Retrieval Augmented Time Series Forecasting

Kutay Tire[*,†,1] **Ege Onur Taga**[*,2] **M. Emrullah Ildiz**[2] **Samet Oymak**[2]
[1]University of Texas at Austin
[2]University of Michigan, Ann Arbor

Abstract

Retrieval-augmented generation (RAG) is a central component of modern LLM systems, particularly in scenarios where up-to-date information is crucial for accurately responding to user queries or when queries exceed the scope of the training data. The advent of time-series foundation models (TSFM), such as Chronos or Moirai, and the need for effective zero-shot forecasting performance across various time-series domains motivates the question: Do the benefits of RAG similarly carry over to time series forecasting? In this paper, we advocate that the dynamic and event-driven nature of time-series data makes RAG a crucial component of TSFMs and introduce a principled RAG framework for time-series forecasting, called *Retrieval Augmented Forecasting* (RAF). Within RAF, we develop efficient strategies for retrieving related time-series examples and incorporating them into the forecast. Through experiments and mechanistic studies, we demonstrate that RAF improves the forecasting accuracy across diverse time series domains and TSFMs, with gains that are more pronounced for larger models.

1 INTRODUCTION

The success of LLMs has motivated a broader push toward developing foundation models for other modalities. Time-series analysis, in particular, stands to directly benefit from recent advancements in sequence modeling techniques. Indeed, there has been significant progress in new time-series architectures (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Li et al., 2019; Nie et al., 2023; Liu et al., 2021; Zhang and Yan, 2023; Du et al., 2021; Rasul et al., 2021; Liu et al., 2023), tokenization strategies (Nie et al., 2023; Chen et al., 2023; Ansari et al., 2024; Talukder et al., 2025), and more recently, time-series foundation models such as Chronos (Ansari et al., 2024). These advances hold the premise of enhancing the accuracy, robustness, and few-shot learning capabilities of future time-series models. On the other hand, there is a notable shift from standalone models to compound AI systems (Nori et al., 2023; Lewis et al., 2020; Lee et al., 2019; Khattab et al., 2021) where LLMs are integrated with external databases and advanced prompting strategies to accomplish complex tasks.

In particular, retrieval augmented generation (RAG) (Lewis et al., 2020), has become a key component of LLM pipelines during recent years (Li et al., 2022). In essence, RAG aims to facilitate factual and up-to-date generation by retrieving query-related documents from external databases. Notably, RAG also mitigates the need for retraining the model to incorporate fresh data or fine-tuning it for individual application domains. In the context of time-series forecasting, we expect RAG to be beneficial for several reasons. First, time-series data is inherently dynamic, heterogeneous, and context-dependent, making it challenging to forecast accurately without access to the relevant external context such as phenomenon-specific conditions. Second, time-series often exhibit complex patterns and dependencies that traditional univariate models (ARIMA (Box et al., 2015), ETS (Hyndman et al., 2002; Hyndman and Athanasopoulos, 2018)) struggle to capture—especially under abrupt, nonlinear regime shifts (e.g., earthquakes, financial crises, elections), where assumptions of linearity and slowly varying structure break down. As a result, these models perform poorly on rare, out-of-distribution events for which historical data offer little precedent. Indeed, celebrated approaches in time series analysis, such as motif discovery, matrix profile, and dynamic time warping (Bailey et al., 2009; Yeh et al., 2016; Müller et al., 2007), are inherently about identifying and matching complex time-series patterns. Incorporating RAG holds the promise of augmenting

[*]Equal contribution. [†]Work done during an internship at the University of Michigan.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

time-series models with these capabilities to utilize external knowledge bases.

In this work, we provide a systematic study of RAG for time-series forecasting. We propose *Retrieval Augmented Forecasting* (RAF) for TSFMs (Figure 1) and observe an *intrinsic retrieval capability* in modern TSFMs: with an appropriately retrieved motif, they align and reuse it in a zero-shot manner to improve forecasts, even without fine-tuning. The benefits are most pronounced on out-of-domain datasets, where domain-specific structure is missing from the base model. For example, a traffic dataset (Wu et al., 2021) has properties unique to transportation, such as the periodicity during a day, which are quite different from those of the NN5 dataset (Godahewa et al., 2021) in finance. RAF facilitates the model to capture these properties by retrieving the best-matching motif(s) from a domain-specific database and appending it to the query context, thereby leveraging the TSFM's in-context learning capability. In doing so, it offers a resource-efficient alternative to fine-tuning. Our main contributions are as follows:

1. We introduce RAF as a principled forecasting framework for TSFMs. We formalize RAF in Section 3, where we establish that transformer-based architectures are capable of RAF and contrast the retrieval performance under various signal-to-noise ratio conditions (see Figure 2). Notably, Chronos Mini fails to provide the correct forecast even if we use the query and its true future values as the retrieved example, whereas Chronos Small and Base can do so. These indicate that retrieval-augmented forecasting is an emerging capability of large time-series foundation models.

2. We study RAF along two axes: (i) *model size*—comparing Chronos Mini vs. Chronos Base (Tables 1 and 2); and (ii) *architecture*—evaluating RAF across four TSFMs (Chronos, Moirai, TimesFM, and Lag-Llama; Table 3). These evaluations show that RAF's gains grow with model capacity, aligning with RAG results in LLMs (Guu et al., 2020; Lewis et al., 2020; Kaplan et al., 2020), and that RAF is effective across diverse TSFMs. We additionally benchmark three non-Transformer baselines—DLinear, GBDT with LightGBM, and ARIMA—and observe negligible or negative average gains with RAF (Table 4), consistent with the expectation that architectures without attention or cross-sequence fusion benefit less from retrieval at inference time.

3. Beyond capacity and architecture, we contrast two RAF modes: (i) *Naive RAF* (hereafter "RAF"), which treats the forecaster as a black box and leaves all weights frozen; and (ii) *Advanced RAF*, which additionally fine-tunes the model for retrieval-augmented forecasting. Both yield substantial gains. We evaluate probabilistic and point forecasts on Chronos, comparing both RAF variants to standard baselines with and without fine-tuning. Advanced RAF outperforms all alternatives, underscoring the effectiveness of our framework and its synergy with fine-tuning.

2 RELATED WORK

Time Series Forecasting. Among the most popular deep learning approaches for time series forecasting are RNN-based and transformer-based models. A line of research in RNN-based models include (Du et al., 2021; Wang et al., 2019; Salinas et al., 2020; Rasul et al., 2021), while transformer-based models feature, among many others (Zhou et al., 2022, 2021; Wu et al., 2021; Zhang and Yan, 2023; Liu et al., 2021, 2023; Lim et al., 2021; Nie et al., 2023), ForecastPFN (Dooley et al., 2023), TimePFN (Taga et al., 2025), and the model employed in this study: Chronos (Ansari et al., 2024). While most references above highlight domain-specific models, there is an emerging trend in zero-shot forecasting. A line of work there includes (Orozco and Roberts, 2020; Oreshkin et al., 2021; Jin et al., 2022; Dooley et al., 2023; Ansari et al., 2024). ForecastPFN (Dooley et al., 2023) and TimePFN (Taga et al., 2025) are trained exclusively on a synthetic dataset using the Prior-data Fitted Networks (PFNs) framework, a concept originally introduced by Müller et al. (2021). Chronos, on the other hand, is trained on real time series data corpora, which is augmented with synthetically generated time series examples via Gaussian processes to improve generalization.

Time Series Foundational Models describe large models trained on extensive datasets, enabling them to recognize patterns across various time series data domains (Liang et al., 2024). A notable line of work includes Moirai (Woo et al., 2024), Moment (Goswami et al., 2024), Lag-Llama (Rasul et al., 2023), TimesFM (Das et al., 2024), TimeGPT-1 (Garza et al., 2023), and Chronos (Ansari et al., 2024). Chronos proposes a scaling and quantization technique to train standard LLM architectures such as T5 (Raffel et al., 2020) and GPT-2 (Radford et al., 2019), demonstrating state-of-the-art zero-shot and few-shot capabilities.

Retrieval-Augmented Generation (RAG) is a key component in modern LLM pipelines, improving domain adaptation in various tasks without retraining. A line of work includes those of (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021; Khattab and Zaharia, 2020; Fan et al., 2021; Borgeaud et al., 2022; Ma

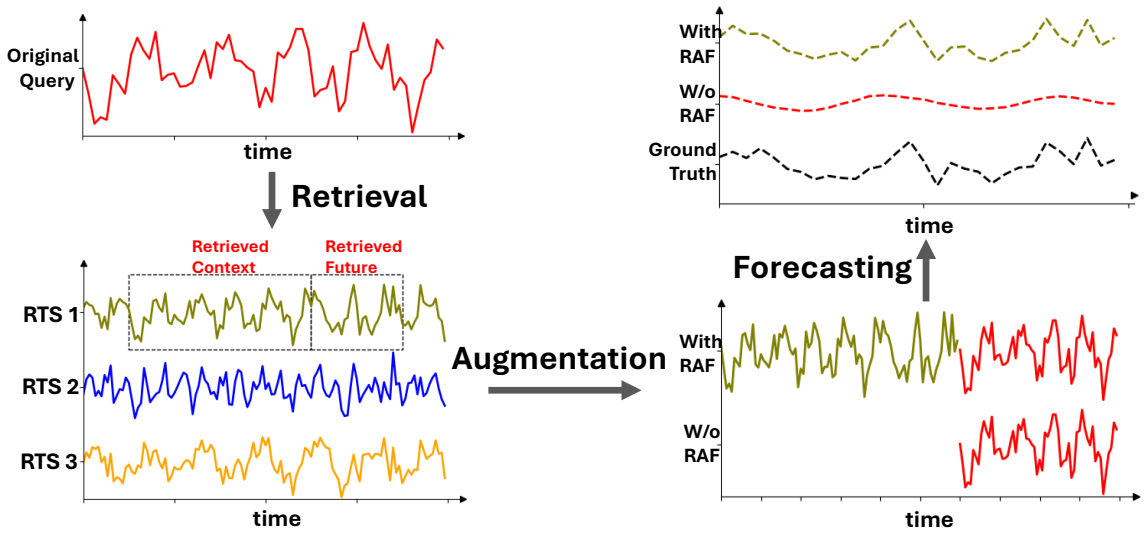

Figure 1: Overview of the *Retrieval Augmented Forecasting* (RAF) framework. *Top left:* The original query is used to retrieve the best-matching time series (RTS 1, RTS 2, RTS 3, ...). *Bottom left:* We utilize the best match (RTS 1) to form the retrieved context and retrieved future. *Bottom right:* These segments are then augmented with the original time series to produce an augmented input for forecasting. *Top right* figure displays the forecasts generated by Chronos Base. RAF outperforms the base (no-retrieval) model and returns a forecast closer to the actual future values.

et al., 2023; Thomas et al., 2024). Furthermore, recent studies have focused on utilizing Retrieval-Augmented Generation (RAG) for time series forecasting (Ravuru et al., 2024; Wang and Cui, 2024; Saveri and Bortolussi, 2024; Han et al., 2025). Ravuru et al. (2024) explores its application in agentic settings, Wang and Cui (2024) targets the domain of customer service, and Saveri and Bortolussi (2024) investigates mining temporal logic specifications from data. Our work sets itself apart by focusing on retrieval augmented generation in the context of time series foundational models.

3 PROBLEM SETUP

Notation. We use lower-case and upper-case bold letters (e.g., $\boldsymbol{a}, \boldsymbol{A}$) to represent vectors and matrices, respectively; a_i denotes the i-th entry of a vector $\boldsymbol{a}$.

Setting. Let $\boldsymbol{x} \in \mathbb{R}^L$ denote a univariate time series of length L. Let $f(\boldsymbol{x}) \in \mathbb{R}^H$ be the horizon-H forecast produced by model f, representing the prediction for the next H time steps given the input series $\boldsymbol{x}$. We use $\boldsymbol{x}[j, j+C-1]$ to denote the sub-series of length C that starts at time j and ends at time $j+C-1$. $(f(\boldsymbol{x}))[j, j+H-1]$ is defined analogously for the horizon-H output of f. A sub-series that approximately repeats within a longer time series is referred to as a **time-series motif**. Given a motif $\boldsymbol{m} \in \mathbb{R}^C$, we say that $\boldsymbol{x}[j, j+C-1]$ *matches* $\boldsymbol{m}$ if $\boldsymbol{x}[j, j+C-1] \approx \boldsymbol{m}$.

3.1 Time-series Retrieval Problem

Retrieval augmented forecasting is inherently related to the model's capability to identify motifs in the time series and utilize this to make inferences. This motivates our Time-Series Retrieval (TS-R) task, which measures the ability to match the current motif (i.e. query) to an earlier similar motif (i.e. key). The model is then asked to retrieve the context surrounding the earlier motif and use it as additional input to form its prediction.

Definition 1 (TS-R problem) *Let $\boldsymbol{m}$ be the motif at the end of the time series, i.e., $\boldsymbol{m} = \boldsymbol{x}[L-C+1, L] \in \mathbb{R}^C$. Suppose there is a unique matching motif in the history: $\boldsymbol{x}[t, t+C-1] = \boldsymbol{m}$ for a unique timestamp $t < L-C+1$. Let $\boldsymbol{\Upsilon} := \boldsymbol{x}[t+C, t+C+H-1] \in \mathbb{R}^H$ denote the length-H segment that follows this past occurrence. We say that a model f solves the TS-R problem if $f(\boldsymbol{x}) = \boldsymbol{\Upsilon}$.*

The TS-R task is inspired in part by the associative recall (AR) and induction heads tasks in language modeling (Olsson et al., 2022; Gu and Dao, 2024). These tasks ask for completing a bigram based on previous occurrences. For instance, if 'Geoff Hinton' occurred earlier in the text, the model should return 'Hinton' after seeing 'Geoff' next time. Similar to this, TS-R measures the ability to make predictions by motif retrieval. Com-

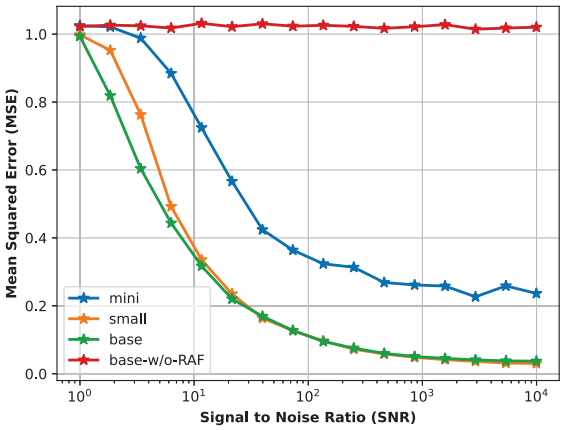

Figure 2: We generated synthetic time-series data by transposing two sinusoidal signals and projecting them via orthogonal projections. We assessed extrapolation behavior using scaled mean squared error (assuming 0 prediction as baseline) and chose a context and forecast length of $C = 30$ and $H = 30$. Evaluations were conducted on Chronos-{mini, small, base}.

pared to the induction heads problem, TS-R has two distinctions due to the continuous nature of time-series data: Rather than cosine similarity between tokens, we are asking for retrieval in Euclidean distance. This is because motifs within the input time series are expected to have distinct norms. Secondly, while tokens/words are clearly delineated in language, TS-R would benefit from intelligently encoding the motifs/sub-series. The following theorem provides a two-layer attention construction to solve the TS-R task.

Theorem 1 (*informal*) *A transformer architecture with two-attention blocks and absolute positional encoding can solve the Time-Series Retrieval (TS-R) problem by employing patch-embeddings with stride length 1 and by suitably encoding the norms and directions of the patches.*

This result shows that the use of transformer-based architectures for time-series retrieval is well-founded. The patching strategy we employ is similar to earlier works (Nie et al., 2023; Taga et al., 2025), however, we use a stride length of 1 to ensure every sub-series is tokenized and can be retrieved. The formal theorem statement is provided under Theorem 2 of Appendix A. Throughout the paper, we denote the motif $\boldsymbol{m}$ and $\boldsymbol{\Upsilon}$ as the retrieved context and retrieved future, respectively.

3.2 Synthetic Retrieval Experiment

In practice, time-series data are noisy, unlike the idealized setting of Definition 1. To assess Chronos' time-series retrieval behavior under noisy conditions, we investigate the following experimental setting. We fix two sinusoidal components with distinct frequencies f_1 and f_2 and sample a random phase $\phi \sim \mathrm{Unif}(0, 2\pi)$, yielding a signal of length $L = C + H$. We then apply a random orthonormal rotation $\mathbf{Q} \in \mathbb{R}^{L\times L}$ (with $\mathbf{Q}^\top\mathbf{Q} = \boldsymbol{I}$) to avoid a trivial axis-aligned structure. The synthetic series is thus generated as

$$\boldsymbol{s} = \mathbf{Q}\big(\sin(\pi f_1 \boldsymbol{t} + \phi) + \sin(\pi f_2 \boldsymbol{t} + \phi)\big).$$

To model noisy retrieval, we form a noisy match $\boldsymbol{s}_r = \boldsymbol{s} + \boldsymbol{z}$ with additive noise $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \sigma_s^2\mathbf{I})$. The variance σ_s^2 determines the signal-to-noise ratio in Figure 2. Based on the retrieved context $\boldsymbol{m}_r := \boldsymbol{s}_r[0,\, C-1]$ (the noisy counterpart of the clean motif $\boldsymbol{m} := \boldsymbol{s}[0,\, C-1]$) and the retrieved future $\boldsymbol{r} := \boldsymbol{s}_r[C,\, C+H-1]$, we construct the RAF query

$$\boldsymbol{s}_{\mathrm{raf}} = [\,\boldsymbol{m}_r \;\; \boldsymbol{r} \;\; \boldsymbol{m}\,] \in \mathbb{R}^{L+C}.$$

We evaluate the forecast $(f(\boldsymbol{s}_{\mathrm{raf}}))[0,\, H-1]$ against the noise-free future $\boldsymbol{s}[C,\, C+H-1]$, which serves as the ground truth, and report the per-step mean squared error.

As demonstrated in Figure 2, the mean squared error decreases as the signal-to-noise ratio increases across all model sizes. However, the convergence behavior varies: larger models exhibit faster convergence and smaller retrieval errors compared to smaller models. Additionally, the Chronos Mini completely fails; that is, even without noise, it is unable to perform retrieval, indicating a failure in the basic TS-R task. It is also noteworthy that without retrieval, no model can extrapolate based solely on a given motif $\boldsymbol{m}$. We provide an extensive discussion of these observations in Appendix A.

4 METHODOLOGY

Indexing and Database Formation. To retrieve the best matching time series as described in Section 3, we construct a database specific to each dataset, since datasets differ in scale, sampling frequency, seasonality, and noise. Consequently, we reserve 20% of the series in each dataset for testing and use the remaining 80% to form the *retrieval database* via a random but fixed split. To prevent information leakage, the database contains only time steps that occur strictly *before* the prediction window of the query series; segments may overlap the context window but never include values from the forecast horizon. From this dataset-specific

database, we then retrieve the best matches using approximate nearest-neighbor search over the chosen representations. An ablation study on retrieval database formation—comparing dataset-specific, same-domain, and cross-domain corpora—is provided in Appendix C.

Matching and Similarity Metric. The selection of the best-matching time series to the original time series is based on embedding similarity. In this approach, we first obtain an embedding of the original time series using the Chronos-Base model's encoder. Since decoder-only models do not have encoders, we use the Chronos-Base encoder as a shared feature extractor to obtain their embeddings as well. Then, we identify the top-n best matches by calculating the ℓ_2 norm between the original time series and the time series in our allocated database. The ℓ_2 norm is given by $||\boldsymbol{m} - \boldsymbol{y}||_{\ell_2} = \sqrt{\sum_{i=1}^{n}(m_i - y_i)^2}$, where $\boldsymbol{m}$ and $\boldsymbol{y}$ represent the vectors corresponding to the embeddings of the original time series and the time series retrieved from the database, respectively. Moreover, we compare ℓ_2, ℓ_1, cosine similarity, and Pearson's correlation in Appendix D.1. On longer-context Benchmark I datasets, ℓ_2 and Pearson's correlation tend to perform best, while on shorter-context Benchmark II datasets, cosine similarity and Pearson's correlation dominate, suggesting that direction-based metrics become more important when scale information is limited.

Instance Normalization. To mitigate the distribution shift effects between training and testing data, we apply instance normalization (Ulyanov et al., 2017; Kim et al., 2022). We normalize each time series instance $\boldsymbol{x}^{(i)}$ with zero mean and unit standard deviation. For the baseline approach, we normalize each $\boldsymbol{x}^{(i)}$ before prediction, and the mean and deviation are added back to the output prediction. On the other hand, original time series $\boldsymbol{x}^{(i)}$ and the retrieved time series $\boldsymbol{x}'^{(i)}$ are normalized separately before being input into the model in the form of *retrieval query formation*. The mean and deviation of $\boldsymbol{x}^{(i)}$ are then added back to the output prediction at the end.

Retrieval Query Formation. The initial step in query formation is identifying the best-matching time series from our database using the given similarity metric. The goal is to find a time series that closely matches the context (historical pattern - motif) of the time series we are working with. This retrieved time series serves as the *retrieved context*. After identifying the retrieved context, we focus on its future portion, which is the segment immediately following the retrieved context. The length of this segment, termed the *retrieved future*, corresponds exactly to the prediction length we aim to forecast. This combination of the retrieved context and retrieved future forms the *retrieved time series*. The retrieved time series, which includes both the retrieved context and the retrieved future, undergoes instance normalization. This step ensures that the data from the retrieved series is scaled appropriately, eliminating any discrepancies that may arise due to varying magnitudes in different time series. The same normalization process is also applied to the original context to maintain consistency.

Once both the original context and the retrieved time series are normalized, we enforce continuity at the join by applying an additive offset to the retrieved segment so that its last value matches the first value of the context. Concretely, we shift the retrieved sequence by this offset and then concatenate the adjusted retrieved segment and the context end-to-start. This alignment prevents any abrupt changes or discontinuities that might negatively affect the performance of the models. After the alignment, we concatenate the retrieved time series and the normalized original context. This combined, augmented time series serves as the input for the TSFMs. Although our current setup employs a single retrieved time series (k=1), we examine the effects of multiple retrievals and present an ablation study comparing our approach with different retrieval settings such as retrieval-only baselines and matched input-length comparisons (see Appendix D.2 - Appendix D.5).

Time-series models. We evaluate RAF on four TSFMs: **Chronos** (Ansari et al., 2024), a probabilistic auto-regressive decoder, for which we report results on both *Mini* and *Base* variants to study size effects; **Moirai** (Woo et al., 2024), pretrained in a unified, cross-domain fashion; **TimesFM** (Das et al., 2024), a decoder-only model trained on massive real-world time-series, providing zero-shot probabilistic forecasts; and **Lag-Llama** (Rasul et al., 2023), a causal decoder-only architecture with lag-token conditioning. In addition, we include three non-Transformer baselines: **DLinear** (Zeng et al., 2022), a decomposition-style linear forecaster; **GBDT (LightGBM)** (Ke et al., 2017), trained on fixed-length lag and calendar features; and **ARIMA**/auto-ARIMA (Box et al., 2015; Hyndman and Khandakar, 2008) as a classical Box–Jenkins baseline.

5 EXPERIMENTS

5.1 Baselines and Evaluation Metrics

Our first set of experiments compares Chronos models augmented with RAF to their non-retrieval baselines, both in the zero-shot setting (Tables 1, 2) and after fine-tuning on the target dataset (Table 17).

For a fair comparison, we evaluate across two benchmarks: Benchmark I sweeps four context lengths $C \in \{50, 75, 100, 150\}$ and three horizons $H \in \{10, 15, 20\}$; Benchmark II uses $C \in \{10, 15, 18, 21\}$ and $H \in \{3, 4, 5\}$.

In the second set of experiments, we focus on architecture-level generality. We pick four datasets from each benchmark (eight total) and evaluate four TSFMs (Chronos, Moirai, TimesFM, Lag-Llama). To isolate architectural effects, we fix C=100, H=10 for Benchmark I and C=21, H=4 for Benchmark II datasets. We then report the average percentage improvement of RAF for each model over the corresponding baseline (Table 3).

For evaluation, we assessed RAF on both probabilistic and point-forecast performance, as recommended by (Ansari et al., 2024). We used the weighted quantile loss (WQL) to measure the quality of probabilistic forecasts; WQL quantifies how closely the predicted distribution matches the observed values across quantile levels. Following (Ansari et al., 2024), we computed WQL at nine evenly spaced quantiles, $\{0.1, 0.2, \ldots, 0.9\}$. For point forecasts, we used the mean absolute scaled error (MASE; Hyndman and Koehler, 2006), which scales the absolute forecast error by the average in-sample error while accounting for seasonality.

Finally, for the first set of experiments, we averaged WQL and MASE over three distinct prediction lengths for each dataset, yielding a single score per context length (as in Ansari et al., 2024). Furthermore, we report per-dataset relative improvements (RAF vs. baseline) for each benchmark, with details in Appendix F.5. For the second set, we simply present WQL and MASE at the fixed H and C values.

5.2 Main Results

We present primary results on two benchmarks—Benchmark I (6 datasets) and Benchmark II (5 datasets). Across both, RAF outperforms the baseline on Chronos Mini and Chronos Base, with larger absolute gains on the larger model, indicating a clear size effect. We further evaluate RAF with four TSFMs—Chronos (Base), Moirai (Base), TimesFM, and Lag-Llama—and observe consistent improvements in both WQL and MASE. Lastly, fine-tuning with retrieval yields additional gains in Benchmark I.

• **Benchmark I** is comprised of 6 datasets selected to evaluate the performance of RAF across various, longer context and prediction lengths. These datasets were not used by Chronos during training. Table 1 summarizes the probabilistic and point forecasting performance of RAF and the baseline across four different context lengths, with average scores across three prediction lengths, based on their MASE and WQL scores, as described in Section 5.1. Full results for each prediction length are provided in the appendix.

In general, RAF demonstrates better performance than the baseline approach in terms of WQL and MASE, with smaller values indicating better predictions. For instance, when RAF is tested on Chronos Mini and Chronos Base, in datasets such as Weather, FRED-MD, and ETTh1, RAF consistently outperforms the baseline across all four context lengths, showing superior performance in both WQL and MASE metrics. The improvements are particularly significant for Chronos Base in ETTh1 and FRED-MD, where RAF achieves lower WQL and MASE values across all context lengths, reinforcing the consistent advantage of RAF.

For datasets like Traffic and Covid Deaths, the improvements made by RAF are relatively small but still notable for both models. In particular, for Covid Deaths, RAF shows a modest but consistent advantage in WQL, especially at shorter context lengths. Meanwhile, for the NN5 dataset, RAF and the baseline show similar performance, with minimal differences in both WQL and MASE across all context lengths on average in Chronos Mini. However, this is not the case for Chronos Base, as NN5 is one of the datasets where the average improvement is significant. This idea is supported by the analysis shown in Figure 2, where the error values for Chronos Base were lower than those for Chronos Mini across various context lengths. Furthermore, Chronos Mini is unable to solve the TS-R problem in the synthetic setting, even in a noiseless environment as depicted in Figure 2. This discrepancy emphasizes that RAF demonstrates enhanced capabilities with larger models, underscoring the importance of model selection in RAF, particularly for handling longer context lengths. Overall, RAF generally exhibits more consistent and superior performance across most datasets in both models, with some exceptions where the performance gap is narrower.

• **Benchmark II** is comprised of 5 datasets selected to evaluate the performance of RAF across various scenarios, but this time with much shorter context and prediction lengths. Table 2 summarizes the average performance of RAF and the baseline on Benchmark II, in terms of MASE and WQL scores. Full results for each prediction length are again provided in the Appendix F.

Similar to Benchmark I, RAF demonstrates superior performance compared to the baseline on Benchmark II. For Chronos Mini, in both the Tourism (Quarterly) and Uber TLC datasets, RAF outperforms the baseline across all four context lengths on average, yielding better results in both WQL and MASE. The same

Table 1: Average performance of RAF on Benchmark I across three prediction lengths, $H \in \{10, 15, 20\}$, with context lengths $C \in \{50, 75, 100, 150\}$, evaluated using two models: Chronos Mini (m) and Chronos Base (b). We employ the WQL and MASE metrics as evaluation criteria.

Datasets		Weather		Traffic		ETTh1		FRED-MD		Covid Deaths		NN5	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
m-RAF	50	**0.168**	**1.677**	0.254	2.114	**0.124**	**1.136**	**0.164**	**0.794**	**0.011**	**12.097**	**0.210**	**0.698**
	75	**0.167**	**1.575**	**0.250**	**2.301**	**0.127**	**1.312**	**0.107**	**0.682**	**0.009**	**14.921**	**0.222**	0.739
	100	**0.167**	**1.505**	**0.256**	**2.798**	**0.095**	**1.167**	**0.113**	**0.685**	**0.008**	11.781	**0.200**	**0.642**
	150	**0.163**	1.094	**0.246**	**1.973**	**0.108**	**0.979**	**0.069**	**0.441**	**0.008**	11.491	0.199	0.618
m-Base.	50	0.173	1.729	**0.253**	**1.994**	0.148	1.250	0.191	0.940	0.015	12.362	0.212	0.702
	75	0.173	1.606	0.266	2.400	0.142	1.369	0.164	0.791	0.012	15.574	0.223	**0.729**
	100	0.172	1.507	0.267	2.854	0.144	1.457	0.131	0.724	0.010	**11.456**	0.205	0.670
	150	0.166	**1.070**	0.261	2.058	0.163	1.253	0.070	0.497	0.012	**11.401**	**0.193**	**0.597**
b-RAF	50	**0.158**	**1.667**	**0.225**	**2.123**	**0.067**	**0.871**	**0.045**	**0.660**	**0.009**	**8.822**	**0.171**	**0.552**
	75	**0.158**	**1.574**	**0.231**	**2.274**	**0.057**	**0.837**	**0.045**	**0.613**	0.008	**13.892**	**0.150**	**0.519**
	100	**0.162**	**1.500**	**0.226**	**2.827**	**0.048**	**0.815**	**0.091**	**0.603**	**0.006**	**10.699**	**0.155**	**0.490**
	150	**0.166**	1.132	**0.223**	2.029	**0.054**	**0.693**	0.036	**0.377**	**0.010**	**10.443**	**0.153**	**0.499**
b-Base.	50	0.158	1.784	0.246	2.319	0.110	1.059	0.209	0.917	0.021	10.037	0.196	0.609
	75	0.160	1.613	0.234	2.443	0.117	1.175	0.148	0.710	**0.007**	15.343	0.180	0.577
	100	0.166	1.569	0.235	2.870	0.127	1.266	0.095	0.675	0.007	11.345	0.180	0.533
	150	0.166	**1.129**	0.225	**2.006**	0.122	1.034	**0.030**	0.464	0.010	10.907	0.185	0.533

Table 2: Average performance of RAF on Benchmark II across three prediction lengths, $H \in \{3, 4, 5\}$, with context lengths $C \in \{10, 15, 18, 21\}$, evaluated using two models: Chronos Mini (m) and Chronos Base (b). The performance of RAF in both models was evaluated across various datasets using WQL and MASE as metrics.

Datasets		Tourism (M.)		Tourism (Q.)		M1 (M.)		Uber TLC		CIF-2016	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
m-RAF	10	**0.190**	**0.797**	**0.146**	**2.319**	**0.174**	**1.277**	**0.221**	**1.234**	**0.055**	1.259
	15	0.576	2.411	**0.092**	**1.238**	0.170	1.324	**0.183**	**1.112**	**0.051**	**1.180**
	18	**0.176**	**1.629**	**0.090**	**1.046**	**0.160**	**1.121**	**0.173**	**0.983**	**0.056**	**0.812**
	21	**0.086**	**1.271**	**0.089**	**1.134**	**0.155**	**1.073**	**0.159**	**0.990**	**0.053**	**0.867**
m-Base.	10	0.613	0.931	0.214	3.066	0.201	1.300	0.271	1.369	0.056	**1.136**
	15	**0.543**	**2.278**	0.134	1.510	**0.168**	**1.300**	0.215	1.235	0.060	1.290
	18	0.357	1.736	0.126	1.263	0.168	1.155	0.203	1.092	0.057	0.963
	21	0.287	1.317	0.095	1.163	0.160	1.120	0.167	1.044	0.066	0.927
b-RAF	10	**0.459**	**0.859**	0.101	1.467	**0.170**	**1.280**	**0.212**	**1.171**	**0.070**	**1.281**
	15	**0.279**	**2.631**	**0.085**	**1.166**	**0.159**	**1.300**	0.188	**1.078**	**0.060**	**1.293**
	18	**0.135**	**1.584**	**0.088**	**1.071**	**0.170**	**1.071**	**0.143**	**0.892**	**0.057**	**1.026**
	21	**0.074**	**1.213**	**0.080**	**1.013**	**0.157**	**0.930**	**0.146**	**0.936**	0.048	**0.724**
b-Base.	10	0.562	0.888	**0.093**	**1.203**	0.180	1.298	0.272	1.394	0.073	1.310
	15	0.320	2.688	0.091	1.188	0.167	1.307	**0.186**	1.134	0.064	1.373
	18	0.371	1.696	0.095	1.124	0.175	1.117	0.188	1.005	0.060	1.044
	21	0.370	1.331	0.086	1.113	0.174	1.077	0.156	0.986	**0.047**	0.924

dominance in performance is also apparent in the Uber TLC dataset in Chronos Base, along with Tourism (Monthly). Even in the other datasets, such as Tourism (Monthly), M1, and CIF-2016, RAF maintains a significantly better performance than the baseline for both models. Notably, the only dataset seen during the training of Chronos was Uber TLC, which may have contributed to RAF's particularly strong performance on this dataset. This observation suggests that while RAF generalizes well to unseen datasets, exposure to a specific dataset during training may further enhance its predictive accuracy for that dataset. This finding prompted us to explore the impact of RAF on fine-tuning models for better performance.

To assess overall performance gains across datasets and both benchmarks, we also report per-dataset relative improvements, as mentioned in Section 5.1. Figures 3 and 4 (Appendix F.5) summarize the aggregate relative MASE and WQL values for Chronos Mini and Chronos Base. For Chronos Mini, datasets in Benchmark II generally benefit more from time-series retrieval than those in Benchmark I, as indicated by lower relative MASE/WQL values in Benchmark II. On the other hand, Chronos Base shows larger gains on Benchmark I, which features longer context and prediction lengths; its greater capacity appears to better exploit retrieved motifs under these settings, yielding lower relative MASE/WQL. This pattern suggests that retrieval is more effective with shorter contexts for smaller models (where nearest matches are easier to identify), while larger models can capitalize on retrieval even when

Table 3: Evaluation of RAF on four TSFMs (Chronos Base, Moirai Base, TimesFM, Lag-Llama) over two benchmarks, each with four datasets. Benchmark I uses a long-context setting ($C = 100$, $H = 10$); Benchmark II uses a short-context setting ($C = 21$, $H = 4$). **Bold** numbers indicate better performance for that model/metric.

	Dataset	Chronos				Moirai				TimesFM				Lag-Llama			
		Baseline		RAF		Baseline		RAF		Baseline		RAF		Baseline		RAF	
		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
Bench. I	ETTh1	0.076	0.851	**0.025**	**0.551**	0.077	1.030	**0.076**	**0.971**	0.094	1.009	**0.090**	**0.911**	0.123	1.388	**0.116**	**1.258**
	FRED-MD	0.095	0.595	**0.074**	**0.572**	0.028	0.592	**0.020**	**0.563**	0.080	0.502	**0.023**	**0.475**	0.115	1.238	**0.106**	**1.188**
	NN5	0.157	0.442	**0.145**	**0.432**	0.154	0.492	**0.142**	**0.455**	**0.679**	**1.843**	0.801	2.046	0.274	0.877	**0.260**	**0.869**
	Covid Deaths	0.004	5.425	**0.004**	**5.293**	**0.006**	7.408	0.007	**6.915**	0.016	11.589	**0.012**	**9.400**	**0.074**	22.087	0.078	**18.164**
Bench. II	Tourism (Q.)	0.090	1.105	**0.088**	**1.035**	0.179	1.905	**0.175**	**1.841**	**0.096**	1.259	0.097	**1.216**	0.240	3.156	**0.203**	**2.799**
	M1	0.172	1.067	**0.140**	**0.880**	**0.169**	1.147	0.174	**1.128**	0.197	1.194	**0.177**	**1.073**	**0.198**	1.335	0.200	**1.282**
	Uber TLC	0.150	0.954	**0.145**	**0.876**	0.206	1.110	**0.202**	**1.104**	0.130	0.710	**0.120**	**0.658**	0.272	1.379	**0.251**	**1.346**
	CIF-2016	0.040	0.954	**0.038**	**0.692**	0.051	0.896	**0.038**	**0.844**	0.073	**1.150**	**0.048**	1.214	0.059	1.294	**0.051**	**1.281**
avg. impr.				**+15.8%**	**+12.4%**			**+6.0%**	**+4.5%**			**+16.7%**	**+4.8%**			**+6.1%**	**+6.4%**

Table 4: Evaluation of RAF on non-Transformer baselines (DLinear, GBDT with LGBM, ARIMA) over two benchmarks, each with four datasets. Benchmark I uses a long-context setting ($C = 100$, $H = 10$); Benchmark II uses a short-context setting ($C = 21$, $H = 4$). **Bold** numbers indicate better performance for that model/metric.

	Dataset	DLinear				GBDT (LGBM)				ARIMA			
		Baseline		RAF		Baseline		RAF		Baseline		RAF	
		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
Bench. I	ETTh1	**0.460**	**3.891**	0.471	3.940	0.271	1.968	0.271	1.968	**0.111**	**1.471**	0.221	1.777
	FRED-MD	0.296	1.120	**0.270**	**1.029**	**0.507**	**1.640**	0.520	1.634	**0.029**	**0.495**	0.056	0.513
	NN5	0.748	1.872	**0.740**	**1.852**	**0.189**	**0.506**	0.190	0.508	0.532	1.492	**0.519**	**1.410**
	Covid Deaths	0.064	**13.942**	**0.058**	14.578	0.059	18.790	0.059	18.790	0.003	7.117	**0.003**	**7.069**
Bench. II	Tourism (Q.)	0.372	3.900	**0.348**	**3.789**	0.115	**1.200**	**0.114**	1.201	0.257	2.679	**0.166**	**1.886**
	M1	**0.248**	**1.622**	0.257	1.692	**0.174**	**1.198**	0.174	1.203	**0.209**	1.215	0.219	**1.157**
	Uber TLC	**0.287**	**1.464**	0.314	1.624	**0.166**	0.943	0.167	**0.942**	**0.188**	**1.023**	0.205	1.249
	CIF-2016	0.256	1.745	**0.248**	**1.740**	0.284	1.630	0.284	1.630	0.124	**0.864**	**0.073**	0.875
avg. impr.				**+1.7 %**	**-1.1 %**			**-0.35 %**	**-0.05 %**			**-15.9 %**	**-0.9 %**

contexts are long. Conversely, Benchmark I exhibits greater variability for Chronos Mini, indicating potential challenges in retrieving optimal matches for longer contexts in certain datasets.

Overall, RAF improves the performance of both Chronos models across benchmarks. For Chronos Mini, the aggregate relative scores (lower is better) are 0.887 (WQL) and 0.950 (MASE) on Benchmark I, and 0.762 (WQL) and 0.925 (MASE) on Benchmark II. For Chronos Base, the corresponding scores are 0.733 (WQL) and 0.880 (MASE) on Benchmark I, and 0.821 (WQL) and 0.944 (MASE) on Benchmark II. The overall averages across all datasets further confirm these gains: 0.823 (WQL) and 0.939 (MASE) for Chronos Mini, and 0.772 (WQL) and 0.908 (MASE) for Chronos Base (Appendix F.5). These results suggest that more complex models, such as Chronos Base, can better leverage retrieved information under longer context lengths.

Across Different TSFMs. The results of RAF across four TSFMs are reported in Table 3. RAF's gains are not confined to a single backbone: it reduces both WQL and MASE for all four models despite their differing design choices. Averaged over datasets, the largest WQL improvements occur with TimesFM (+16.7%) and Chronos (+15.8%), suggesting that retrieval particularly benefits large models that were not fine-tuned on these benchmarks. For MASE, the largest relative reductions are on Chronos (+12.4%) and Lag-Llama (+6.4%), indicating that RAF also sharpens point forecasts. These cross-architecture gains imply that retrieval provides complementary, task-specific signals beyond each model's inductive biases, exposing patterns not retained from pretraining.

Still, per-dataset results show that the magnitude of improvement varies with the data regime: large, seasonal datasets such as ETTh1, M1, and Tourism benefit strongly and consistently, whereas noisier series (e.g., Covid Deaths) exhibit smaller or mixed effects. This pattern aligns with our hypothesis that RAF excels when "nearby" motifs exist in the time-series pool and the base model can exploit them once surfaced. Overall, the averaged gains confirm that RAF is a broadly ap-

plicable plug-in: it lowers both probabilistic and point errors across diverse TSFMs without any architecture-specific engineering.

Across Non-Transformer Baselines. Table 4 shows that adding RAF to classic non-attention models yields little to no average benefit and can even hurt performance. On DLinear, RAF slightly lowers WQL (+1.7%) yet raises MASE (-1.1%). GBDT with LGBM is effectively unchanged: WQL (-0.35%) and MASE (-0.05%). ARIMA degrades on average: WQL (-15.9%) and MASE (-0.9%). Per-dataset outcomes are mixed as a few datasets show small gains, but the pattern is not consistent. This aligns with our expectation that retrieval helps when the backbone can attend to and fuse retrieved motifs at inference time. Linear filters such as DLinear, tree ensembles over fixed windows such as GBDT, and parametric ARIMA processes do not natively integrate cross-series context; retrieved segments behave as weak covariates and can disturb inductive biases, which often affects probabilistic calibration more than point accuracy, hence the larger swings in WQL. Complete experimental setups for the models are provided in Appendix B.

Fine-tuning evaluations. Motivated by the strong zero-shot performance of RAF, we fine-tuned Chronos Base and Mini on each Benchmark I dataset, comparing runs *with* and *without* retrieval (details in Appendix E). Each model was fine-tuned only on the dataset on which it was evaluated.

As shown in Table 17 (Appendix E), *Advanced RAF* achieves the lowest WQL/MASE on most datasets—e.g., on NN5, MASE drops from 0.563 to 0.401 for Chronos Mini and from 0.481 to 0.378 for Chronos Base—demonstrating that retrieval and fine-tuning are complementary. Exceptions exist: on Covid Deaths (Chronos Mini), Baseline FT outperforms Advanced RAF on both metrics. Notably, even without fine-tuning, the baseline exceeds Naïve RAF for this dataset under the chosen (C, H), suggesting that fine-tuning may be unnecessary when retrieval adds limited signal.

Robustness to data corruption. To evaluate how RAF behaves when the input data are imperfect, we run controlled stress tests with Gaussian noise and missing-at-random (MCAR) corruption on Chronos-Base (Appendix G). RAF continues to outperform the baseline under moderate corruption, specifically for noise levels up to $\sigma \leq 0.6$ and for missing-data rates up to 60%. Beyond these levels, performance declines gradually rather than failing abruptly. Interestingly, a small amount of noise ($\sigma = 0.2$) gives the largest relative gain, reducing WQL by 9.64%, which suggests that slight perturbations may improve retrieval diversity. Overall, the main limit is not the noise level itself, but whether the corrupted context still preserves enough structural information for meaningful nearest-neighbor retrieval

6 CONCLUSION

In this paper, we have introduced the RAF framework for time series foundation models, which leverages retrieval augmentation and fine-tuning to enhance forecast accuracy. By incorporating external, domain-specific knowledge during inference through retrieval, RAF provides substantial improvements in both probabilistic and point forecasting tasks. Furthermore, we have validated the generality of RAF across four TSFMs—Chronos (Mini/Base), Moirai, TimesFM, and Lag-Llama—observing consistent improvements in both WQL and MASE. Finally, we have explored two variants: Naive RAF, which uses TSFMs as black boxes without modifying their weights, and Advanced RAF, which fine-tunes models for enhanced retrieval integration. We have evaluated the performance of both Naive and Advanced RAF across diverse datasets and settings.

Our experimental results demonstrate that both Naive and Advanced RAF outperform the standard baseline approach on average. These findings highlight the flexibility and robustness of the RAF framework in improving time series forecasting performance, particularly in scenarios that require adapting to varying historical data contexts and forecasting needs. Importantly, our study has also revealed that model size matters: Chronos Mini fails to solve simple synthetic retrieval tasks and larger models benefit more from retrieval-augmented forecasting both in synthetic and real experiments. As a future perspective, we propose expanding the RAF framework to handle multi-channel predictions.

Author Contributions

K.T. implemented and ran the real-data retrieval experiments and ablations, and wrote the methodology, experiments, and most of the appendix. E.O.T. conceived the project, developed the methodology and retrieval mechanism, co-supervised the project, developed the theory, designed and ran the synthetic retrieval experiments, and wrote the introduction, problem setup, and synthetic experiments. M.E.I. contributed to the theoretical formulation and proofs, including the setting in Figure 2. S.O. supervised the project throughout.

Acknowledgements

This work is supported by the National Science Foundation grants CCF-2046816, CCF-2403075, CCF-2212426, the Office of Naval Research grant N000142412289, a gift from Open Philanthropy, and an Adobe Data Science Research Award. The computational aspects of the research are generously supported by computational resources provided by the Amazon Research Award on Foundation Model Development.

References

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Wilson, A. G., Bohlke-Schneider, M., and Wang, Y. (2024). Chronos: Learning the language of time series.

Athanasopoulos, G., Hyndman, R. J., Song, H., and Wu, D. C. (2011). The tourism forecasting competition. *International Journal of Forecasting*, 27(3):822–844.

Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., Ren, J., Li, W. W., and Noble, W. S. (2009). Meme suite: tools for motif discovery and searching. *Nucleic acids research*, 37(suppl_2):W202–W208.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pages 359–370.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G. B., Lespiau, J.-B., Damoc, B., Clark, A., et al. (2022). Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., and Pfister, T. (2023). Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*.

Das, A., Kong, W., Sen, R., and Zhou, Y. (2024). A decoder-only foundation model for time-series forecasting. In *Forty-first international conference on machine learning*.

Dooley, S., Khurana, G. S., Mohapatra, C., Naidu, S. V., and White, C. (2023). Forecastpfn: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36:2403–2426.

Du, Y., Wang, J., Feng, W., Pan, S., Qin, T., Xu, R., and Wang, C. (2021). Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 402–411.

Fan, A., Gardent, C., Braud, C., and Bordes, A. (2021). Augmenting transformers with knn-based composite memory for dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99.

Garza, A., Challu, C., and Mergenthaler-Canseco, M. (2023). Timegpt-1. *arXiv preprint arXiv:2310.03589*.

Godahewa, R., Bergmeir, C., Webb, G. I., Hyndman, R. J., and Montero-Manso, P. (2021). Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., and Dubrawski, A. (2024). Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*.

Gu, A. and Dao, T. (2024). Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. (2020). Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Han, S., Lee, S., Cha, M., Arik, S. O., and Yoon, J. (2025). Retrieval augmented time series forecasting. In *Forty-second International Conference on Machine Learning*.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 27(3):1–22.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., and Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of forecasting*, 18(3):439–454.

Izacard, G. and Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

Jin, X., Park, Y., Maddix, D., Wang, H., and Wang, Y. (2022). Domain adaptation for time series forecasting

via attention sharing. In *International Conference on Machine Learning*, pages 10280–10297. PMLR.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Khattab, O., Potts, C., and Zaharia, M. (2021). Baleen: Robust multi-hop reasoning at scale via condensed retrieval. *Advances in Neural Information Processing Systems*, 34:27670–27682.

Khattab, O. and Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. (2022). Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

Lee, K., Chang, M.-W., and Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 6086–6096.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Li, H., Su, Y., Cai, D., Wang, Y., and Liu, L. (2022). A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., and Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32.

Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., Pan, S., and Wen, Q. (2024). Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6555–6565.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4):1748–1764.

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., and Dustdar, S. (2021). Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. (2023). itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Ma, X., Gong, Y., He, P., Zhao, H., and Duan, N. (2023). Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Makridakis, S. and Hibon, M. (1979). Accuracy of forecasting: An empirical investigation. *Journal of the Royal Statistical Society: Series A (General)*, 142(2):97–125.

Müller, M. et al. (2007). Dynamic time warping. *Information retrieval for music and motion*, 69:84.

Müller, S., Hollmann, N., Arango, S. P., Grabocka, J., and Hutter, F. (2021). Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*.

Nori, H., Lee, Y. T., Zhang, S., Carignan, D., Edgar, R., Fusi, N., King, N., Larson, J., Li, Y., Liu, W., Luo, R., McKinney, S. M., Ness, R. O., Poon, H., Qin, T., Usuyama, N., White, C., and Horvitz, E. (2023). Can generalist foundation models outcompete special-purpose tuning? case study in medicine.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. (2022). In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. (2021). Meta-learning framework with applications to zero-shot time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9242–9250.

Orozco, B. P. and Roberts, S. J. (2020). Zero-shot and few-shot time series forecasting with ordinal regression recurrent neural networks.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models

are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., et al. (2023). Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*.

Rasul, K., Seward, C., Schuster, I., and Vollgraf, R. (2021). Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, pages 8857–8868. PMLR.

Ravuru, C., Sakhinana, S. S., and Runkana, V. (2024). Agentic retrieval-augmented generation for time series analysis. *arXiv preprint arXiv:2408.14484*.

Sakoe, H. and Chiba, S. (2003). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. (2020). Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191.

Saveri, G. and Bortolussi, L. (2024). Retrieval-augmented mining of temporal logic specifications from data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 315–331. Springer.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. (2023). Roformer: Enhanced transformer with rotary position embedding.

Taga, E. O., Ildiz, M. E., and Oymak, S. (2025). Timepfn: Effective multivariate time series forecasting with synthetic data.

Talukder, S., Yue, Y., and Gkioxari, G. (2025). Totem: Tokenized time series embeddings for general time series analysis.

Thomas, V., Ma, J., Hosseinzadeh, R., Golestan, K., Yu, G., Volkovs, M., and Caterini, A. (2024). Retrieval & fine-tuning for in-context tabular models. *Advances in Neural Information Processing Systems*, 37:108439–108467.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Instance normalization: The missing ingredient for fast stylization.

Wang, T. and Cui, G. (2024). Ratsf: Empowering customer service volume management through retrieval-augmented time-series forecasting. *arXiv preprint arXiv:2403.04180*.

Wang, Y., Smola, A., Maddix, D., Gasthaus, J., Foster, D., and Januschowski, T. (2019). Deep factors for forecasting. In *International conference on machine learning*, pages 6607–6617. PMLR.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. (2024). Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*.

Wu, H., Xu, J., Wang, J., and Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430.

Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Silva, D. F., Mueen, A., and Keogh, E. (2016). Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. Ieee.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2022). Are transformers effective for time series forecasting?

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

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. (2022). Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR.

Zhu, Y., Zimmerman, Z., Senobari, N. S., Yeh, C.-C. M., Funning, G., Mueen, A., Brisk, P., and Keogh, E. (2016). Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 739–748. Ieee.

Checklist

1. For all models and algorithms presented, check if you include:

(a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**]. The setting, assumptions, and algorithm are stated clearly. in the main body and in the appendix

(b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Not Applicable**]

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**]. We provide an anonymized source code with required dependencies.

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. [**Yes**]

(b) Complete proofs of all theoretical results. [**Yes**]

(c) Clear explanations of any assumptions. [**Yes**]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**]. The details are given in Appendix C.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**] We conducted all experiments with a fixed random seed and evaluated using MASE and WQL. The explanation of the metrics is provided in the main paper.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**]. The details are given in Appendix 9.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [**Yes**]

(b) The license information of the assets, if applicable. [**Yes**]

(c) New assets either in the supplemental material or as a URL, if applicable. [**Yes**]. Codebase is given in Supplementary Materials.

(d) Information about consent from data providers/curators. [**Not Applicable**]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [**Not Applicable**]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]

Supplementary Materials

A Theoretical Results

A.1 TS-R Problem

In Section 3, we asserted that a two layer transformer architecture can solve the TS-R problem with mild assumptions employed by various transformer-based time-series architectures. In the below theorem, we prove that with patching, a 2-layer transformer architecture can indeed solve the TS-R problem. Note that our proof is based on the literature on the *nearest neighbor retrieval*, where a previous line of work (Olsson et al., 2022; Gu and Dao, 2024) has shown that softmax attention can implement nearest neighbor retrieval.

Setting. We use the definitions in Section 3. Given $\boldsymbol{x} \in \mathbb{R}^L$ denoting a univariate time series of length L, and with C denoting the context length, we extract patches of size C with a sliding window of size 1. Furthermore, we assume $H \leq C$. Without loss of generality, from here on we assume $H = C$ as we can trim the output after the retrieval. Overall, we get a patched input sequence $X = [x_1 \quad x_2 \quad ... \quad x_{L-C+1}]$. Moreover, we embed each x_i to $\bar{x}_i := [\frac{x_i^T}{\|x_i\|_{\ell_2}} \quad \|x_i\|_{\ell_2}]^T \in \mathbb{R}^{C+1}$. That is, we embed each x_i so that we store the direction and the magnitude in seperate dimensions (there is a clear bijective mapping that is inverse of this embedding, denote by $\boldsymbol{g}$). Thus, with this mapping, we define the $\bar{X} := [\bar{x}_1 \quad \bar{x}_2 \quad ... \quad \bar{x}_{L-C+1}]$. Based on $\bar{X}$, we define the token embedding matrix as $\boldsymbol{X} := [\bar{x}_1 \quad \bar{x}_2 \quad ... \quad \bar{x}_{L-C+1}]^T \in \mathbb{R}^{(L-C+1)\times(C+1)}$. Note that based on X, we can recover each x_i.

Moreover, let $\boldsymbol{p}_i$ be fixed positional encodings, denoting the positions of the tokens. Define $\boldsymbol{X}_{PE} := [\bar{x}_1 + \boldsymbol{p}_1 ... \quad \bar{x}_{L-C+1} + \boldsymbol{p}_{L-C+1}]^T$. For the ease of notation, we use $\boldsymbol{X} = \boldsymbol{X}_{PE}$ from here on.

Assumption 1 *We assume that the positional encodings $(\boldsymbol{p}_i)_{i=1}^{L-C+1}$ have unit ℓ_2 norm and are unique. Moreover, we assume that positional encodings are orthogonal to tokens $\boldsymbol{p}_i^T \bar{x}_j = 0$ (if there are no such token positional encodings, without loss of generality, we can just concatenate $\boldsymbol{x}_i$ with 0 vectors of required size). In addition to that, we assume that retrieved token positions are rotated versions of the value positions. That is, there is a unitary matrix $\boldsymbol{R}$ such that $\boldsymbol{p}_{i+C} = \boldsymbol{R}\boldsymbol{p}_i$ for all $1 \leq i \leq L - 2C + 1$.*

As in the retrieval, given a matching motif patch, the value is in C forward patches, we put above rotational assumption with rotation value as C. Note that this idea is also employed in one of the most popular positional embedding strategies, namely Rotational Positional Encoding (RoPE) (Su et al., 2023).

Moreover, denote the projection matrix associated to the token embeddings via $\boldsymbol{\Phi}$, where $\boldsymbol{\Phi}\bar{x}_1 = \bar{x}_1$ and $\boldsymbol{\Phi}\boldsymbol{p}_i = 0$. This means $\boldsymbol{\Phi}^\perp = \boldsymbol{I} - \boldsymbol{\Phi}$, implying that $\boldsymbol{\Phi}^\perp \bar{x}_1 = 0$ and $\boldsymbol{\Phi}^\perp \boldsymbol{p}_i = \boldsymbol{p}_i$.

Attention model. We consider a 2-layer attention as $\boldsymbol{X}_{tr} = \boldsymbol{X}_{0:L-C,:}$, that is the truncated version of $\boldsymbol{X}$ where we remove the last row. $\boldsymbol{N}$ is the diagonal normalization matrix that normalizes each row of $\boldsymbol{X}_{tr}$ to be unit norm. Moreover, In the first attention layer, we write $\tilde{\boldsymbol{x}} = f_1(x_{L-C+1}) = \boldsymbol{X}_{tr}^T \mathbb{S}(\boldsymbol{N}\boldsymbol{X}_{tr}\boldsymbol{W}_1 \frac{x_{L-C+1}}{\|x_{L-C+1}\|_{\ell_2}})$ and for the second attention layer we write $f_2(\tilde{\boldsymbol{x}}) = \boldsymbol{X}_{tr}^T \mathbb{S}(\boldsymbol{X}_{tr}\boldsymbol{W}_2\tilde{\boldsymbol{x}})$.

Theorem 2 (TS-R Problem) *Consider the TS-R problem as described in Section 3, Definition 1. Moreover, assume the setting, assumptions, and the attention model above. That is, given a time series of length L, i.e. $\boldsymbol{x} \in \mathbb{R}^L$, that is patched and ending with motif $\boldsymbol{x}_{L-C+1}$ of size C and has a unique matching motif in the time series, followed by Υ of size $C = H$. Moreover:*

1. *Set $\boldsymbol{W}_1 = c.\boldsymbol{\Phi}$*
2. *Set $\boldsymbol{W}_2 = c.\boldsymbol{\Phi}^\perp \boldsymbol{R} \boldsymbol{\Phi}^\perp$*

As $c \to \infty$, we have $\boldsymbol{g}(f_2(f_1(x_{L-C+1}))) \to \Upsilon$.

Proof 1 *Realize that with* $\boldsymbol{W}_1$*, the positional encodings will be ignored and only token embeddings will remain. Moreover,* $\boldsymbol{N}\boldsymbol{X}_{tr}\boldsymbol{\Phi}\frac{x_{L-C+1}}{\|x_{L-C+1}\|_{\ell_2}}$ *will return a vector* $\boldsymbol{v}$ *of size* $L-C$ *with the largest element at* j*, for index* j*, corresponding to the matching retrieval motif. As* $c\to\infty$*, the softmax will be saturated. Thus,* $\mathbb{S}(c.\boldsymbol{v})=\boldsymbol{e}_j$ *as* $c\to\infty$*. We get,* $X_{tr}\boldsymbol{e}_j=x_j$*. In the second layer, realize that due to* $\boldsymbol{\Phi}^\perp$*, this time token embeddings will be ignored and* $\boldsymbol{R}$ *aligns the token embeddings so that the future context of the retrieved element index can be returned. That is,* $\boldsymbol{X}_{tr}\boldsymbol{\Phi}^\perp\boldsymbol{R}\boldsymbol{\Phi}^\perp x_j$ *will return a vector* $\bar{\boldsymbol{v}}$ *of size* $L-C$ *with the largest element at* $j+C$*. As* $c\to\infty$*, the softmax will be saturated, resulting in* $\mathbb{S}(c.\bar{\boldsymbol{v}})=\boldsymbol{e}_{j+C}$*. Hence, we get* $X_{tr}\boldsymbol{e}_{j+C}=x_{j+C}$*. Note that* $\boldsymbol{g}(x_{j+C})=\Upsilon$*, completing the proof.*

This theorem provides a construction of a 2-layer attention model that retrieves Υ.

A.2 Synthetic Retrieval Experiment

Here, we provide details about our synthetic retrieval experiment, as illustrated in Figure 2. We introduce randomness in our experimental setup through $\mathbf{Q}$, a randomly sampled orthonormal matrix for each $\boldsymbol{s}$. Consequently, each data generation process involves L^2 learnable parameters, ensuring that for time series of length $C\leq L$, the data remains essentially random for Chronos. This effect is clearly observable in the non-RAF results presented in Figure 2. However, when RAF is employed, even the smaller Chronos models with retrieval capabilities exhibit significantly enhanced performance, despite the context length being relatively small compared to L^2.

Note that since Chronos is stochastic, we sampled forecasts from Chronos 20 times and took their mean values for each query, an approach also suggested by the Chronos paper. We assessed the retrieval performances under varying signal-to-noise ratios, as depicted in Figure 2. For each signal, we sampled many time series signals and averaged the metrics for all of them. From Figure 2, it is clear why Chronos-base outperforms Chronos-mini on retrieval tasks. In noisy settings, we generally observe that larger models perform retrieval better than smaller models, a trend also evident in experiments with real data.

B Experimental Details

B.1 Experimental Setup for Fine-tuning the Chronos Models

Table 5: Summary of hyperparameter settings used for Chronos fine-tuning

Hyper-parameter Name	**Baseline FT**	**Advanced RAF**
Database Formation/Validation/Testing	0.7 / 0.1 / 0.2	0.7 / 0.1 / 0.2
Chronos Models	Mini, Base	Mini, Base
Prediction Length	10	10
Context Length	75	160
Minimum Past	30	60
Number of Epoch	400 (Mini)/ 1000 (Base)	400 (Mini)/ 1000 (Base)
Learning Rate	0.00001	0.00001
Number of Generated Samples	20	20
LR Scheduler	Linear	Linear
Optimizer	AdamW	AdamW
Gradient Accumulation Steps	1	1
Dropout Probability	0.2	0.2

The experimental details for the fine-tuning process are given in Table 5. The table provides a summary of the hyperparameter settings used for fine-tuning two approaches: Baseline FT and Advanced RAF. Both approaches employ a data split of 70% for database formation (used only for Advanced RAF), 10% for validation on which the approaches are fine-tuned, and 20% for testing. The Chronos Mini and Chronos Base models are used in both setups, with a prediction length of 10. However, there are key differences between the two approaches: the Baseline FT approach uses a context length of 75, while the Advanced RAF uses a longer context length of 160 due to the concatenation of the retrieved context and retrieved future. Still, the approaches are evaluated on the same samples during the test time. Additionally, the minimum past time steps considered are 30 for Baseline FT

and 60 for Advanced RAF. This represents the minimum number of time steps that must be retained prior to the introduction of NaN values during the training process.

Despite these differences, other hyperparameters remain consistent between the two approaches. Both approaches fine-tune Chronos Mini for 400 epochs and Chronos Base for 1000 epochs, with a learning rate of 0.00001. Each approach generates 20 samples during fine-tuning, as maintained in Ansari et al. (2024), and employs a linear learning rate scheduler. The optimizer for both of them is AdamW, which incorporates weight decay for regularization. Gradient accumulation occurs after each step (set to 1), and a dropout probability of 0.2 is applied to both approaches. The experiments were conducted using NVIDIA A100 40GB and L40S 48GB GPUs. Finally, for the reproducibility of the results, the seed for dataset splitting is fixed at 42 throughout every experiment.

B.2 Experimental Setup for other TSFMs

We evaluate three pretrained TSFMs using a common retrieval pipeline in which *all* retrieval embeddings are produced by the Chronos encoder for consistency. For Moirai , we instantiate the Mixture-of-Experts checkpoint `Salesforce/moirai-1.1-R-large` with automatic patching (`patch_size=auto`), and sample 20 trajectories per query; dynamic feature dimensions are passed directly from the dataset wrapper. For Lag-Llama and TimesFM, we load the official checkpoints (Lag-Llama from the authors' repository; TimesFM via `google/timesfm-2.0-500m-pytorch`) and apply them in zero-shot mode, again conditioning on the same Chronos-derived retrieval embeddings. Unless stated otherwise, we do not perform additional fine-tuning on these TSFMs; all models are evaluated on the common test windows used throughout the paper.

B.3 Experimental Setup for Non-Transformer Baselines

For DLinear, where no public checkpoint was available, we fine-tuned the model on each dataset's validation split for 100 epochs with a learning rate of 10^{-3} and then used the resulting checkpoint for comparing RAF and the baseline. This process was done separately for each dataset we evaluated. For GBDT (LightGBM), we constructed lag features $\{1, 2, 3, 7, 14, 28\}$ from the validation split and trained a single multi-output regressor per dataset with `n_estimators`= 1000 and `max_depth`= 64 while keeping other settings at defaults. For ARIMA, we fit one model per series using StatsForecast AutoARIMA with season length inferred from the data frequency.

C Ablation Study on Retrieval Database Formation

In many real-world forecasting applications, the target dataset may contain too few series to assemble an effective retrieval database solely from in-dataset examples. To address this limitation, we evaluate two alternative retrieval strategies: (i) drawing the best-matching motif from a different dataset within the same domain (same-domain retrieval), and (ii) using a fully cross-domain dataset. This comparison allows us to determine whether related datasets can serve as practical substitutes when in-dataset retrieval is infeasible, and to quantify the performance trade-offs as domain alignment weakens.

Table 6: Performance of RAF on the *Traffic* dataset with prediction length $H = 15$, evaluated across four context lengths using various retrieval databases. Traffic and Uber TLC both belong to the transportation domain (same domain), whereas NN5 originates from the finance domain (cross-domain). Datasets achieving the **first** and second best scores have been highlighted.

	Baseline (no retrieval)		**RAF-Traffic** (same dataset)		**RAF-Uber TLC** (same domain)		**RAF-NN5** (cross domain)	
Context C	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
$C_1 = 50$	0.233	2.162	**0.213**	**1.852**	0.229	2.081	0.242	2.201
$C_2 = 75$	0.228	2.914	**0.215**	**2.249**	0.217	2.863	0.226	2.915
$C_3 = 100$	0.204	2.234	0.199	2.225	**0.194**	**2.166**	0.199	2.197
$C_4 = 150$	0.215	1.872	**0.200**	**1.752**	0.202	1.842	0.203	1.823

Table 7: Performance of RAF on the *CIF-2016* dataset with prediction length $H = 4$, evaluated across four context lengths using various retrieval databases. CIF-2016 and FRED-MD both belong to the finance domain (same domain), whereas Tourism (M.) originates from the tourism domain (cross-domain). Datasets achieving the **first** and second best scores have been highlighted.

	Baseline (no retrieval)		**RAF-CIF- 2016** (same dataset)		**RAF-FRED-MD** (same domain)		**RAF-Tourism (M.)** (cross domain)	
Context C	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
$C_1 = 10$	0.063	1.162	0.062	1.256	**0.061**	**1.081**	0.068	1.357
$C_2 = 15$	0.043	1.412	**0.041**	1.360	0.041	**1.338**	0.044	1.391
$C_3 = 18$	0.047	0.907	**0.039**	0.862	0.043	**0.709**	0.046	0.963
$C_4 = 21$	0.040	0.954	**0.038**	**0.692**	0.045	0.767	0.055	0.947

Table 6 shows that retrieval from the *same dataset* yields the largest performance gains. However, when in-dataset retrieval is unavailable, a *different dataset within the same domain* still provides benefits—reducing WQL by approximately 4% and MASE by around 2.5%—whereas *cross-domain* retrieval offers only marginal improvements (about 1% in WQL and under 1% in MASE) and can even degrade accuracy at shorter contexts.

Similarly, Table 7 reports the results on CIF-2016. Here, retrieval from FRED-MD (same domain) matches or slightly exceeds in-dataset CIF-2016 retrieval in most settings, confirming FRED-MD as a viable alternative. In contrast, Tourism (cross-domain) yields only modest gains and, at certain context lengths, underperforms the baseline, underscoring the limited value of cross-domain augmentation when domain characteristics diverge.

Together, these findings imply that, in the absence of sufficient in-dataset series, selecting another dataset from the same domain (e.g. Uber TLC for Traffic, or FRED-MD for CIF-2016) offers a reliable surrogate that outperforms fully cross-domain sources (NN5 or Tourism). This trend reflects the degree of contextual alignment: same-dataset corpora capture both task-specific patterns and domain characteristics; same-domain different-dataset corpora preserve domain structure but lack exact task alignment; and cross-domain corpora share neither.

D Analysis of Retrieval Components

This section evaluates how each element of RAF influences overall performance.

D.1 Choice of Similarity Metric

We compare several similarity metrics, namely the L2 norm, the L1 norm, cosine similarity, and Pearson's correlation, to evaluate their performance under different context and prediction lengths. The L2 norm and L1 norm measure dissimilarity by summing over point-wise differences of the time-series embeddings in slightly different manners, while cosine similarity captures the orientation between the vectors, and Pearson's correlation quantifies their linear dependence. For the Benchmark I datasets (see Table 8), which involve a longer context and a prediction length, the performance of these metrics is mixed; although L2 and Pearson's correlation tend to perform better overall, the optimal choice ultimately depends on the specific dataset under consideration. In contrast, for the Benchmark II datasets (see Table 9), where the context and prediction length are shorter, cosine similarity and Pearson's correlation generally outperform the other metrics, indicating a higher accuracy in these shorter-horizon scenarios.

Table 8: Comparison of different similarity metrics for the selected Benchmark I datasets when the prediction length $H = 10$ and context length $C = 75$.

	L2 Norm		L1 Norm		Cosine Sim.		Pearson's Corr.	
Dataset	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
ETTh1	0.040	**0.625**	0.039	0.706	**0.034**	0.660	0.039	0.761
FRED-MD	0.019	0.500	0.011	0.501	0.012	**0.488**	**0.009**	0.494
NN5	**0.134**	**0.417**	0.137	0.426	0.138	0.430	0.138	0.431
Covid Deaths	0.006	5.124	0.006	5.312	0.006	5.183	**0.005**	**5.092**

Table 9: Comparison of different similarity metrics for the selected Benchmark II datasets when the prediction length $H = 4$ and context length $C = 18$.

	L2 Norm		L1 Norm		Cosine Sim.		Pearson's Corr.	
Dataset	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
Tourism (Q.)	**0.091**	1.054	0.094	1.071	0.098	1.056	0.098	**1.043**
M1	0.165	1.064	0.164	1.056	**0.161**	1.054	0.162	**1.042**
Uber TLC	0.163	0.983	0.160	0.982	**0.159**	0.969	0.160	**0.964**
CIF-2016	**0.039**	0.862	0.039	0.873	0.040	**0.758**	0.041	0.858

D.2 Ablation Study on Retrieval

Tables 10 and 11 present an ablation study comparing five retrieval variants: (1) *RAF* (ours), which retrieves the most relevant segment using the learned similarity and concatenates it with alignment; (2) *Retrieval w/o Alignment*, which performs the same retrieval but *pastes the window without enforcing this boundary alignment*; (3) *Retrieval w/o R.F.*, which retrieves only the *matching context motif* without the retrieved future segment; (4) *Random Retrieval*, where key patches are drawn at random; and (5) *Baseline (w/o Retrieval)*, which performs no retrieval.

From the results, random retrieval is typically worse than no retrieval at all, as seen in higher WQL/MASE across both Benchmark I (Table 10) and Benchmark II (Table 11), underscoring that *relevance* is crucial. Removing the retrieved future (w/o R.F.) consistently degrades performance relative to RAF—often approaching the baseline—indicating that supplying the model with the neighbor's immediate *future* (not just its past motif) is an important driver of the gains.

Likewise, dropping alignment generally hurts—or at best yields comparable—performance in several Benchmark I datasets, as the unaligned paste introduces a boundary discontinuity that weakens retrieval. By contrast, in

Table 10: Ablation study on retrieval mechanism for selected Benchmark I datasets when the prediction length $H = 10$ and context length $C = 75$.

	ETTh1		FRED-MD		NN5		Covid Deaths	
Method	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
RAF	**0.040**	**0.625**	**0.019**	**0.500**	**0.134**	0.417	0.006	**5.124**
Retrieval w/o Alignment	0.041	0.659	0.021	0.510	0.136	**0.414**	0.009	6.336
Retrieval w/o R.F.	0.064	0.787	0.103	0.539	0.225	0.608	**0.004**	6.051
Random Retrieval	0.125	1.171	0.074	0.577	0.155	0.476	0.010	9.146
Baseline	0.074	0.800	0.112	0.577	0.154	0.456	0.007	5.492

Table 11: Ablation study on retrieval mechanism for selected Benchmark II datasets when the prediction length $H = 4$ and context length $C = 18$.

	Tourism (Q.)		M1		Uber TLC		CIF-2016	
Method	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
RAF	0.091	1.054	0.165	1.063	0.163	0.983	**0.039**	**0.862**
Retrieval w/o Alignment	**0.080**	**1.048**	**0.158**	**0.952**	**0.153**	**0.922**	0.050	1.027
Retrieval w/o R.F.	0.126	1.364	0.168	1.171	0.187	1.074	0.045	0.882
Random Retrieval	0.096	1.282	0.189	1.255	0.218	1.195	0.045	1.093
Baseline	0.092	1.072	0.167	1.065	0.194	1.093	0.047	0.954

multiple Benchmark II datasets, enforcing alignment slightly backfires, suggesting that with shorter context lengths, alignment can be treated as a design choice rather than a required component of the pipeline. Overall, both components—alignment and inclusion of the retrieved future—contribute to RAF's improvements.

D.3 Choice of Number of Retrievals k

In Table 12, one of the key hyperparameters analyzed is k, which represents the number of retrieved examples. In our setup, the retrieved examples are concatenated before the original query, thereby extending the model's input context. As can be observed, increasing k leads to a slight improvement in performance. However, it is important to note that these experiments were conducted in a univariate setting; we believe that employing multivariate models, especially for this setup, may yield even more pronounced benefits.

Table 12: Effect of the number of retrievals (k) on performance or selected Benchmark I datasets when the prediction length $H = 10$ and context length $C = 100$.

	k=1		k=2		k=5		k=10	
Dataset	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
ETTh1	**0.025**	0.551	0.033	**0.548**	0.033	0.549	0.033	0.550
FRED-MD	0.074	0.572	0.037	0.554	0.035	0.548	**0.024**	**0.500**
NN5	0.145	0.432	0.133	0.419	**0.130**	**0.404**	0.137	0.423
Covid Deaths	0.004	5.293	0.004	5.564	0.004	5.294	**0.004**	**5.200**

D.4 Retrieval-Only Baselines

To isolate the contribution of retrieval itself, we evaluate three *retrieval-only* baselines that do not use a TSFM. Given a query context of length C, each method retrieves the closest segment(s) from a training bank and *copies their immediate future* as the forecast for horizon H. For $k > 1$, we aggregate the k futures using rank–order centroid (ROC) weights, which reduce sensitivity to exact neighbor ordering while keeping a simple, parameter-free scheme. This design answers a narrow question: *how far can pure match-and-copy go without a learnable forecaster?*

KNN. Our KNN baseline computes z-normalized ℓ_2 distance over the context window. As expected, k=1 yields sharp but volatile predictions, while larger k averages nearby regimes and reduces variance. Across datasets, KNN performs reasonably when local dynamics are stationary and scale-aligned; however, misalignments degrade accuracy. For instance, performance degrades significantly on `Covid Deaths`, which exhibits sharp and rapidly changing dynamics.

Dynamic Time Warping (DTW) and Matrix Profile (MP). Dynamic Time Warping (DTW) (Sakoe and Chiba, 2003; Berndt and Clifford, 1994) relaxes strict phase alignment by allowing local time warps within the context, which helps when similar shapes occur at slightly different times or speeds. In our results, DTW narrows the gap to RAF on datasets with misaligned seasonality, yet the gains become inconsistent when noise dominates or when the future depends on covariates that are not visible in the context. The Matrix Profile (MP) (Yeh et al., 2016; Zhu et al., 2016) baseline performs an AB-join via z-normalized cross-correlation and retrieves globally similar motifs. It works well for motif-rich series where shape similarity matters more than exact timing, and in those cases it can match or exceed DTW. Nevertheless, both DTW and MP ultimately copy a neighbor's future, and copying proves brittle under distribution shift or when the next steps depend on exogenous drivers.

Table 13: Retrieval-only baselines for H=10, C=75 on Benchmark I datasets. RAF with Chronos-Base is compared to KNN/DTW/MP with $k \in \{1, 5, 10\}$.

	RAF		**KNN** (k=1)		**KNN** (k=5)		**KNN** (k=10)		**DTW** (k=1)	
Dataset	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
`ETTh1`	**0.040**	**0.625**	0.463	3.834	0.472	3.918	0.472	3.905	0.537	4.497
`FRED-MD`	**0.019**	**0.500**	0.404	1.543	0.409	1.543	0.418	1.554	2.957	0.866
`NN5`	**0.134**	**0.417**	0.296	0.775	0.296	0.775	0.296	0.775	1.062	2.775
`Covid Deaths`	**0.006**	**5.124**	0.094	26.832	0.094	26.744	0.095	26.705	0.112	26.928
	DTW (k=5)		**DTW** (k=10)		**MP** (k=1)		**MP** (k=5)		**MP** (k=10)	
Dataset	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
`ETTh1`	0.298	2.092	0.304	2.162	0.339	2.639	0.348	2.745	0.365	2.927
`FRED-MD`	0.505	2.151	0.521	2.140	0.611	1.563	0.612	3.837	0.808	7.394
`NN5`	0.415	1.070	0.438	1.130	2.258	5.500	1.749	4.288	1.672	4.120
`Covid Deaths`	0.190	41.377	0.179	39.820	0.028	25.852	0.059	26.877	0.048	25.544

As Tables 13 and 14 demonstrate, pure retrieval is a strong *match-and-copy* heuristic, but it cannot infer unseen dynamics, model uncertainty, or integrate contextual signals. In contrast, RAF uses retrieval to provide informative priors while letting a TSFM learn how to transform them—rescaling, re-timing, and combining retrieved evidence with the model's internal dynamics. The consistent gaps between RAF and retrieval-only baselines indicate that retrieval without a model is a poor substitute for retrieval with a model.

Table 14: Retrieval-only baselines for H=4, C=18 on Benchmark II datasets. RAF with Chronos-Base is compared to KNN/DTW/MP with $k \in \{1, 5, 10\}$.

	RAF		**KNN** (k=1)		**KNN** (k=5)		**KNN** (k=10)		**DTW** (k=1)	
Dataset	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
Tourism (Q.)	**0.091**	**1.054**	0.322	4.297	0.295	3.777	0.283	3.569	0.280	3.077
M1	**0.165**	**1.064**	0.200	1.444	0.211	1.367	0.214	1.354	0.195	1.282
Uber TLC	**0.163**	**0.983**	0.346	1.766	0.334	1.743	0.331	1.739	0.271	1.335
CIF-2016	**0.039**	**0.862**	0.061	1.204	0.058	1.167	0.056	1.159	0.081	1.276
	DTW (k=5)		**DTW** (k=10)		**MP** (k=1)		**MP** (k=5)		**MP** (k=10)	
Dataset	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
Tourism (Q.)	0.394	4.232	0.342	3.802	0.722	6.975	2.022	16.560	2.315	18.814
M1	0.263	1.834	0.217	1.507	0.337	2.568	0.319	4.201	0.341	5.001
Uber TLC	0.428	2.120	0.402	1.972	0.634	3.131	0.525	2.685	0.516	2.678
CIF-2016	0.275	1.873	0.220	1.576	0.087	2.337	0.357	9.510	0.414	11.137

D.5 Matched Input-Length Comparison

A natural question is whether RAF's gains simply come from seeing more input tokens. RAF augments the input with retrieved evidence: the model is fed the original context of length C, plus a retrieved context (length C) and retrieved future (length H), i.e., total sequence length $2C + H$. To rule out the possibility that a baseline given the same number of input tokens might close the gap by simply using more past history, we introduce a **Long-Context Baseline** that uses only additional past observations from the same series to match RAF's total input length. Concretely, for each (C, H), the baseline context length is set to $L = 2C + H$ (e.g., $C = 50, H = 10 \Rightarrow L = 110$), so the TSFM receives the same length input as RAF but without retrieval. We evaluate this on Benchmark I, which contains long histories, using Chronos-Base.

Table 15: Matched input-length comparison on Benchmark I datasets for prediction length $H = 10$. Each RAF row is paired with a Long-Context Baseline (L.C.B.) row whose input length $L = 2C + H$ matches RAF's total token budget.

		Weather		**ETTh1**		**FRED-MD**		**Covid Deaths**		**NN5**	
Method	**Length**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
RAF	50	**0.152**	**1.247**	**0.041**	**0.741**	**0.018**	0.513	0.005	5.713	0.170	0.514
	75	**0.151**	1.200	**0.040**	**0.625**	**0.019**	0.500	**0.006**	**5.124**	**0.134**	**0.417**
	100	**0.155**	1.209	**0.025**	**0.551**	0.074	0.572	**0.004**	**5.293**	**0.145**	0.432
	150	**0.155**	**1.068**	0.038	**0.543**	0.031	**0.369**	0.010	**5.301**	**0.127**	**0.389**
L.-C. B.	110	0.158	1.544	0.075	0.858	0.109	0.608	**0.004**	9.937	**0.167**	**0.500**
	160	0.180	**1.197**	0.084	0.753	0.065	**0.492**	0.010	9.062	0.150	0.421
	210	0.178	**1.263**	0.087	0.763	**0.075**	**0.585**	0.010	10.615	0.150	**0.423**
	310	0.179	1.164	**0.082**	0.819	**0.025**	0.413	-	-	0.150	0.445

Table 16: Matched input-length comparison on Benchmark I datasets for prediction length $H = 20$. Each RAF row is paired with a Long-Context Baseline (L.C.B.) row whose input length $L = 2C + H$ matches RAF's total token budget.

		Weather		ETTh1		FRED-MD		Covid Deaths		NN5	
Method	**Length**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**	**WQL**	**MASE**
RAF	50	**0.164**	1.909	0.075	0.885	0.093	0.808	0.014	12.462	**0.186**	0.646
	75	**0.165**	**1.924**	**0.047**	**0.867**	**0.083**	**0.643**	0.012	21.807	**0.155**	**0.538**
	100	**0.168**	**1.807**	**0.044**	**0.835**	0.125	0.582	**0.008**	**17.229**	**0.150**	**0.487**
	150	**0.175**	**1.175**	0.055	**0.714**	**0.040**	**0.400**	**0.009**	**15.990**	0.162	0.538
L.-C. B.	120	0.172	**1.934**	**0.130**	**1.152**	**0.163**	**0.747**	**0.010**	15.394	0.212	**0.626**
	170	0.176	2.762	0.123	0.942	0.094	0.844	**0.010**	**16.357**	0.194	0.584
	220	0.173	1.852	0.108	0.917	**0.066**	**0.590**	0.012	20.087	0.185	0.567
	320	0.211	1.203	**0.074**	0.805	0.033	0.452	-	-	**0.181**	**0.579**

As can be seen, RAF yields consistent relative improvements over the long-context baseline. We summarize the matched length gains using the average percent change, computed as (RAF − Baseline)/Baseline × 100. For prediction length $H = 10$, RAF achieves an average percent change of $-\mathbf{15.54}\%$ in MASE and $-\mathbf{26.47}\%$ in WQL (relative reductions of 15.54% and 26.47%). For $H = 20$, the average percent change is $-\mathbf{7.48}\%$ in MASE and $-\mathbf{10.48}\%$ in WQL (relative reductions of 7.48% and 10.48%).

These results indicate that, even under the same token budget, simply extending the past context may add *irrelevant* history and does not ensure exposure to the specific motif needed for the next prediction window. RAF instead injects targeted evidence by retrieving a similar motif and its continuation, which can be substantially more informative. Moreover, for Benchmark II / short series regimes, longer history is often not available, so "just increase context" is not a viable alternative.

E Results for RAF with Fine-tuning

Chronos Mini and Chronos Base were fine-tuned separately on each Benchmark I dataset (without cross-dataset mixing) and subsequently tested on the same datasets used for fine-tuning. Chronos Mini ran for 400 epochs and Chronos Base for 1000 epochs. We used a dataset-agnostic schedule with an initial learning rate of 1×10^{-5}, linearly decayed to 0. The context and horizon were fixed to $C = 75$ and $H = 10$.

We compare three regimes: (i) *Baseline FT* (no retrieval), (ii) *Naïve RAF* (retrieval-augmented inputs, weights frozen), and (iii) *Advanced RAF* (retrieval-augmented inputs *and* fine-tuned weights).

Table 17: Comparative analysis of Chronos Mini and Chronos Base models fine-tuned on Benchmark I datasets, evaluated with and without time series retrieval. Prediction and context lengths are set at $H = 10$ and $C = 75$, respectively. Advanced RAF refers to RAF with fine-tuning.

Approach		**Baseline**		**Naive RAF**		**Baseline FT**		**Advanced RAF**	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
Chronos Mini	Weather	0.170	1.308	0.166	1.265	0.163	1.200	**0.159**	**1.176**
	Traffic	0.234	1.561	0.225	1.524	0.157	1.013	**0.154**	**0.930**
	ETTh1	0.089	0.893	0.079	1.020	0.081	0.800	**0.073**	**0.736**
	FRED-MD	0.085	0.592	0.055	0.582	0.028	0.689	**0.019**	**0.566**
	Covid Deaths	0.007	8.765	0.009	9.229	**0.005**	**8.620**	0.008	8.910
	NN5	0.217	0.680	0.175	0.563	0.152	0.460	**0.125**	**0.401**
Chronos Base	Weather	0.154	1.226	0.151	1.200	**0.149**	1.151	0.150	**1.131**
	Traffic	0.171	1.443	0.184	1.608	**0.151**	**1.237**	0.160	1.331
	ETTh1	0.074	0.800	0.040	0.625	0.072	0.759	**0.036**	**0.580**
	FRED-MD	0.112	0.577	0.019	0.500	0.077	0.552	**0.017**	**0.475**
	Covid Deaths	0.006	5.492	0.006	**5.124**	0.003	8.793	**0.002**	8.275
	NN5	0.156	0.524	0.135	0.481	0.129	0.386	**0.115**	**0.378**

Table 17 shows that *Advanced RAF* typically delivers the best WQL/MASE scores. For Chronos Mini, for example, Weather improves from WQL 0.163→0.159 and MASE 1.200→1.176, and ETTh1 from 0.081→0.073 and 0.800→0.736. For Chronos Base, ETTh1 drops from 0.072→0.036 (WQL) and 0.759→0.580 (MASE), while FRED-MD improves from 0.077→0.017 and 0.552→0.475. Traffic is a mild counterexample: for Chronos Base, its MASE increases slightly (1.237→1.331). Together with the Covid Deaths case, this suggests that when the retrieved motifs are weak or poorly aligned with the target series, retrieval can add noise, and fine-tuning may not help.

Nevertheless, across the remaining datasets, Advanced RAF is consistently best, and Naïve RAF almost always lies between Baseline FT and Advanced RAF—indicating that (i) adding retrieved context is beneficial, and (ii) end-to-end adaptation is what unlocks the full gain.

F Extended Results for Chronos Models

F.1 Chronos Mini Results on Benchmark I

Table 18: Performance of RAF on Benchmark I when the prediction length $H = 10$ with context lengths, $C \in \{50, 75, 100, 150\}$

Datasets		Weather		Traffic		ETTh1		FRED-MD		Covid Deaths		NN5 (Daily)	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	50	**0.168**	**1.341**	0.223	1.434	**0.095**	0.958	**0.095**	**0.577**	0.011	**6.380**	0.221	0.713
	75	**0.166**	**1.265**	**0.225**	**1.524**	**0.079**	1.020	**0.055**	**0.582**	0.009	9.229	**0.174**	**0.563**
	100	**0.165**	**1.237**	**0.219**	**1.677**	**0.059**	**0.756**	**0.087**	**0.592**	**0.008**	8.309	**0.181**	**0.543**
	150	**0.160**	1.072	**0.198**	**1.285**	**0.079**	0.807	**0.046**	**0.407**	**0.006**	8.572	**0.187**	**0.561**
Baseline	50	0.170	1.380	**0.215**	**1.183**	0.099	**0.843**	0.114	0.629	**0.010**	6.647	**0.203**	**0.645**
	75	0.170	1.308	0.234	1.561	0.089	**0.893**	0.085	0.592	**0.007**	**8.765**	0.217	0.680
	100	0.169	1.289	0.230	1.748	0.083	0.905	0.091	0.598	0.010	**7.602**	0.204	0.635
	150	0.162	**1.049**	0.223	1.446	0.087	**0.801**	0.056	0.413	0.010	**8.337**	0.209	0.627

Table 19: Performance of RAF on Benchmark I when the prediction length $H = 15$ with context lengths, $C \in \{50, 75, 100, 150\}$

Datasets		Weather		Traffic		ETTh1		FRED-MD		Covid Deaths		NN5 (Daily)	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	50	**0.166**	**1.823**	**0.243**	2.096	**0.167**	**1.378**	**0.176**	**0.866**	**0.012**	14.968	**0.183**	**0.556**
	75	**0.165**	**1.562**	**0.234**	**2.467**	**0.159**	**1.490**	**0.099**	**0.673**	**0.007**	**14.549**	**0.163**	**0.514**
	100	**0.167**	1.472	**0.248**	**2.803**	**0.163**	**1.621**	**0.073**	**0.754**	**0.007**	**9.794**	**0.183**	**0.599**
	150	**0.163**	1.116	**0.262**	**2.242**	**0.095**	**0.861**	0.061	**0.449**	**0.008**	9.948	0.179	0.564
Baseline	50	0.173	1.895	0.243	**1.998**	0.169	1.406	0.211	0.936	0.014	**12.324**	0.193	0.619
	75	0.172	1.585	0.257	2.629	0.183	1.663	0.164	0.798	0.013	14.701	0.204	0.645
	100	0.172	**1.442**	0.263	2.810	0.194	1.827	0.131	0.771	0.007	9.811	0.185	0.606
	150	0.167	**1.082**	0.264	2.267	0.232	1.613	**0.045**	0.483	0.012	**9.862**	**0.173**	**0.523**

Table 20: Performance of RAF on Benchmark I when the prediction length $H = 20$ with context lengths, $C \in \{50, 75, 100, 150\}$

Datasets		Weather		Traffic		ETTh1		FRED-MD		Covid Deaths		NN5 (Daily)	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	50	**0.171**	**1.867**	**0.296**	2.813	**0.112**	**1.071**	**0.222**	**0.939**	**0.009**	**14.943**	**0.226**	**0.824**
	75	**0.169**	**1.898**	**0.292**	**2.911**	**0.143**	**1.426**	**0.167**	**0.792**	**0.010**	**20.984**	0.330	1.139
	100	**0.169**	1.805	**0.302**	**3.915**	**0.063**	**1.125**	0.178	**0.707**	**0.011**	17.239	0.236	0.782
	150	**0.165**	1.095	**0.278**	**2.391**	**0.151**	**1.269**	**0.099**	**0.468**	**0.011**	**15.954**	0.231	0.728
Baseline	50	0.176	1.911	0.301	**2.801**	0.176	1.500	0.247	1.254	0.021	18.115	0.239	0.838
	75	0.176	1.924	0.307	3.012	0.154	1.550	0.242	0.982	0.010	23.256	**0.249**	**0.861**
	100	0.176	**1.789**	0.309	4.003	0.155	1.640	**0.169**	0.802	0.013	**16.956**	**0.226**	**0.767**
	150	0.170	**1.080**	0.295	2.461	0.170	1.344	0.109	0.595	0.014	16.003	**0.197**	**0.641**

F.2 Chronos Mini Results on Benchmark II

Table 21: Performance of RAF on Benchmark II when the prediction length $H = 3$ with context lengths, $C \in \{10, 15, 18, 21\}$

Datasets		Tourism (M.)		Tourism (Q.)		M1 (M.)		Uber TLC		CIF-2016	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	10	**0.115**	**0.737**	**0.131**	**1.729**	**0.176**	**1.199**	**0.250**	**1.108**	**0.053**	1.211
	15	0.682	2.252	**0.105**	**1.402**	0.192	1.362	**0.177**	**1.223**	0.053	**1.223**
	18	**0.171**	**1.627**	**0.100**	**1.072**	**0.182**	**1.102**	**0.172**	**0.809**	0.051	**0.747**
	21	**0.075**	**1.233**	**0.095**	**1.214**	**0.162**	**1.037**	**0.162**	**1.034**	**0.043**	**0.735**
Baseline	10	0.744	0.932	0.223	3.162	0.225	1.217	0.311	1.188	0.053	**1.079**
	15	**0.677**	**2.178**	0.105	1.596	**0.184**	**1.266**	0.210	1.302	**0.053**	1.255
	18	0.596	1.714	0.105	1.310	0.185	1.118	0.209	0.850	**0.045**	0.924
	21	0.513	1.446	0.103	1.293	0.177	1.071	0.171	1.110	0.052	0.836

Table 22: Performance of RAF on Benchmark II when the prediction length $H = 4$ with context lengths, $C \in \{10, 15, 18, 21\}$

Datasets		Tourism (M.)		Tourism (Q.)		M1 (M.)		Uber TLC		CIF-2016	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	10	**0.161**	**0.755**	**0.122**	**2.331**	**0.194**	**1.268**	**0.201**	**1.271**	**0.040**	1.258
	15	0.562	2.436	**0.088**	**1.155**	**0.178**	**1.324**	**0.224**	**1.155**	**0.045**	**1.163**
	18	**0.248**	**1.539**	**0.081**	**0.972**	**0.169**	1.121	**0.188**	**1.117**	0.043	**0.780**
	21	**0.088**	1.284	**0.085**	**1.009**	0.185	**1.090**	**0.170**	1.067	**0.039**	**0.896**
Baseline	10	0.594	0.875	0.200	2.913	0.198	1.295	0.251	1.465	0.049	**1.102**
	15	**0.557**	**2.227**	0.140	1.445	0.179	1.328	0.226	1.234	0.053	1.457
	18	0.279	1.592	0.130	1.210	0.177	**1.083**	0.218	1.241	**0.038**	0.935
	21	0.238	**1.231**	0.090	1.090	**0.176**	1.126	0.177	**1.044**	0.040	0.981

Table 23: Performance of RAF on Benchmark II when the prediction length $H = 5$ with context lengths, $C \in \{10, 15, 18, 21\}$

Datasets		Tourism (M.)		Tourism (Q.)		M1 (M.)		Uber TLC		CIF-2016	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	10	**0.294**	**0.898**	**0.185**	**2.897**	**0.154**	**1.364**	**0.210**	**1.323**	0.072	1.308
	15	0.483	2.545	**0.083**	**1.158**	**0.139**	**1.287**	**0.149**	**0.957**	**0.055**	**1.153**
	18	**0.110**	**1.722**	**0.088**	**1.095**	**0.128**	**1.138**	**0.159**	**1.023**	**0.074**	**0.908**
	21	**0.094**	1.296	**0.087**	1.179	**0.119**	**1.092**	**0.146**	**0.867**	**0.075**	0.971
Baseline	10	0.502	0.987	0.220	3.125	0.178	1.386	0.251	1.453	**0.067**	**1.226**
	15	**0.397**	**2.430**	0.156	1.489	0.141	1.304	0.208	1.167	0.075	1.157
	18	0.197	1.903	0.142	1.268	0.140	1.264	0.184	1.184	0.090	1.030
	21	0.111	**1.275**	0.092	**1.106**	0.127	1.160	0.152	0.979	0.106	**0.963**

F.3 Chronos Base Results on Benchmark I

Table 24: Performance of RAF on Benchmark I when the prediction length $H = 10$ with context lengths, $C \in \{50, 75, 100, 150\}$

Datasets		Weather		Traffic		ETTh1		FRED-MD		Covid Deaths		NN5 (Daily)	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	50	**0.152**	**1.247**	**0.178**	**1.789**	**0.041**	**0.741**	**0.018**	**0.513**	**0.005**	**5.713**	**0.170**	**0.514**
	75	**0.151**	**1.200**	0.183	1.608	**0.040**	**0.625**	**0.019**	**0.500**	**0.006**	**5.124**	**0.134**	**0.417**
	100	0.155	**1.209**	**0.194**	**2.479**	**0.025**	**0.551**	**0.074**	**0.572**	**0.004**	**5.293**	**0.145**	**0.432**
	150	**0.155**	**1.068**	**0.193**	2.014	**0.038**	**0.543**	0.031	**0.369**	0.010	**5.301**	**0.127**	0.389
Baseline	50	0.155	1.322	0.205	1.976	0.090	0.799	0.113	0.647	0.012	6.220	0.181	0.520
	75	0.154	1.226	**0.171**	**1.443**	0.074	0.800	0.112	0.577	0.007	5.492	0.154	0.456
	100	**0.154**	1.251	0.205	2.490	0.076	0.851	0.095	0.595	0.004	5.425	0.157	0.442
	150	0.156	1.097	0.196	**1.888**	0.077	0.752	**0.022**	0.419	**0.009**	5.772	0.128	**0.378**

Table 25: Performance of RAF on Benchmark I when the prediction length $H = 15$ with context lengths, $C \in \{50, 75, 100, 150\}$

Datasets		Weather		Traffic		ETTh1		FRED-MD		Covid Deaths		NN5 (Daily)	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	50	**0.156**	**1.845**	**0.213**	**1.852**	**0.086**	**0.986**	**0.025**	**0.659**	**0.008**	**8.291**	**0.156**	**0.495**
	75	**0.158**	1.598	**0.215**	**2.249**	**0.083**	**1.017**	**0.038**	**0.689**	**0.005**	**14.743**	**0.160**	**0.539**
	100	**0.163**	**1.485**	**0.199**	**2.225**	**0.074**	**1.058**	0.073	**0.656**	**0.006**	**9.577**	**0.171**	**0.549**
	150	**0.167**	1.151	**0.200**	**1.752**	**0.068**	**0.820**	0.037	**0.362**	0.010	**10.040**	**0.169**	**0.567**
Baseline	50	0.159	1.885	0.233	2.162	0.121	1.169	0.248	0.939	0.020	10.089	0.166	0.507
	75	0.162	**1.571**	0.228	2.914	0.150	1.453	0.128	0.722	0.005	16.238	0.191	0.592
	100	0.176	1.516	0.204	2.234	0.164	1.602	**0.070**	0.682	0.006	10.364	0.193	0.571
	150	0.169	**1.147**	0.215	1.872	0.155	1.239	**0.024**	0.475	**0.009**	10.118	0.218	0.608

Table 26: Performance of RAF on Benchmark I when the prediction length $H = 20$ with context lengths, $C \in \{50, 75, 100, 150\}$

Datasets		Weather		Traffic		ETTh1		FRED-MD		Covid Deaths		NN5 (Daily)	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	50	0.164	**1.909**	**0.282**	**2.727**	**0.075**	**0.885**	**0.093**	**0.808**	**0.014**	**12.462**	**0.186**	**0.646**
	75	**0.165**	**1.924**	**0.294**	**2.966**	**0.047**	**0.867**	**0.083**	**0.643**	0.012	**21.807**	**0.155**	**0.538**
	100	0.168	**1.807**	**0.284**	**3.776**	**0.044**	**0.835**	0.125	**0.582**	**0.008**	**17.229**	**0.150**	**0.487**
	150	0.175	1.175	0.275	2.321	**0.055**	**0.714**	**0.040**	**0.400**	**0.009**	**15.990**	**0.162**	**0.538**
Baseline	50	**0.161**	2.146	0.299	2.820	0.120	1.209	0.268	1.163	0.031	13.800	0.242	0.798
	75	0.165	2.042	0.301	2.972	0.128	1.274	0.203	0.829	**0.011**	24.301	0.192	0.613
	100	**0.168**	1.940	0.294	3.886	0.143	1.346	**0.119**	0.749	0.009	18.247	0.189	0.584
	150	**0.174**	**1.144**	**0.263**	**2.257**	0.133	1.109	0.043	0.497	0.011	16.831	0.208	0.613

F.4 Chronos Base Results on Benchmark II

Table 27: Performance of RAF on Benchmark II when the prediction length $H = 3$ with context lengths, $C \in \{10, 15, 18, 21\}$

Datasets		Tourism (M.)		Tourism (Q.)		M1 (M.)		Uber TLC		CIF-2016	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	10	**0.470**	**0.836**	**0.072**	1.406	**0.172**	**1.162**	**0.252**	**1.040**	0.083	**1.202**
	15	**0.209**	2.521	**0.091**	**1.174**	0.181	**1.235**	0.207	**1.162**	**0.020**	**1.339**
	18	**0.198**	**1.643**	**0.080**	**0.973**	**0.182**	**1.029**	**0.142**	**0.745**	0.082	**1.011**
	21	**0.070**	**1.118**	**0.072**	**0.908**	**0.169**	**0.867**	**0.148**	**1.004**	**0.026**	**0.639**
Baseline	10	0.643	0.901	0.075	**1.172**	0.204	1.192	0.295	1.257	**0.079**	1.282
	15	0.230	**2.421**	0.097	1.190	**0.178**	1.260	**0.205**	1.205	0.025	1.395
	18	0.560	1.679	0.099	1.093	0.185	1.115	0.212	0.899	**0.078**	1.039
	21	0.512	1.352	0.073	0.973	0.174	1.015	0.166	1.079	0.027	0.793

Table 28: Performance of RAF on Benchmark II when the prediction length $H = 4$ with context lengths, $C \in \{10, 15, 18, 21\}$

Datasets		Tourism (M.)		Tourism (Q.)		M1 (M.)		Uber TLC		CIF-2016	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	10	**0.485**	0.829	0.111	1.323	0.203	**1.354**	**0.191**	**1.198**	**0.062**	1.256
	15	0.395	2.703	**0.081**	**1.146**	**0.178**	1.415	0.185	**1.100**	**0.041**	**1.360**
	18	**0.107**	**1.559**	**0.091**	**1.054**	**0.165**	**1.063**	**0.163**	**0.983**	**0.039**	**0.862**
	21	**0.068**	**1.173**	**0.088**	**1.035**	**0.140**	**0.880**	**0.145**	**0.876**	**0.038**	**0.692**
Baseline	10	0.554	**0.827**	**0.090**	**1.113**	**0.201**	1.358	0.261	1.463	0.063	**1.162**
	15	**0.359**	**2.514**	0.082	1.151	0.194	**1.392**	**0.178**	1.122	0.043	1.412
	18	0.336	1.584	0.092	1.072	0.167	1.065	0.194	1.093	0.047	0.907
	21	0.370	1.289	0.090	1.105	0.172	1.067	0.150	0.954	0.040	0.954

Table 29: Performance of RAF on Benchmark II when the prediction length $H = 5$ with context lengths, $C \in \{10, 15, 18, 21\}$

Datasets		Tourism (M.)		Tourism (Q.)		M1 (M.)		Uber TLC		CIF-2016	
Metric		WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE	WQL	MASE
RAF	10	**0.422**	**0.911**	0.121	1.671	0.136	**1.323**	**0.195**	**1.274**	**0.066**	**1.385**
	15	**0.233**	**2.669**	**0.083**	**1.177**	**0.118**	**1.250**	**0.172**	**0.972**	**0.120**	**1.181**
	18	**0.101**	**1.551**	**0.092**	**1.188**	**0.162**	**1.120**	**0.125**	**0.949**	**0.052**	1.206
	21	**0.085**	**1.349**	**0.081**	**1.096**	**0.162**	**1.042**	**0.145**	0.927	0.081	**0.840**
Baseline	10	0.488	0.937	**0.113**	**1.326**	**0.134**	1.344	0.261	1.463	0.077	1.487
	15	0.372	3.131	0.093	1.222	0.129	1.269	0.176	1.076	0.122	1.310
	18	0.217	1.826	0.093	1.206	0.174	1.170	0.158	1.023	0.054	**1.186**
	21	0.227	1.353	0.094	1.261	0.177	1.148	0.150	**0.925**	**0.075**	1.024

F.5 Aggregate Relative MASE and WQL Scores

We summarize per-dataset improvements using a relative score. For a dataset d and context length C, let $s^{\text{RAF}}_{d,C}$ and $s^{\text{base}}_{d,C}$ denote the (averaged) WQL or MASE of the Chronos model with and without retrieval. Then, the relative score is

$$r_{d,C} = \frac{s^{\text{RAF}}_{d,C}}{s^{\text{base}}_{d,C}},$$

so values below 1 indicate improvement. To aggregate across the multiple context lengths used for a dataset, we take the geometric mean. Figures 3 and 4 plot these aggregated relative scores (baseline normalized to 1) for Chronos Mini and Chronos Base.

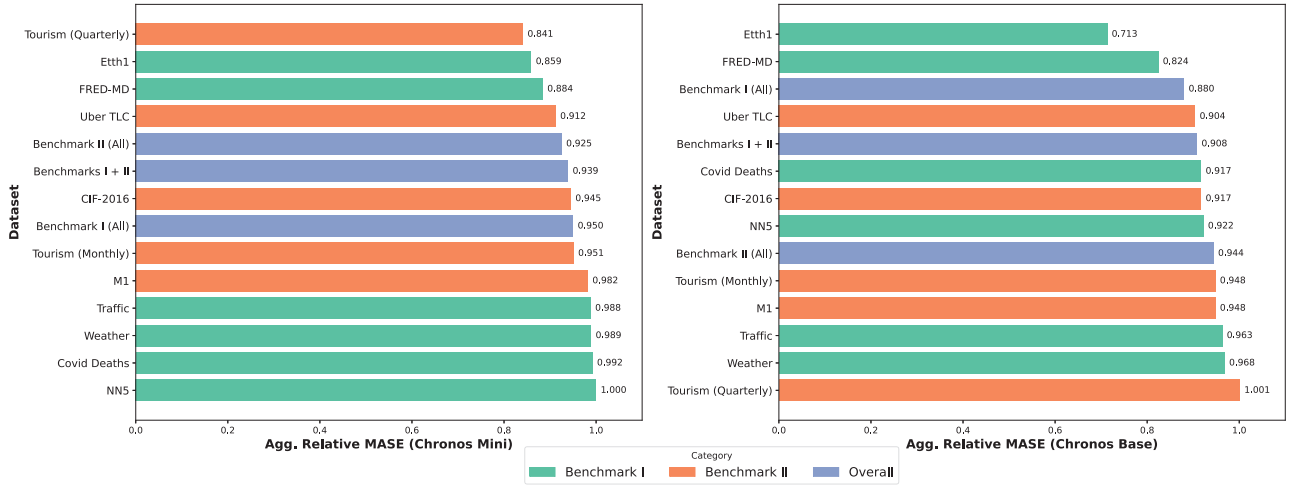

Figure 3: Aggregated Relative MASE performance for Chronos Mini and Chronos Base across datasets and benchmarks. This figure illustrates the comparative analysis of MASE for two configurations—Chronos Mini and Chronos Base, highlighting the relative performance improvements when using RAF within each model configuration.

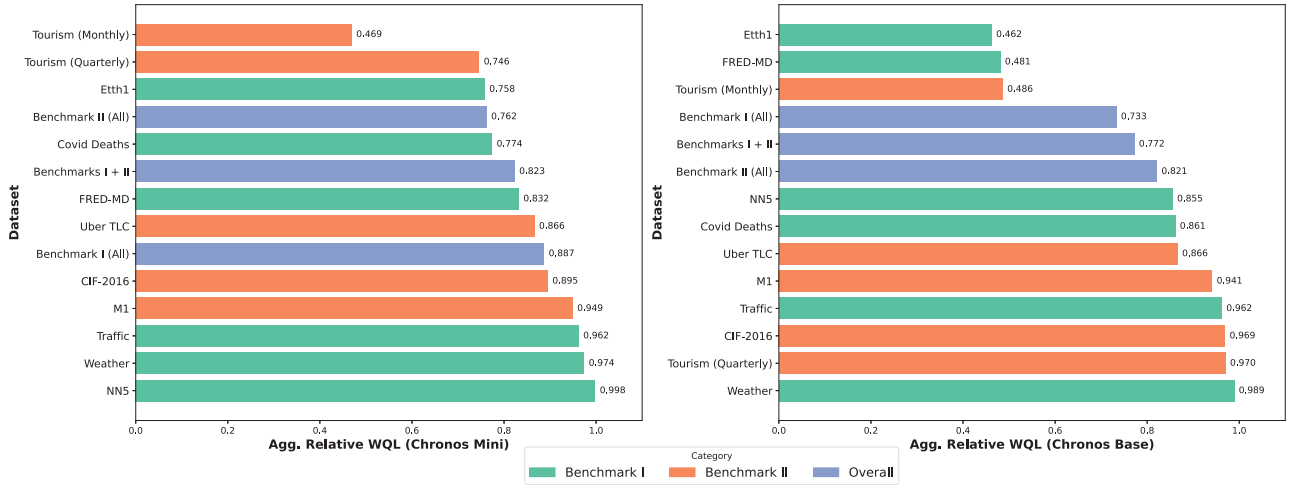

Figure 4: Aggregated Relative WQL performance for Chronos Mini and Chronos Base across datasets and benchmarks. This figure illustrates the comparative analysis of WQL for two configurations—Chronos Mini and Chronos Base, highlighting the relative performance improvements when using RAF within each model configuration.

G Robustness Analysis

RAF's gains depend on the quality of the retrieved context. To probe this dependency, we run controlled synthetic stress tests: we generate univariate series of length 240 and evaluate across 240 rolling windows with C=100 and H=20 on Chronos-Base. We report

$$\%\Delta \text{ mean WQL} = \left(\frac{\text{WQL}_{\text{RAF}}}{\text{WQL}_{\text{Base}}} - 1\right) \times 100$$

alongside the **failure rate**, defined as the fraction of windows where RAF increases WQL.

Noise sensitivity. We corrupt series with i.i.d. Gaussian noise $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$. Table 30 shows that RAF improves mean WQL at every noise level, but the margin narrows from -7.62% (σ=0) to -5.27% (σ=0.8) as noisier contexts yield less reliable retrievals. The failure rate remains between 30–41%.

Table 30: Gaussian noise sweep.

σ	WQL (Base)	WQL (RAF)	%Δ	Fail %
0.0	0.01245	**0.01150**	−7.62	41.25
0.2	0.06462	**0.05840**	−9.64	35.00
0.4	0.11160	**0.10372**	−7.06	32.08
0.6	0.16024	**0.15029**	−6.21	30.42
0.8	0.20608	**0.19523**	−5.27	31.25

Sparsity. We drop observations at random with probability p (MCAR) and fill gaps via linear interpolation before retrieval. Table 31 shows graceful degradation up to p=0.6; at p=0.8 the interpolated context is too unreliable for similarity matching and retrieval slightly hurts (+1.67%, failure rate ≈50%).

Table 31: MCAR sparsity sweep.

p	WQL (Base)	WQL (RAF)	%Δ	Fail %
0.2	0.09576	**0.09187**	−4.06	41.67
0.4	0.24416	**0.22022**	−9.80	34.58
0.6	0.38024	**0.37236**	−2.07	45.42
0.8	**0.41410**	0.42100	+1.67	49.58

Taken together, these results reveal that RAF is notably resilient to moderate corruption. Under noise, the best relative improvement actually occurs at σ=0.2 (-9.64%), suggesting that mild noise can even improve retrieval diversity by breaking ties among near-identical candidates. The failure rate simultaneously drops to its lowest point (30–32%) at moderate noise levels ($\sigma \in \{0.4, 0.6\}$), indicating that while individual WQL values worsen, the retrievals that *do* match become more reliably beneficial. The sparsity results tell a complementary story that linear interpolation preserves enough structure for effective retrieval up to 60% missingness, and the sharp transition at p=0.8 marks the point at which the interpolated signal no longer resembles the true context. This suggests that RAF's practical boundary is determined not by noise magnitude, but by whether the corrupted context retains sufficient shape information for meaningful nearest-neighbor matching.

H Qualitative Results

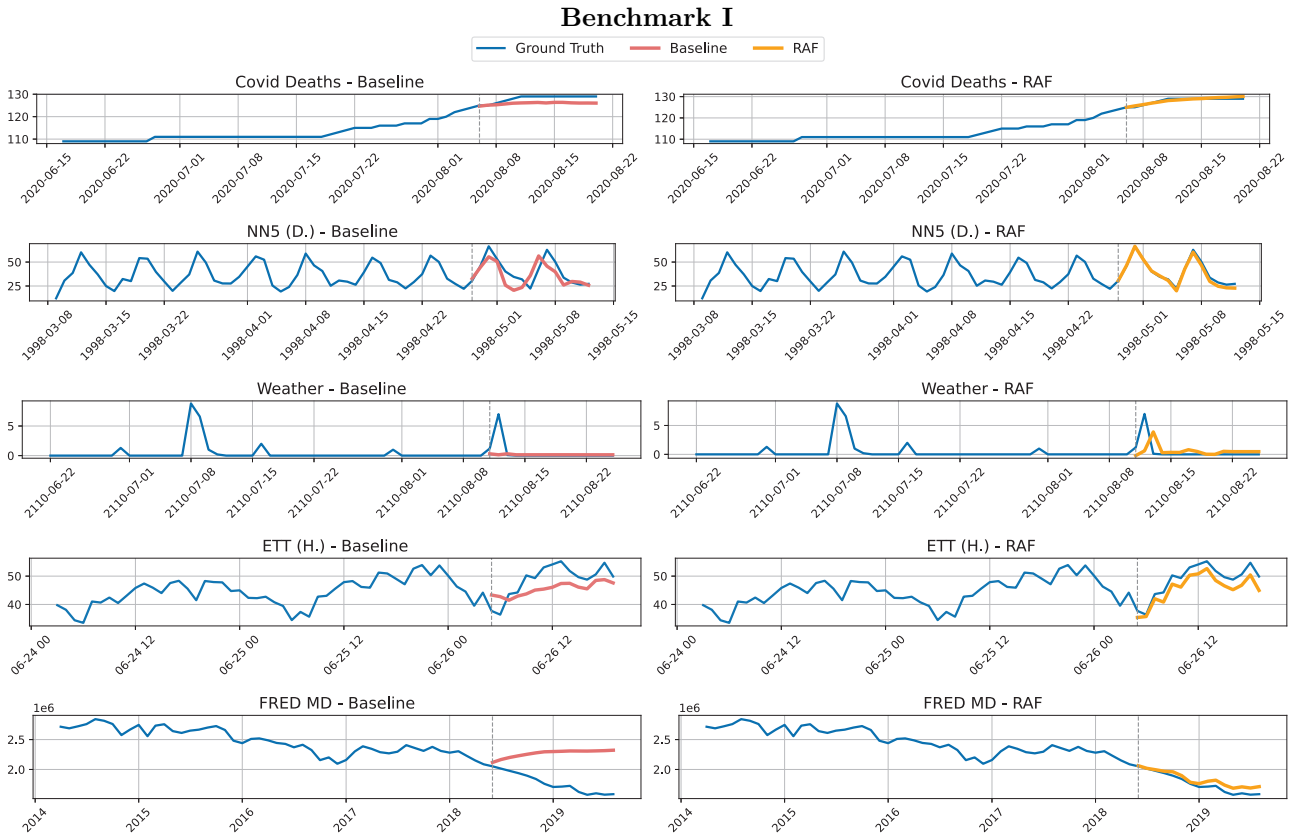

Figure 5: Qualitative results for Benchmark I datasets with $C = 50$ and $H = 15$ on Chronos Base.

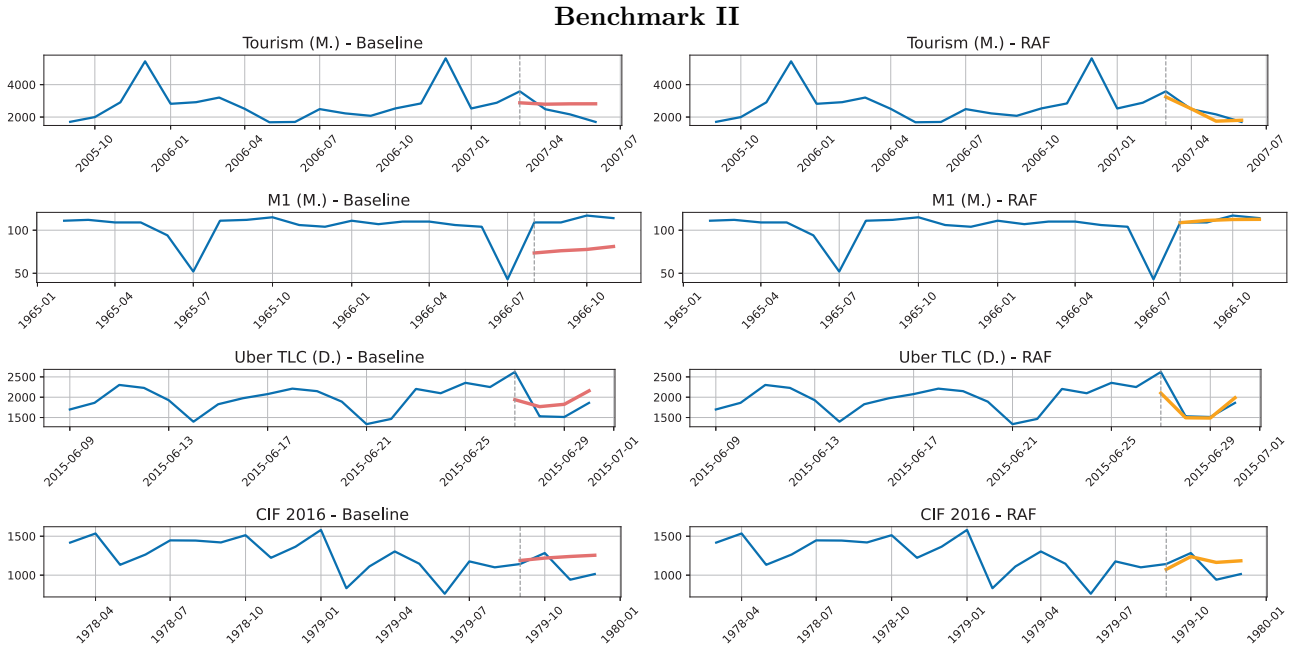

Figure 6: Qualitative results for Benchmark II datasets with $C = 18$ and $H = 4$ on Chronos Base.

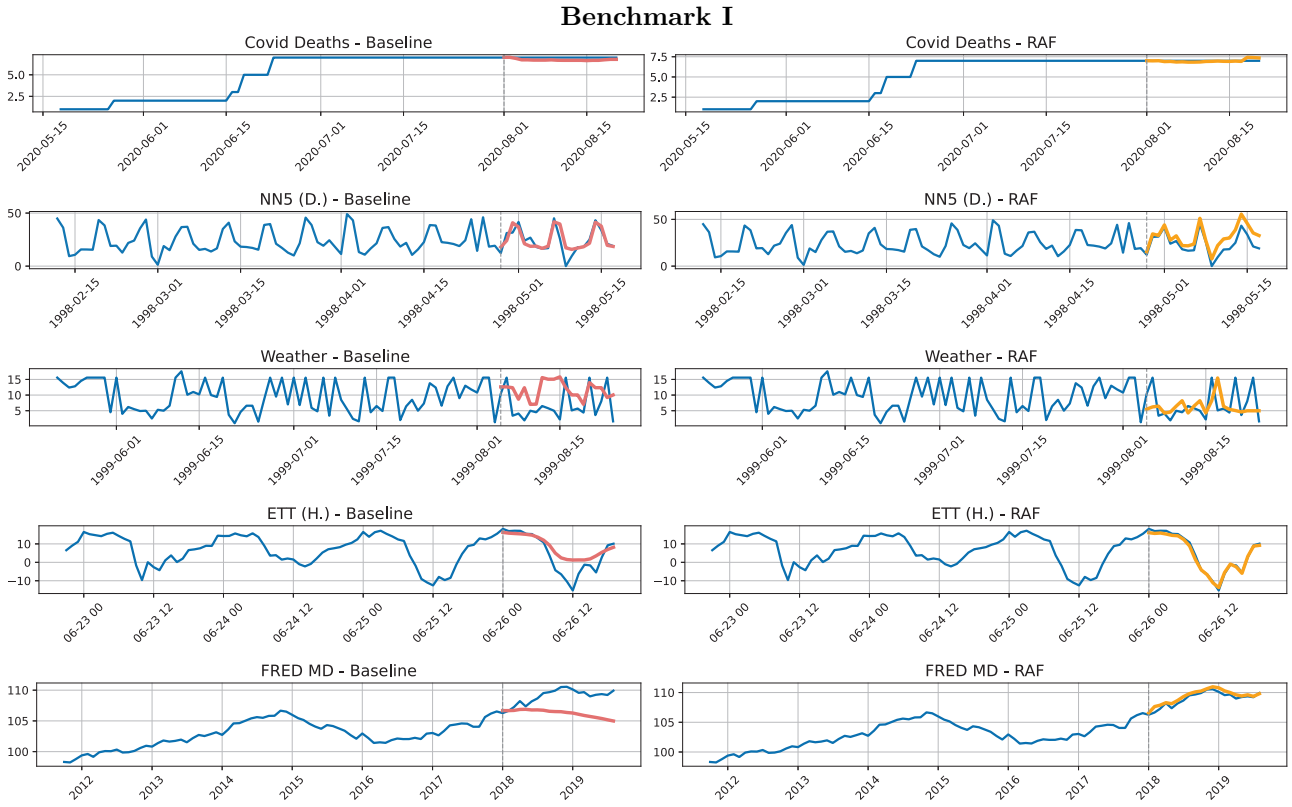

Figure 7: Qualitative results for Benchmark I datasets with $C = 75$ and $H = 20$ on Chronos Base.

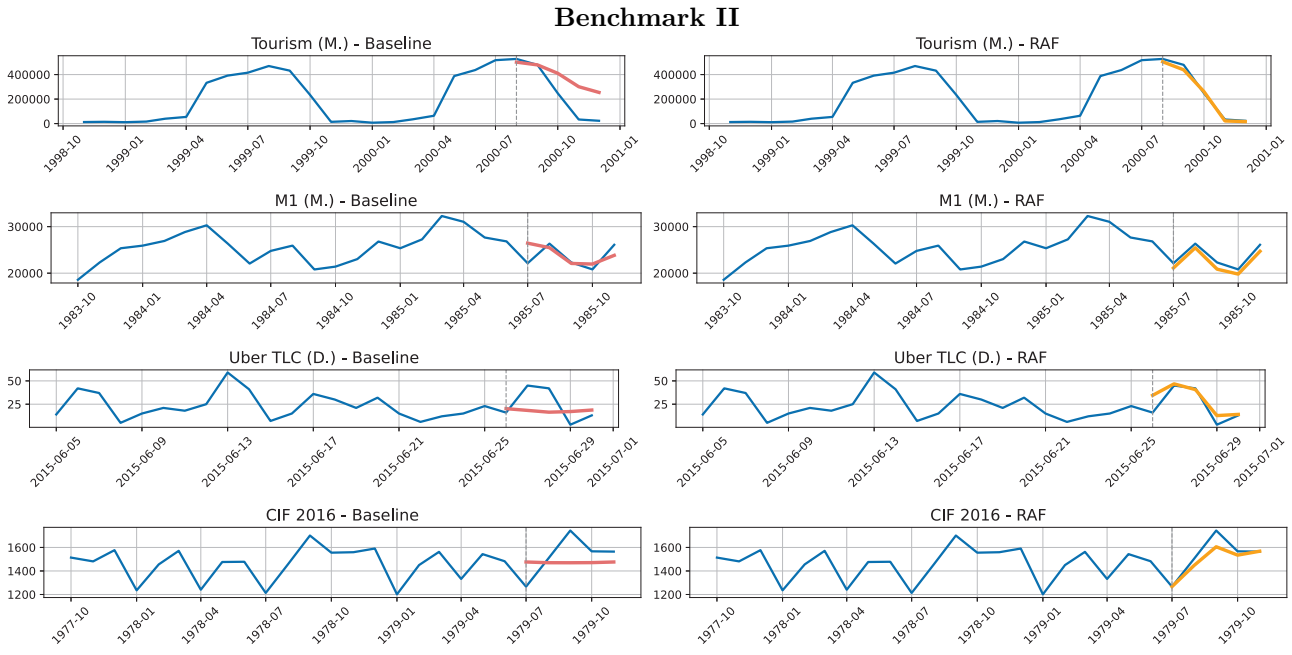

Figure 8: Qualitative results for Benchmark II datasets with $C = 21$ and $H = 5$ on Chronos Base.

I Datasets

Table 32: Overview of the Benchmark Datasets

	Dataset	Domain	Frequency	Number of Series
Benchmark I	Weather	Nature	1D	3010
	Traffic	Transport	1H	862
	ETT (Hourly)	Energy	1H	14
	FRED-MD	Finance	1M	107
	Covid Deaths	Health	1D	266
	NN5 (Daily)	Finance	1D	111
Benchmark II	Tourism (Monthly)	Tourism	1M	366
	Tourism (Quarterly)	Tourism	1Q	427
	M1 (Monthly)	Finance	1M	617
	Uber TLC (Daily)	Transport	1D	262
	CIF-2016	Finance	1M	72

I.1 Benchmark I Datasets

Weather dataset (Godahewa et al., 2021) contains daily time series data with 3010 series for rainfall, recorded at various weather stations across Australia.

Traffic dataset (Godahewa et al. (2021)) consists of 862 hourly time series representing road occupancy rates on freeways in the San Francisco Bay area, covering the period from 2015 to 2016.

ETT dataset (Zhou et al. (2021)) includes 14 time series about oil temperatures and additional covariates of electrical transformers from two stations in China, recorded at 1-hour intervals.

FRED-MD (Godahewa et al. (2021)) contains 107 monthly time series showing a set of macro-economic indicators from the Federal Reserve Bank starting from 01/01/1959.

Covid Deaths (Godahewa et al. (2021)) includes 266 daily time series representing the total number of COVID-19 deaths in various countries and states, covering the period from January 22, 2020, to August 20, 2020. The data was sourced from the Johns Hopkins repository.

NN5 dataset (Godahewa et al. (2021)) consists of 111 daily time series of cash withdrawals from Automated Teller Machines (ATMs) in the UK, and was utilized in the NN5 forecasting competition.

I.2 Benchmark II Datasets

Tourism dataset (Godahewa et al. (2021); Athanasopoulos et al. (2011)), derived from a Kaggle competition, includes 366 monthly and 427 quarterly tourism-related time series.

M1 (Godahewa et al. (2021); Makridakis and Hibon (1979)) contains 617 time series used in the M1 forecasting competition, covering areas such as microeconomics, macroeconomics, and demographics.

Uber TLC contains 262 time series with daily frequency, representing the number of Uber pick-ups from various locations in New York, between January and June 2015. Data obtained from https://github.com/fivethirtyeight/uber-tlc-foil-response.

CIF-2016 (Godahewa et al. (2021)) consists of banking data used in the CIF-2016 forecasting competition. It includes 24 real-time series, while the remaining 48 are artificially generated.