448	0.458
	Avg	0.351	0.392	0.344	0.391	0.363	0.406	0.384	0.409	0.992	0.708	0.406	0.422	0.457	0.445	0.493	0.488	0.421	0.431
ETTh1 ↓ ETTm2	96	0.218	0.310	0.165	0.256	0.222	0.315	0.236	0.320	0.646	0.563	0.236	0.315	0.247	0.319	0.255	0.357	0.239	0.313
	192	0.274	0.344	0.224	0.294	0.302	0.366	0.265	0.353	0.934	0.654	0.287	0.342	0.293	0.350	0.338	0.413	0.291	0.342
	336	0.319	0.369	0.276	0.330	0.339	0.389	0.337	0.376	1.157	0.728	0.341	0.374	0.364	0.419	0.425	0.465	0.342	0.371
	720	0.432	0.438	0.360	0.384	0.419	0.434	0.429	0.430	4.730	1.531	0.435	0.422	0.534	0.470	0.640	0.573	0.434	0.419
	Avg	0.311	0.365	0.256	0.316	0.321	0.376	0.317	0.370	1.867	0.869	0.325	0.363	0.360	0.390	0.415	0.452	0.327	0.361
ETTh2 ↓ ETTh1	96	0.496	0.468	0.430	0.445	0.472	0.474	0.618	0.515	1.130	0.777	0.732	0.577	0.854	0.606	0.689	0.555	0.848	0.601
	192	0.575	0.512	0.440	0.450	0.522	0.501	0.715	0.570	1.242	0.820	0.578	0.559	0.863	0.615	0.707	0.568	0.860	0.610
	336	0.617	0.542	0.478	0.483	0.590	0.555	0.636	0.523	1.328	0.864	0.759	0.578	0.867	0.626	0.710	0.577	0.867	0.626
	720	0.578	0.534	0.620	0.572	0.758	0.638	0.683	0.553	4.145	1.461	0.781	0.597	0.887	0.654	0.704	0.596	0.887	0.648
	Avg	0.567	0.514	0.492	0.488	0.586	0.542	0.663	0.540	1.961	0.981	0.757	0.578	0.868	0.625	0.703	0.574	0.865	0.621
ETTh2 ↓ ETTm2	96	0.205	0.299	0.205	0.296	0.206	0.299	0.258	0.326	0.646	0.563	0.253	0.329	0.244	0.356	0.240	0.336	0.248	0.324
	192	0.261	0.331	0.264	0.333	0.278	0.346	0.303	0.342	0.934	0.654	0.293	0.346	0.291	0.356	0.395	0.369	0.296	0.352
	336	0.355	0.395	0.311	0.360	0.332	0.382	0.356	0.383	1.157	0.728	0.347	0.376	0.351	0.391	0.345	0.397	0.353	0.383
	720	0.398	0.417	0.397	0.417	0.406	0.422	0.440	0.434	4.730	1.531	0.446	0.429	0.452	0.451	0.432	0.442	0.471	0.466
	Avg	0.305	0.361	0.294	0.352	0.306	0.362	0.339	0.371	1.867	0.869	0.335	0.370	0.335	0.382	0.328	0.386	0.342	0.376
ETTm1 ↓ ETTh2	96	0.308	0.364	0.307	0.363	0.388	0.404	0.355	0.403	0.510	0.576	0.353	0.392	0.371	0.407	0.365	0.415	0.377	0.407
	192	0.365	0.400	0.374	0.398	0.414	0.419	0.449	0.450	0.523	0.586	0.443	0.437	0.463	0.458	0.454	0.462	0.471	0.453
	336	0.406	0.430	0.377	0.411	0.406	0.427	0.479	0.467	0.640	0.637	0.469	0.461	0.481	0.485	0.496	0.494	0.472	0.484
	720	0.456	0.469	0.400	0.434	0.430	0.455	0.477	0.476	2.296	1.034	0.466	0.468	0.503	0.482	0.541	0.529	0.495	0.482
	Avg	0.384	0.417	0.365	0.402	0.410	0.426	0.440	0.449	0.992	0.708	0.433	0.439	0.455	0.458	0.464	0.475	0.457	0.454
ETTm1 ↓ ETTm2	96	0.173	0.264	0.178	0.262	0.183	0.270	0.218	0.271	0.646	0.563	0.217	0.294	0.219	0.305	0.221	0.314	0.222	0.295
	192	0.226	0.299	0.232	0.297	0.254	0.313	0.288	0.335	0.934	0.654	0.277	0.327	0.277	0.347	0.286	0.359	0.288	0.337
	336	0.288	0.336	0.278	0.330	0.290	0.337	0.322	0.355	1.157	0.728	0.331	0.360	0.354	0.378	0.357	0.406	0.341	0.367
	720	0.384	0.398	0.359	0.382	0.369	0.389	0.414	0.409	4.730	1.531	0.429	0.413	0.426	0.420	0.476	0.476	0.436	0.418
	Avg	0.268	0.324	0.262	0.318	0.274	0.327	0.311	0.343	1.867	0.869	0.313	0.348	0.319	0.363	0.335	0.389	0.322	0.354
ETTm2 ↓ ETTh2	96	0.296	0.358	0.304	0.356	0.310	0.359	0.334	0.416	0.510	0.576	0.360	0.401	0.347	0.401	0.333	0.391	0.360	0.401
	192	0.368	0.400	0.376	0.403	0.386	0.411	0.439	0.441	0.523	0.586	0.434	0.437	0.438	0.444	0.441	0.456	0.434	0.437
	336	0.398	0.427	0.414	0.433	0.416	0.441	0.455	0.457	0.640	0.637	0.460	0.459	0.459	0.464	0.505	0.503	0.460	0.459
	720	0.431	0.461	0.423	0.449	0.442	0.470	0.488	0.479	2.296	1.034	0.485	0.477	0.485	0.477	0.543	0.534	0.485	0.477
	Avg	0.373	0.412	0.379	0.410	0.389	0.420	0.429	0.448	0.992	0.708	0.435	0.443	0.432	0.447	0.455	0.471	0.435	0.443
ETTm2 ↓ ETTm1	96	0.412	0.400	0.406	0.413	0.426	0.420	0.488	0.445	1.179	0.781	0.747	0.558	0.619	0.564	0.570	0.490	0.747	0.558
	192	0.461	0.438	0.499	0.467	0.557	0.501	0.555	0.464	1.327	0.846	0.781	0.560	0.685	0.565	0.590	0.506	0.781	0.560
	336	0.496	0.456	0.455	0.443	0.608	0.532	0.608	0.538	1.478	0.902	0.778	0.578	0.792	0.578	0.706	0.567	0.778	0.578
	720	0.511	0.460	0.555	0.502	0.959	0.667	0.699	0.566	3.749	1.408	0.769	0.573	0.727	0.579	0.706	0.584	0.769	0.573
	Avg	0.470	0.439	0.479	0.456	0.638	0.530	0.588	0.503	1.933	0.984	0.769	0.567	0.706	0.572	0.649	0.537	0.769	0.567

C.4 TARGET-TRAINED EVALUATION ON GIFT-EVAL

We additionally evaluate ForeSeek on two datasets from GIFT-Eval (Aksu et al., 2024), following its official evaluation protocol. All task-specific models are trained on the corresponding target dataset. Thus, the controlled evidence comes from comparisons between ForeSeek and its matched backbone, while Chronos-Bolt serves only as a pretrained TSFM reference.

Table 12: Target-trained results on two GIFT-Eval datasets. Lower values are better. Aggregate metrics are geometric means across the two datasets. Chronos-Bolt is included as a pretrained TSFM reference rather than a training-matched baseline.

Model	**US Births MASE / CRPS**	**Saugeen MASE / CRPS**	**Aggregate MASE ↓**	**Aggregate CRPS ↓**
Time-LLM + ForeSeek	**0.225 / 0.224**	**0.893 / 0.760**	**0.449**	**0.412**
Time-LLM	0.244 / 0.242	0.998 / 0.849	0.493	0.453
S2IP-LLM + ForeSeek	0.266 / 0.265	**0.896 / 0.762**	**0.488**	**0.449**
S2IP-LLM	**0.241 / 0.240**	1.003 / 0.853	0.492	0.452
Chronos-Bolt (Base)	0.260 / 0.217	0.832 / 0.578	0.465	0.354
PatchTST	0.261 / 0.208	0.961 / 0.697	0.501	0.381
iTransformer	0.281 / 0.223	0.982 / 0.715	0.525	0.399
TiDE	0.335 / 0.259	0.926 / 0.679	0.557	0.419
DeepAR	0.287 / 0.230	1.263 / 0.978	0.602	0.474
DLinear	0.346 / 0.344	1.231 / 1.048	0.652	0.600
Seasonal Naive	1.000 / 1.000	1.000 / 1.000	1.000	1.000

D Efficiency and Computational Cost

D.1 Metric Definitions and Measurement Protocols

To improve clarity and reproducibility, we summarize the definitions of all efficiency-related metrics used in our evaluation in Table 13. These metrics characterize different aspects of inference efficiency, including token usage, LLM-only computation, end-to-end computational cost, runtime latency, memory footprint, and throughput. Unless otherwise specified, all latency, memory, and throughput measurements are obtained during inference under the same hardware setting.

Table 13: Definitions and measurement protocols of efficiency-related metrics.

Metric	Definition	Unit	Measurement protocol
Token-level efficiency			
ρ	Retained token ratio for the LLM backbone. It is only applied to ForeSeek, while baselines use all tokens.	–	Treated as an external hyperparameter.
Tokens	Number of tokens fed into the LLM backbone.	count	Obtained from `inputs_embeds.shape[1]` using a forward hook.
Computational cost			
LLM FLOPs (G)	FLOPs of the LLM backbone only.	FLOPs (G)	Measured by `fvcore.FlopCountAnalysis` and reported as MACs $\times 2$.
LLM FLOPs Save	Relative reduction in LLM FLOPs with respect to the corresponding baseline.	%	$\left(1 - \frac{\text{LLM}}{\text{LLM}_{\text{baseline}}}\right) \times 100$.
Total FLOPs (G)	FLOPs of the full model for one forward pass.	FLOPs (G)	Measured by `fvcore.FlopCountAnalysis` and reported as MACs $\times 2$.
Total FLOPs Save	Relative reduction in total FLOPs with respect to the corresponding full-context backbone baseline.	%	$\left(1 - \frac{\text{Total}}{\text{Total}_{\text{Time-LLM}}}\right) \times 100$.
Runtime and memory			
LLM Latency	Runtime of one LLM-backbone forward pass.	ms	Averaged over 30 runs with CUDA synchronization.
End-to-end Latency	Runtime of one full-model forward pass.	ms	Measured by `time.perf_counter()` with CUDA synchronization and averaged over 30 runs.
Peak Memory	Peak GPU memory allocated during inference.	MB	Obtained from `torch.cuda.max_memory_allocated()`.
Throughput	Number of processed samples per second.	samples/s	Batch size divided by end-to-end latency in seconds.

D.2 Inference Efficiency across Retention Ratios

Table 14 reports the raw values underlying Figure 5. Normalized values in the figure are computed as the ForeSeek value divided by the corresponding full-context backbone value, multiplied by 100.

Table 14: Efficiency analysis of **ForeSeek** under different Top-K ratios. Lower values are better for Tokens, FLOPs (G), Latency, and Peak Memory, while higher values are better for Save (%) and Throughput.

Model	Top-K Ratio	Tokens ↓	LLM Cost		Overall Cost		Latency		Peak Memory ↓	Throughput ↑
			FLOPs (G) ↓	Save (%) ↑	FLOPs (G) ↓	Save (%) ↑	LLM ↓	Total ↓		
Time-LLM	–	191	236.79	0.00	319.71	0.00	11.23	26.95	1119.6	37.11
Time-LLM (ForeSeek)	0.10	19	22.71	90.41	105.67	66.95	**5.37**	19.54	**776.9**	51.18
Time-LLM (ForeSeek)	0.20	38	45.61	80.74	128.58	59.78	5.86	22.53	**776.9**	44.39
Time-LLM (ForeSeek)	0.30	57	68.69	70.99	151.68	52.56	5.87	**19.27**	819.7	**51.90**
Time-LLM (ForeSeek)	0.40	76	91.96	61.16	174.96	45.27	8.10	22.59	844.1	44.27
Time-LLM (ForeSeek)	0.50	96	116.66	50.73	199.67	37.54	7.69	20.71	879.8	48.28
Time-LLM (ForeSeek)	0.70	134	164.15	30.68	247.19	22.68	8.95	23.22	969.0	43.07
S2IP-LLM	–	68	82.14	0.00	84.56	0.00	8.00	23.09	634.9	43.31
S2IP-LLM (ForeSeek)	0.10	7	8.35	89.84	10.84	87.18	**5.83**	**21.61**	**528.3**	**46.28**
S2IP-LLM (ForeSeek)	0.20	14	16.72	79.65	19.21	77.28	6.22	22.04	542.5	45.36
S2IP-LLM (ForeSeek)	0.30	20	23.91	70.89	26.41	68.77	6.14	22.20	559.0	45.04
S2IP-LLM (ForeSeek)	0.40	27	32.33	60.64	34.83	58.81	6.03	21.69	563.4	46.09
S2IP-LLM (ForeSeek)	0.50	34	40.77	50.38	43.27	48.83	6.22	22.15	570.8	45.14
S2IP-LLM (ForeSeek)	0.70	48	57.74	29.71	60.23	28.78	6.82	23.02	595.0	43.43

D.3 Training Cost and Teacher Overhead

As shown in Table 15, full teacher supervision increases wall-clock step time relative to the no-teacher ForeSeek variant because it requires an additional full-context forward pass and attention extraction. Nevertheless, the complete ForeSeek training graph uses fewer estimated training FLOPs than the corresponding vanilla backbone: 546.17 G versus 653.51 G for Time-LLM, and 232.18 G versus 252.32 G for S2IP-LLM. The teacher is discarded after training and therefore introduces no additional deployment module.

Table 15: Training-cost analysis of ForeSeek. "No teacher" removes the training-only full-context teacher, whereas "full training" uses the complete teacher supervision. Step time is reported as mean $\pm$ standard deviation.

Backbone	Variant	Train FLOPs / sample	Step time (ms)	Peak memory (MB)
Time-LLM	Vanilla	653.51 G	55.69 ± 1.69	2719.6
	ForeSeek, no teacher	309.86 G	58.69 ± 1.53	1596.2
	ForeSeek, full training	**546.17 G**	$\mathbf{69.06 \pm 0.97}$	**1953.0**
S2IP-LLM	Vanilla	252.32 G	33.70 ± 7.40	1350.7
	ForeSeek, no teacher	183.68 G	26.85 ± 2.44	1384.3
	ForeSeek, full training	**232.18 G**	$\mathbf{48.70 \pm 3.81}$	**1395.0**

E Model Analysis

E.1 Matched-Budget Selector Comparison

We compare ForeSeek with fixed and query-conditioned selectors under the same backbone, segment construction, retention ratio, and realized unique-token budget. The results are averaged over four prediction horizons, while full context denotes the uncompressed reference.

As shown in Table 16, ForeSeek achieves the lowest MSE and MAE on all four datasets. Its consistent advantage over random selection, recent truncation, uniform-interval selection, and cosine-similarity retrieval shows that context reduction alone does not explain the improvement. The comparison with horizon-only similarity further indicates that the recent state and target horizon provide complementary routing signals. reference.

Table 16: Matched-budget selector comparison. All compressed variants use a retention ratio of 0.3 and the same realized unique-token budget. Results are averaged over four prediction horizons. Lower values are better.

Selector	Trainable	Recent State	Horizon	ETTh2 MSE / MAE	ETTm2 MSE / MAE	Weather MSE / MAE	Traffic MSE / MAE
Full context	No	–	–	0.358 / 0.401	0.266 / 0.322	0.240 / 0.278	0.419 / 0.298
Random selection	No	No	No	0.370 / 0.408	0.278 / 0.330	0.238 / 0.275	0.427 / 0.303
Recent truncation	No	Implicit	No	0.360 / 0.401	0.260 / 0.321	0.247 / 0.282	0.422 / 0.301
Uniform-interval selection	No	No	No	0.365 / 0.405	0.283 / 0.334	0.235 / 0.273	0.425 / 0.299
Cosine-similarity retrieval	No	Yes	No	0.372 / 0.409	0.282 / 0.333	0.250 / 0.283	0.429 / 0.305
Horizon-only similarity	No	No	Yes	0.347 / 0.393	0.257 / 0.317	0.244 / 0.278	0.416 / 0.294
Teacher-attention Top-K	No	Yes	Yes	0.355 / 0.400	0.261 / 0.318	0.240 / 0.271	0.417 / 0.292
ForeSeek	**Yes**	**Yes**	**Yes**	**0.343 / 0.392**	**0.253 / 0.315**	**0.233 / 0.268**	**0.412 / 0.291**

E.2 Selected-Evidence Categories

We analyze the token-category distribution of the evidence selected by ForeSeek. Figure 6 reports the fine-grained categories of prompt-token overlap hits on the ETTh2 test set. A hit is recorded when a selected evidence segment overlaps with a prompt-token position.

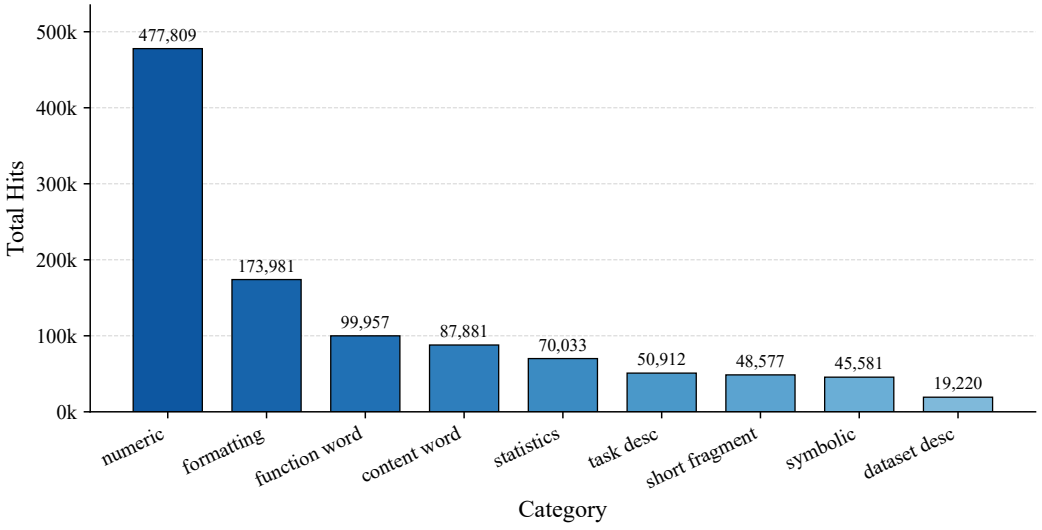


Figure 6: Fine-grained category distribution of retrieved prompt-token overlap hits on the ETTh2 test set.A hit is counted when a selected evidence segment overlaps with a prompt-token position.Numeric and formatting tokens dominate the overlap distribution, suggesting that mixed prompt-patch contexts contain substantial low-level structural and numerical redundancy.

E.3 Teacher Supervision and Component Ablations

Teacher attention is used as a training-time information-routing surrogate rather than causal ground truth. We separately ablate attention alignment and prediction consistency to evaluate their contributions.

Table 17: Ablation of the two teacher-supervision objectives. Alignment denotes router–teacher attention alignment, and consistency denotes prediction-level distillation. Lower values are better.

Variant	**Alignment**	**Consistency**	**ETTh2 MSE / MAE**	**ETTm2 MSE / MAE**
No teacher	No	No	0.372 / 0.408	0.260 / 0.322
Prediction distillation only	No	Yes	0.359 / 0.400	0.275 / 0.333
Attention alignment only	Yes	No	0.353 / 0.399	0.264 / 0.325
Full ForeSeek	**Yes**	**Yes**	**0.343 / 0.392**	**0.253 / 0.315**

As shown in Table 17, neither supervision objective alone reproduces the performance of the complete model. Combining attention alignment with prediction consistency yields the lowest MSE and MAE on both ETTh2 and ETTm2, supporting the complementary roles of distribution-level routing supervision and prediction-level distillation.

E.4 Stability and Sensitivity

We examine the robustness of ForeSeek from three complementary perspectives: variation across random seeds, sensitivity to the evidence-retention ratio, and sensitivity to segment construction.

E.4.1 Random-Seed Stability

Table 18 reports the mean and standard deviation over three independent runs for ForeSeek and its corresponding backbones. The small standard deviations across datasets and metrics indicate that the performance of ForeSeek is stable under different random initializations. The improvements over the original backbones are therefore not attributable to a particular random seed.

Table 18: Performance comparison of LLM forecasters with and without ForeSeek. "Original" denotes the vanilla Time-LLM and S2IP-LLM baselines. Results are reported as the mean ± standard deviation over three independent random seeds.

Methods	ForeSeek (Ours)				Original			
	Time-LLM		S2IP-LLM		Time-LLM		S2IP-LLM	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.421 ± 0.004	0.439 ± 0.007	0.430 ± 0.003	0.442 ± 0.004	0.426 ± 0.008	0.435 ± 0.007	0.439 ± 0.006	0.450 ± 0.005
ETTh2	0.343 ± 0.002	0.392 ± 0.004	0.345 ± 0.001	0.392 ± 0.002	0.361 ± 0.003	0.398 ± 0.005	0.364 ± 0.002	0.408 ± 0.003
ETTm1	0.366 ± 0.002	0.399 ± 0.003	0.359 ± 0.003	0.382 ± 0.006	0.354 ± 0.003	0.384 ± 0.002	0.346 ± 0.003	0.380 ± 0.006
ETTm2	0.253 ± 0.001	0.315 ± 0.002	0.259 ± 0.002	0.319 ± 0.002	0.275 ± 0.002	0.334 ± 0.004	0.266 ± 0.002	0.328 ± 0.001

E.4.2 Evidence-Retention Ratio

The retention ratio ρ determines the proportion of candidate segments preserved for evidence-path forecasting. We vary ρ under the 5% few-shot setting to examine the accuracy–compression trade-off. As shown in Figure 7, forecasting performance does not improve monotonically as more evidence is retained. A very small ratio may remove useful historical cues, whereas a large ratio can reintroduce redundant or weakly relevant context. The results suggest that moderate evidence retention provides a favorable balance between information preservation and context compression.

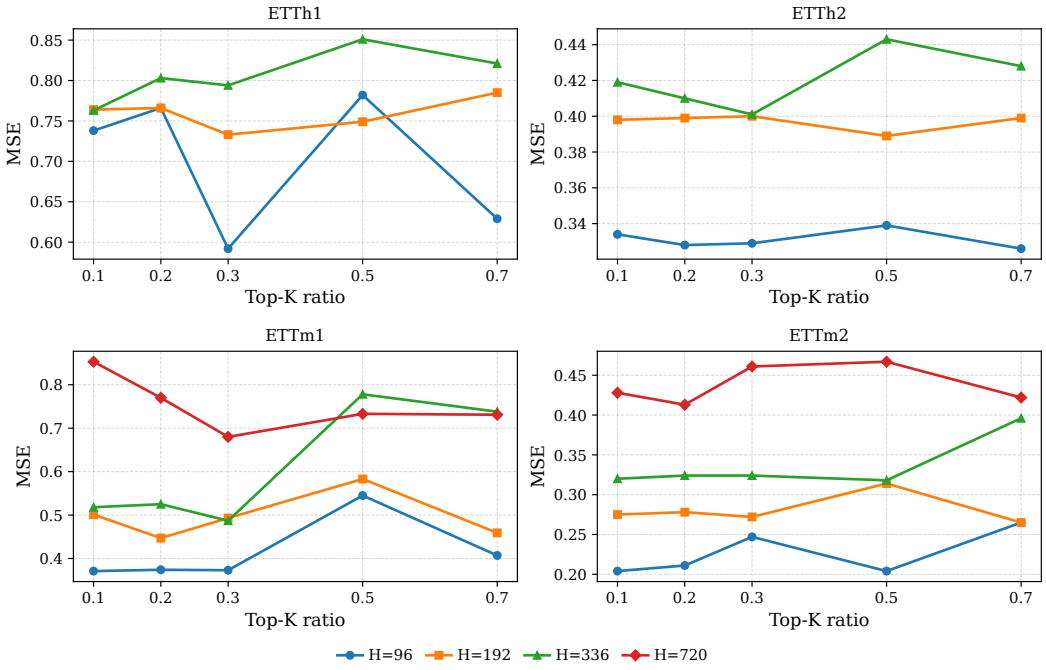


Figure 7: Sensitivity of ForeSeek to the evidence-retention ratio under the 5% few-shot setting.

E.4.3 Segment Length and Stride

We further vary the segment length and stride while fixing the retention ratio at $\rho = 0.3$. Table 19 shows that ForeSeek remains competitive across the evaluated configurations, indicating that its effectiveness is not restricted to a single segmentation scheme. Segment construction nevertheless affects the granularity and redundancy of the candidate evidence: shorter or highly overlapping segments provide finer-grained selection but increase the number of routing candidates, whereas longer segments reduce routing resolution and may retain additional irrelevant tokens. The default configuration offers a practical balance between forecasting accuracy and routing complexity.

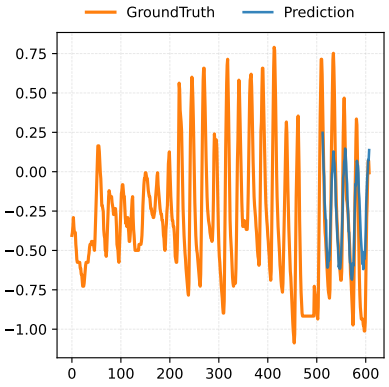
Table 19: Sensitivity to segment construction under the 5% few-shot setting. The retention ratio is fixed at 0.3. "Unique Tokens" counts the union of retained token positions after resolving segment overlap. Lower values are better.

Length	Stride	Overlap	Unique Tokens	ETTh2		ETTm2		Weather		Traffic	
				MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
1	1	0%	58	**0.391**	0.419	**0.298**	**0.345**	0.269	0.320	**0.442**	**0.310**
2	2	0%	58	0.393	0.420	0.308	0.353	**0.268**	**0.319**	0.443	0.311
4	4	0%	56	0.397	**0.416**	0.319	0.361	0.270	0.321	0.452	0.319
4	2	50%	112	0.393	0.422	0.309	0.353	0.271	0.322	0.445	0.314

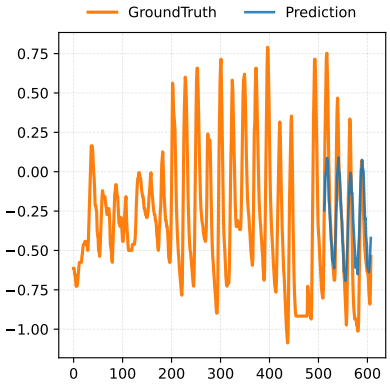
F QUALITATIVE VISUALIZATIONS

We provide representative visualization examples to complement the quantitative results. These cases illustrate how the proposed method tracks both local variations and overall temporal trends across different datasets.

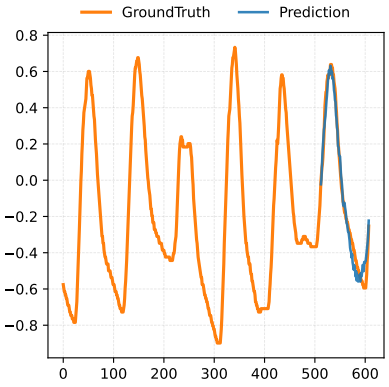
F.1 FORECASTS ON ETT AND ELECTRICITY



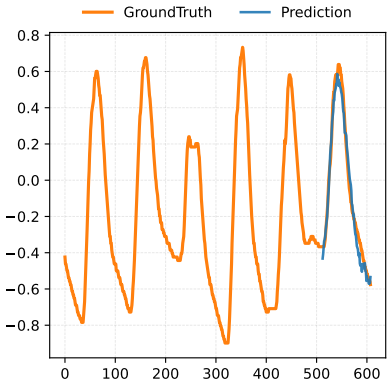
(a) ETTh2 dataset example 1



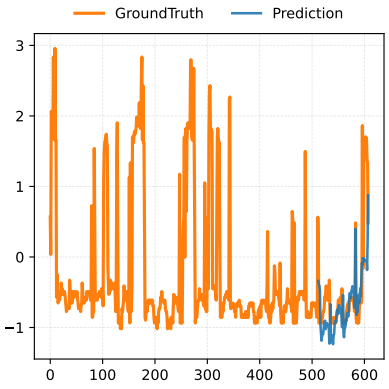
(b) ETTh2 dataset example 2



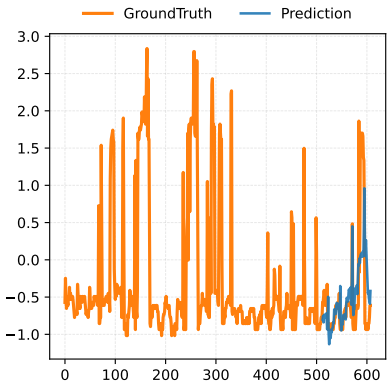
(c) ETTm2 dataset example 1



(d) ETTm2 dataset example 2



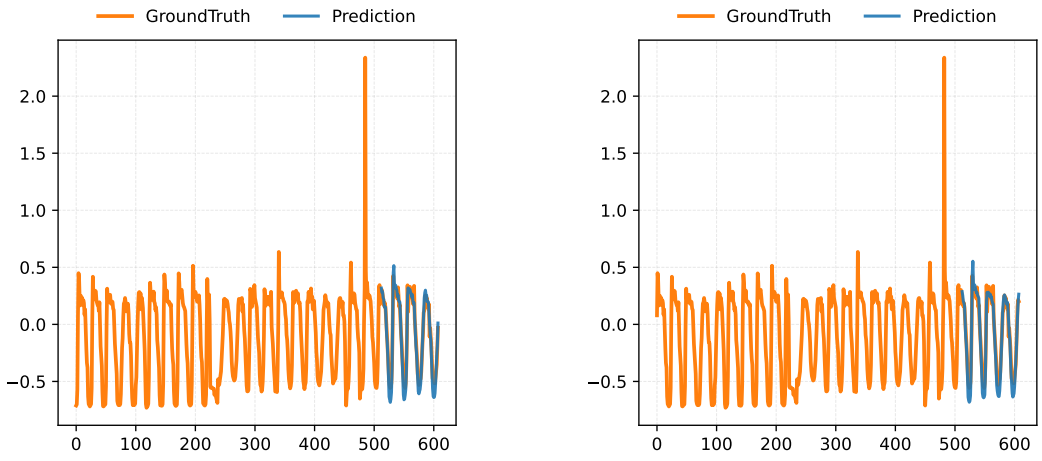
(e) Electricity dataset example 1



(f) Electricity dataset example 2

Figure 8: Sequential visualization examples of prediction results on the ETTh2, ETTm2, and Electricity datasets. The blue lines represent predictions, while the orange lines indicate ground truth.

F.2 FORECASTS ON TRAFFIC



(a) Traffic dataset example 1

(b) Traffic dataset example 2

Figure 9: Sequential visualization examples of prediction results on the Traffic dataset. The blue lines represent predictions, while the orange lines indicate ground truth.

F.3 EVIDENCE-ROUTING CASE STUDIES

Figure 10 presents representative periodic, regime-shift, and abrupt-spike cases. Each case jointly shows the forecast, router–teacher agreement, and the evidence retained by different matched-budget selectors.

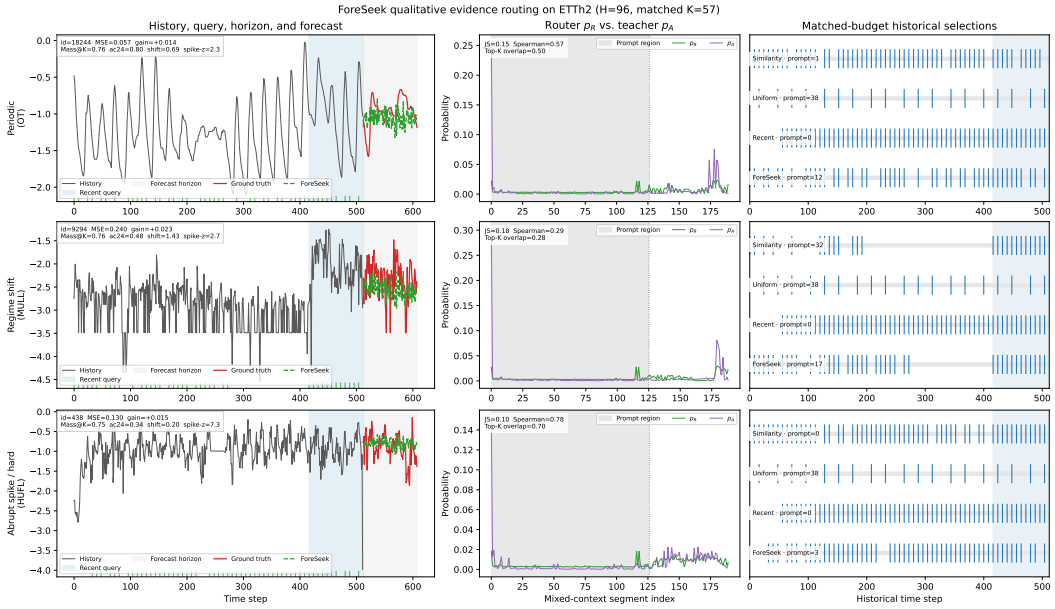


Figure 10: Qualitative evidence-routing cases on ETTh2 with a prediction horizon of 96 and a matched evidence budget of 57 tokens. Rows show periodic, regime-shift, and abrupt-spike cases. The left column presents the historical input, recent query, forecast horizon, ground truth, and ForeSeek prediction. The middle column compares the router distribution with the full-context teacher distribution using JS divergence, Spearman correlation, and Top-K overlap. The right column visualizes the evidence retained by ForeSeek and matched-budget recent, uniform-interval, and cosine-similarity selectors.

In the periodic case, ForeSeek retains both recent and distant phase-aligned evidence. In the regime-shift case, its selections concentrate on observations compatible with the active regime rather than uniformly covering the pre-shift history. In the abrupt-spike case, ForeSeek preserves sparse prior excursions and locally matched levels. These examples illustrate adaptive routing behaviors that fixed recent or uniform selection cannot express.