Not All Retrievals are Useful: Cross-Attention for Input-Aware RAG in Time Series Forecasting

Seunghan Lee
LG AI Research
Seoul, South Korea

Jaehoon Lee
LG AI Research
Seoul, South Korea

Jun Seo
LG AI Research
Seoul, South Korea

Sungdong Yoo
LG AI Research
Seoul, South Korea

Minjae Kim
LG AI Research
Seoul, South Korea

Tae Yoon Lim
LG AI Research
Seoul, South Korea

Dongwan Kang
LG AI Research
Seoul, South Korea

Hwanil Choi
LG AI Research
Seoul, South Korea

SoonYoung Lee
LG AI Research
Seoul, South Korea

Wonbin Ahn
LG AI Research
Seoul, South Korea

Abstract

Retrieval-augmented generation (RAG) enhances zero-shot time series (TS) forecasting by leveraging external knowledge bases, yet existing approaches overlook input-level relevance when fusing retrieved samples with the query. We argue that *not all retrievals are equally useful*, and irrelevant ones can degrade performance. To this end, we propose **Cross-RAG**, a zero-shot RAG-based forecasting framework that *selectively* attends to query-relevant retrieved samples via query–retrieval *cross-attention*. By modeling input-level relevance between the query and retrieved samples, Cross-RAG jointly incorporates three sources of information: 1) the query itself, 2) the retrieved samples, and 3) their relational interactions. In particular, this input-aware design enables Cross-RAG to remain stable as the number of retrieved samples k grows, whereas prior methods without cross-attention require careful k tuning to avoid degradation from irrelevant retrievals. Extensive experiments demonstrate that Cross-RAG consistently improves zero-shot forecasting performance across multiple TSFM backbones and various RAG methods, with additional analyses confirming its effectiveness across various retrieval scenarios. Code is available at: https://github.com/seunghan96/Cross-RAG.

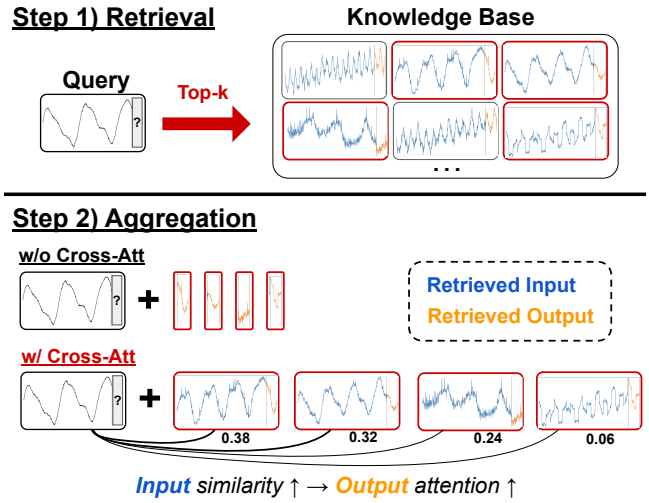


Figure 1: Cross-attention btw query & retrieved inputs. Cross-RAG performs *input-aware* fusion with *cross-attention* to weight retrieved samples based on the input similarity.

1 Introduction

Time series (TS) forecasting is widely adopted across diverse domains, including finance [25] and traffic systems [5]. Deep learning–based approaches, such as recurrent [32], convolutional [20], graph-based [11], and transformer-based models [17], effectively capture temporal dependencies, while their training remains largely confined to domain-specific datasets.

Recent studies investigate time series foundation models (TSFMs) to enable more generalizable forecasting through large-scale pretraining [2, 6, 12, 14, 28]. Despite their expressive capacity, these models exhibit limited robustness in zero-shot settings due to distribution mismatch between the pretraining and unseen datasets, incur substantial adaptation cost, and lack mechanisms for dynamic domain conditioning and interpretability [21].

Recently, retrieval-augmented generation (RAG) enhances context awareness across diverse tasks by retrieving relevant contextual evidence and incorporating it into model inputs. Several studies extend this paradigm to TS forecasting [10, 21, 34], seeking to leverage historical patterns through retrieval mechanisms. However, existing approaches for TS forecasting either rely on dataset-specific adaptation rather than zero-shot inference or fail to fully exploit the input structure of retrieved samples.

To this end, we propose **Cross-RAG**, a zero-shot retrieval-augmented forecasting framework which models input-level relevance between the query and retrieved samples by attending to their inputs via (query–retrieval) *cross-attention*, as illustrated in Figure 1. Specifically, the prediction integrates three sources of information: 1)

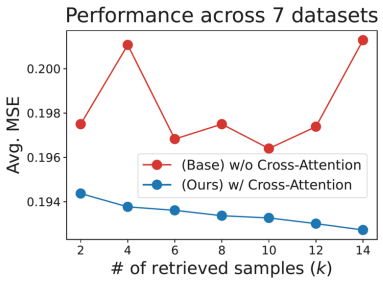
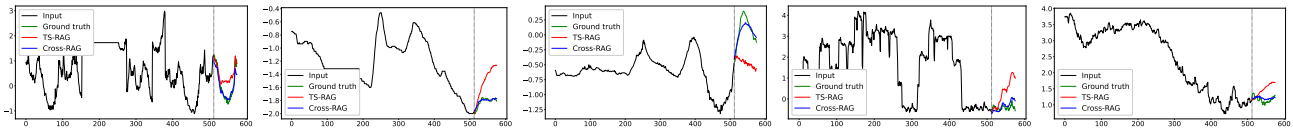


Figure 2: Performance by k. Selective attention enables Cross-RAG to benefit from larger k, even when irrelevant samples are included.

Method		1. How to retrieve?	2. What to retrieve?	3. How to fuse?
Full-shot	RAFT (ICML 2025)	Q vs. R_x	R_y	Concatenate
	RATD (NeurIPS 2024)	Q vs. $h(R_x)$	(R_x, R_y)	Cross-Attention[1]
	TimeRAG (ICASSP 2025)	Q vs. R_x	R_x	✗[2]
Zero-shot	RAF (arXiv 2025)	Q vs. $h(R_x)$	(R_x, R_y)	✗[3]
	TS-RAG (NeurIPS 2025)	Q vs. $h(R_x)$	R_y	Self-Attention
	Cross-RAG (Ours)	Q vs. R_x	(R_x, R_y)	Cross-Attention

[1] The cross-attention mechanism in **RATD** is fundamentally different from ours, as it attends over retrieved samples using both (R_x, R_y) as keys and dataset-level auxiliary statistics as values, rather than modeling query–retrieval interactions.
[2] **TimeRAG** converts retrieved input sequences into text prompts and feeds them into an LLM backbone.
[3] **RAF** forms a single sequence by prepending retrieved samples to the query without modeling interaction between them.

Table 1: RAG in TS forecasting. (1) *How to retrieve?* Defines the similarity space for retrieval, depending on retrieval encoder $h(\cdot)$. (2) *What to retrieve?* Specifies the retrieved content R. (3) *How to fuse?* Describes how the retrieved info is integrated with query Q.



(a) ETTm1 **(b) ETTm2** **(c) Weather** **(d) Electricity** **(e) Exchange**

Figure 3: Visualization of zero-shot TS forecasting results. The figure shows the forecasting results across five datasets for models with cross-attention (Ours) and without cross-attention (TS-RAG), demonstrating that Cross-RAG follows the ground truth more closely than TS-RAG across diverse domains.

the query itself, 2) the retrieved samples themselves, and 3) the relational information between the query and retrieved samples.

As shown in Figure 2, Cross-RAG consistently improves performance (average MSE across seven datasets) as the number of retrieved samples (k) increases by selectively leveraging relevant samples even when irrelevant ones may be included, whereas without cross-attention [21], performance does not improve monotonically with increasing k, requiring careful selection of k. Figure 3 visualizes the zero-shot forecasting results, showing that incorporating cross-attention more closely follows the ground truth than the variant without cross-attention [21] across diverse datasets.

The main contributions are as follows:

- We propose **Cross-RAG**, a plug-in method that enhances retrieval-augmented forecasting by *selectively* attending to retrieved samples according to their relevance to the query, using *cross-attention* between the query and the inputs of the retrieved samples.
- To the best of our knowledge, Cross-RAG is the first zero-shot RAG approach to model relationships between the query and retrieved inputs, as shown in Table 1.
- We validate Cross-RAG through extensive experiments, achieving state-of-the-art (SoTA) performance across diverse datasets and backbones. The proposed method supports zero-shot forecasting, enabling efficient integration without requiring additional retraining of target datasets.
- We conduct in-depth analyses under various retrieval conditions, demonstrating its ability to selectively attend to relevant samples while remaining robust to irrelevant samples.

2 Related Works

Time series foundation models (TSFMs). TSFMs have emerged as a scalable alternative to LLM-based forecasters, which incur high computational costs and domain adaptation challenges [3, 22, 27, 33, 36]. Early works introduce large-scale pretraining for TS forecasting [8, 24], followed by Transformer-based extensions [18, 19]. Recent models commonly adopt patch-based tokenization to embed continuous TS, where TimesFM [6] and MOMENT [9] model patch-level representations to enable forecasting across diverse datasets, while TTM [7] improves efficiency through compact architectures with reduced parameterization. Time-MoE [26] further enhances scalability by adopting a sparse mixture-of-experts design. Beyond deterministic forecasting, Moirai [30] learns mixtures of predictive distributions to capture uncertainty. Chronos [2] formulates forecasting as a language modeling task by discretizing scaled TS into categorical tokens, and Chronos-Bolt replaces discrete tokenization with patch-based inputs. Despite strong generalization, these models rely on internal representations and lack mechanisms to dynamically incorporate external knowledge, which limits interpretability and zero-shot adaptability [21].

RAG in TS. RAG [15] provides a principled way to incorporate external knowledge and improve robustness, and recent studies extend this paradigm to TS forecasting. ReTime [13] introduces relational retrieval with content synthesis, RATD [16] leverages retrieved historical series to guide diffusion-based denoising, and RAFT [10] combines retrieval with multi-resolution forecasting. TimeRAG [34] incorporates retrieved sequences through a frozen

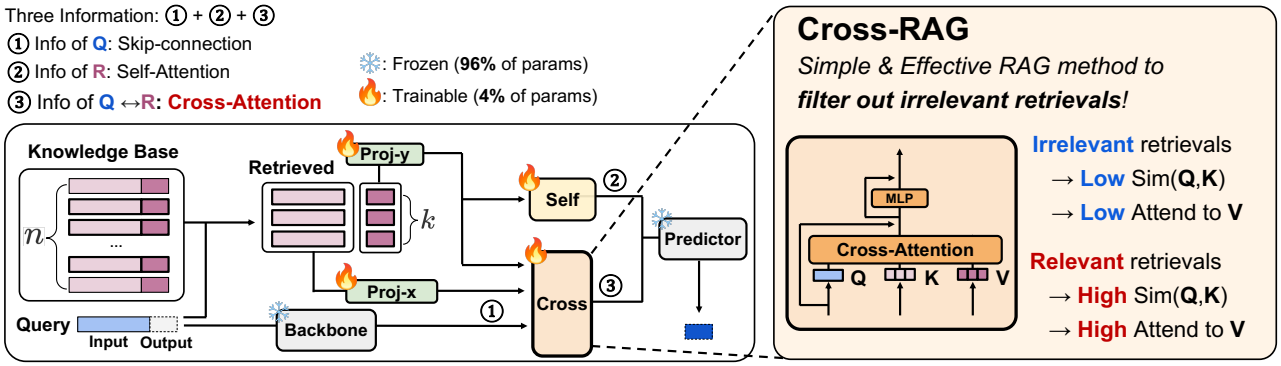


Figure 4: Framework of Cross-RAG. Cross-RAG models *query–retrieval relevance* through *cross-attention*, which selectively aggregates retrieved outputs conditioned on their relevance to the query. Additionally, retrieval self-attention is employed to provide (query-independent) contextual information among retrieved samples. The backbone and predictor are frozen.

LLM backbone with a trainable reprogramming layer to align modalities. However, these approaches are limited in that they *require fine-tuning on the target set* for adaptation to unseen datasets.

Recently, zero-shot[1] RAG methods have also been introduced for TS forecasting. RAF [29] constructs an augmented input by concatenating retrieved sequences with the query sequence in the raw data space. TS-RAG [21] retrieves TS segments using a pretrained encoder and conducts self-attention with concatenated query and retrieved outputs. However, these approaches *do not explicitly account for the relationship between the query and the retrieved inputs* when aggregating retrieved samples.

3 Methodology

TS forecasting with RAG. We consider the task of TS forecasting with retrieval augmentation. Let $\mathcal{D} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{n}$ denote a retrieval knowledge base, where $\boldsymbol{x}_j \in \mathbb{R}^T$ is a input window of length T and $\boldsymbol{y}_j \in \mathbb{R}^L$ is the corresponding forecasting horizon of length L. Given a query input $\boldsymbol{x}_i$, our goal is to predict $\hat{\boldsymbol{y}}_i$ by leveraging both the query input and a set of retrieved samples from $\mathcal{D}$.

In this section, we organize our retrieval-augmented forecasting framework into three key aspects:

- Section 3.1 (***How to Retrieve***) covers the choice of space and metric for selecting relevant samples.
- Section 3.2 (***What to Retrieve***) specifies which parts of the retrieved samples (X or Y) are used.
- Section 3.3 (***How to Fuse***) describes how query and retrieved information are aggregated to produce a final representation.

The overall framework of our method is shown in Figure 4.

3.1 How to Retrieve

A key design choice in RAG for TS forecasting is the retrieval space, where we analyze retrieval performed in the *data space* and the *latent space*. Table 2 reports the average MSE across seven datasets using four different similarity metrics, showing that our method remains robust across retrieval spaces and metrics. This result contrasts with prior studies [16, 21], which report inferior performance with data-space retrieval. In contrast, our method achieves comparable performance in both spaces, while data-space retrieval avoids the need for an additional retrieval encoder and thus offers a more efficient alternative, which is further discussed in Section 5.3. Based on this observation, we adopt data-space retrieval by selecting the top-k samples from $\mathcal{D}$ that are most similar to $\boldsymbol{x}_i$ under cosine similarity in the input space, denoted as $\mathcal{R}_i$.

[1]Zero-shot in RAG refers to settings where no data from the *target task dataset* is used for parameter updates, and models are pretrained on *task-independent TS corpora*.

Table 2: *1) How to retrieve?* Robustness to similarity space and retrieval metric (Avg. MSE across seven datasets).

Retrieval space & Metric		Retrieval encoder	Avg. MSE
Data	Cosine	✗	0.191
	Euclidean		0.193
	Correlation		0.194
Latent		✓	0.193

3.2 What to Retrieve

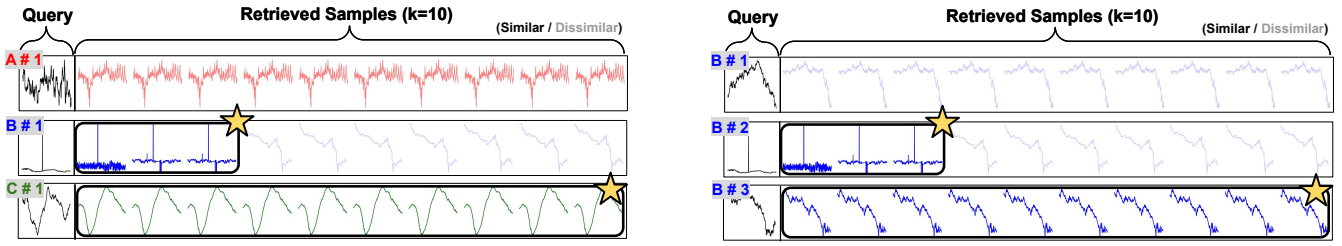
Selecting a fixed k samples can be problematic, as optimal k depends on how many samples in the knowledge base are similar to the query, and a fixed k may include *irrelevant ones* when this number varies. Figure 3 illustrates this issue, where Cross-RAG follows the ground truth more closely than the previous method [21].

As shown in Figure 5a, the optimal k differs *across* datasets. Moreover, Figure 5b shows that even *within* the same dataset, the number of samples similar to a given query varies across queries, highlighting the necessity to *selectively attend to relevant samples* among the retrieved ones.

To address this issue, we fuse information between the query and the retrieved samples by explicitly modeling the relationship between the query and the retrieved inputs. Each retrieved element in $\mathcal{R}_i$ is represented as an input–output pair $(\boldsymbol{x}_j, \boldsymbol{y}_j)$, which are mapped into d-dimensional latent spaces via MLP-based projectors. Formally, we compute

$$\boldsymbol{r}_{x,j} = f_x(\boldsymbol{x}_j), \qquad \boldsymbol{r}_{y,j} = f_y(\boldsymbol{y}_j) \tag{1}$$

for each retrieved pair $(\boldsymbol{x}_j, \boldsymbol{y}_j) \in \mathcal{R}_i$, where $f_x(\cdot)$ and $f_y(\cdot)$ denote the input and output projectors.



(a) Optimal k varies across datasets.

(b) Optimal k varies within a dataset.

Figure 5: Necessity of selective attention. The figure shows that the optimal k varies both *across* and *within* datasets ([A] ETTh1, [B] Exchange, [C] Weather), highlighting the necessity of selectively attending to samples among the retrieved samples.

3.3 How to Fuse

Given a query $\boldsymbol{x}$ and its retrieved set $\mathcal{R} = \{(\boldsymbol{r}_{x,j}, \boldsymbol{r}_{y,j})\}_{j=1}^{k}$, we integrate query and retrieved information by explicitly modeling their input-level relevance. For simplicity, we omit the query index in this section. Let $\boldsymbol{h} \in \mathbb{R}^d$ denote the query representation produced by TSFM backbone, and let $\boldsymbol{R}_x \in \mathbb{R}^{k\times d}$ and $\boldsymbol{R}_y \in \mathbb{R}^{k\times d}$ denote the stacked representations of retrieved inputs and outputs.

(Query–Retrieval) Cross-Attention. The proposed method applies cross-attention to selectively incorporate retrieval information relevant to the query, using the query representation as the *query* and the retrieved inputs and outputs as the *keys* and *values* of attention-mechanism, respectively:

$$\boldsymbol{c} = \mathrm{Attn}\big(\boldsymbol{h},\ \boldsymbol{R}_x,\ \boldsymbol{R}_y\big) + \boldsymbol{h}. \tag{2}$$

Here, $\mathrm{Attn}(\cdot)$ denotes a multi-head attention block, and the residual connection preserves the information of the query itself. The attended representation is further processed by a feed-forward network with a residual connection:

$$\tilde{\boldsymbol{c}} = \mathrm{FFN}_{\mathrm{cross}}(\boldsymbol{c}) + \boldsymbol{c}. \tag{3}$$

This operation selectively extracts information from the retrieved outputs by weighting them according to the relevance of their corresponding inputs to the query.

(Retrieval) Self-Attention. To summarize the retrieved samples independently of the query, Cross-RAG applies self-attention over their outputs:

$$\boldsymbol{s} = \mathrm{Attn}\big(\boldsymbol{R}_y,\ \boldsymbol{R}_y,\ \boldsymbol{R}_y\big) + \boldsymbol{R}_y, \tag{4}$$

followed by a feed-forward network and aggregation:

$$\tilde{\boldsymbol{s}} = \mathrm{Pool}(\mathrm{FFN}_{\mathrm{self}}(\boldsymbol{s}) + \boldsymbol{s}), \tag{5}$$

where $\mathrm{Pool}(\cdot)$ denotes mean pooling over the retrieved samples. This operation captures retrieval-driven information by aggregating the outputs of the top-k retrieved samples.

Finally, we combine the two sources of information as:

$$\boldsymbol{z} = \lambda\tilde{\boldsymbol{c}} + (1-\lambda)\tilde{\boldsymbol{s}}, \tag{6}$$

where $\lambda \in [0, 1]$ is a fixed hyperparameter controlling the relative contribution of each source of information. The fused representation $\boldsymbol{z}$, which encodes (i) the query, (ii) the retrieved samples, and (iii) their relational information, is passed to the forecasting head to produce the prediction $\hat{\boldsymbol{y}}$.

4 Experiments

4.1 Experimental Setup

Pretraining datasets. For pretraining, we use the Chronos pretraining dataset [2], from which 50M data points are uniformly sampled, following the setup of prior work [21]. A subset of 5M data points is further selected to construct a retrieval knowledge base. Both the pretraining dataset and the retrieval knowledge base are segmented using a fixed input window, resulting in 26M pretraining pairs and 2.8M retrieval pairs. Specifically, the 50M data points are sampled from the full Chronos pretraining corpus, which includes both real-world and synthetic datasets, and 5M of them are randomly selected to construct the retrieval knowledge base. The training splits of downstream evaluation datasets are excluded from the retrieval knowledge base to ensure fair zero-shot evaluation.

Evaluation datasets. Zero-shot evaluation is conducted on widely used time series benchmarks spanning diverse domains, four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) [35], Weather, Electricity, and Exchange [31]. We maintain chronological order when separating the training, validation, and test sets, with split ratios of 6:2:2 for the ETT datasets and 7:1:2 for the remaining datasets. Details of datasets are provided in Appendix A.1.

Baseline methods. We compare our method with several TSFMs, including Chronos, Chronos-Bolt [2], MOMENT [9], TTM [7], Moirai [30], TimesFM [6], and Time-MoE [26]. Details of baselines are provided in Appendix B.

Experimental settings. To remain consistent with standard settings, we fix the input length and forecasting horizon to 512 and 64, respectively, for both pretraining and evaluation, and use Chronos-Bolt as the TSFM backbone. For the evaluation metrics, we use mean squared error (MSE) and mean absolute error (MAE), and set $k = 15$ and $\lambda = 0.7$. Note that we use $k = 5$ for TS-RAG, which yields the best performance among values in $[1, 15]$. Details are reported in Appendix A.2.

4.2 Zero-shot Forecasting

We validate the effectiveness of Cross-RAG from three perspectives:

- (1) Comparison with TSFMs
- (2) Comparison with RAG methods
- (3) Application to various TSFMs

The baseline results are obtained both by reproducing experiments using the official implementations and by taking the reported results from the original paper [21].

Table 3: Results of zero-shot forecasting across TSFMs. The proposed method outperforms existing TSFMs across diverse datasets. The 1st and 2nd results are indicated by bold and underline, respectively.

Methods	Cross-RAG (Ours)		Chronos-bolt (TMLR 2024)		MOMENT (ICML 2024)		TTM (NeurIPS 2024)		Moirai (ICML 2024)		TimesFM (ICML 2024)		Chronos (TMLR 2024)		Time-MoE (ICLR 2025)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.341**	0.367	0.362	**0.365**	0.392	0.411	0.362	0.371	0.369	0.384	0.425	0.383	0.422	0.381	0.362	0.367
ETTh2	**0.243**	**0.299**	0.252	**0.299**	0.274	0.333	0.253	0.303	0.255	0.305	0.289	0.323	0.266	0.314	0.252	0.322
ETTm1	**0.290**	**0.319**	0.311	**0.319**	0.351	0.383	0.315	0.325	0.540	0.432	0.332	0.333	0.394	0.370	0.321	0.334
ETTm2	**0.143**	**0.224**	0.149	**0.224**	0.170	0.258	0.151	0.241	0.196	0.269	0.170	0.255	0.166	0.252	0.157	0.254
Weather	**0.144**	**0.178**	0.153	0.183	0.180	0.238	0.154	0.189	0.171	0.191	—	—	0.190	0.211	0.149	0.184
Electricity	**0.112**	**0.200**	0.113	**0.200**	0.197	0.303	0.172	0.264	0.183	0.281	—	—	0.146	0.224	0.114	0.203
Exchange	**0.064**	0.175	0.067	0.178	0.098	0.206	0.066	**0.173**	0.066	0.172	0.070	0.180	0.083	0.188	0.085	0.206
Average	**0.191**	**0.252**	0.201	0.253	0.237	0.304	0.210	0.267	0.254	0.291	—	—	0.238	0.277	0.206	0.267

Table 4: Results of zero-shot forecasting across TS RAG methods. The proposed method outperforms existing RAG methods.

Method	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Exchange	Average
RAF (arXiv 2025)	0.366	0.252	0.306	0.148	0.178	0.119	**0.063**	0.205
TS-RAG (NeurIPS 2025)	0.357	0.246	0.298	0.147	0.151	**0.112**	0.071	0.197
Cross-RAG (Ours)	**0.341**	**0.243**	**0.290**	**0.143**	**0.144**	**0.112**	0.064	**0.191**

Table 5: Application to various TSFM backbones. Cross-RAG generalizes across both Chronos-Bolt and MOMENT, consistently outperforming the standalone backbones and TS-RAG.

Backbone	Method	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Exchange	Average
Chronos-Bolt	(standalone)	0.362	0.252	0.311	0.149	0.153	0.113	0.067	0.201
	+ TS-RAG	0.357	0.246	0.298	0.147	0.151	**0.112**	0.071	0.197
	+ Cross-RAG	**0.341**	**0.243**	**0.290**	**0.143**	**0.144**	**0.112**	**0.064**	**0.191**
MOMENT	(standalone)	0.392	0.274	**0.351**	0.170	0.180	0.197	0.098	0.237
	+ TS-RAG	0.370	0.272	0.366	0.171	0.164	**0.154**	0.095	0.227
	+ Cross-RAG	**0.367**	**0.269**	0.358	**0.166**	**0.157**	0.159	**0.089**	**0.224**

[1] Comparison with TSFMs. We compare our method with various TSFMs under the zero-shot forecasting setting. All models are evaluated without any task-specific fine-tuning, following the standard protocol of zero-shot forecasting in RAG. Table 3 presents the results, demonstrating the effectiveness of our method in incorporating retrieval information for zero-shot forecasting and showing consistent improvements over TSFMs that rely solely on pretrained representations.

[2] Comparison with RAG methods. We compare our method with RAG-basedshot forecasting across approaches under the same zero-shot forecasting setting. Table 4 shows the results, highlighting the effectiveness of selectively attending to query-rshot forecasting acrosselevant retrieved samples. As visualized in Figure 3, Cross-RAG follows the ground truth more closely than TS-RAG. Additional visualizations are provided in Section M.

[3] Application to various TSFMs. To verify that Cross-RAG is backbone-agnostic, we evaluate it with two TSFM backbones: Chronos-Bolt [2] and MOMENT [9]. Table 5 presents the results, comparing each backbone without RAG, with TS-RAG, and with Cross-RAG. Cross-RAG achieves the best average MSE across both backbones (0.191 for Chronos-Bolt and 0.224 for MOMENT), consistently outperforming both baselines across most datasets. These results confirm that the proposed cross-attention mechanism generalizes beyond a single backbone and is effective across different TSFMs.

4.3 Ablation Studies

We conduct ablation studies to evaluate the contributions of Cross-RAG's components: query skip-connection (Q), retrieval self-attention (R), and query–retrieval cross-attention ($Q \leftrightarrow R$). Table 6 shows the results, demonstrating that using only the query representation (Q) yields limited performance, while adding either retrieval self-attention (R) or query–retrieval cross-attention ($Q \leftrightarrow R$) consistently improves results. Combining all three components achieves the best performance across all datasets, highlighting the importance of jointly modeling the query, retrieved samples, and their input-level interactions. Additional ablation combinations are reported in Section D.

Table 6: Ablation on fusion components. Q denotes the *skip-connection* from the query representation, R denotes retrieval *self-attention* that summarizes retrieved samples, and $Q \leftrightarrow R$ denotes query–retrieval *cross-attention*.

Q	R	$Q \leftrightarrow R$	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Exchange	Average
✓			0.423	0.321	0.359	0.204	0.195	0.184	0.172	0.265
✓	✓		0.361	0.249	0.308	0.148	0.151	0.113	0.067	0.200
✓		✓	0.360	0.246	0.300	0.148	0.147	0.113	**0.066**	0.197
✓	✓	✓	**0.341**	**0.243**	**0.290**	**0.143**	**0.144**	**0.112**	**0.064**	**0.191**

Table 7: (R1) MSE/Rank under various ks. Cross-attention enables Cross-RAG to benefit from larger k.

k	2	4	6	8	10	12	14
TS-RAG	.1975 4	.2011 6	.1968 2	.1975 4	.1967 1	.1974 3	.2013 7
Cross-RAG	**.1944** 7	**.1938** 6	**.1936** 5	**.1934** 4	**.1933** 3	**.1930** 2	**.1927** 1

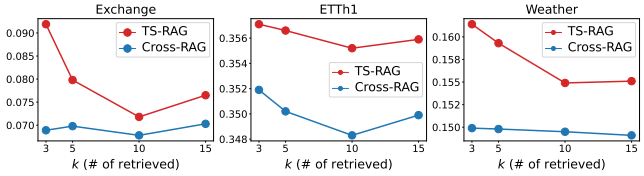


Figure 6: (R2) MSE under random retrieval. Selective attention enables Cross-RAG to filter out irrelevant samples even under random retrieval.

Table 8: (R3) MSE under smaller knowledge base. Cross-RAG remains effective when only a fraction of the training data is available for retrieval.

Train period (%)				Methods	
Q1	Q2	Q3	Q4	TS-RAG	Cross-RAG
✓	✓	✓	✓	.1470	**.1435**
✗	✓	✓	✓	.1485	**.1463**
✗	✗	✓	✓	.1486	**.1458**
✗	✗	✗	✓	.1489	**.1467**

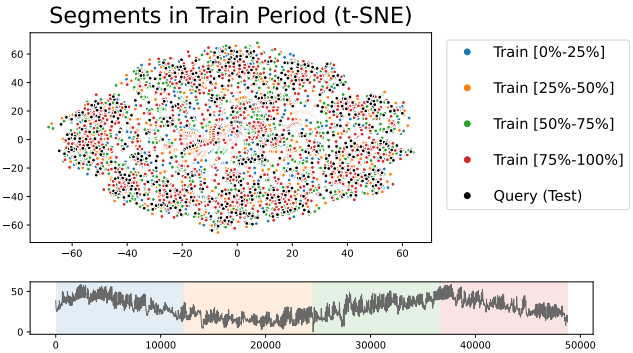


Figure 7: (R3) Visualization of TS segments. The figure illustrates that historical TS segments contain redundant patterns, supporting Cross-RAG's robustness under smaller KBs.

5 Analysis

5.1 Various Retrieval Scenarios

To validate the effectiveness of our method under various retrieval scenarios, we evaluate it across four different settings as follows:

- (R1) Retrieval with **various numbers of samples (k)**
- (R2) Retrieval of **random samples**
- (R3) Retrieval using a **smaller knowledge base**
- (R4) Retrieval from a **different dataset**

(R1) Retrieval with various ks. Table 7 and Figure 2 show the average MSE over seven datasets for various numbers of retrieved samples (k). Unlike prior work [21], Cross-RAG consistently improves as k increases. These results indicate that simply increasing the number of retrieved samples does not improve performance unless relevance is considered, and that Cross-RAG effectively leverages additional samples by focusing on relevant ones. Full results for all datasets are reported in Section E.

(R2) Retrieval of random samples. To evaluate Cross-RAG's ability to selectively attend to relevant retrieved samples, we conduct an experiment using *random* samples from the knowledge base, rather than selecting them based on similarity. Figure 6 presents the MSE over three datasets, showing that Cross-RAG consistently outperforms the SoTA method [21] for all values of k.

(R3) Retrieval using a smaller knowledge base. To evaluate the robustness of Cross-RAG under limited knowledge, we construct knowledge bases (KBs) using *different fractions of the training dataset* using ETTm2, as shown in Figure 7. Table 8 demonstrates that Cross-RAG consistently outperforms TS-RAG by effectively leveraging the available information even when the KB is small.

Furthermore, we attribute this to the high redundancy of similar segments in historical data, as illustrated in Figure 7. The figure shows that as recent segments cover a large portion of the relevant patterns, even a reduced KB provides sufficient context for accurate retrieval, allowing Cross-RAG to maintain strong performance. We further conduct a sensitivity analysis by varying the KB size from 5% to 100% across all datasets, reported in Appendix H. The results confirm that Cross-RAG degrades gracefully even when the KB is reduced to 5% of its original size.

Table 9: (R4) Retrieval from a different dataset. Selective attention enables Cross-RAG to transfer useful signals across datasets, supporting retrieval-augmented forecasting even without target-domain KBs.

KB → Target (KB: Knowledge Base)		Target			
		ETTh1	ETTh2	ETTm1	ETTm2
KB	ETTh1	**0.341**	0.246	0.325	0.149
	ETTh2	0.351	**0.243**	0.318	0.148
	ETTm1	0.370	0.250	**0.290**	0.146
	ETTm2	0.363	0.252	0.304	**0.143**
w/o Cross-RAG		0.362	0.252	0.311	0.149

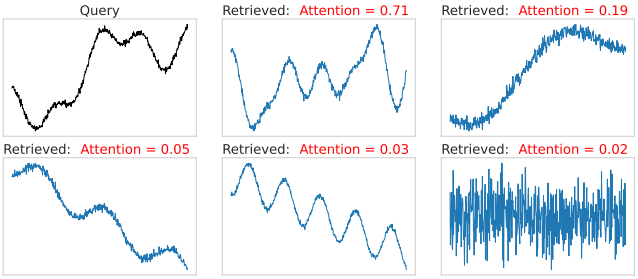


Figure 8: Attention of query and retrieved samples. Cross-attention assign higher attention to more relevant samples.

(R4) Retrieval from a different dataset. We further evaluate Cross-RAG in a setting where *no knowledge base is available from the target dataset* by constructing the knowledge base from *different* datasets. Specifically, we treat one of the four ETT datasets as the target and use the remaining three as the knowledge base. Table 9 reports the results, where Cross-RAG consistently outperforms the strong baseline (Chronos-Bolt) in most cases. In addition, stronger performance is observed when the knowledge base comes from the same or a similar domain (e.g., ETT hourly or minutely), indicating more effective information transfer across aligned datasets. An extended transfer analysis with additional target datasets (Weather, Electricity, Exchange) is provided in Appendix K.

5.2 Other Analyses

Performance across various T, Ls. Table 10 reports the average MSE across various input windows (T) and forecast horizons (L). We fix $L = 64$ when varying the input window size (T), and fix $T = 512$ when varying the forecast horizon (L). The proposed method consistently outperforms TS-RAG across all settings, with improvements pronounced for smaller input windows and longer forecast horizons. We further extend the evaluation to longer L with a fixed T=512, as reported in Appendix J. Cross-RAG maintains consistent improvements across all extended horizons.

Attention weights across retrieved samples. To examine how our proposed method attends to each retrieved sample for a given query, we construct a toy dataset and manually compose the retrieved set to include both relevant and irrelevant samples for each query. Figure 8 visualizes the results along with the (cross-) attention weights, showing that samples similar to the query receive higher weights, while irrelevant or noisy samples receive lower weights. More examples are visualized in Section L.

Table 10: Average MSE across various T,L. Cross-RAG consistently improves over TS-RAG across all Ts and Ls.

(1) **Input window** (T) with L=64				
T	64	128	256	512
TS-RAG	0.463	0.376	0.217	0.197
Cross-RAG	**0.258**	**0.246**	**0.196**	**0.191**
Improvement (%)	**+44.3**	**+34.6**	**+9.7**	**+3.0**
(2) **Forecast horizon** (L) with T=512				
L	24	36	48	64
TS-RAG	0.140	0.164	0.181	0.197
Cross-RAG	**0.138**	**0.163**	**0.178**	**0.191**
Improvement (%)	**+1.4**	**+0.6**	**+1.7**	**+3.0**

Table 11: Robustness to similarity metrics. Performance remains stable across metrics.

	Data space			Latent
	Cosine	Euclidean	Correlation	
ETTh1	0.341	0.343	0.348	0.347
ETTh2	0.243	0.248	0.245	0.244
ETTm1	0.290	0.294	0.297	0.294
ETTm2	0.143	0.143	0.144	0.144
Weather	0.144	0.146	0.146	0.145
Electricity	0.112	0.114	0.115	0.114
Exchange	0.064	0.067	0.069	0.067
Average	**0.191**	0.193	0.194	0.193

Robustness to similarity metrics. To examine the robustness to the choice of similarity metric used for retrieval, we conduct experiments under various similarity metrics across seven datasets. Specifically, we consider cosine similarity, Euclidean distance, correlation, and a latent-space metric (Euclidean). Table 11 reports the results, demonstrating that the performance remains stable across similarity metrics, suggesting that retrieval in the data space can be conducted without introducing an additional retrieval encoder.

Robustness to λ. To evaluate the robustness to λ, which controls the fusion of retrieved samples and the relational information between the query and these samples, we perform a sensitivity analysis across various λs. We also test a learnable version of this gating mechanism, where two types of representations are concatenated and passed through an MLP followed by a sigmoid function.

Table 12 reports the results, demonstrating that the performance is stable across λ and that the fixed gating mechanism outperforms the learnable version. We attribute this to the high variability introduced by instance-level gating: while a learnable λ can in principle

Table 12: Robustness to λ. A simple fixed λ outperforms learnable gating and remains stable across a wide range of values.

λ	Fixed				
	0.4	0.5	0.6	0.7	0.8
MSE	0.196	0.194	0.193	**0.191**	0.194
	Learnable				
MSE	0.196				

Table 13: Efficiency - Retrieval. Data-space retrieval achieves a 339× speed-up by performing retrieval directly on raw inputs, without any embedding overhead.

Space	Procedures	Time (s)
Latent	Embedding	381.6
	Search	2.1
	Total	383.7
Data	Embedding	—
	Search	1.1
	Total	1.1
Speed-up		**339**×

adapt to each query, the limited training signal per instance leads to noisy gradient estimates that prevent stable convergence. In contrast, a fixed λ acts as a regularizer that stabilizes the fusion across diverse retrieval scenarios. A detailed per-dataset analysis of the learnable gating mechanism is provided in Appendix I.

5.3 Efficiency Analyses

Efficiency analysis 1 - Retrieval. Table 13 compares retrieval in the data space and latent space using ETTh1. The results show that data-space retrieval is significantly faster, achieving a 339.5× speed-up over latent-space retrieval, as it does not require any embedding computation. Search with FAISS corresponds to the offline construction of the retrieval index, where similar historical TS across all timestamps and variables are precomputed and stored. During training and inference, only the stored index is accessed, without performing FAISS search repeatedly. Results for all datasets are discussed in Section F.

Efficiency analysis 2 - Inference. Table 14 compares inference efficiency on ETTh1 under different combinations of Cross-RAG components in terms of 1) inference time per instance (ms, averaged across 1000 runs) and 2) FLOPs. The inference cost remains nearly unchanged when incorporating the proposed components, demonstrating minimal computational overhead.

Efficiency analysis 3 - Parameters. As shown in Table 15, the proposed modules (e.g., attention and projector) account for approximately 4% of the total parameters, with all other modules frozen, ensuring high parameter efficiency. These learnable parameters are pretrained *only on the pretraining datasets* and remain fixed across all target datasets, consistent with the zero-shot setting.

Table 14: Efficiency - Inference. Cross-RAG incurs negligible inference overhead, demonstrating that input-aware fusion can be achieved without sacrificing efficiency.

Q	R	$Q \leftrightarrow R$	Time (ms)	FLOPS
✓			8.90	751.3M
✓	✓		9.69	755.5M
✓		✓	9.68	751.8M
✓	✓	✓	9.73	756.5M
TS-RAG			9.91	754.2M

Table 15: Efficiency - Parameters. With only ~4% trainable parameters, Cross-RAG operates as a lightweight plug-in over the frozen TSFM backbone.

Components			# Params	(%)
Patch Embedding		Input / Output	7.1M	3.3
Backbone		Enc. / Dec.	198M	92.6
+Ours	Attention	Self	3.5M	**1.6**
		Cross	3.5M	**1.6**
	Projector	Input	1.0M	**0.5**
		Output	0.6M	**0.3**

6 Conclusion

In this work, we introduce Cross-RAG, a zero-shot retrieval-augmented framework for TS forecasting that selectively attends to query-relevant samples and models their input-level interactions via cross-attention. Our results show consistent improvements in zero-shot settings, demonstrating effective and efficient selective retrieval that is robust to retrieval settings, distance metrics, and the choice of top-k. We further validate generalizability across TSFM backbones (e.g., Chronos-Bolt and MOMENT) and provide analyses on knowledge base sensitivity and extended forecast horizons.

Limitation and future works. While Cross-RAG relies on input-level relationships for selective retrieval, it does not exploit auxiliary information such as metadata. Furthermore, following the common evaluation protocols used in prior RAG-based time series forecasting methods, our evaluation focuses on conventional forecasting datasets, and extending to large-scale benchmarks (e.g., GIFT-Eval [1]) would further strengthen the findings. Future work includes adaptive retrieval and interaction mechanisms that incorporate such auxiliary information across datasets. We also provide an empirical analysis showing that learnable gating introduces instability due to noisy per-instance gradients, and a deeper theoretical understanding of the optimal fusion dynamics remains an open problem. Finally, as Cross-RAG is a backbone-agnostic framework, we hope it will naturally extend to future TSFMs, as it relies on input-level interactions without requiring architecture-specific modifications.

References

[1] Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. 2024. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393* (2024).

[2] 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. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815* (2024).

[3] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. 2023. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *CoRR* (2023).

[4] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. 2023. Tsmixer: An all-mlp architecture for time series forecasting. *TMLR* (2023).

[5] Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. 2022. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2900–2913.

[6] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. *ICML* (2024).

[7] Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam 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. *NeurIPS* 37 (2024), 74147–74181.

[8] Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. 2023. TimeGPT-1. *arXiv preprint arXiv:2310.03589* (2023).

[9] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. Moment: A family of open time-series foundation models. In *ICML*.

[10] Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. 2025. Retrieval augmented time series forecasting. *ICML* (2025).

[11] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. 2023. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems* 36 (2023), 46885–46902.

[12] Yue Jiang, Yile Chen, Xiucheng Li, Qin Chao, Shuai Liu, and Gao Cong. 2025. FSTLLM: Spatio-Temporal LLM for Few Shot Time Series Forecasting. In *ICML*.

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

[14] Seunghan Lee, Taeyoung Park, and Kibok Lee. 2026. Dataset-Driven Channel Masks in Transformers for Multivariate Time Series. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2401–2405.

[15] 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. *NeurIPS* 33 (2020), 9459–9474.

[16] Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. 2024. Retrieval-augmented diffusion models for time series forecasting. *NeurIPS* 37 (2024), 2766–2786.

[17] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*.

[18] Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2025. Sundial: A family of highly capable time series foundation models. *ICML* (2025).

[19] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024. Timer: Generative Pre-trained Transformers Are Large Time Series Models. In *ICML*.

[20] Donghao Luo and Xue Wang. 2024. Moderntcn: A modern pure convolution structure for general time series analysis. In *The twelfth international conference on learning representations*. 1–43.

[21] Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Yiming Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. 2025. Ts-rag: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. *NeurIPS* (2025).

[22] Wenzhe Niu, Zongxia Xie, Yanru Sun, Wei He, Man Xu, and Chao Hao. 2025. LangTime: A Language-Guided Unified Model for Time Series Forecasting with Proximal Policy Optimization. *ICML* (2025).

[23] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[24] 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*.

[25] Hadi Rezaei, Hamidreza Faaljou, and Gholamreza Mansourfar. 2021. Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications* 169 (2021), 114332.

[26] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. 2025. Time-moe: Billion-scale time series foundation models with mixture of experts. *ICLR* (2025).

[27] Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. 2024. Test: Text prototype aligned embedding to activate llm's ability for time series. *ICLR* (2024).

[28] Mingtian Tan, Mike Merrill, Vinayak Gupta, Tim Althoff, and Tom Hartvigsen. 2024. Are language models actually useful for time series forecasting? *NeurIPS* 37 (2024), 60162–60191.

[29] Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. 2024. Retrieval augmented time series forecasting. *arXiv preprint arXiv:2411.08249* (2024).

[30] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified training of universal time series forecasting transformers. In *ICML*.

[31] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*.

[32] Zhen Xu, Xiqing Liu, Qingqing Xu, Xin Su, Xiaojun Guo, and Yixian Wang. 2025. Time series forecasting with attention-augmented recurrent networks: A financial market application. In *Proceedings of the 2025 2nd International Conference on Computer and Multimedia Technology*. 155–159.

[33] Hao Xue and Flora D Salim. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *TKDE* 36, 11 (2023), 6851–6864.

[34] Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. 2025. Timerag: Boosting llm time series forecasting via retrieval-augmented generation. In *ICASSP*. IEEE, 1–5.

[35] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*.

[36] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. In *NeurIPS*.

A Experimental Settings

A.1 Dataset Statistics

Inference dataset for zero-shot forecasting. We conduct experiments on seven datasets spanning diverse domains, with dataset statistics summarized in Table A.1, where C and T denote the number of channels and timesteps, respectively. The training split is used to construct the knowledge base, while the test split is used for zero-shot inference.

Table A.1: Statistics of inference dataset for zero-shot forecasting.

Dataset	C	T	$(N_{\text{train}}, N_{\text{val}}, N_{\text{test}})$
ETTh1 [35]	7	17420	(8545, 2881, 2881)
ETTh2 [35]	7	17420	(8545, 2881, 2881)
ETTm1 [35]	7	69680	(34465, 11521, 11521)
ETTm2 [35]	7	69680	(34465, 11521, 11521)
Exchange [31]	8	7588	(5120, 665, 1422)
Weather [31]	21	52696	(36792, 5271, 10540)
Electricity [31]	321	26304	(18317, 2633, 5261)

A.2 Experimental Details

We follow the experimental settings of TS-RAG [21], where the backbone TSFM remains frozen during pretraining, and only the parameters introduced by Cross-RAG, including attention modules and projectors, are updated. Chronos-Bolt [2] serves as the backbone, and we adopt its original quantile regression loss. All experiments are conducted on a single NVIDIA L40-48G GPU. Details of training hyperparameters are summarized in Table A.2.

Table A.2: Details of training hyperparameters.

Optimizer	AdamW
Learning rate	3×10^{-4}
Weight decay	0.01
Batch size	256
Training steps	10,000
Dropout rate	0.2
Retrieved sequences (k)	15

B Baseline Methods

- **Chronos** [2]: A probabilistic TSFM that tokenizes quantized TS and processes them with a T5 backbone [23].
- **Chronos-Bolt** [2]: A Chronos variant that supports patch-based modeling and generates multi-step quantile forecasts using decoder representations.
- **MOMENT** [9]: A zero-shot TSFM applies masked modeling by appending a masked forecast horizon to the lookback sequence.
- **TTM** [7]: A compact TSFM based on TSMixer [4] with adaptive patching and resolution-aware pretraining.
- **Moirai** [30]: A Transformer encoder pretrained on LOTSA via horizon masking and reconstruction across target channels.
- **TimesFM** [6]: A decoder-style attention model pretrained on large-scale real and synthetic TS.
- **Time-MoE** [26]: A decoder-only TSFM with a mixture-of-experts (MoE) architecture for autoregressive forecasting with long contexts.

C Similarity Metrics

For the experiments, we use four different similarity metrics for retrieving similar samples:

- **Cosine similarity (data space)** measures the similarity between two TS based on the angle between their vectors in the data space, capturing pattern-level similarity while being invariant to scale.
- **Euclidean distance (data space)** computes the ℓ_2 distance between raw TS, directly reflecting point-wise magnitude differences.
- **Pearson correlation (data space)** quantifies the linear relationship between two TS, providing a normalized similarity measure invariant to mean and variance.
- **Euclidean distance (latent space)** measures the ℓ_2 distance between latent representations produced by a retrieval encoder, capturing similarity in the learned representation space.

For cosine similarity and Euclidean distance in the data space, each TS is first normalized to the $[0, 1]$ range using min–max scaling before computing the similarity. As robustness is consistently observed with these similarity metrics, Dynamic Time Warping (DTW) is omitted due to its high computational complexity.

D Ablation Studies

We conduct an ablation study to assess the effect of the query skip-connection (Q), retrieval self-attention (R), and query–retrieval cross-attention ($Q \leftrightarrow R$). Table D.1 reports the full results, where combining all three components yields the best performance.

Table D.1: Full results of ablation on fusion components. Q denotes the *skip-connection* from the query representation, R denotes retrieval *self-attention* that summarizes retrieved samples.

Q	R	$Q \leftrightarrow R$	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Exchange	Average
✓			0.423	0.321	0.359	0.204	0.195	0.184	0.172	0.265
	✓		0.717	0.364	0.697	0.284	0.263	0.848	0.364	0.505
		✓	0.405	0.282	0.348	0.177	0.171	0.168	0.128	0.240
✓	✓		0.361	0.249	0.308	0.148	0.151	0.113	0.067	0.200
✓		✓	0.360	0.246	0.300	0.148	0.147	0.113	0.066	0.197
	✓	✓	0.393	0.279	0.329	0.171	0.164	0.164	0.115	0.231
✓	✓	✓	**0.341**	**0.243**	**0.290**	**0.143**	**0.144**	**0.112**	**0.064**	**0.191**

E Various Number of Retrieved Sequences (ks)

To assess zero-shot forecasting performance under different numbers of retrieved samples (k), we conduct an evaluation with varying values of k. Table E.1 reports the results, showing that the average performance across seven datasets improves as k increases.

Table E.1: Zero-shot forecasting performance (MSE) under various ks.

k	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity	Exchange	Average
1	0.3513	0.2457	0.2986	0.1441	0.1462	0.1158	0.0656	0.1953
2	0.3476	0.2464	0.2976	0.1435	0.1458	0.1145	0.0652	0.1944
3	0.3466	0.2467	0.2966	0.1433	0.1454	0.1143	0.0648	0.1940
5	0.3461	0.2468	0.2945	0.1435	0.1455	0.1141	0.0653	0.1937
8	0.3444	0.2469	0.2925	0.1436	0.1449	0.1139	0.0658	0.1934
10	0.3451	0.2467	0.2922	0.1425	0.1447	0.1142	0.0675	0.1933
12	0.3449	0.2462	0.2912	0.1434	0.1450	0.1141	0.0663	0.1930
15	0.3411	0.2427	0.2896	0.1435	0.1437	0.1123	0.0645	**0.1911**

F Efficiency Analysis

Table F.1 compares retrieval in the data space and latent space using five different datasets. Since ETTh1 and ETTh2 share the same data size, as do ETTm1 and ETTm2, we report only one dataset from each pair. The results demonstrate that data-space retrieval is significantly faster than latent-space retrieval.

Table F.1: Efficiency analysis under five different datasets.

Space	Retrieval procedures	Dataset				
		ETTh1	ETTm1	Exchange	Weather	ECL
Latent space (Z)	1) Embedding	381.6	1547.4	196.7	3530.1	27514.5
	2) Search (FAISS)	2.1	19.4	0.7	37.1	253.5
	Total	383.7	1566.8	197.4	3587.2	27768.0
Data space (X)	1) Embedding	—	—	—	—	—
	2) Search (FAISS)	1.1	16.5	0.4	22.2	129.3
	Total	1.1	16.5	0.4	22.2	129.3

G Electricity Data Leakage Analysis

The full Electricity dataset is included in the Chronos pretraining corpus, which may raise concerns about unfair evaluation. To address this, we remove all Electricity-related data from the retrieval knowledge base and re-evaluate on the remaining six datasets. Table G.1 compares the results with and without Electricity in the knowledge base. The results on all six non-Electricity datasets are nearly identical, confirming that the presence of Electricity data in the knowledge base does not inflate performance on other datasets.

Table G.1: Data leakage analysis. Results with and without Electricity in the knowledge base.

KB Setting	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Exchange
Full KB	0.341	0.244	0.290	0.144	0.144	0.065
w/o Electricity	0.341	0.243	0.290	0.143	0.144	0.064

H Knowledge Base Sensitivity Analysis

To assess the sensitivity of Cross-RAG to the size of the retrieval knowledge base, we vary the KB size from 5% to 100% of the full corpus and evaluate across all seven datasets. Table H.1 presents the results. Performance degrades gracefully as the KB size decreases: even at 5% of the original KB, the average MSE is 0.212, compared to 0.191 at 100%—representing only a ∼10% relative degradation. Notably, ETTm2 is almost unaffected across all KB sizes, while Electricity shows the most sensitivity due to its large number of channels and diverse patterns.

Table H.1: KB sensitivity analysis. MSE across varying knowledge base sizes (α% of the full KB).

α (%)	5	10	20	25	50	75	100
ETTh1	0.365	0.360	0.355	0.352	0.349	0.354	0.341
ETTh2	0.251	0.246	0.246	0.247	0.248	0.250	0.243
ETTm1	0.310	0.318	0.309	0.307	0.306	0.303	0.290
ETTm2	0.147	0.146	0.147	0.147	0.146	0.147	0.143
Weather	0.157	0.154	0.149	0.155	0.152	0.148	0.144
Electricity	0.175	0.220	0.200	0.217	0.226	0.177	0.112
Exchange	0.079	0.070	0.075	0.078	0.069	0.074	0.064
Average	0.212	0.216	0.212	0.215	0.213	0.207	**0.191**

I Learnable λ Analysis

We provide a per-dataset breakdown of the learnable vs. fixed λ comparison in Table I.1. While the learnable λ achieves competitive results on certain datasets (e.g., ETTh2: 0.243 vs. 0.246, Electricity: 0.113 vs. 0.113), it underperforms on others (e.g., ETTh1: 0.355 vs. 0.341). We attribute this to the high variance of instance-level gating gradients: since each training instance produces a different gating signal, the gradient updates for the gating network are noisy, preventing stable convergence. In contrast, a fixed λ provides consistent regularization across instances.

Table I.1: Per-dataset comparison of learnable vs. fixed λ.

λ	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Elec.	Exch.	Avg.
Learnable	0.355	**0.243**	0.295	0.147	0.147	0.113	0.065	0.195
Fixed (λ=0.7)	**0.341**	**0.243**	**0.290**	**0.143**	**0.144**	**0.112**	**0.064**	**0.191**

J Various Forecast Horizons

To evaluate Cross-RAG under extended forecast horizons, we conduct experiments with $L \in \{96, 128, 192\}$ using a fixed context length of T=512. Table J.1 reports the results. As expected, MSE increases with horizon length, but the degradation is smooth and the method maintains reasonable forecasting quality even at L=192. These results demonstrate that Cross-RAG is applicable beyond the default setting and generalizes to varying prediction requirements.

Table J.1: Extended forecast horizons. MSE with T=512 and varying prediction length L.

L	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Elec.	Exch.	Avg.
64	0.341	0.243	0.290	0.143	0.144	0.112	0.064	0.191
96	0.374	0.285	0.326	0.174	0.168	0.160	0.094	0.226
128	0.398	0.311	0.356	0.202	0.188	0.187	0.124	0.252
192	0.447	0.353	0.408	0.245	0.217	0.245	0.188	0.300

K Extended Transfer Learning Analysis

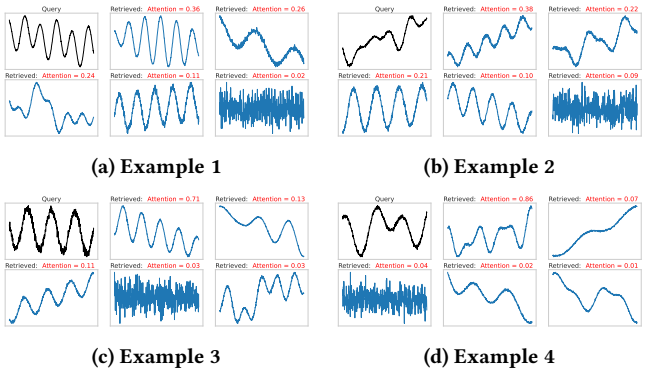
We extend the transfer learning analysis from Section 5.1 by evaluating Cross-RAG with same-domain knowledge bases for Weather, Electricity, and Exchange datasets, in addition to the ETT datasets already reported. Table K.1 presents the results. Using same-domain knowledge bases yields competitive performance across most datasets, with Weather and Electricity showing slight degradation compared to the full cross-domain KB, indicating that these datasets benefit from the diversity of the cross-domain knowledge. The ETT datasets perform nearly identically under both settings, confirming that the model generalizes well to target-specific knowledge bases.

Table K.1: Extended transfer learning. MSE with same-domain vs. full cross-domain knowledge base.

KB Setting	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Elec.	Exch.
Same-domain	0.345	0.247	0.296	0.146	0.154	0.157	0.066
Cross-domain (full KB)	0.341	0.243	0.290	0.143	0.144	0.112	0.064

L Visualization of Inputs of Query and Retrieved Samples

To analyze how Cross-RAG assigns attention to retrieved samples for a given query, we design a toy dataset containing both relevant and irrelevant retrieved sequences. Figure L.1 presents the resulting cross-attention weights, where retrieved samples that resemble the query receive higher weights, whereas unrelated or noisy samples are assigned lower weights.



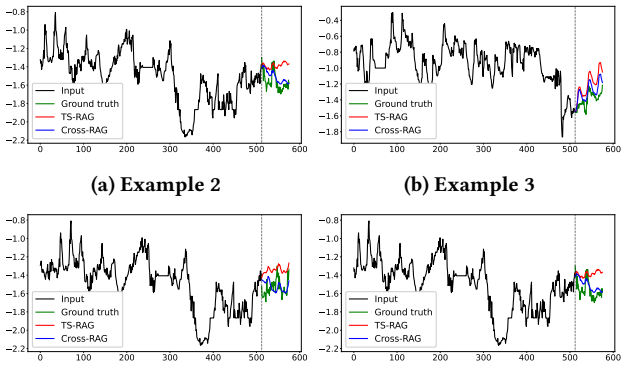
(a) Example 1 (b) Example 2

(c) Example 3 (d) Example 4

Figure L.1: Visualization of inputs of query and retrieved samples with toy datasets.

M Visualization of Zero-shot Forecasting

We visualize the predicted results on six different datasets and compare them with TS-RAG[21].



(a) Example 2 (b) Example 3

(c) Example 4

Figure M.1: Visualization of zero-shot forecasting on ETTh1.

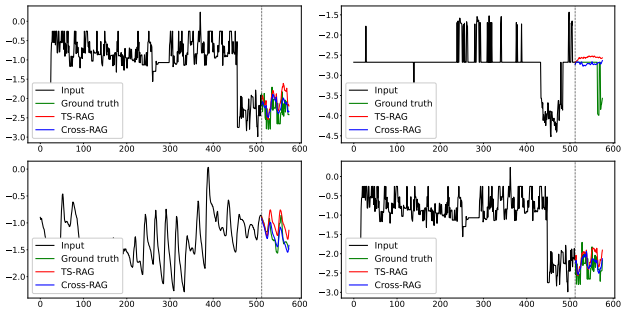


Figure M.2: Visualization of zero-shot forecasting on ETTh2.

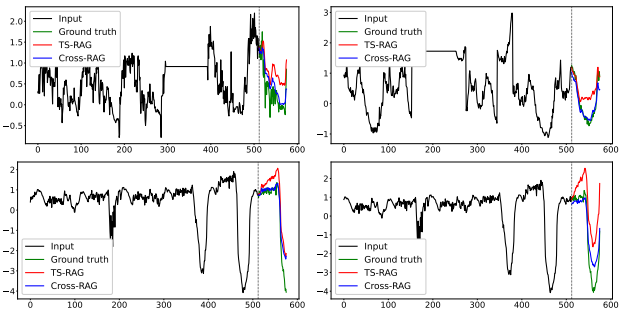


Figure M.3: Visualization of zero-shot forecasting on ETTm1.

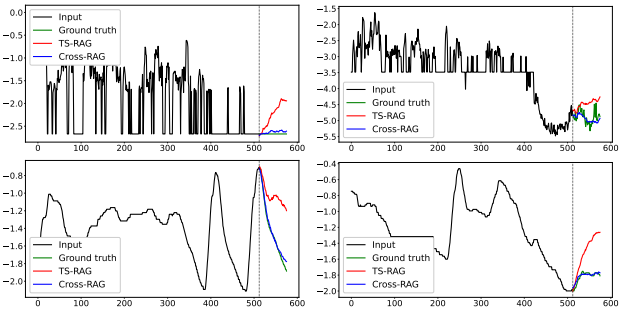


Figure M.4: Visualization of zero-shot forecasting on ETTm2.

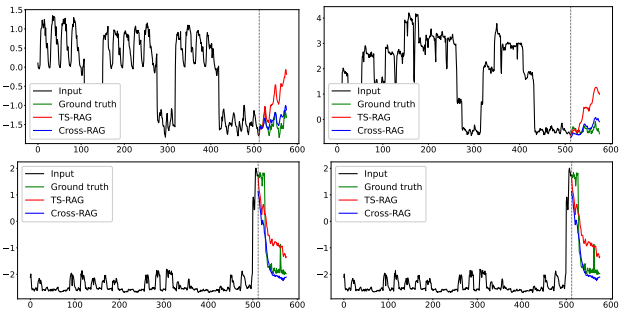


Figure M.5: Visualization of zero-shot forecasting on Electricity.

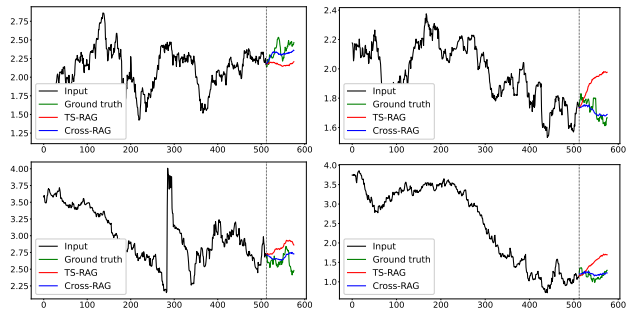


Figure M.6: Visualization of zero-shot forecasting on Exchange.

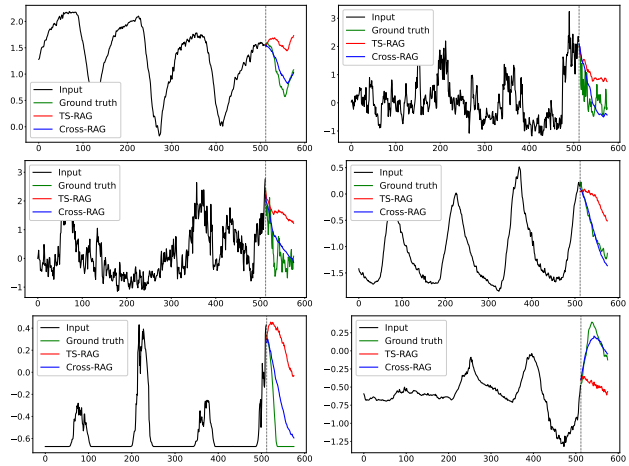


Figure M.7: Visualization of zero-shot forecasting on Weather.