QuiZSF: A Retrieval-Augmented Framework for Zero-Shot Time Series Forecasting

Shichao Ma
University of Science and Technology of China
Hefei, China
mashichao@mail.ustc.edu.cn

Zhengyang Zhou*
University of Science and Technology of China
Hefei, China
Suzhou Institute for Advanced Research, USTC
Suzhou, China
zzy0929@ustc.edu.cn

Qihe Huang
University of Science and Technology of China
Hefei, China
hqh@mail.ustc.edu.cn

Binwu Wang
University of Science and Technology of China
Hefei, China
wbw2024@ustc.edu.cn

Yang Wang*
University of Science and Technology of China
Hefei, China
Suzhou Institute for Advanced Research, USTC
Suzhou, China
angyan@ustc.edu.cn

Abstract

Accurate forecasting of sequential data streams is a cornerstone of modern Web services, supporting applications such as traffic management, user behavior modeling, and online anomaly prevention. However, in many Web environments, new domains emerge rapidly and labeled history data is scarce, which makes zero-shot forecasting particularly challenging. Existing time-series pre-trained models (TSPMs) show promise but they lack the ability to dynamically incorporate external knowledge, while conventional retrieval-augmented generation (RAG) methods are rarely extended beyond text. In this work, we present **QuiZSF**, a retrieval-augmented forecasting framework that integrates search and forecasting for time series data. The framework performs search by retrieving structurally similar sequences from a large-scale time-series database, and it performs forecasting by integrating the retrieved knowledge into the target sequence. Specifically, QuiZSF introduces a **ChronoRAG Base**, a hierarchical tree-structured database that enables scalable and domain-aware retrieval, a **Multi-grained Series Interaction Learner** that captures fine- and coarse-grained dependencies between target and retrieved sequences, and a **Model Cooperation Coherer** that adapts retrieved knowledge to TSPMs. This design teaches models to actively perform search, align auxiliary information across modalities, and leverage it for more accurate forecasting. Extensive experiments on five public benchmarks demonstrate that QuiZSF consistently outperforms strong baselines, ranking first in up to **87.5%** of zero-shot forecasting settings while maintaining high efficiency.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence.**;

Keywords

Time Series Forecasting, Retrieval-Augmented Generation, Information Retrieval

ACM Reference Format:
Shichao Ma, Zhengyang Zhou, Qihe Huang, Binwu Wang, and Yang Wang. 2026. QuiZSF: A Retrieval-Augmented Framework for Zero-Shot Time Series Forecasting. In . ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Modern Web services rely heavily on sequential data streams, ranging from traffic monitoring and user behavior modeling to anomaly prevention in large-scale online platforms. Accurate forecasting of such time series is therefore a cornerstone for building reliable and intelligent Web applications. However, Web environments are highly dynamic and data-scarce: new domains and conditions frequently emerge, while historical labels are limited or unavailable. These challenges make *zero-shot time-series forecasting* (ZSF), which aims to infer future trends of unseen sequences without domain-specific supervision, a crucial yet difficult problem [9, 13, 19, 36].

Recent progress in *time series pre-trained models* (TSPMs) has shown promising results for ZSF by transferring knowledge across datasets, analogous to pre-training in NLP and CV. TSPMs can be categorized into two groups [9]: (i) numerical Non-LLM based TSPMs such as Moment [12], TTM [9], TimesFM [7], and Moirai [35],

*Dr. Zhengyang Zhou and Prof. Yang Wang are corresponding authors.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and (ii) textual LLM based models that reconstruct language models for time series forecasting, such as LLMTime [13], Time-LLM [19], and GPT4TS [42]. Although both categories achieve strong forecasting accuracy, they face two critical limitations. First, real-world sequences are continually evolving, yet TSPMs cannot efficiently incorporate new knowledge without costly fine-tuning [31]. Second, time series often exhibit structural similarity across domains [32], but TSPMs lack mechanisms to retrieve and reuse such auxiliary patterns to improve generalization.

To address these limitations, we propose integrating TSPMs with retrieval-augmented generation (RAG), a paradigm that has shown great success in natural language processing but remains underexplored for time series forecasting. In this approach, a search module first retrieves relevant sequences from a large-scale temporal database, and a forecasting module then integrates the retrieved knowledge into the target sequence prediction process. As illustrated in Figure 1, retrieved auxiliary sequences can provide valuable contextual signals such as periodicity, seasonal shifts, or abrupt transitions, which enhance forecasting quality and reduce hallucinations in zero-shot scenarios.

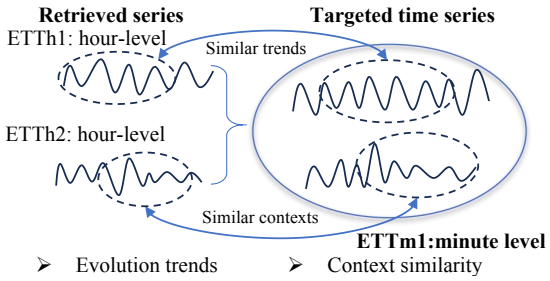


Figure 1: Motivation. Time series across domains often exhibit similar temporal patterns, which can be retrieved and reused as auxiliary knowledge.

Nevertheless, combining retrieval with forecasting introduces three core challenges: (i) **Efficient and domain-sensitive storage and retrieval**: indexing millions of sequences for fast and relevant cross-domain search remains difficult at Web scale; (ii) **Multi-level feature extraction**: retrieved sequences vary in scale, domain, and noise, requiring expressive yet lightweight interaction modeling; (iii) **Modality-aligned representation integration**: numerical Non-LLM based TSPMs demand feature-level fusion, while textual LLM-based models require structured prompts that translate retrieved sequences into language-compatible inputs.

To address these challenges, we propose the **Qui**ck **Z**ero-shot time-series **S**earch and **F**orecasting framework (**QuiZSF**). QuiZSF is a retrieval-augmented forecasting framework that integrates scalable storage, cross-series interaction learning, and modality-aware adaptation. Our main contributions are summarized as follows:

- We build the **ChronoRAG Base (CRB)**, a hierarchical tree-structured temporal database, together with a **Hybrid and Hierarchical TS Retrieval (HHTR)** strategy for fast and domain-sensitive search.
- We design the **Multi-grained Series Interaction Learner (MSIL)**, a module that captures fine-grained dependencies and coarse-grained trends from retrieved sequences to improve target sequence understanding.
- We develop the **Model Cooperation Coherer (MCC)**, a dual-branch adapter that integrates retrieved information into both Non-LLM and LLM-based TSPMs.
- We conduct extensive experiments on public benchmarks, where QuiZSF outperforms state-of-the-art baselines. Using Non-LLM TSPMs, QuiZSF achieves top performance in **75%** of ZSF settings; with LLM-based TSPMs, it ranks first in **87.5%** of settings, while maintaining high efficiency in memory usage and inference speed.

2 Related Work

2.1 TSPMs for Zero-shot Forecasting

Recent advancements in TSPMs for ZSF have garnered significant attention. These models can be broadly categorized into two types. The first is **pre-trained models** designed specifically for TSF. These models learn generalizable temporal representations from large-scale TS datasets.Representative methods include Moment [12], TTM [9], TimesFM [7], Moirai [35], and Lag-LLaMA [29]. The second type leverages **pre-trained large language models (LLMs)** by framing time-series forecasting as a form of cross-domain transfer learning. These models, including LLMTime [13], Time-LLM [19], and GPT4TS [42], transform numerical sequences into natural language prompts, enabling ZSF.

Despite promising results, these models rely heavily on large-scale pre-training or fine-tuning across diverse datasets. They often lack scalability and the ability to incorporate real-time, open-world knowledge. As illustrated in Figure 1, time series from similar domains frequently share common temporal patterns, which can serve as valuable references in low-resource settings. With the advancement of retrieval techniques, integrating external sequence-level knowledge into pre-trained models enables dynamic forecasting, mitigates hallucinations, and constrains forecasting within realistic boundaries. Enhancing TSPMs with retrieval capabilities has therefore emerged as a promising direction for improving zero-shot time-series forecasting.

2.2 Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) combines models with information retrieval to enhance model performance by leveraging external knowledge. Early work, such as REALM [14], introduces retrieval-augmented models by incorporating unsupervised retrieval modules, allowing models to access relevant passages during generation. RAG [22] further advances this by combining retrieval and generation in an end-to-end framework, improving tasks like open-domain question answering. Subsequent approaches, including T5+RAG [22] and Fusion-in-Decoder (FiD) [16], enhance the fusion of retrieved information, achieving better coherence and relevance. Applying RAG to time series forecasting offers promising benefits. Recent studies [20, 32, 37] have introduced retrieval mechanisms into forecasting. However, they often overlook the cost and scalability of external databases in large-scale settings, leading to inefficiencies. While TimeRAF [39] explores zero-shot forecasting with retrieval, it lacks a clear mechanism for modeling interactions between pre-trained models and retrieved sequences.

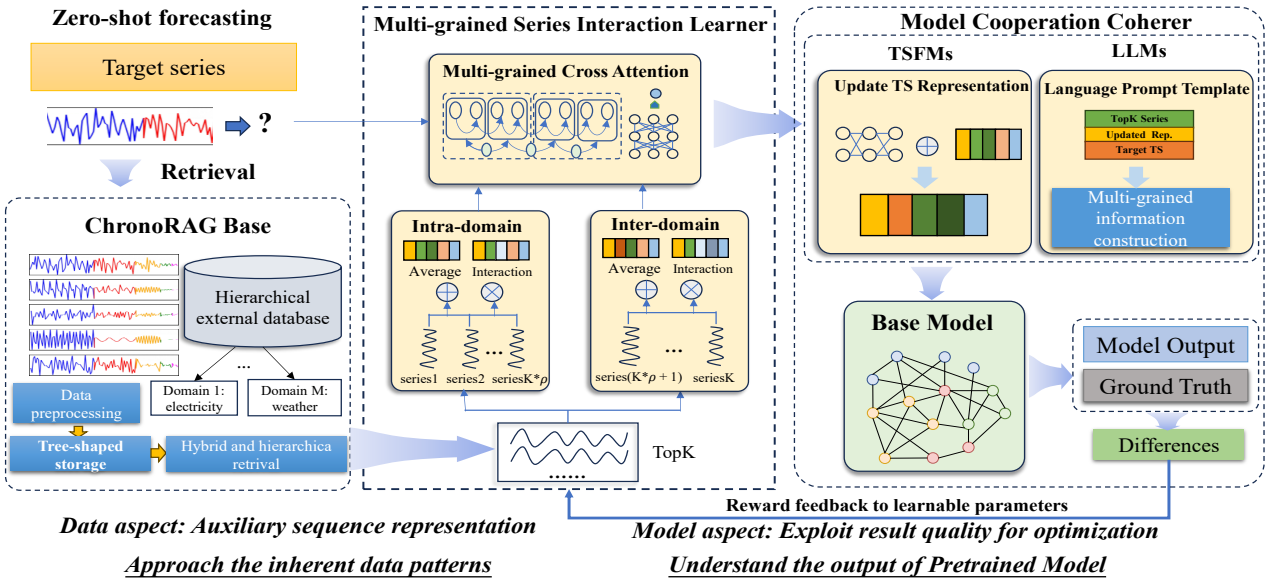


Figure 2: Overview of QuiZSF.

These pioneering studies highlight the potential value of combining pre-trained models with retrieval processes.

3 Preliminaries and Problem Definition

We focus on zero-shot forecasting (ZSF) with Time Series Pre-trained Models (TSPMs), which can be categorized into Non-LLM based (processing numerical inputs) and LLM based (operating on language inputs). We conduct separate ZSF formulations to accommodate their distinct modalities.

3.1 Retrieval-Augmented Non-LLM based TSPMs

Let $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$ denote multiple distinct TS domains, where each domain $\mathcal{G}_k$ contains series $\mathcal{G}_k = \{\mathbf{X}^k \mid x_1^k, \ldots, x_T^k\}$. These series are compiled into a dynamic auxiliary database $\mathcal{D}$.

Given a target sequence $\mathbf{X}^T$, we retrieve the top-K relevant auxiliary series $\mathbb{X}^R = \{\mathbf{X}_1^R, \ldots, \mathcal{X}_K^R\} \subset \mathcal{D}$. Since Non-LLM models do not support textual prompts, the target and retrieved series are fused into a new representation, $\widehat{\mathbf{X}}^T = \mathcal{F}_{\mathrm{N}}^*(\mathbf{X}^T, \mathbb{X}^R)$, $\widehat{\mathbf{Y}}^T = \mathcal{M}_1^*(\widehat{\mathbf{X}}^T)$, where $\mathcal{M}_1$ is the modified learning framework with Non-LLM based TSPM.

3.2 Retrieval-Augmented LLM based TSPMs

In contrast, for LLM based TSPMs, we first process the target sequence and retrieved auxiliary series to produce both a structured input and a corresponding textual prompt:

$$\widehat{\mathbf{X}}^T, \mathcal{P} = \mathcal{F}_{\mathrm{L}}^*(\mathbf{X}^T, \mathbb{X}^R), \quad \widehat{\mathbf{Y}}^T = \mathcal{M}_2^*(\widehat{\mathbf{X}}^T \mid \mathcal{P}), \tag{1}$$

where $\mathcal{M}_2$ is the modified learning framework and $\mathcal{P}$ is the generated prompt fusing the retrieved knowledge.

4 Methodology

4.1 Framework overview

Our zero-shot learning framework consists of three major components, as illustrated in Figure 2. (i) a **ChronoRAG Base** with efficient hierarchical retrieval, (ii) a **Multi-grained Series Interaction Learner** captures series-level interactions between the target and retrieved series, offering valuable information for zero-shot forecasting, (iii) a **Model Cooperation Coherer** enables modality-aligned integration between learned representations and both Non-LLM based and LLM based TSPMs.

4.2 ChronoRAG Base

ChronoRAG Base (CRB) comprises 27 time series datasets categorized into seven domains: Web, Energy, Health, IoT, Nature, Transport, and Environment. It primarily integrates five open data sources: UTSD [24], TSER Archive [30], Monash [11], TDBrain [34], and UCR Time Series Archive [8]. These datasets exhibit diverse sampling frequencies, ranging from macro-level intervals (e.g., daily) to finer granularities (e.g., hourly or minute-level). Notably, some datasets demonstrate exceptionally high sampling rates, such as the TDBrain dataset, which operates at a millisecond-level frequency. Based on these datasets, we constructed three versions of the database: CRB-Small, CRB-Medium, and CRB-Large, containing 34M, 48M, and 143M time points, respectively. Detailed information about CRB-Large can be found in Table 1. Each database covers all seven domains, and the smaller versions are subsets of the larger ones. There are three important database designs, include a well-designed data protocol for ChronRAG for processing high-quality datasets, a Hierarchical Series Tree for efficient series-level storage as well as a Hybrid and Hierarchical Time-series Retrieval for efficient retrieval.

Table 1: CRB-Large Detailed Descriptions: Domain indicates the field to which the dataset belongs. Datasets refer to the specific datasets included. Time Series represents the number of time series contained in the dataset after processing. Frequency denotes the sampling interval of time points, where "-" indicates either the absence of timestamps or irregular intervals. Time Points represents the total number of time points in the dataset. Source specifies the original paper or resource from which the dataset is obtained.

Domain	Datasets	Time Series	Frequency	Time Points	Source
Web	kaggle_web_traffic_dataset_without_missing_values	141444	Daily	72419328	Monash [11]
Energy	wind_4_seconds_dataset	1	4 Sec	512	Monash [11]
	australian_electricity_demand_dataset	5	30 Min	2560	Monash [11]
	london_smart_meters_dataset_without_missing_values	5556	Hourly	2844672	Monash [11]
Health	SelfRegulationSCP1	3366	0.004 Sec	1723392	UCR Time Series Archive [8]
	MotorImagery	24192	0.001 Sec	12386304	UCR Time Series Archive [8]
	PigCVP	312	-	159744	UCR Time Series Archive [8]
	PigArtPressure	312	-	159744	UCR Time Series Archive [8]
	SelfRegulationSCP2	2660	0.004 Sec	1361920	UCR Time Series Archive [8]
	AtrialFibrillation	60	0.008 Sec	30720	UCR Time Series Archive [8]
	IEEEPPG	15480	0.008 Sec	7925760	TSER archive [30]
	BIDMC32HR	12278	-	6286336	TSER archive [30]
	TDBrain	28644	0.002 Sec	14665728	TDBrain [34]
IoT	baian	918	0.02 Sec	470016	UTSD [24]
Nature	StarLightCurves	9236	-	4728832	UCR Time Series Archive [8]
	Phoneme	2110	-	1080320	UCR Time Series Archive [8]
	EigenWorms	1554	-	795648	UCR Time Series Archive [8]
	Worms	258	0.033 Sec	132096	UCR Time Series Archive [8]
	us_births_dataset	1	Daily	512	Monash [11]
	kdd_cup_2018_dataset_without_missing_values	270	Hourly	138240	Monash [11]
	temperature_rain_dataset_without_missing_values	32072	Daily	16420864	Monash [11]
	Sunspot_dataset_without_missing_values	1	Daily	512	Monash [11]
	saugeenday_dataset	1	Daily	512	Monash [11]
Transport	pedestrian_counts_dataset	66	Hourly	33792	Monash [11]
Environment	AustraliaRainfall	3	Hourly	1536	TSER archive [30]
	BenzeneConcentration	8	Hourly	4096	TSER archive [30]
	BeijingPM25Quality	9	Hourly	4608	TSER archive [30]

4.2.1 Data protocol for ChronoRAG.. To ensure consistency and scalability, we design a unified data protocol including sliding-window segmentation, linear interpolation for missing values, channel-independent processing for dimensional alignment, and standardized metadata. All sequences are stored in the ARROW format [28] for efficient access. Further Details are provided in Appendix A.1.

4.2.2 Hierarchical Series Tree. It supports efficient indexing and retrieval in CRB by partitioning the database into domain-based groups and using k-means clustering. This hierarchical structure accelerates approximate nearest-neighbor search and supports dynamic updates, ensuring the index evolves incrementally over time. Details are provided in Appendix A.2.

4.2.3 Hybrid and Hierarchical Time-series Retrieval. To support both high-accuracy and scalable search, we propose a hybrid and hierarchical time-series retrieval (HHTR) strategy. It combines local domain-specific matching and global prototype comparison, leveraging the hierarchical index structure built in CRB. The strategy is shown in Figure 3 (a).

Top-K Series Retrieval. Given a query time series $\mathbf{X}_T$, the objective is to retrieve its most relevant neighbors from the prototype database. This is a classical nearest-neighbor retrieval task, tailored to the time-series domain through hybrid matching strategies. When the domain of $\mathbf{X}_T$ is known and exists in CRB, we apply a combination of local and global prototype matching. The final Top-K set is computed as:

$$\text{Top } K = \rho \cdot \text{Top } K_{\text{local}} + (1-\rho) \cdot \text{Top } K_{\text{global}}, \tag{2}$$

where $\rho \in [0,1]$ controls the balance between local and global contributions. When the domain is not identified or not present in CRB, retrieval is performed across all stored cluster prototypes,

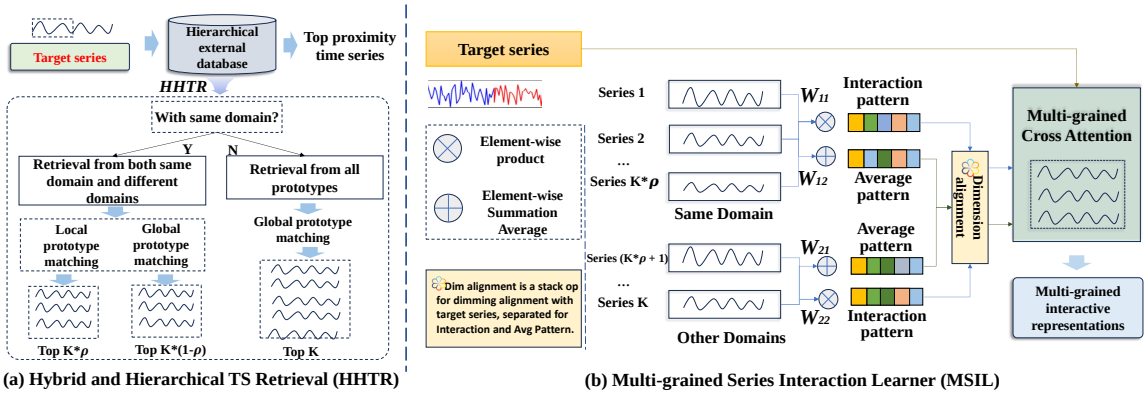
$$\text{Top } K = \text{Top } K_{\text{global}}, \tag{3}$$

Compared with existing methods, the proposed hybrid design exhibits superior performance in retrieval accuracy while maintaining higher computational efficiency.

Distance Metric Design. To quantify similarity between the query $\mathbf{X}_T$ and a candidate sequence $\mathbf{X}_i$, we define a compound similarity score:

$$\text{Sim}(\mathbf{X}_T, \mathbf{X}_i) = \cos(\mathbf{X}_T, \mathbf{X}_i) + \frac{1}{\text{dist}(\mathbf{X}_T, \mathbf{X}_i)}, \tag{4}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity and $\text{dist}(\cdot,\cdot)$ is the Euclidean distance. This dual metric emphasizes trend alignment through cosine similarity while capturing geometric proximity via Euclidean distance.



(a) Hybrid and Hierarchical TS Retrieval (HHTR) (b) Multi-grained Series Interaction Learner (MSIL)

Figure 3: (a) HHTR: Domain-aware and global retrieval via a hierarchical index. (b) MSIL: Interaction and average patterns extracted from retrieved sequences are fused with the target via cross-attention.

4.3 Multi-grained Series Interaction Learner

The core challenge after retrieving relevant series lies in effectively integrating them with the target series to enhance forecasting performance. Simply concatenating or averaging the retrieved series can lead to suboptimal results due to heterogeneity across domains. To this end, we propose the **Multi-grained Series Interaction Learner (MSIL)**, which is designed to extract robust and informative representations by modeling fine-grained interactions and global trends simultaneously.

MSIL is motivated by three considerations: (1) retrieved series may come from different domains with distinct dynamics; (2) fine-grained dependencies across series are often crucial for forecasting; and (3) domain-specific context and global knowledge should be fused in a unified representation. As illustrated in Figure 3 (b), MSIL achieves this by computing two representative patterns, namely an interaction pattern and an average pattern, which are then fused with the target sequence through a cross-attention mechanism.

Given a target time series $\mathbf{T} \in \mathbb{R}^{N \times D}$ with N time steps and D channels, and a set of retrieved series $\{\mathbf{S}_1, \ldots, \mathbf{S}_n\}$, we first divide the retrieved set into same-domain and cross-domain subsets:

$$\mathbf{S}_i \in \begin{cases} \mathbb{S}_{\text{same}} & \text{if domain}(\mathbf{S}_i) = \text{domain}(\mathbf{T}), \\ \mathbb{S}_{\text{cross}} & \text{if domain}(\mathbf{S}_i) \neq \text{domain}(\mathbf{T}) \end{cases} \quad \forall i \in n \tag{5}$$

To ensure numerical consistency, we normalize both target and retrieved sequences using the scaler module associated with their respective base models:

$$\mathbf{T}^{\text{norm}}, \text{loc}_T, \text{scale}_T = \text{scaler}(\mathbf{T}), \tag{6}$$

$$\mathbf{S}_i^{\text{norm}}, \text{loc}_{S_i}, \text{scale}_{S_i} = \text{scaler}(\mathbf{S}_i), \quad \forall i = 1, 2, \ldots, K \tag{7}$$

with

$$\mathbf{T}^{\text{norm}} = \frac{\mathbf{T} - \text{loc}_T}{\text{scale}_T}, \quad \mathbf{S}_i^{\text{norm}} = \frac{\mathbf{S}_i - \text{loc}_{S_i}}{\text{scale}_{S_i}} \tag{8}$$

Based on the normalized retrieved sequences, MSIL computes two complementary patterns: **1) Interaction Pattern** ($\mathbf{P}_{\text{int}}$) captures fine-grained dependencies via element-wise product, followed by a non-linear projection. **2) Average Pattern** ($\mathbf{P}_{\text{avg}}$) encodes global trends through mean pooling and transformation. These are defined as:

$$\mathbf{P}_{\text{int}} = \text{MLP}_1\left(\frac{\prod_{i=1}^{n} \mathbf{S}_i^{\text{norm}}}{\left\|\prod_{i=1}^{n} \mathbf{S}_i^{\text{norm}}\right\|}\right), \qquad \mathbf{P}_{\text{avg}} = \text{MLP}_2\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{S}_i^{\text{norm}}\right) \tag{9}$$

To fuse these patterns with the target sequence, we use a multi-grained cross-attention mechanism, where the target serves as query and the patterns as key/value:

$$\mathbf{Q} = \mathbf{W}_q \mathbf{T}^{\text{norm}}, \quad \mathbf{K} = \mathbf{W}_k \mathbf{P}_{\text{avg}}, \quad \mathbf{V} = \mathbf{W}_v \mathbf{P}_{\text{int}}, \tag{10}$$

$$\mathbf{R}_{\text{fused}} = \text{Softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d}}\right) \cdot \mathbf{V} \tag{11}$$

The resulting representation $\mathbf{R}_{\text{fused}}$ combines domain-aware alignment and trend-aware feature fusion. MSIL enables rich interaction modeling between the target and retrieved series, and its multi-granular design improves generalization, especially in zero-shot forecasting scenarios.

4.4 Model Cooperation Coherer

In retrieval-augmented zero-shot time-series forecasting, it is crucial to effectively connect retrieved knowledge with diverse TSPMs. These models fall into two categories: Non-LLM-based (numerical input) and LLM-based (textual input), each requiring tailored integration strategies. To fully leverage the representations produced by MSIL, we design a unified cooperation mechanism with two branches: one for numerical models and one for language models, both supporting feedback-driven optimization.

Numerical Coherer for Non-LLM based TSPMs. For numerical TS pre-trained models, we apply a residual connection to fuse the normalized target sequence $\mathbf{T}^{\text{norm}}$ with the MSIL-fused representation $\mathbf{R}_{\text{fused}}$, then feed it into the forecasting model:

$$\widehat{\mathbf{T}} = \mathcal{F}_{\text{N}}(\mathbf{R}_{\text{fused}}, \mathbf{T}^{\text{norm}}) \tag{12}$$

where $\mathcal{F}_{\text{N}}$ is a residual module [15] that enhances expressiveness and mitigates gradient vanishing, and the output $\widehat{\mathbf{T}}$ denotes the forecasted sequence.

Language Coherer for LLM based TSPMs. Language models operate exclusively on text, making direct use of numeric features infeasible. To bridge this modality gap, we convert MSIL outputs

Table 2: Long sequence forecasting results. The experimental setup follows TTM [9]. Best results are in bold; second best are underlined. Full-shot results are obtained from the Moirai [35] where the authors draw similar comparison.

		Zero-shot forecasting				Full-shot forecasting						
		QuiZSF$_T$ (Ours)	TTM$_B$ (2024)	Moirai$_B$ (2024)	TimesFM (2024)	iTransformer (2024)	Crossformer (2023)	DLinear (2023)	TimesNet (2023)	PatchTST (2023)	TiDE (2023)	FEDformer (2022)
ETTh1	96	**0.361**	0.364	0.384	0.421	0.386	0.423	0.386	0.384	0.414	0.479	0.376
	192	**0.384**	0.388	0.425	0.472	0.441	0.471	0.437	0.436	0.46	0.525	0.420
	336	**0.398**	0.402	0.456	0.51	0.487	0.570	0.481	0.491	0.501	0.565	0.459
	720	**0.468**	0.471	0.470	0.514	0.503	0.653	0.519	0.521	0.500	0.594	0.506
	Avg	**0.403**	0.406	0.434	0.479	0.454	0.529	0.456	0.458	0.469	0.541	0.440
ETTh2	96	**0.276**	0.279	0.277	0.326	0.297	0.745	0.333	0.34	0.302	0.4	0.358
	192	**0.334**	**0.334**	0.34	0.400	0.380	0.877	0.477	0.402	0.388	0.528	0.429
	336	**0.364**	0.366	0.371	0.434	0.428	1.043	0.594	0.452	0.426	0.643	0.496
	720	0.407	0.408	**0.394**	0.451	0.427	1.104	0.831	0.462	0.431	0.874	0.463
	Avg	**0.345**	0.347	0.346	0.403	0.383	0.942	0.559	0.414	0.388	0.611	0.437
ETTm1	96	0.369	0.359	0.335	0.357	0.334	0.404	0.345	0.338	**0.329**	0.364	0.379
	192	0.377	0.376	0.367	0.411	**0.337**	0.450	0.38	0.374	0.367	0.398	0.426
	336	**0.397**	0.407	0.398	0.442	0.426	0.532	0.413	0.41	0.409	0.428	0.445
	720	0.441	0.446	**0.434**	0.507	0.491	0.666	0.474	0.478	0.481	0.487	0.543
	Avg	0.395	0.397	**0.383**	0.429	0.397	0.513	0.403	0.400	0.397	0.419	0.448
ETTm2	96	0.176	0.178	0.195	0.205	0.18	0.287	0.193	0.187	**0.175**	0.207	0.203
	192	**0.238**	**0.238**	0.247	0.293	0.25	0.414	0.284	0.249	0.241	0.290	0.269
	336	**0.292**	0.300	0.293	0.366	0.311	0.597	0.369	0.321	0.305	0.377	0.325
	720	0.390	0.41	**0.365**	0.472	0.412	1.730	0.554	0.408	0.402	0.558	0.421
	Avg	**0.274**	0.282	0.275	0.334	0.288	0.757	0.350	0.291	0.281	0.358	0.305
Weather	96	**0.153**	0.158	0.167	-	0.174	0.158	0.196	0.172	0.177	0.202	0.217
	192	**0.194**	0.206	0.209	-	0.221	0.206	0.237	0.219	0.225	0.242	0.276
	336	**0.251**	0.260	0.256	-	0.278	0.272	0.283	0.280	0.278	0.287	0.339
	720	**0.324**	0.330	0.325	-	0.358	0.398	0.345	0.365	0.354	0.351	0.403
	Avg	**0.231**	0.239	0.239	-	0.258	0.259	0.265	0.259	0.259	0.271	0.309

$(\mathbf{P}_{\text{int}}, \mathbf{P}_{\text{avg}}, \mathbf{T}^{\text{norm}})$ into structured textual summaries. These summaries, along with an instruction-style prompt, guide the language model in generating forecasting outputs. See Appendix B and Figure 9 for prompt construction details.

5 Experiments

5.1 Experiments Setups

For **evaluation datasets**, we use five public datasets: ETTh1, ETTh2, ETTm1, ETTm2, and Weather, which are widely used in prior state-of-the-art works [19, 26, 42]. Standard error metric is MSE. For **CRB selection**, our ChronoRAG Base comes in three versions: CRB-Small, CRB-Medium, and CRB-Large. Given the balance between computing time and experimental results, we select CRB-Medium as the RAG Base. For **model comparison**, we evaluate against thirteen state-of-the-art forecasting methods, which can be classified into the following categories: (a) **Non-LLM based TSPMs:** TTM [9], Moirai [35], and TimesFM [7]; (b) **LLM based TSPMs:** TimeLLM [19], LLMTime [13], and GPT4TS [42]; (c) **Other architectures:** iTransformer [23], Crossformer [40], DLinear [38], TimesNet [36], TiDE [6], PatchTST [26], and FEDformer [41].

5.2 Implementation Details

In the time series domain, **zero-shot time-series forecasting** refers to evaluating models on unseen datasets without direct supervision or fine-tuning. Given the two types of TSPMs, current ZSF setups are categorized accordingly: the **multi-source generalization setup** for Non-LLM based models, and the **single-source transfer setup** for LLM based models. To ensure comprehensive evaluation, we adopt both setups in our experiments, resulting in **QuiZSF$_T$** and **QuiZSF$_L$**, respectively.

Multi-source generalization zero-shot setup (for Non-LLM based TSPMs) This setup trains on diverse source datasets and enables zero-shot forecasting by directly applying the model to unseen target datasets. This approach is widely used in Non-LLM based TSPMs like TTM [9], TimesFM [7], and Moirai [35]. We use this setup for **QuiZSF$_T$**, with TTM-Base [9] as the base model. For training, we use a comprehensive dataset covering 38.7 million time points, curated from multiple public benchmarks to ensure diversity. Evaluation is conducted on the held-out ETT and Weather datasets, ensuring strict zero-shot conditions without data leakage. Details of the training datasets can be found in Appendix C.

Single-source transfer setup (for LLM based TSPMs) This setup trains a model on a single source dataset (e.g., ETTh1) and evaluates it on an unseen target dataset (e.g., ETTm2), highlighting the model's ability to transfer knowledge across domains. Commonly used in LLM based TSPMs [19, 42], this setup involves fine-tuning or prompting LLMs on a specific dataset and then applying them to novel inputs from a different domain. We use this setup for **QuiZSF$_L$**, with TimeLLM [19] as the base model and LLaMA-7B [33] as the backbone. The evaluation is carried out in multiple train-test splits according to the experimental setup of TimeLLM zero-shot forecasting (Table 3).

Table 3: Zero-shot forecasting results under the single-source transfer setup. Following the setting of TimeLLM [19], results are averaged over prediction lengths {96, 192, 336, 720}. Best scores are in bold, second best are underlined.

	QuiZSF$_L$	Time-LLM	LLMTime	GPT4TS	DLinear	PatchTST	TimesNet	Autoformer
ETTh1 → ETTh2	**0.352**	0.356	0.992	0.406	0.493	0.380	0.421	0.582
ETTh1 → ETTm2	**0.272**	0.277	1.867	0.325	0.415	0.314	0.327	0.457
ETTh2 → ETTh1	0.535	**0.521**	1.961	0.757	0.703	0.545	0.865	0.757
ETTh2 → ETTm2	**0.269**	0.271	1.867	0.335	0.328	0.325	0.342	0.366
ETTm1 → ETTh2	**0.382**	0.394	0.992	0.433	0.464	0.439	0.457	0.470
ETTm1 → ETTm2	**0.281**	0.296	1.867	0.313	0.335	0.291	0.322	0.469
ETTm2 → ETTh2	**0.351**	0.354	0.992	0.435	0.455	0.409	0.435	0.423
ETTm2 → ETTm1	**0.414**	0.418	1.993	0.769	0.649	0.568	0.769	0.755

Table 4: Ablation studies on zero-shot settings

	ETTm1 → ETTh2	ETTm1 → ETTm2
QuiZSF$_L$-w/o-RAG	0.394	0.296
QuiZSF$_L$-w/o-MSIL	0.387	0.286
QuiZSF$_L$-w/o-Coherer	0.388	0.289
QuiZSF$_L$	**0.382**	**0.281**

5.3 Peformance Comparison

The prediction results of **QuiZSF$_T$** and **QuiZSF$_L$** are shown in Table 2 and Table 3, respectively. The best performance is marked in bold, and the second best is underlined.

QuiZSF$_T$. Currently, there is limited work on zero-shot forecasting. To provide a more comprehensive assessment, we compare QuiZSF$_T$ not only with zero-shot methods but also with several strong full-shot forecasting models. These comparisons fall under the category of long-sequence forecasting. As shown in Table 2, **"zero-shot"** refers to the prediction results of various base models without any pre-training on the test datasets, while **"full-shot"** denotes the performance of benchmark models that have been fully trained on each dataset. QuiZSF$_T$, trained solely under the zero-shot setting, outperforms not only existing zero-shot baselines but also full-shot models, demonstrating strong generalization capabilities even without access to target-domain training data. QuiZSF$_T$ ranks Top1 in **75%** of ZSF settings. However, we observe that QuiZSF$_T$ performs particularly well on relatively coarse-grained datasets but shows limited effectiveness on short-term, minute-scale forecasting tasks (e.g., ETTm1). This may be due to the difficulty in aligning retrieved information with fine-grained fluctuations in the target sequence. Leveraging coarse-grained knowledge to guide fine-grained prediction presents an interesting direction for future research. Combining complexity and efficiency comparison in Figure 4 (a), we consider QuiZSF$_T$ to be an excellent lightweight zero-shot forecasting framework, which benefits from the retrieval-enhanced representation and active feedback.

To further validate the generalization capability of QuiZSF$_T$, we also evaluate its performance when the domain of the target series is not within the seven domains of CRB. For this purpose, we use the Weather dataset (Meteorological Domain) as the test set for experimentation, with results shown in Table 2. QuiZSF$_T$ achieves the best performance across all prediction horizons, demonstrating its strong generalization capability.

QuiZSF$_L$. As shown in Table 3, QuiZSF$_L$, equipped with retrieved augmented series, outperforms the majority of competing methods, achieving the best results in 7 out of 8 prediction settings. Additionally, our approach enhances the performance based on the pre-trained model. With the continuous advancement of pre-trained models, QuiZSF$_L$ will bring about further performance improvements when adapting to new model frameworks.

5.4 Ablation study

Ablative variants. 1) QuiZSF$_L$-w/o-RAG. We remove the auxiliary sequence retrieval and only utilize the LLM for ZSF to verify the motivation for RAG, which degenerates to LLMTime [13]. **2) QuiZSF$_L$-w/o-MSIL.** We remove the Multi-grained Series Interaction Learner, performing only an average calculation on the retrieved time series instead of feature extraction. **3) QuiZSF$_L$-w/o-Coherer.** We remove the structured prompt template and directly concatenate MSIL-extracted features with the target sequence, without adapting them into LLM-compatible textual inputs.

Main results. Results are presented in Table 4, with key findings summarized as follows: (i) The most significant performance drop occurs when removing the retrieval module (RAG), confirming its importance in providing external contextual signals for zero-shot forecasting. Compared to the full model, this variant sees a performance decline of 3.14%–5.34% (*line 1 vs. line 4*). (ii) Removing MSIL and using simple averaging instead also degrades performance. This is because the model cannot learn multi-level representations and capture relationships between sequences, resulting in performance degradation (*line 2 vs. line 4*). (iii) Additionally, without converting numerical time series into LLM-understandable tokens using the prompt template, the LLM's performance is inferior to the full QuiZSF$_L$, with a drop of about 2% (*line 3 vs. line 4*). These results demonstrate the effectiveness of our integrated approach.

5.5 Detailed model analysis

Complexity analysis. We report empirical comparisons of model size and inference time in Figure 4 (a). QuiZSF$_T$ maintains competitive efficiency while outperforming most baselines. Although it introduces minor computational overhead due to retrieval and interaction modules, the added cost is minimal. Full analysis is provided in Appendix D.1.

Hyperparameter analysis. Key hyperparameters include the size of the retrieval database, the number of retrieved sequences

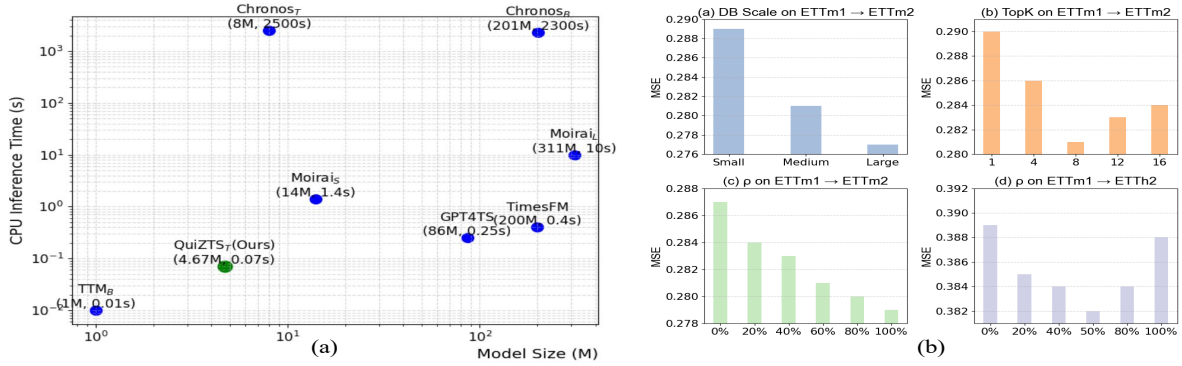


Figure 4: (a) Size and time overview of QuiZTST vs. pre-trained TS benchmarks. Plot each model based on model size and the CPU inference time per batch. (b) Hyperparameter analysis on ETTm1 -> ETTm2.

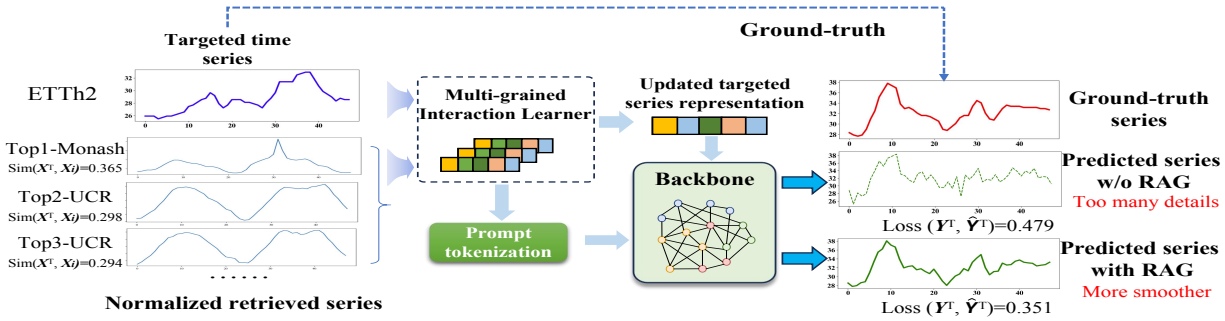


Figure 5: Case studies on ETTh2 prediction.

K, and the local-domain proportion ρ. As shown in Figure 4 (b), performance is sensitive to these choices. CRB-Medium, $K = 8$, and $\rho = 60\%$ strike a good balance between accuracy and efficiency. Detailed experiments and discussion are provided in Appendix D.2.

5.6 Case study

In order to demonstrate how the retrieval sequence improves the prediction effect, we conduct an intuitive analysis of the intermediate results. Taking the target sequence of the ETTh2 dataset as an example (as shown in Figure 5), we screen out the Top-8 sequences that are closest to the target sequence through hybrid and hierarchical time-series retrieval and marked their similarities. These sequences have similar patterns and evolution models as the target sequence. Subsequently, we update the sequence representation of the target sequence by combining the auxiliary sequence with the target sequence through MSIL and input it into the LLM after generating prompts. We visualized and compared the output of the LLM with RAG and the output w/o RAG.

The results show that RAG can reveal the average pattern of the retrieval sequence, making the prediction results smoother and avoiding overfitting; while the output w/o RAG fluctuates more and contains more inaccurate details. This indicates that RAG effectively suppresses the time-series hallucination of the LLM. Our analysis enhances the interpretability of the model, deepens the understanding of zero-shot forecasting, and highlights the contribution of RAG in enhancing prediction.

6 Conclusion

We presented **QuiZSF**, a retrieval-augmented framework for zero-shot time series forecasting that unifies search and forecasting in dynamic Web environments. QuiZSF combines a tree-structured temporal database, hybrid retrieval strategies, multi-level interaction learning, and modality-aware adaptation, enabling models to retrieve auxiliary sequences and incorporate them for more accurate prediction. Experiments on public benchmarks show that QuiZSF achieves state-of-the-art performance with both Non-LLM and LLM-based TSPMs, while maintaining efficiency in memory and inference. Beyond forecasting, this work illustrates how retrieval-augmented AI can extend beyond text to time series, offering new perspectives for the WWW community and opening opportunities for adaptive and intelligent Web-scale systems.

7 acknowledgements

This paper is partially supported by the National Natural Science Foundation of China (No.62502488, No.12227901), Natural Science Foundation of Jiangsu Province (BK20240460), the grant from State Key Laboratory of Resources and Environmental Information System.

References

[1] Sahand Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. Time-series clustering–A decade review. *Information Systems* 53 (2015), 16–38.

[2] Manos Athanassoulis and Anastasia Ailamaki. 2014. BF-tree: approximate tree indexing. In *Proceedings of the 40th International Conference on Very Large Databases*.

[3] Stefan Berchtold, Daniel A Keim, and Hans-Peter Kriegel. 1996. The X-tree: An index structure for high-dimensional data. (1996).

[4] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is "nearest neighbor" meaningful?. In *International Conference on Database Theory*. Springer, 217–235.

[5] Feng Cao, Martin Ester, Weining Qian, and Aoying Zhou. 2006. Density-based clustering over data stream. In *Proceedings of the 2006 SIAM International Conference on Data Mining*. SIAM, 328–339.

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

[7] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*.

[8] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.

[9] Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H Nguyen, Wesley M Gifford, Chandra Reddy, and Jayant Kalagnanam. 2024. Tiny Time Mixers (TTMs): Fast Pre-trained Models for Enhanced Zero/Few-Shot Forecasting of Multivariate Time Series. *CoRR* (2024).

[10] Milton Friedman. 1962. The interpolation of time series by related series. *J. Amer. Statist. Assoc.* 57, 300 (1962), 729–757.

[11] Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. 2021. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643* (2021).

[12] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885* (2024).

[13] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2024. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems* 36 (2024).

[14] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[16] Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* (2020).

[17] Christian S Jensen, Dan Lin, and Beng Chin Ooi. 2004. Query and update efficient B+-tree based indexing of moving objects. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 768–779.

[18] Søren Kejser Jensen, Torben Bach Pedersen, and Christian Thomsen. 2017. Time series management systems: A survey. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2581–2600.

[19] 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. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728* (2023).

[20] Baoyu Jing, Si Zhang, Yada Zhu, Bin Peng, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. 2022. Retrieval based time series forecasting. *arXiv preprint arXiv:2209.13525* (2022).

[21] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).

[22] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

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

[24] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. [n. d.]. Timer: Generative Pre-trained Transformers Are Large Time Series Models. In *Forty-first International Conference on Machine Learning*.

[25] Kasper Overgaard Mortensen, Fatemeh Zardbani, Mohammad Ahsanul Haque, Steinn Ymir Agustsson, Davide Mottin, Philip Hofmann, and Panagiotis Karras. 2023. Marigold: Efficient k-Means Clustering in High Dimensions. *Proceedings of the VLDB Endowment* 16, 7 (2023), 1740–1748.

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

[27] Tuomas Pelkonen, Scott Franklin, Justin Teller, Paul Cavallaro, Qi Huang, Justin Meza, and Kaushik Veeraraghavan. 2015. Gorilla: A fast, scalable, in-memory time series database. *Proceedings of the VLDB Endowment* 8, 12 (2015), 1816–1827.

[28] Johan Peltenburg, Jeroen Van Straten, Lars Wijtemans, Lars Van Leeuwen, Zaid Al-Ars, and Peter Hofstee. 2019. Fletcher: A framework to efficiently integrate FPGA accelerators with apache arrow. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 270–277.

[29] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, et al. 2023. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

[30] Chang Wei Tan, Christoph Bergmeir, François Petitjean, and Geoffrey I Webb. 2021. Time series extrinsic regression: Predicting numeric values from time series data. *Data Mining and Knowledge Discovery* 35, 3 (2021), 1032–1060.

[31] Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. 2024. Are language models actually useful for time series forecasting?. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[32] Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. 2024. Retrieval Augmented Time Series Forecasting. *arXiv preprint arXiv:2411.08249* (2024).

[33] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[34] Hanneke Van Dijk, Guido Van Wingen, Damiaan Denys, Sebastian Olbrich, Rosalinde Van Ruth, and Martijn Arns. 2022. The two decades brainclinics research archive for insights in neurophysiology (TDBRAIN) database. *Scientific data* 9, 1 (2022), 333.

[35] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592* (2024).

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

[37] Chin-Chia Michael Yeh, Huiyuan Chen, Xin Dai, Yan Zheng, Junpeng Wang, Vivian Lai, Yujie Fan, Audrey Der, Zhongfang Zhuang, Liang Wang, et al. 2023. An efficient content-based time series retrieval system. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4909–4915.

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

[39] Huanyu Zhang, Chang Xu, Yi-Fan Zhang, Zhang Zhang, Liang Wang, Jiang Bian, and Tieniu Tan. 2024. TimeRAF: Retrieval-Augmented Foundation model for Zero-shot Time Series Forecasting. *arXiv preprint arXiv:2412.20810* (2024).

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

[41] Tian Zhou, Ziqing Ma, Chi-Man Leung, et al. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *ICLR* (2022).

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

A Detailed Design of ChronoRAG Base

A.1 Data protocol for ChronoRAG

Large-scale time series datasets are essential for retrieval tasks, but constructing the ChronoRAG Base (CRB) presents challenges such as inconsistent lengths, dimensionality mismatches, missing values, metadata diversity, and storage scalability. To tackle these issues, we design a unified data protocol that ensures consistent preprocessing, metadata unification, and efficient storage. All data is stored using the ARROW format [28], which is optimized for deep learning frameworks and enables efficient retrieval and access.

To address the issue of **data length inconsistencies across datasets**, we adopt a sliding window approach. Let $\mathbf{X} \in \mathbb{R}^{N \times D}$ be a multivariate time series with N time steps and D channels. The j-th channel is denoted as $\mathbf{x}_j = [x_{1j}, x_{2j}, \cdots, x_{Nj}]$. Following the foundation model settings [9], we set window size w and step size s ($w \leq N$), and segment each channel into uniform windows:

$$\mathbf{X}_{kj} = [x_{(k-1)s+1,j}, \cdots, x_{(k-1)s+w,j}] \tag{13}$$

where $(k-1)s + w \leq N$. This preserves local patterns and improves retrieval efficiency.

To address the issue of **varying dimensionality**, we use a channel-independent strategy, which treats each dimension separately and has been validated by PatchTST [26], TimeLLM [19], and TTM [9]. For each channel $\mathbf{x}_i$, we apply a shared function f:

$$\mathbf{y}_i = f(\mathbf{x}_i) \tag{14}$$

This simplifies database construction and fusion, while enhancing scalability and cross-domain adaptability.

To address the issue of **missing values**, which may impair data integrity and affect retrieval, we apply linear interpolation [10] to complete incomplete sequences.

To address the issue of **diverse metadata**, we define a unified metadata protocol by standardizing key attributes such as item ID, start time, end time, frequency, domain, and sequence values (see Table 5). This ensures consistent integration across multi-source datasets.

To address the issue of **large volume and variety of sequence data**, which make efficiently storing and retrieving a significant challenge, we implement a hierarchical tree-like storage structure [2, 3, 17]. This enables efficient storage and indexing for large-scale datasets and seamless integration into deep learning frameworks. Details are provided in Section A.2.

Table 5: Structural key-value instance in CRB

	Meta information					Deterministic observation
Key	Domain Category	Item_id	Start	End	Freq	Target
Value	Nature	us_births_ dataset_0_0	20000101	20010527	Daily	[9083,8006,11136,……]

A.2 Hierarchical Series Tree

To support efficient indexing and retrieval in ChronoRAG Base (CRB), we design a hierarchical tree structure with pre-clustering. Traditional FIFO-based linear storage [18, 27] suffers from inefficiency when scaling to millions of time series. Linear retrieval requires one-by-one comparisons with time complexity $T_{\text{linear}} = O(N)$, which becomes prohibitive at large scale.

To mitigate this, we construct a tree-shaped structure inspired by database indexing techniques [2, 3, 17]. At the top level of the structure, the database is partitioned by domain: given a dataset $\mathcal{X} = \{X_1, \ldots, X_N\}$, we separate it into K disjoint domain-based groups $\{\mathcal{D}_k\}_{k=1}^{K}$, such that:

$$\mathcal{X} = \bigcup_{k=1}^{K} \mathcal{D}_k, \quad \mathcal{D}_i \cap \mathcal{D}_j = \emptyset \text{ for } i \neq j. \tag{15}$$

Each domain group $\mathcal{D}_k$ is then recursively divided using the k-means algorithm, where each cluster contains at most $N = 256$ time series. This setting follows the clustering granularity used in prior work such as Marigold [25], where small prototype groups (typically $N \leq 256$) are shown to improve retrieval quality and update flexibility. The number of clusters M_k in each domain is thus determined by the size of $\mathcal{D}_k$, i.e., $M_k = \lceil \frac{|\mathcal{D}_k|}{N} \rceil$.

Each cluster $C_m^{(k)}$ is formed by minimizing the standard k-means objective function:

$$\mathcal{L}_k = \sum_{X_i \in \mathcal{D}_k} \|X_i - C_{j(i)}^{(k)}\|^2, \tag{16}$$

where $C_{j(i)}^{(k)}$ is the centroid of the cluster to which X_i belongs. For each cluster, the prototype is selected as the sequence closest to its centroid:

$$X_{\text{proto}}^{(k,m)} = \arg \min_{X_i \in C_m^{(k)}} \|X_i - C_m^{(k)}\|^2. \tag{17}$$

While tree-based structures have a theoretical average-case complexity of $O(\log_b N)$, this does not always hold in high-dimensional time series due to the curse of dimensionality [4]. Hence, instead of relying solely on theoretical claims, we report empirical improvements in retrieval speed and memory in Section 5.5.

This hierarchical prototype-based structure supports domain-level filtering and accelerates approximate nearest-neighbor search. During retrieval, a query is first matched against cluster prototypes, and then only a small number of candidate clusters are examined in full. This greatly reduces computation compared to flat comparisons.

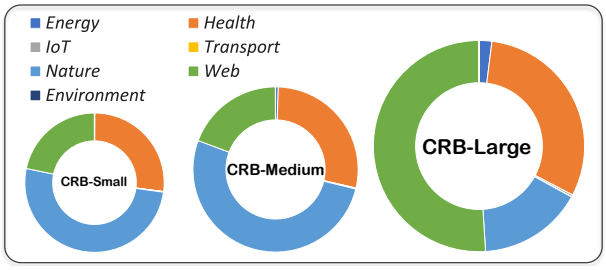


Figure 6: Three Versions of CRB.

Finally, the tree structure supports dynamic updates. When a new sequence X_{new} arrives, it is first matched to the nearest prototype using:

$$X_{\text{proto}}^{*} = \arg \min_{X_{\text{proto}}} \|X_{\text{new}} - X_{\text{proto}}\|^2, \tag{18}$$

and inserted into the corresponding cluster $C_m^{(k)}$. If the cluster exceeds the predefined maximum size N, local re-clustering is triggered within the affected subtree.

This local re-clustering process reassigns the sequences in the overflowed cluster by minimizing the intra-cluster distance:

$$\min_{\{C_i^{(k)}\}} \sum_{X_j \in \bigcup C_i^{(k)}} \left\| X_j - C_{i(j)}^{(k)} \right\|^2, \tag{19}$$

where the optimization is restricted to the current subtree, and $C_{i(j)}^{(k)}$ denotes the centroid of the sub-cluster assigned to X_j. This ensures that updates remain computationally tractable and localized.

The above update strategy allows the index to evolve incrementally over time without full reorganization, and is inspired by dynamic clustering methods in data streams [1, 5].

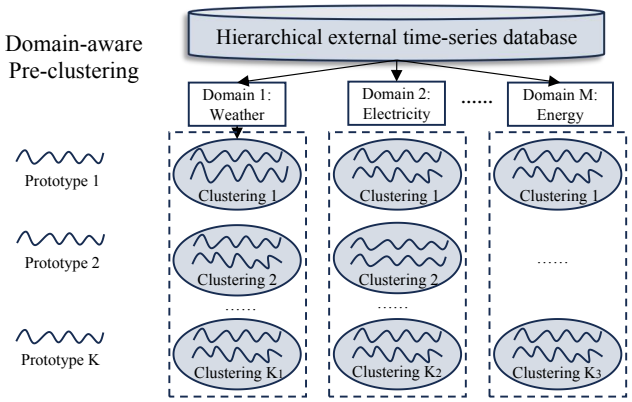


Figure 7: Tree-shaped hierarchical series organization with pre-clustering. Prototypes represent cluster centroids.

B Language Coherer for LLM based TSPMs

Unlike numerical models, language models operate exclusively on text. Direct integration of numeric features is thus infeasible due to modality mismatch. To bridge this gap, we transform the MSIL-derived representations (i.e., $\mathbf{P}_{\text{int}}$, $\mathbf{P}_{\text{avg}}$, and $\mathbf{T}^{\text{norm}}$) into structured textual summaries (see Figure 9). These summaries are combined with an instruction prompt to guide the language models in generating forecasting outputs.

The transformation process involves converting numerical features into a format that can be understood by language models. This is achieved by creating textual summaries that capture the essential characteristics of the retrieved time series. The summaries are then combined with an instruction prompt that provides context and guidance for the language model to generate accurate forecasts.

This approach ensures that the language model can effectively leverage the retrieved knowledge, even though it operates on a different modality. By converting numerical features into structured text, we enable seamless integration with the language model, allowing it to generate more accurate and reliable forecasts.

C Training Dataset Details

Detailed information of the training set is shown in Figure 8. The x-axis denotes domains. In addition to the seven core domains (Table 1), extra domains are included to enhance generalization. The y-axis shows the number of time points and datasets per domain.

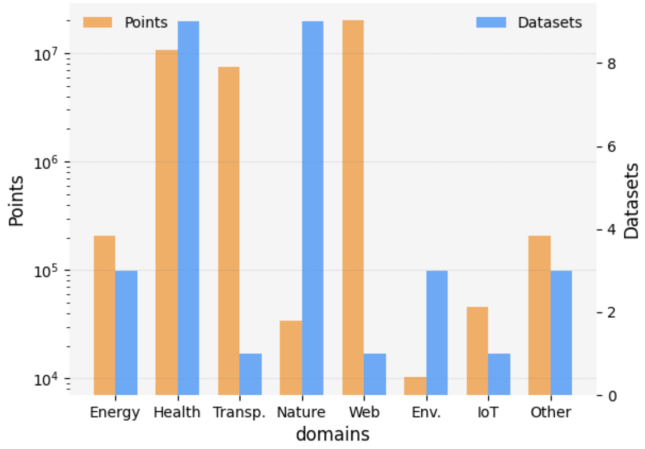


Figure 8: Detailed information of the training set.

D Additional Experimental Analyses

D.1 Complexity Analysis

We conduct a detailed empirical analysis and present a comparison of the complexity and efficiency of the QuiZSF$_T$ in Figure 4 (a). The comparison is mainly carried out from two aspects: the model size (in mebibytes, MiB) and the CPU inference time per batch (s/iter). QuiZSF$_T$ shows a notable performance in both metrics, with a clear advantage over most comparative models, falling just slightly short of TTM$_B$. Upon further investigation, it is found that this is due to the introduction of learnable modules in the retrieval and feature extraction process based on Retrieval-Augmented Generation (RAG). However, it is worth mentioning that these modules introduced by QuiZSF$_T$ are of a lightweight design, with very few additional parameters. Moreover, the retrieval and feature extraction processes rely on dot product calculations, which are highly efficient and do not significantly extend inference time. The empirical results clearly indicate that while the introduction of QuiZSF$_T$ brings certain memory and time overheads, these overheads are within an acceptable range, and at the same time, the model's performance is significantly enhanced. We further believe that by properly adjusting the hyperparameters in the retrieval process and the learnable weights, the number of model parameters can be reduced, thus further optimizing the model.

D.2 Hyperparameter Study and Analysis

We specify three key hyperparameters to explore how to achieve their optimal performance on QuiZSF$_L$. Firstly, "Database Scale", which includes three scales of CRB, namely {Small, Medium, Large}. Secondly, it is the number of time series retrieved from the auxiliary sequence dataset, with values in the range of $K = \{1, 4, 8, 12, 16\}$. Thirdly, it is the proportion of retrieved time series that are in the

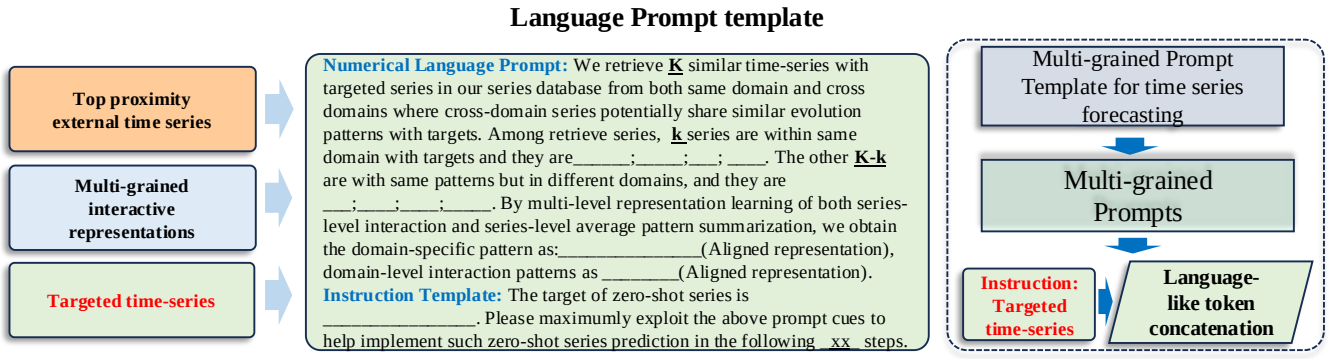


Figure 9: Prompt construction framework for LLM-based forecasting.

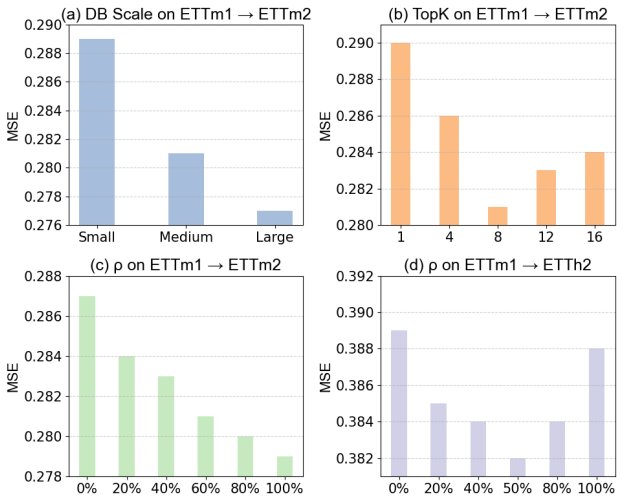


Figure 10: Hyperparameter analysis.

same domain as the target series, that is, the Local prototype ρ, with values of {0%, 20%, 40%, 60%, 80%, 100%}. Due to space limitations, the first two experiments are only elaborated in the ETTm1 → ETTm2 task, while the last experiment is described in both the ETTm1 → ETTm2 and ETTm1 → ETTh2 tasks, as shown in Figure 10.

The CRB_Small scale performs worst (Figure 10 (a)). As the scale decreases, the external knowledge it provides declines, leading to poor performance. This partly verifies the scaling law [21] in time series. In the hyperparameter experiment for the retrieved number K, TopK = 8 yields the best results. Retrieving more sequences may introduce more noise, while fewer sequences carry less information (Figure 10 (b)). In the hyperparameter experiment for the Local prototype ρ, two tasks show different trends. A too-high proportion of the same domain limits data feature diversity, over-emphasizing single-dimension features. The model performs well on samples fitting this feature (Figure 10 (c)), but poorly on non-matching samples due to the lack of auxiliary correction from other dimensions, resulting in a polarized outcome (Figure 10 (d)). To balance performance and efficiency, we select CRB_Medium, set Top$K$ to 8, and set the Local prototype ρ at 60%.

D.3 Case Study

In order to demonstrate how the retrieval sequence improves the prediction effect, we conduct an intuitive analysis of the intermediate results. Taking the target sequence of the ETTh2 dataset as an example (as shown in Figure 5), we screen out the Top-8 sequences that are closest to the target sequence through hybrid and hierarchical time-series retrieval and marked their similarities. These sequences have similar patterns and evolution models as the target sequence. Subsequently, we update the sequence representation of the target sequence by combining the auxiliary sequence with the target sequence through MSIL and input it into the LLM after generating prompts. We visualized and compared the output of the LLM with RAG and the output w/o RAG.

The results show that RAG can reveal the average pattern of the retrieval sequence, making the prediction results smoother and avoiding overfitting; while the output w/o RAG fluctuates more and contains more inaccurate details. This indicates that RAG effectively suppresses the time-series hallucination of the LLM. Our analysis enhances the interpretability of the model, deepens the understanding of zero-shot forecasting, and highlights the contribution of RAG in enhancing prediction.

D.4 Experiments compute resources

All experiments were conducted on a single NVIDIA A100 GPU with 40GB memory.

E Limitations and Future Work

QuiZSF, while advancing zero-shot time series forecasting, faces limitations: its effectiveness relies on pre-trained model quality, which is limited in data-scarce domains; large-scale retrieval efficiency remains a challenge, potentially addressable with sparse indexing; and multi-granularity sequence learning for cross-pattern transfer is needed to enhance generalization across heterogeneous streams, ultimately strengthening QuiZSF and contributing to scalable retrieval-augmented AI systems for diverse Web environments.