Retrieval Augmented Time Series Forecasting

Kutay Tire[1*] **Ege Onur Taga**[2] **M. Emrullah Ildiz**[2] **Samet Oymak**[2]

[1]Bilkent University, Ankara
kutay.tire@ug.bilkent.edu.tr [2]University of Michigan, Ann Arbor
{egetaga, eildiz, oymak}@umich.edu

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, and the need for effective zero-shot forecasting performance across various time-series domains motivates the question: Do 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 indeed improves the forecasting accuracy across diverse time series domains and the improvement is more significant for larger TSFM sizes.

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. 2023a; Liu et al. 2022; Zhang and Yan 2023; Du et al. 2021; Rasul et al. 2021; Liu et al. 2024), tokenization strategies (Nie et al. 2023a; Chen et al. 2023; Ansari et al. 2024; Persak et al. 2024; Talukder, Yue, and Gkioxari 2024), 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 (Leblond et al. 2023; Nori et al. 2023; Lewis et al. 2021; Lee, Chang, and Toutanova 2019; Khattab, Potts, and Zaharia 2022) where LLMs are integrated with external databases and advanced prompting strategies to accomplish complex tasks.

In particular, retrieval augmented generation (RAG) (Lewis et al. 2021), 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. Second, time-series data often exhibits complex patterns and dependencies that are difficult to capture via traditional forecasting methods – particularly in scenarios where rare events such as earthquakes, financial crises, or elections are of interest. 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 2007), are inherently about identifying and matching complex time-series patterns. Incorporating RAG holds the promise of augmenting time-series models with these capabilities to utilize external knowledge bases.

In this work, we provide a thorough investigation of the benefits of RAG in time-series forecasting. We introduce the *Retrieval Augmented Forecasting* (RAF) framework for time-series foundation models described in Figure 1. We find that RAF substantially enhances the forecast accuracy, especially in out-of-domain datasets (novel scenarios) where the TSFM may lack domain-specific information. 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 M1 dataset (Godahewa et al. 2021) in finance. RAF facilitates the model to capture these properties by augmenting the query with top-matching patterns and harnessing the in-context learning capability of the TSFM. This way, it also provides a resource-efficient alternative to fine-tuning methods. Our main contributions are as follows:

1. We introduce RAF as a principled forecasting framework for TSFMs. We formalize RAF in Section , 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). No-

[*]Equal contribution.

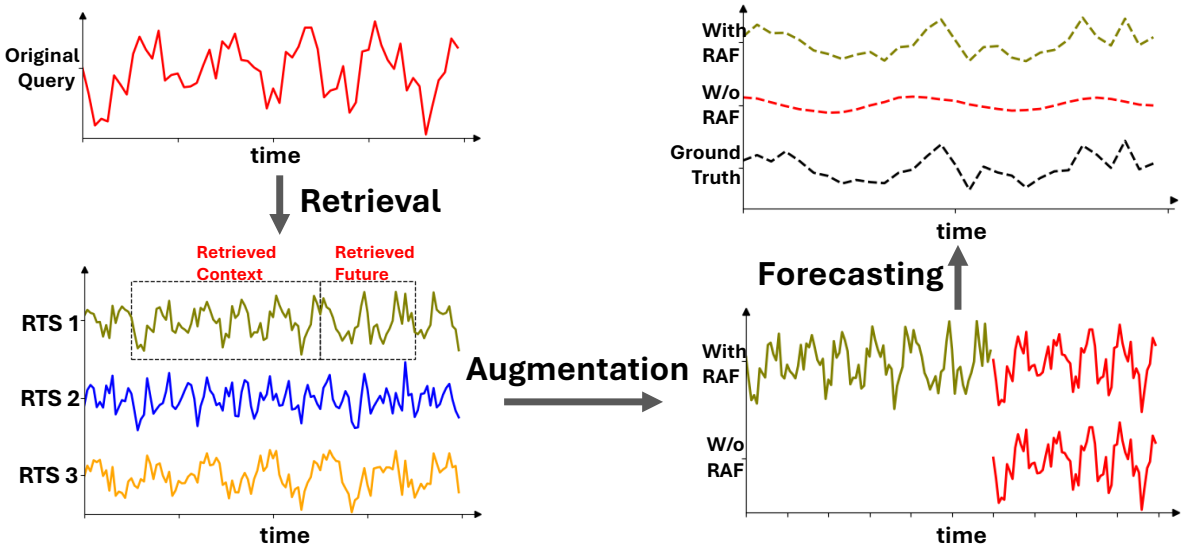

Figure 1: Overview of the *Retrieval Augmented Forecasting* (RAF) framework. *Top left:* The original query is used to retrieve the best-matching time series (RTS 1, RTS 2, RTS 3, . . .). *Bottom left:* We utilize the best match (RTS 1) to form the retrieved context and retrieved future. *Bottom right:* 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 model and returns a forecast closer to the actual future values.

tably, 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 forecast is an emerging capability of large time-series foundation models.

2. We describe and study two variants of RAF: Naive RAF, which we will simply refer to as RAF and Advanced RAF. RAF employs the model as a black-box without adjusting the model weights, whereas Advanced RAF additionally fine-tunes the model for retrieval-augmented generation, both yielding significant performance gains. As detailed in Table 3, we assess probabilistic and point forecasting performances on Chronos, comparing both RAF types against standard methods with and without fine-tuning. Advanced RAF surpasses all baselines, highlighting the effectiveness of our framework and its synergy with fine-tuning. Additionally, RAF closely matches the performance of fine-tuned models, providing a new method to improve TSFM performance.

3. We examine RAF with two model sizes – Chronos Mini and Base – shedding light on its effectiveness across different model capabilities, as shown in Figures 3 and 4 in Appendix and Tables 1 and 2. These evaluations reveal that the relative improvement of RAF increases as the model size grows, in line with the empirical findings of RAG in large language models (Guu et al. 2020; Lewis et al. 2021; Kaplan et al. 2020).

Related Works

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. 2022, 2024; Lim et al. 2021; Nie et al. 2023b), ForecastPFN (Dooley et al. 2023), TimePFN (Taga, Ildiz, and Oymak 2024), 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. 2020; Jin et al. 2022; Dooley et al. 2023; Ansari et al. 2024). ForecastPFN (Dooley et al. 2023) and TimePFN (Taga, Ildiz, and Oymak 2024) 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 describes 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), Lag-Llama (Rasul et al. 2024), TimeGPT-1 (Garza, Challu, and Mergenthaler-Canseco 2024), and

Chronos (Ansari et al. 2024). Chronos proposes a scaling and quantization technique to train standard LLM architectures such as T5 (Raffel et al. 2023) 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 model generation in knowledge intensive tasks. A line of work includes those of (Lewis et al. 2021; Guu et al. 2020; Izacard and Grave 2020; Khattab and Zaharia 2020; Fan et al. 2021; Borgeaud et al. 2022; Ma et al. 2023). Furthermore, recent studies have focused on utilizing Retrieval-Augmented Generation (RAG) for time-series forecasting (Ravuru, Sakhinana, and Runkana 2024; Wang and Cui 2024; Saveri and Bortolussi 2024). (Ravuru, Sakhinana, and Runkana 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. To the best of our knowledge, this is the first study to explore RAG in TSFMs.

Problem Setup

Let us first introduce the basic 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 $\boldsymbol{a}$. Let $\boldsymbol{x} \in \mathbb{R}^L$ denote a univariate time series of length L. Let $f(\boldsymbol{x}) \in \mathbb{R}^H$ represent the forecast of the time series given model f and input $\boldsymbol{x}$. We also use $\boldsymbol{x}[j, C]$ to denote the sub-series of length C that starts from time j and ends at time $j+C-1$. $(f(\boldsymbol{x}))[j, H]$ is defined similarly. A sub-series that approximately repeats within a longer time-series is also referred to as **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 $\boldsymbol{x}[j, j+C-1] = \boldsymbol{m}$.

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 output it as its prediction.

Definition 1 (TS-R problem) *Let $\boldsymbol{m}$ be the motif at the end of the time-series sequence i.e. $\boldsymbol{m} = \boldsymbol{x}[L-C+1, L] \in \mathbb{R}^C$. Suppose that there is a unique matching motif in the history, namely, we have $\boldsymbol{x}[t, t+C-1] = \boldsymbol{m}$ for a unique timestamp $t < L-C+1$. Let $\Upsilon \in \mathbb{R}^H$ be a motif followed by $\boldsymbol{x}[t, t+C-1]$. We say that a model f solves TS-R problem if- for all $\boldsymbol{m} \in \mathbb{R}^C$ and $\Upsilon \in \mathbb{R}^H$, we have $f(\boldsymbol{x}) = \Upsilon$.*

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 prediction by motif retrieval. Compared 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.

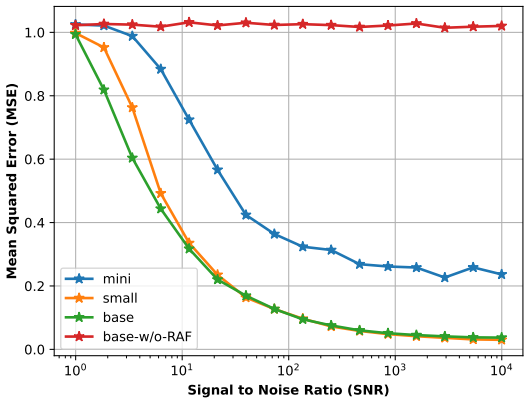

Figure 2: We generated synthetic time-series data by transposing two sinusoidal signals and projecting them via orthogonal projections. We assessed extrapolation behavior using scaled mean squared error (assuming 0 prediction as baseline) and chose a context and forecast length of $C = 30$ and $H = 30$.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

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. 2023a; Taga, Ildiz, and Oymak 2024), however, we use stride length 1 to ensure every sub-series is tokenized and can be retrieved. The formal theorem statement is provided under Theorem 2 of Appendix . Throughout the paper, we denote the motif $\boldsymbol{m}$ and Υ as the retrieved context and retrieved future, respectively.

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 transpose two sinusoidal signals with different frequencies, designated as f_1 and f_2 (both fixed), creating a signal of length $L = C + H$. This signal is then randomly rotated by $\mathbf{Q} \in \mathbb{R}^{L \times L}$, which is a

random unitary matrix obeying $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$. Overall, we generate the synthetic retrieval data as follows:

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

To model noisy retrieval, we retrieve the matching motif $\boldsymbol{s}_r = \boldsymbol{s} + \boldsymbol{z}$ with additive noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma_s^2 \mathbf{I})$. The variance σ_s^2 determines the signal-to-noise ratio in Figure 2. Based on the retrieved context $\boldsymbol{m}_r := \boldsymbol{s}_r[0, C]$, which is the matching motif in noiseless setting to $\boldsymbol{m} := \boldsymbol{s}[0, C]$, and the retrieved future $\boldsymbol{r} := \boldsymbol{s}_r[C, C + H]$, we create a RAF query $\boldsymbol{s}_{raf} = [\boldsymbol{m}_r \quad \boldsymbol{r} \quad \boldsymbol{m}] \in \mathbb{R}^{L+C}$. We evaluate the forecasts $f(\boldsymbol{s}_{raf})[0, H]$ with respect to the original future $\boldsymbol{s}[C, C + H]$.

As demonstrated in Figure 2, the mean squared error converges to zero 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 .

Methodology

The time-series retrieval problem discussed in Section and the experimental results shown in Figure 2 demonstrate that transformer-based time-series models are well-equipped for retrieval-augmented forecasting. However, these assume the existence of a retrieval mechanism and a time-series model. Here, we describe the design choices and implementation details of the RAF framework.

Indexing and Database Formation. To retrieve the best matching time series as described in Section , we construct a database specific to each data domain (dataset) because different data domains exhibit distinct characteristics. Consequently, we allocate 20% of the time series in each dataset for testing and use the remaining 80% to form the database through a random but fixed split for all evaluations. From each dataset-specific database, we then retrieve the best matches.

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 the embedding of the original time series using the encoder of the chosen model. 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.

Instance Normalization. To mitigate the distribution shift effects between training and testing data, we apply instance normalization (Ulyanov, Vedaldi, and Lempitsky 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 top-1 best-matching time series from our database using our 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 what we call the *retrieved time series*. The retrieved time series, which includes both the retrieved context and 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, a smooth transition between them must be ensured. To do this, the ending point of the retrieved time series is aligned with the beginning point of the original context. This alignment prevents any abrupt changes or discontinuities that might negatively affect the performance of the Chronos 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 Chronos models.

Time series models. We utilize Chronos Mini and Base as our evaluation models. Similar to LLMs (Ansari et al. 2024), Chronos is probabilistic, allowing for multiple future trajectories to be generated by autoregressively sampling from the predicted distribution, $p_\theta(\mathbf{z}_{C+h+1} \,|\, \mathbf{z}_{1:C+h})$.

Experiments

This section provides an overview of the baselines, and evaluation metrics, followed by our main results and evaluations.

Baselines and Evaluation Metrics

Our experiments compare Chronos models augmented with RAF to the baseline Chronos models without retrieval. We also compare both approaches for zero-shot forecasting (Tables 1 and 2) as well as forecasting after the model is fine-tuned on the target dataset (Table 3). For a fair comparison, we evaluated RAF against the baseline across two different benchmark datasets, as explained earlier. For each dataset, we evaluated the results using four different context lengths, $C \in \{50, 75, 100, 150\}$ for Benchmark I and

$C \in \{10, 15, 18, 21\}$ for Benchmark II, across three different prediction lengths: $H \in \{10, 15, 20\}$ for Benchmark I and $H \in \{3, 4, 5\}$ for Benchmark II. We then compared the average results across these three prediction lengths for each benchmark.

For evaluation, we assessed RAF based on both probabilistic and point forecast performance, as recommended by (Ansari et al. 2024). We utilized the weighted quantile loss (WQL) to measure the quality of our probabilistic forecasts. WQL measures how closely the predicted distribution aligns with the actual observed values across various quantile levels. In our evaluation, we calculated WQL at nine evenly spaced quantile levels: $\{0.1, 0.2, \ldots, 0.9\}$, following the approach outlined by (Ansari et al. 2024). We used the mean absolute scaled error (MASE, (Hyndman and Koehler 2006)) to evaluate point forecast performance. MASE is computed by taking the absolute forecast error and scaling it by the average historical error, accounting for seasonality in the time series. For each dataset, we first averaged the MASE and WQL metrics across three distinct prediction lengths which yielded an average score for each context length similar to (Ansari et al. 2024). After calculating the average score for each context length, we evaluate RAF on each dataset by computing the relative performance of RAF defined as *the ratio of the average Chronos score with RAF to the baseline average Chronos score*, where no time series retrieval was applied (w/o RAG). The relative scores were then aggregated using the geometric mean to account for the varying context lengths, yielding the final results displayed.

Main Results

We present our primary results based on two sets of datasets: Benchmark I, consisting of 6 datasets, and Benchmark II, consisting of 5 datasets. On average, RAF outperforms the baseline approach across both benchmarks when tested on both Chronos Base and Chronos Mini. Additionally, the fine-tuning of the models with time series retrieval demonstrates superior performance in Benchmark I as seen in Table 3.

• **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 . 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 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 when implemented 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 .

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 consistently outperforms the baseline across all four context lengths, yielding better results in both WQL and MASE. The same 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 RAF's training was Uber TLC, which may have contributed to its 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 observe the overall improvement in performance across various datasets and the two benchmarks, we calculated the geometric mean of the relative scores for each context length, as explained in Section . To determine the average improvement for the benchmarks as a whole, we also used the geometric mean across the different datasets in each benchmark. Figure 4 provides a comprehensive view of how well RAF performed on average in terms of WQL across the different datasets for both models.

The results indicate that the datasets in Benchmark II gen-

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	**0.164**	**1.330**	0.254	2.114	**0.124**	**1.136**	**0.164**	**0.794**	**0.011**	**12.097**	**0.210**	**0.698**
	75	**0.166**	**1.535**	**0.250**	**2.301**	**0.127**	**1.312**	**0.107**	**0.682**	**0.009**	**14.921**	**0.222**	0.739
	100	**0.167**	**1.545**	**0.256**	**2.798**	**0.095**	**1.167**	**0.113**	**0.685**	**0.008**	11.781	**0.200**	**0.642**
	150	**0.167**	**1.441**	**0.246**	**1.973**	**0.108**	**0.979**	**0.069**	**0.441**	**0.008**	11.491	0.199	0.618
m-Base.	50	0.168	1.342	**0.253**	**1.994**	0.148	1.250	0.191	0.940	0.015	12.362	0.212	0.702
	75	0.172	1.553	0.266	2.400	0.142	1.369	0.164	0.791	0.012	15.574	0.223	**0.729**
	100	0.173	1.558	0.267	2.854	0.144	1.457	0.131	0.724	0.010	**11.456**	0.205	0.670
	150	0.171	1.458	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**	**0.044**	**0.616**	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**	0.008	**10.335**	**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**	11.085	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 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		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**	**0.055**	**1.259**	**0.174**	**1.277**	**0.221**	**1.234**	**0.055**	1.259
	15	0.576	2.411	**0.051**	**1.180**	0.170	1.324	**0.183**	**1.112**	**0.051**	**1.180**
	18	**0.176**	**1.629**	**0.056**	**0.812**	**0.160**	**1.121**	**0.173**	**0.983**	**0.056**	**0.812**
	21	**0.086**	**1.271**	**0.053**	**0.867**	**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**	0.061	**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	**0.056**	0.924

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.

erally benefit more from time series retrieval compared to those in Benchmark I for Chronos Mini, whereas the situation is reversed for Chronos Base, as evidenced by the lower relative MASE and WQL values. This suggests that time series retrieval is more effective with shorter context lengths for simpler models, as it is easier to identify matching time series using the similarity metric. Conversely, Benchmark I exhibited greater variability in performance for Chronos Mini, indicating potential challenges or limitations in retrieving the optimal matching time series for longer context lengths in certain datasets. Therefore, more complex models, such as Chronos Base, can be employed to address this issue and better utilize the information provided by retrieval for longer context lengths.

Nevertheless, RAF enhanced the performance of the Chronos models in both benchmarks, as evidenced by the

	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	0.515	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 3: 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 finetuning.

aggregate relative scores for WQL of 0.887 and 0.762, and for MASE of 0.950 and 0.925 for Benchmarks I and II, respectively, in Chronos Mini; and WQL of 0.733 and 0.821, and MASE of 0.880 and 0.944 for Benchmarks I and II in Chronos Base. The overall improvement across all datasets is also apparent, with scores of 0.823 for WQL and 0.939 for MASE in Chronos Mini, and 0.772 for WQL and 0.908 for MASE in Chronos Base (See Appendix).

Fine-tuning evaluations. Inspired by the strong performance of RAF, we conducted an initial study to fine-tune Chronos Base and Mini on individual datasets from Benchmark I, comparing the results with and without the integration of RAF. For this experiment, Chronos Mini was fine-tuned for 400 epochs, while Chronos Base was fine-tuned for 1000 epochs. Both models were fine-tuned in a dataset-agnostic manner, starting with an initial learning rate of 0.00001, which was linearly reduced to 0. The context length was set to 75, and the prediction length to 10.

Table 3 illustrates the impact of RAF during the fine-tuning process. The final column (Advanced RAF) represents the performance of Chronos models that are fine-tuned on datasets augmented through time series retrieval. As it can be observed, Advanced RAF outperforms the baseline FT in almost all cases, where fine-tuning was performed without time series retrieval. Overall, Advanced RAF delivers superior performance across most datasets, achieving the lowest WQL and MASE values in both models.

Notably, for datasets such as Weather, ETTh1, FRED-MD, and NN5, Advanced RAF shows significant improvement over both the Baseline and Naive RAF in both models. For example, in the NN5 dataset, the MASE value decreases from 0.563 (Naive RAF) to 0.401 (Advanced RAF) in Chronos Mini, and from 0.481 (Naive RAF) to 0.378 (Advanced RAF) in Chronos Base, representing a substantial performance boost. While these improvements are clear in many datasets, there are exceptions as well. In the Covid Deaths dataset for Chronos Mini, Baseline FT outperforms Advanced RAF in both WQL and MASE metrics. Interestingly, even without fine-tuning, the baseline performs better than Naive RAF in this dataset with the chosen context and prediction lengths. A similar pattern is observed in the Traffic dataset for Chronos Base, suggesting that fine-tuning may not be optimal for certain scenarios where time series retrieval does not significantly benefit the model.

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 explored two variations: 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 on 11 diverse datasets, grouped into two distinct benchmarks, each with varying context and prediction lengths. 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 and retrieve multiple samples from external sources. Combining

these retrieved samples in a structured manner could further enhance the models' forecasting capabilities, particularly in complex, multi-variate time-series scenarios.

Acknowledgements

This work was supported in part by the NSF grants CCF-2046816, CCF-2403075, the Office of Naval Research grant N000142412289, and gifts by Open Philanthropy and Google Research.

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. arXiv:2403.07815.

Athanasopoulos, G.; Hyndman, R.; 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.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G. B.; Lespiau, J.-B.; Damoc, B.; Clark, A.; De Las Casas, D.; Guy, A.; Menick, J.; Ring, R.; Hennigan, T.; Huang, S.; Maggiore, L.; Jones, C.; Cassirer, A.; Brock, A.; Paganini, M.; Irving, G.; Vinyals, O.; Osindero, S.; Simonyan, K.; Rae, J.; Elsen, E.; and Sifre, L. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 2206–2240. PMLR.

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:2303.06053.

Dooley, S.; Khurana, G. S.; Mohapatra, C.; Naidu, S. V.; and White, C. 2023. ForecastPFN: Synthetically-Trained Zero-Shot Forecasting. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 2403–2426. Curran Associates, Inc.

Du, Y.; Wang, J.; Feng, W.; Pan, S.; Qin, T.; Xu, R.; and Wang, C. 2021. AdaRNN: Adaptive Learning and Forecasting of Time Series. arXiv:2108.04443.

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. 2024. TimeGPT-1. arXiv:2310.03589.

Godahewa, R.; Bergmeir, C.; Webb, G. I.; Hyndman, R. J.; and Montero-Manso, P. 2021. Monash Time Series Forecasting Archive. arXiv:2105.06643.

Gu, A.; and Dao, T. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M.-W. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. arXiv:2002.08909.

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

Izacard, G.; and Grave, E. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Jin, X.; Park, Y.; Maddix, D.; Wang, H.; and Wang, Y. 2022. Domain Adaptation for Time Series Forecasting via Attention Sharing. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 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. arXiv:2001.08361.

Khattab, O.; Potts, C.; and Zaharia, M. 2022. Baleen: Robust Multi-Hop Reasoning at Scale via Condensed Retrieval. arXiv:2101.00436.

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*, 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*.

Leblond, L.; et al. 2023. AlphaCode 2 Technical Report. Technical report, DeepMind.

Lee, K.; Chang, M.-W.; and Toutanova, K. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. arXiv:1906.00300.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; tau Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

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. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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*, volume 619 of *KDD '24*, 6555–6565. ACM.

Lim, B.; Arik, S. O.; 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. 2022. 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. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. arXiv:2310.06625.

Ma, X.; Gong, Y.; He, P.; Zhao, H.; and Duan, N. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Makridakis, S.; Hibon, M.; and Moser, C. 1979. Accuracy of Forecasting: An Empirical Investigation. *Journal of the Royal Statistical Society. Series A (General)*, 142(2): 97–145.

Müller, M. 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.; H. Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023a. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023b. 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. arXiv:2311.16452.

Olsson, C.; Elhage, N.; Nanda, N.; Joseph, N.; DasSarma, N.; Henighan, T.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; Conerly, T.; Drain, D.; Ganguli, D.; Hatfield-Dodds, Z.; Hernandez, D.; Johnston, S.; Jones, A.; Kernion, J.; Lovitt, L.; Ndousse, K.; Amodei, D.; Brown, T.; Clark, J.; Kaplan, J.; McCandlish, S.; and Olah, C. 2022. In-context Learning and Induction Heads. arXiv:2209.11895.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2020. Meta-learning framework with applications to zero-shot time-series forecasting. arXiv:2002.02887.

Orozco, B. P.; and Roberts, S. J. 2020. Zero-shot and few-shot time series forecasting with ordinal regression recurrent neural networks. arXiv:2003.12162.

Persak, E.; Anjos, M. F.; Lautz, S.; and Kolev, A. 2024. Multiple-Resolution Tokenization for Time Series Forecasting with an Application to Pricing. arXiv:2407.03185.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683.

Rasul, K.; Ashok, A.; Williams, A. R.; Ghonia, H.; Bhagwatkar, R.; Khorasani, A.; Bayazi, M. J. D.; Adamopoulos, G.; Riachi, R.; Hassen, N.; Biloš, M.; Garg, S.; Schneider, A.; Chapados, N.; Drouin, A.; Zantedeschi, V.; Nevmyvaka, Y.; and Rish, I. 2024. Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting. arXiv:2310.08278.

Rasul, K.; Seward, C.; Schuster, I.; and Vollgraf, R. 2021. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. arXiv:2101.12072.

Ravuru, C.; Sakhinana, S. S.; and Runkana, V. 2024. Agentic Retrieval-Augmented Generation for Time Series Analysis. *arXiv preprint arXiv:2408.14484*.

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*, 315–331. Springer.

Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; and Liu, Y. 2023. RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv:2104.09864.

Taga, E. O.; Ildiz, M. E.; and Oymak, S. 2024. TimePFN: Effective Multivariate Time Series Forecasting with Synthetic Data. In *NeurIPS Workshop on Time Series in the Age of Large Models*.

Talukder, S.; Yue, Y.; and Gkioxari, G. 2024. TOTEM: TOkenized Time Series EMbeddings for General Time Series Analysis. arXiv:2402.16412.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2017. Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv:1607.08022.

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 Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 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. arXiv:2402.02592.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*.

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)*, 1317–1322. Ieee.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *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 *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, 11106–11115. AAAI Press.

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 *Proc. 39th International Conference on Machine Learning (ICML 2022)*.

Theoretical Results

TS-R Problem

In Section , 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 . 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 separate dimensions (there is a clear bijective mapping that is inverse of this embedding, denoted 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 the 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 Definition 2. 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.

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.

Experimental Details

Hyper-parameter Name	**Baseline**	**Naive 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

Table 4: Summary of hyper-parameter settings used for Chronos fine-tuning

The experimental details for the fine-tuning process are given in Table 4. The table provides a summary of the hyper-parameter settings used for fine-tuning two approaches: Baseline and Naive RAF. Both approaches employ a data split of 70% for database formation (used only for Naive 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 approach uses a context length of 75, while Naive 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 and 60 for Naive 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 hyper-parameters 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.

Extended Results

Chronos Mini Results

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 5: Performance of RAF on Benchmark I when the prediction lengths $H = 10$ 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	**0.166**	**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 6: Performance of RAF on Benchmark I when the prediction lengths $H = 15$ 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	**0.171**	**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**

Table 7: Performance of RAF on Benchmark I when the prediction lengths $H = 20$ with context lengths, $C \in \{50, 75, 100, 150\}$

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 8: Performance of RAF on Benchmark II when the prediction lengths $H = 3$ 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	**0.161**	**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	12	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 9: Performance of RAF on Benchmark II when the prediction lengths $H = 4$ 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	**0.294**	**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**

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

Chronos Base Results

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**	**0.515**	**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**	0.011	**4.197**	**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	**0.006**	4.464	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 11: Performance of RAF on Benchmark I when the prediction lengths $H = 10$ 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	**0.156**	**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 12: Performance of RAF on Benchmark I when the prediction lengths $H = 15$ 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	0.164	**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

Table 13: Performance of RAF on Benchmark I when the prediction lengths $H = 20$ with context lengths, $C \in \{50, 75, 100, 150\}$

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 14: Performance of RAF on Benchmark II when the prediction lengths $H = 3$ 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	**0.485**	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**	0.075	**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	**0.065**	0.954

Table 15: Performance of RAF on Benchmark II when the prediction lengths $H = 4$ 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	**0.422**	**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

Table 16: Performance of RAF on Benchmark II when the prediction lengths $H = 5$ with context lengths, $C \in \{10, 15, 18, 21\}$

Aggregate Relative MASE and WQL Scores

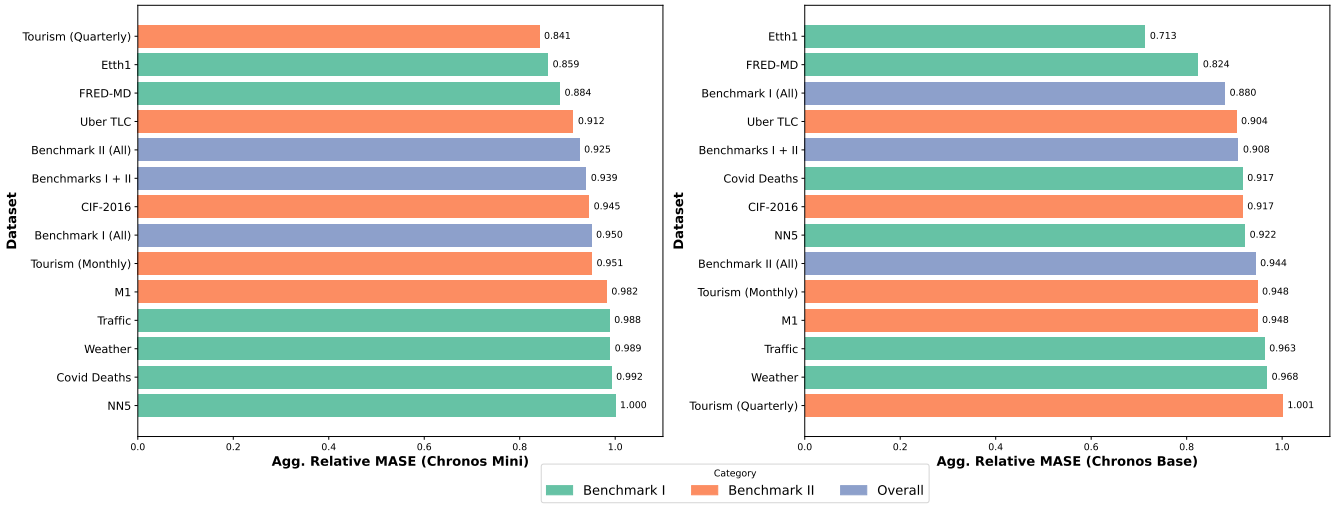

Figure 3: Aggregated Relative MASE performance for Chronos Mini and Chronos Base across datasets and benchmarks. This figure illustrates the comparative analysis of MASE for two configurations—Chronos Mini and Chronos, highlighting the relative performance improvements when using RAF within each model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

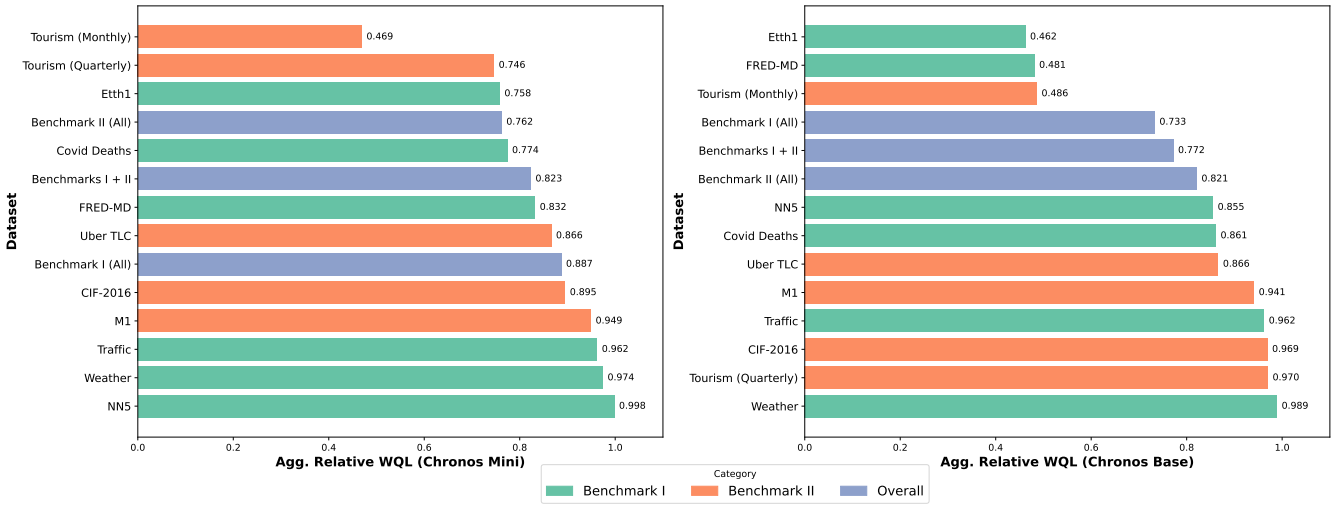

Figure 4: Aggregated Relative WQL performance for Chronos Mini and Chronos Base across datasets and benchmarks. This figure illustrates the comparative analysis of WQL for two configurations—Chronos Mini and Chronos, highlighting the relative performance improvements when using RAF within each model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

Qualitative Results

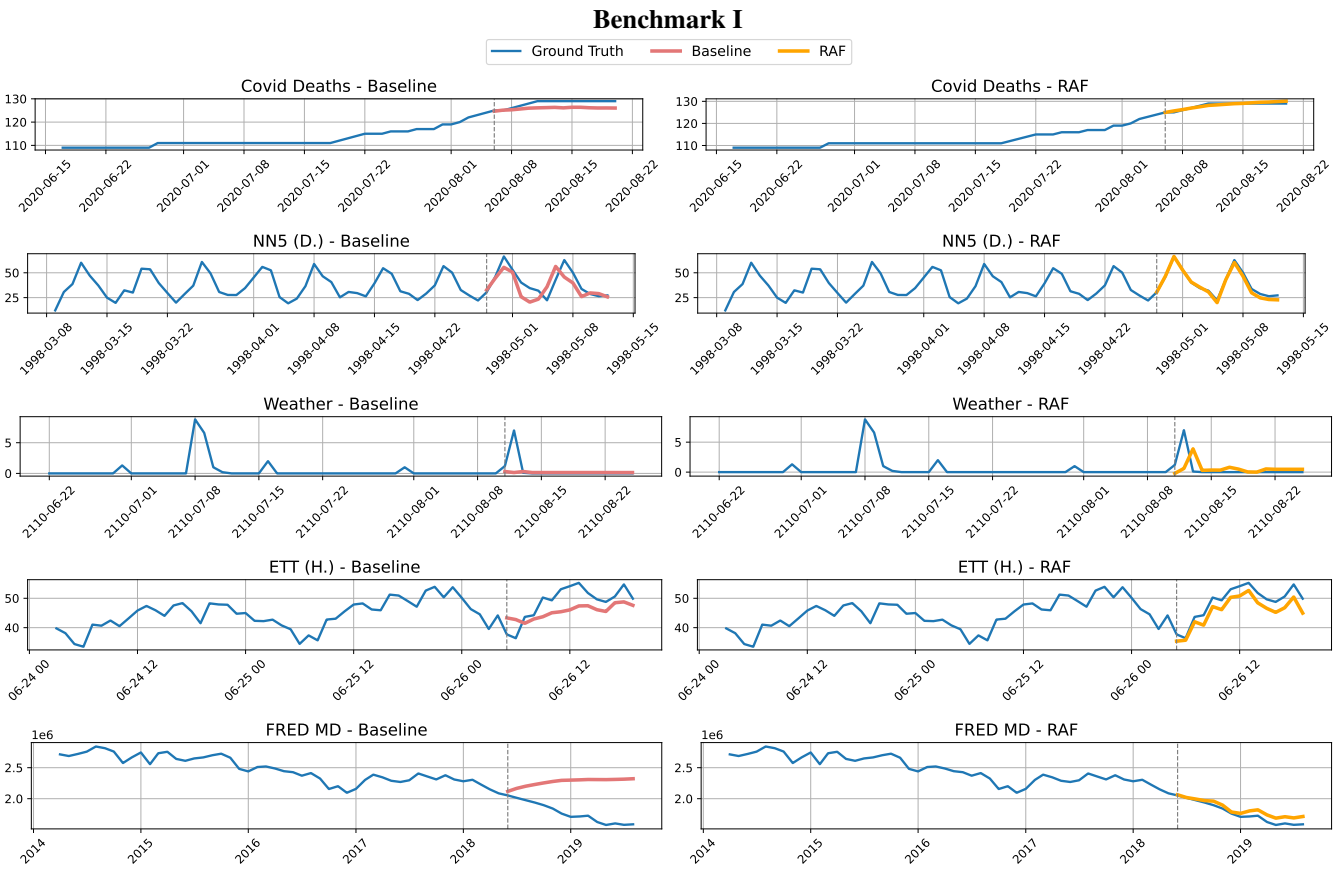

Figure 5: Qualitative results for Benchmark I datasets with $C = 50$ and $H = 15$ on Chronos Base.

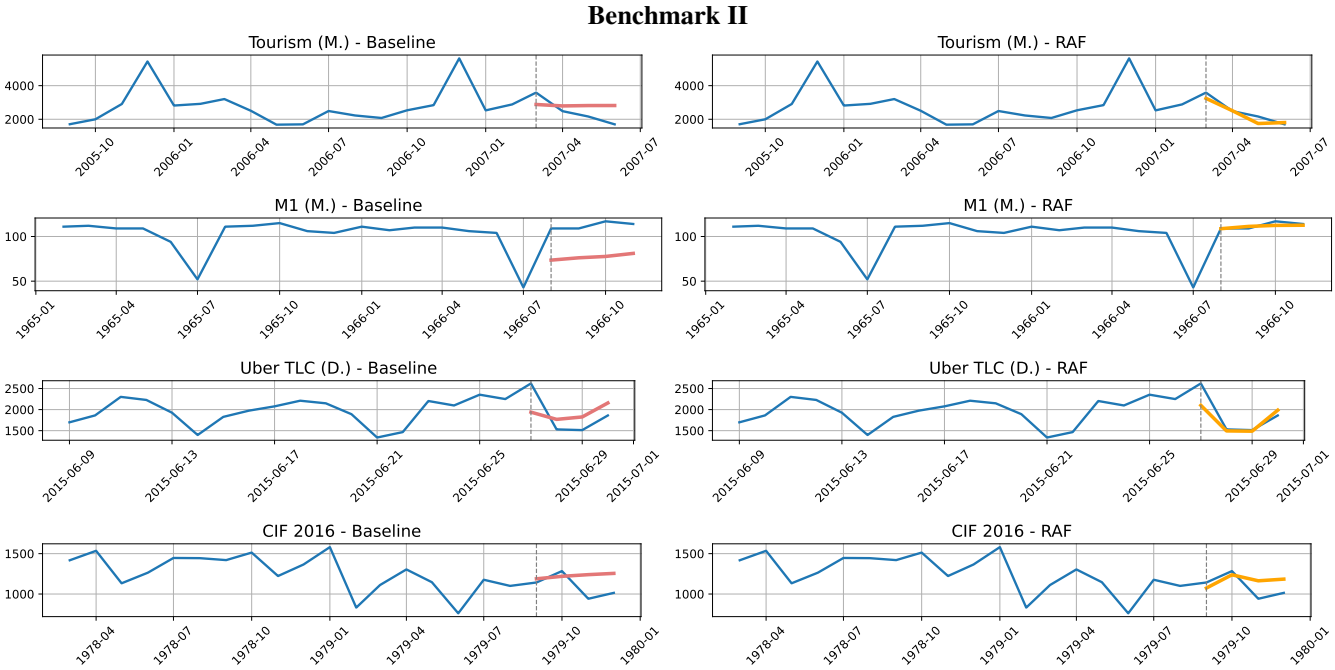

Figure 6: Qualitative results for Benchmark II datasets with $C = 18$ and $H = 4$ on Chronos Base.

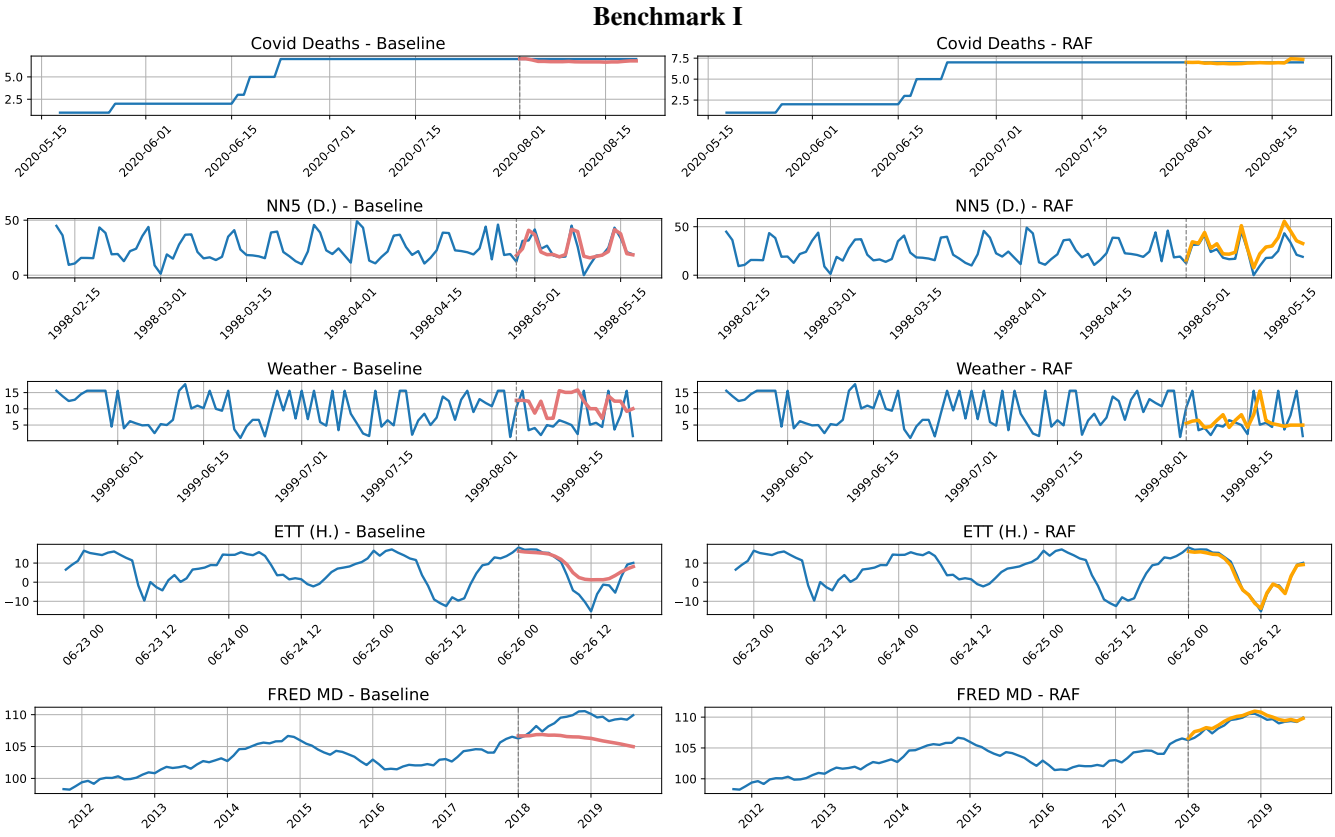

Figure 7: Qualitative results for Benchmark I datasets with $C = 75$ and $H = 20$ on Chronos Base.

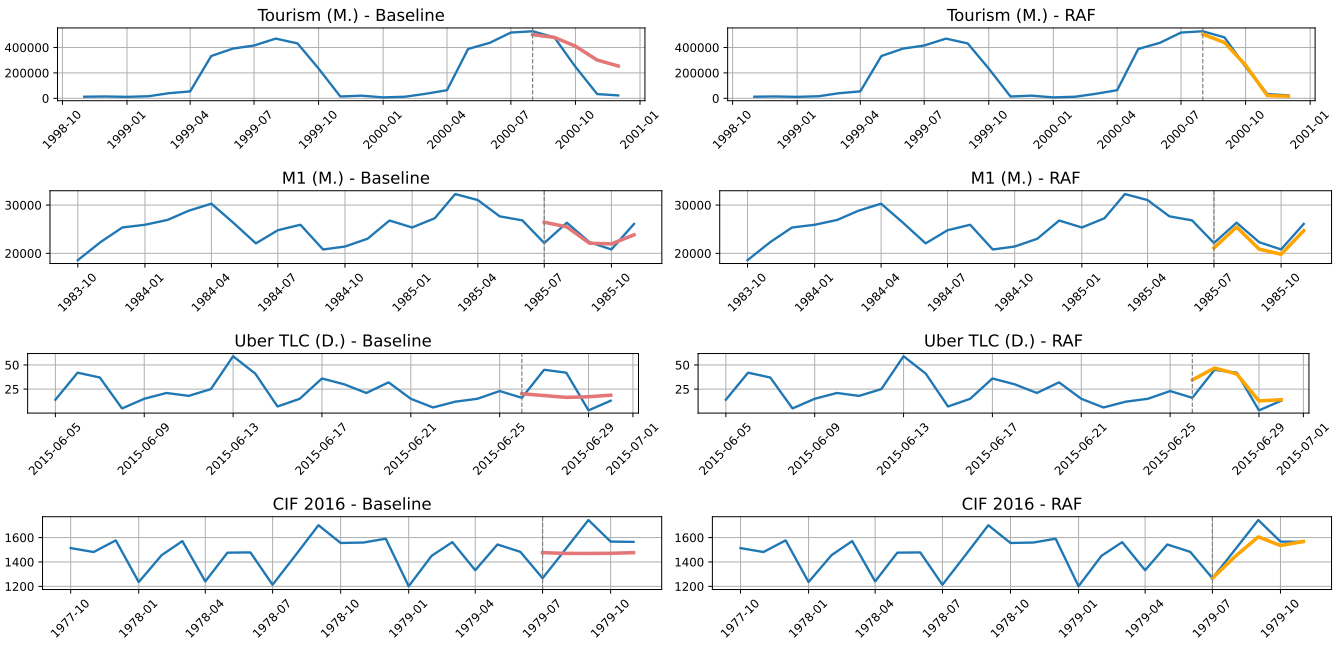

Figure 8: Qualitative results for Benchmark II datasets with $C = 21$ and $H = 5$ on Chronos Base.

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

Table 17: Overview of the Benchmark Datasets

To evaluate the performance of RAF, we have analyzed two distinct benchmark datasets: (a) Benchmark I datasets with long context and prediction lengths (6 datasets), and (b) Benchmark II datasets with short context and prediction lengths (5 datasets). In selecting the datasets, we adhere to the list used in the zero-shot and in-domain evaluation of Chronos to ensure that Chronos has no prior exposure to the data during training, thus guaranteeing a fair evaluation. For both Benchmarks I and II, we use the last H observations of each time series as the test set, following (Ansari et al. 2024). The prediction length $H \in \{10, 15, 20\}$ and context length $C \in \{50, 75, 100, 150\}$ are used for Benchmark I, while $H \in \{3, 4, 5\}$ and $C \in \{10, 15, 18, 21\}$ are used for Benchmark II. Both benchmarks exhibit a wide range of characteristics, such as dataset size and frequency, making them valuable benchmarks that closely resemble real-world scenarios. The specifications and lists of the datasets are provided below.

Benchmark I Datasets

Weather dataset ((Godahewa et al. 2021)) contains hourly 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.

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, Hibon, and Moser 1979)) contains 617 time series used in the M1 forecasting competition, covering areas such as microeconomic, 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.

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.