Predicting the Future by Retrieving the Past

Dazhao Du, Tao Han, Song Guo*

Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
{dduab, thanad}@connect.ust.hk, songguo@cse.ust.hk

Abstract

Deep learning models such as MLP, Transformer, and TCN have achieved remarkable success in univariate time series forecasting, typically relying on sliding window samples from historical data for training. However, while these models implicitly compress historical information into their parameters during training, they are unable to explicitly and dynamically access this global knowledge during inference, relying only on the local context within the lookback window. This results in an underutilization of rich patterns from the global history. To bridge this gap, we propose **Predicting the Future by Retrieving the Past (PFRP)**, a novel approach that explicitly integrates global historical data to enhance forecasting accuracy. Specifically, we construct a **Global Memory Bank (GMB)** to effectively store and manage global historical patterns. A retrieval mechanism is then employed to extract similar patterns from the GMB, enabling the generation of global predictions. By adaptively combining these global predictions with the outputs of any local prediction model, PFRP produces more accurate and interpretable forecasts. Extensive experiments conducted on seven real-world datasets demonstrate that PFRP enhances the average performance of advanced univariate forecasting models by 8.4%.

Introduction

Time series forecasting (TSF) has broad applications across various domains, including weather (Chen et al. 2023), finance (Sonkavde et al. 2023), transportation (Jiang and Luo 2022), and energy (Olivares et al. 2023). Recently, many advanced deep learning models have been proposed, such as Transformer-based (Zhou et al. 2021; Wang et al. 2024b), TCN-based (Wang et al. 2023), and MLP-based (Zeng et al. 2023) architectures. Despite their architectural differences, these models generally follow a common training and testing paradigm (Lim and Zohren 2021). As illustrated in Figure 1, an entire time series is typically partitioned into three intervals for training, validation, and test sets. A sliding window approach is then employed to create samples, each consisting of a lookback window sequence and a prediction horizon sequence. The training objective is to learn a mapping function from the lookback window sequence to the prediction horizon sequence. Upon completion of training, while the global historical information is implicitly compressed into the model's parameters, the original training data is typically discarded after training. Consequently, during inference, the trained model can only leverage the limited context within the current lookback window. These models are referred to as *local prediction models*, as they lack the ability to explicitly reference specific, relevant patterns from the entire historical sequence to inform the prediction at hand.

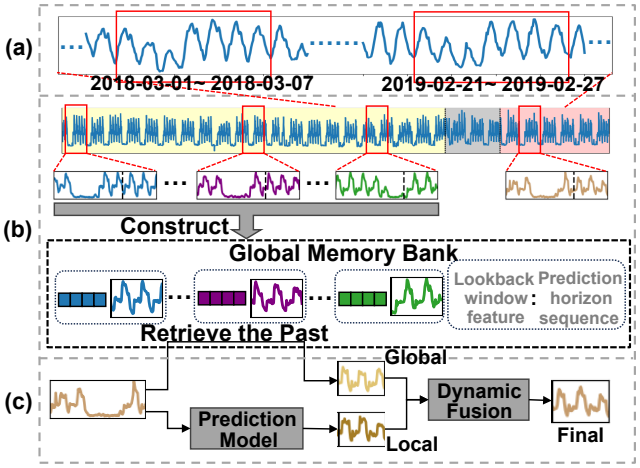

Figure 1: (a) Time series often contain highly similar subsequences across different periods. (b) The GMB is constructed from historical sliding window samples, containing pairs of lookback window features and their corresponding prediction horizon sequences. (c) During inference, relevant patterns are retrieved from the GMB to generate global prediction, which are then dynamically fused with local prediction from any prediction model to yield the final result.

However, we observe that time series often contain subsequences from different periods that exhibit remarkably similar patterns. For instance, in the household electricity consumption dataset (Zhou et al. 2021), the consumption pattern from a week in 2019 closely resembles that of a week in 2018, as shown in Figure 1(a). Both exhibit daily periodic fluctuations, with peaks gradually declining over the first three days and stabilizing at a high level for the subsequent four days. This observation suggests that TSF should

*Corresponding Author.

not only rely solely on the local lookback window but also leverage the global historical sequence as a reference. Inspired by the retrieval-augmented generation technique in NLP (Lewis et al. 2020; Guu et al. 2020), we propose a novel framework, Predicting the Future by Retrieving the Past (PFRP), which explicitly stores historical data and retrieves relevant sequences to enhance forecasting accuracy. PFRP addresses two key challenges: (1) *how to effectively retrieve relevant historical sequences* and (2) *how to integrate the retrieved sequences to improve current predictions*.

To address the first challenge, we construct a Global Memory Bank (GMB) (Chang et al. 2018) to store the lookback window features and corresponding prediction horizon sequences from historical training samples, as shown in Figure 1(b). The lookback window features serve as retrieval keys, while the prediction horizon sequences act as the associated values to be retrieved. During forecasting, we hypothesize that if the current lookback window sequence closely resembles a historical lookback window sequence, their respective prediction horizon sequences should also exhibit similar patterns. For instance, in Figure 1(a), the first three days of two weeks are similar, and their subsequent four days also exhibit similar patterns.

To address the second challenge, a naive approach is to directly copy the retrieved prediction horizon sequence as the current prediction result. However, this naive strategy performs suboptimally due to the inherent randomness and uncertainty in TSF. Furthermore, referencing multiple historical sequences is desirable to enhance robustness. To this end, we retrieve the top-k most similar lookback window features and compute a weighted combination of their corresponding prediction horizon sequences based on similarity scores. Additionally, we introduce a confidence gate and an output gate to dynamically adjust the weighting coefficients and modulate the scale and shift of the global prediction output. Although PFRP can independently generate global predictions, we find that dynamically fusing these global predictions with the local predictions of any other local prediction model yields superior prediction results.

Our contributions can be summarized as follows: (1) We propose the Global Memory Bank, a novel mechanism for explicitly storing and utilizing global historical data. (2) Building upon GMB, we introduce PFRP, a retrieval-based forecasting method that generates global predictions by retrieving and leveraging relevant historical patterns. (3) By seamlessly integrating global predictions with the outputs of any local prediction model, our model-agnostic approach significantly enhances univariate TSF performance.

Related Work

Time Series Forecasting

In the era of deep learning, numerous time series forecasting models have been proposed. From the perspective of model architectures, these models can be broadly categorized into RNN-based (Salinas et al. 2020), MLP-based (Oreshkin et al. 2020; Wang et al. 2024a), CNN-based (Wu et al. 2023; Wang et al. 2023), and Transformer-based (Lim et al. 2021; Du, Su, and Wei 2023; Zhou et al. 2022) approaches. These methods focus on designing effective strategies for temporal modeling. Autoformer (Wu et al. 2021) employs autocorrelation mechanisms, while DLinear (Zeng et al. 2023) utilizes simple linear mapping. From the data perspective, prediction models fall into two main categories: univariate and multivariate. Univariate models are designed to capture temporal patterns within a single variable, whereas multivariate models (Liu et al. 2024b) additionally address the inter-variable dependencies. However, PatchTST (Nie et al. 2023) demonstrated that repeatedly applying univariate forecasting along the variable dimension can effectively achieve multivariate forecasting. Our method also focuses on univariate TSF. Furthermore, recent years have witnessed a surge in models based on LLMs (Zhou et al. 2023; Jin et al. 2024) and pre-trained foundation models (Woo et al. 2024). Although various models have been developed, most are essentially local prediction models, as they rely solely on limited lookback windows for forecasting, without fully utilizing the entire historical sequence. In contrast, our method integrates global historical data into the forecasting process.

Method	Retrieval Criterion	Efficiency	Plug-and-Play
RATD	Feature Similarity	✗	✗
TimeRAF	Feature Similarity	✗	✗
TimeRAG	DTW Distance	✗	✗
RAFT	Pearson Correlation	✗	✗
PFRP	Feature Similarity	✓	✓

Table 1: Comparison between PFRP and existing RAG-based forecasting methods. Note that RATD, TimeRAF, and TimeRAG require diffusion models, time series foundation models, and LLMs respectively, resulting in significantly lower prediction efficiency. While RAFT requires retrieval across the entire training dataset, PFRP operates solely on a fixed-size memory bank, yielding higher retrieval efficiency.

Retrieval-Augmented Generation

In NLP, retrieval-augmented generation (RAG) combines pre-trained models with an external knowledge retrieval mechanism, allowing the model to dynamically access and integrate relevant information to enhance generation tasks (Lewis et al. 2020; Guu et al. 2020). Recently, this technique has been extended to time series forecasting. RATD (Liu et al. 2024a) retrieves the most relevant historical time series to guide the denoising process in diffusion models. RAF (Tire et al. 2024) and TimeRAF (Zhang et al. 2024) leverage retrieval-augmented techniques to enhance the zero-shot forecasting capabilities of time series foundation models, while TimeRAG (Yang et al. 2024) focuses on improving LLM-based time series forecasting using retrieval-augmented methods. RAFT (Han et al. 2025) leverages retrieval from training data to augment the input. Table 1 lists the comparison between PFRP and these approaches. Compared to them, PFRP offers a more efficient retrieval mechanism and prediction generation strategy. Additionally, PFRP is model-agnostic and adaptable for enhancing any existing univariate forecasting model.

Methodology

Definition and Overview

For univariate time series forecasting, suppose the training set comprises N historical sliding-window samples, denoted as $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(N)}, y^{(N)})\}$, where $x^{(i)} \in \mathbb{R}^L$ represents the lookback window sequence and $y^{(i)} \in \mathbb{R}^H$ denotes the prediction horizon sequence. Typically, a local prediction model is trained using these samples to learn the mapping function between $x^{(i)}$ and $y^{(i)}$. During inference, the current lookback window sequence x is input into the trained model to generate the corresponding prediction y. Consequently, the training set is discarded once the prediction model is trained. In contrast, our method retains part of the training set and explicitly leverages the rich historical information it contains.

Our proposed method consists of two main stages. The first stage involves constructing a *Global Memory Bank (GMB)* to store historical information. As illustrated in Figure 2, we introduce *Predictive Contrastive Learning (PCL)* to train a encoder that encodes the lookback window sequences of all historical samples into high-level features. To reduce redundancy and improve retrieval efficiency, we apply *K-medoids clustering* in the feature space, retaining only the cluster medoids. The second stage focuses on prediction through GMB retrieval, i.e., predicting the future by retrieving the past, as shown in Figure 3. The global prediction retrieved from the GMB is dynamically fused with the local prediction produced by any prediction model to generate the final forecast. These two stages are detailed below.

Global Memory Bank

Predictive Contrastive Learning To retrieve relevant historical samples based on the lookback window, some methods directly measure the similarity of lookback window sequences using DTW or MSE (Yang et al. 2024; Zhang et al. 2024). In contrast, we measure similarity at the feature level. To achieve this, we introduce contrastive learning (Chen et al. 2020) to train an MLP-based feature encoder for the lookback window sequences. We propose a new strategy for selecting positive and negative samples. Specifically, instead of selecting based on the MSE between lookback window sequences, we select them based on the MSE between their corresponding prediction horizon sequences. Intuitively, this training objective encourages lookback window sequences with more similar future to be closer in the feature space, which facilitates the retrieval of historical samples that are more helpful for the current prediction. We refer to this method as Predictive Contrastive Learning (PCL), as shown in Figure 2(a). Specifically, for i-th sample $(x^{(i)}, y^{(i)})$ in a training batch $\{(x^{(1)}, y^{(1)}), \cdots, (x^{(B)}, y^{(B)})\}$, where B is the batch size, its positive sample index i^+ is:

$$i^+ = \arg \min_{1 \le j \le B, j \neq i} \|y_i - y_j\|_2^2. \quad (1)$$

Other samples in the same batch can be regarded as negative samples. We pass the lookback window sequence $x^{(i)}$ of each sample through the encoder to obtain the feature $\epsilon^{(i)}$.

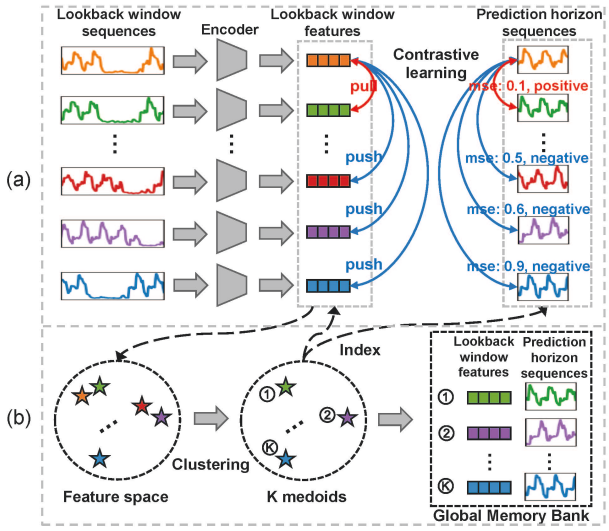

Figure 2: Construction of GMB. (a) *Predictive Contrastive Learning*. Positive sample pairs are identified as those whose prediction horizon sequences exhibit the lowest MSE. PCL aims to pull the encoded lookback window sequences of positive pairs closer in feature space. (b) *K-medoids Clustering*. Retain only K representative medoids in feature space to construct GMB, which stores both their lookback window features and corresponding prediction horizon sequences.

The objective function for PCL is then defined as:

$$\mathcal{L}_{pcl} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp\left(\epsilon^{(i)} \cdot \epsilon^{(i^+)} / \tau\right)}{\sum_{j=1, j \neq i}^{B} \exp\left(\epsilon^{(i)} \cdot \epsilon^{(j)} / \tau\right)}, \quad (2)$$

where τ is the temperature.

K-medoids Clustering Using the time series encoder (a MLP) trained by PCL to encode the lookback window sequences of all training samples, we can obtain a new set $\{(\epsilon^{(1)}, y^{(1)}), (\epsilon^{(2)}, y^{(2)}), \cdots, (\epsilon^{(N)}, y^{(N)})\}$, where each sample is represented as a pair of a lookback window feature $\epsilon^{(i)}$ and its corresponding prediction horizon sequence $y^{(i)}$. To reduce redundancy and improve retrieval efficiency, we apply K-medoids clustering (Park and Jun 2009) to the lookback window features, retaining only the samples corresponding to the K cluster medoids, as shown in Figure 2(b). A key advantage of K-medoids over alternatives like K-means is its use of actual historical samples as cluster centroids rather than synthetic averages. This exemplar-based approach is essential for our task, as it ensures the patterns stored in our GMB represent authentic and coherent historical sequences. These selected samples are stored in the GMB, which can be formalized as $\{(\epsilon^{(1)}, y^{(1)}), (\epsilon^{(2)}, y^{(2)}), \cdots, (\epsilon^{(K)}, y^{(K)})\}$.

Predicting the Future by Retrieving the Past

Retrieving from Global Memory Bank Firstly, the current lookback window sequence x is encoded by the encoder into a feature vector ϵ. This vector serves as the query and

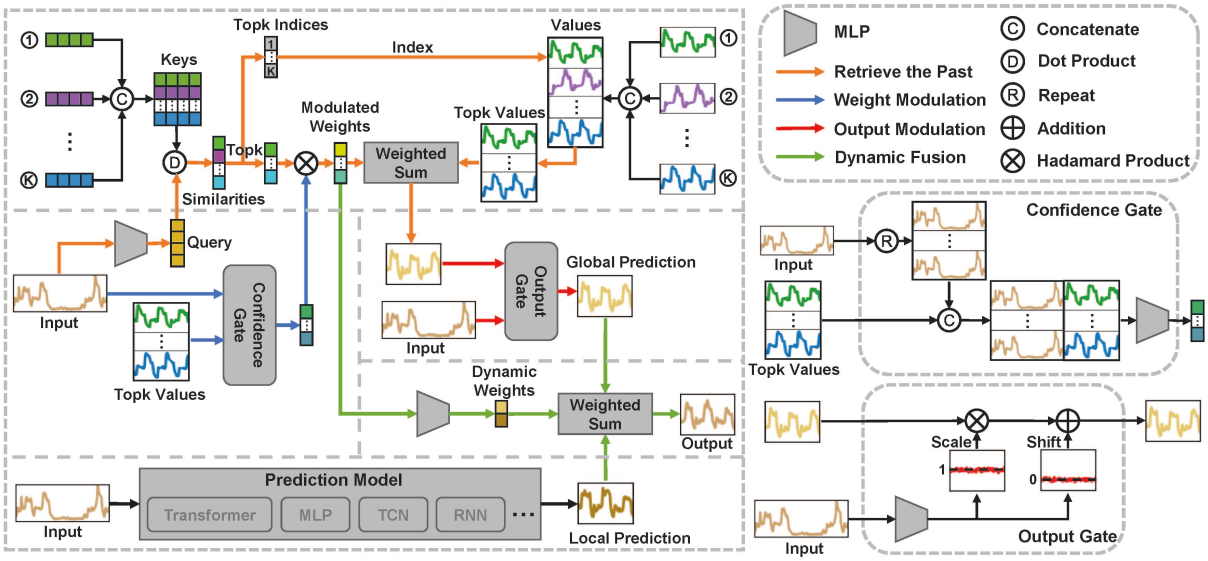

Figure 3: Schematic diagram of PFRP. The diagram can be interpreted through the following key processes: (1) Orange arrows represent the retrieval process from GMB. (2) Blue arrows denote the weight modulation process. (3) Red arrows illustrate the output modulation process, which generates the global prediction. (4) The local prediction process is depicted at the bottom. (5) Green arrows indicate the fusion of global and local predictions based on dynamic weights. (6) The bottom-right section details the structures of the confidence gate and output gate.

is used to compute cosine similarity with the K historical lookback window features (keys) stored in the GMB:

$$w^{(i)} = \epsilon \cdot \epsilon^{(i)}, \quad i = \{1, \cdots, K\}. \tag{3}$$

Assume the indices of the top-k biggest similarities are given by $\{a_1, \cdots, a_k\}$. The top-k biggest similarities are $\{w^{(a_1)}, \cdots, w^{(a_k)}\}$, and the corresponding prediction horizon sequences (values) are $\{y^{(a_1)}, \cdots, y^{(a_k)}\}$.

The naive implementation of PFRP can be adopting an attention-like (Vaswani et al. 2017) operation, where the query-key similarities are regarded as weights to compute a weighted sum of the values to generate the output. To adaptively regulate the entire process and enhance model capacity, we further introduce two learnable components: (i) a *confidence gate* that adaptively adjusts attention weights, and (ii) an *output gate* that modulates the global prediction.

Confidence Gate Historical lookback window sequences that are more similar to x do not necessarily indicate that their corresponding prediction horizon sequences are more likely to represent the future of x, nor do they warrant larger weights. Therefore, we design a confidence gate to modulate the weights. If the retrieved historical prediction horizon sequence $y^{(a_i)}$ better matches the current lookback window sequence x, then the complete sequence formed by concatenating them over time is more likely to exist, suggesting that the corresponding $y^{(a_i)}$ is more likely to represent the true future of x. To achieve this, we concatenate the top-k retrieved values (i.e., the k historical prediction horizon sequences) with x to form k complete sequences $\{[x; y^{(a_1)}], [x; y^{(a_2)}], \cdots, [x; y^{(a_k)}]\}$. Next, we use an MLP with a sigmoid activation function to output the existence probability for each of these k complete sequences:

$$p_i = \text{Sigmoid}(\text{MLP}([x; y^{(a_i)}])), \quad p_i \in (0, 1). \tag{4}$$

These probabilities are then used to directly modulate the weights by multiplying them with the original top-k weights $\{w^{(a_1)}, w^{(a_2)}, \cdots, w^{(a_k)}\}$:

$$\overline{w}^{(a_1)}, \cdots, \overline{w}^{(a_k)} = \text{Softmax}(w^{(a_1)} \cdot p_1, \cdots, w^{(a_k)} \cdot p_k). \tag{5}$$

Output Gate The retrieved values are aggregated using a weighted sum based on the modulated weights to obtain the initial global prediction $\overline{y}_1$:

$$\overline{y}_1 = \sum_{i=1}^{k} \overline{w}^{(a_i)} \cdot y^{(a_i)}. \tag{6}$$

However, the future sequence to be predicted may exhibit a pattern similar to a certain past sequence but might not align perfectly in terms of scale and shift. To address this, we employ an output gate to dynamically modulate the output based on the current lookback window sequence x. The output gate refines the initial global prediction $\overline{y}_1$ via learnable scale and shift. Specifically, x is fed into an MLP that outputs two sequences, $\alpha \in \mathbb{R}^H$ and $\beta \in \mathbb{R}^H$. α represents the scale of the prediction horizon sequence and is initialized to all ones, while β represents the shift and is initialized to all zeros. Then the global prediction y_1 is formulated as:

$$y_1 = \alpha \cdot \overline{y}_1 + \beta. \tag{7}$$

Dynamic Fusion If no highly similar sequences exist in the historical data for the current lookback window, relying solely on GMB may lead to inaccurate predictions, and the model should reduce reliance on the

Models	SparseTSF		+PFRP		DLinear		+PFRP		PatchTST		+PFRP		TimesNet		+PFRP	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	0.2404	0.3082	**0.1919**	**0.2710**	0.2778	0.3660	**0.1793**	**0.2720**	0.1797	0.2680	**0.1712**	**0.2416**	0.2165	0.3065	**0.1799**	**0.2631**
Electricity	0.4968	0.5207	**0.3561**	**0.4214**	0.3951	0.4579	**0.3666**	**0.4351**	0.4200	0.4620	**0.3869**	**0.4456**	0.4264	0.4644	**0.3950**	**0.4517**
Weather	0.6452	0.5724	**0.6365**	**0.5659**	0.7250	0.5961	**0.6751**	**0.5735**	0.6542	0.5719	**0.6426**	**0.5663**	0.6808	0.5853	**0.6382**	**0.5676**
ETTh1	0.0841	0.2247	**0.0766**	**0.2141**	0.1160	0.2594	**0.1122**	**0.2564**	0.0802	0.2180	**0.0769**	**0.2145**	0.0785	0.2172	**0.0766**	**0.2147**
ETTh2	0.2024	0.3537	**0.1915**	**0.3437**	0.2242	0.3694	**0.2147**	**0.3621**	0.2008	0.3531	**0.1893**	**0.3411**	0.1926	0.3459	**0.1917**	**0.3442**
ETTm1	0.0536	0.1752	**0.0523**	**0.1710**	0.0637	0.1838	**0.0627**	**0.1828**	0.0534	0.1735	**0.0527**	**0.1719**	0.0537	0.1744	**0.0526**	**0.1717**
ETTm2	0.1263	0.2684	**0.1203**	**0.2570**	0.1255	0.2622	**0.1239**	**0.2596**	0.1244	0.2621	**0.1218**	**0.2585**	0.1236	0.2606	**0.1214**	**0.2584**

Table 2: Univariate forecasting results for four baselines with and without PFRP. Lower metric values indicate better performance, and the better results are highlighted in bold. The lookback window length L is set to 96. The results are averaged across all prediction horizons $H = \{96, 192, 336, 720\}$.

global prediction. In such cases, the modulated weights $\{\overline{w}^{(a_1)}, \overline{w}^{(a_2)}, \cdots, \overline{w}^{(a_k)}\}$ tend to be relatively small. Therefore, we input the current lookback window sequence x into a local prediction model to generate a local prediction y_2. By dynamically fusing the global prediction y_1 with the local prediction y_2, we achieve a more accurate forecast. Specifically, the modulated weights $\{\overline{w}^{(a_1)}, \overline{w}^{(a_2)}, \cdots, \overline{w}^{(a_k)}\}$ which reflect the importance of global versus local predictions are fed into an MLP followed by a Softmax activation function to dynamically compute the fusion weights:

$$w_1, w_2 = \text{Softmax}(\text{MLP}(\overline{w}^{(a_1)}, \overline{w}^{(a_2)}, \cdots, \overline{w}^{(a_k)})). \quad (8)$$

Then the global prediction y_1 and local prediction y_2 are aggregated using a weighted sum based on the dynamic fusion weights w_1, w_2:

$$y = w_1 \cdot y_1 + w_2 \cdot y_2, \quad (9)$$

where y is the final prediction result.

Experiments

Datasets We perform extensive experiments on seven datasets across three domains, including Traffic, Electricity, Weather, and four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2). While these datasets include multiple variables, our study focuses exclusively on univariate TSF. Accordingly, for each dataset, we consider only the time series of the last variable. Adhering to the standard forecasting setup (Wu et al. 2021), we fix the lookback window length L at 96, with the prediction horizon H varying across $\{96, 192, 336, 720\}$. Forecasting performance is evaluated using the MSE and MAE metrics.

Baselines We select four SOTA local prediction models as the main baselines: two MLP-based models (DLinear (Zeng et al. 2023) and SparseTSF (Lin et al. 2024)), one Transformer-based model (PatchTST (Nie et al. 2023)), and one CNN-based model (TimesNet (Wu et al. 2023)). We further investigate whether PFRP can enhance the predictive performance of state-of-the-art large time-series models, including the LLM-based models (TimeCMA (Liu et al. 2025a)) and time series foundation models (Moirai (Woo et al. 2024) and Sundial (Liu et al. 2025b)). For these large models with strong zero-shot capabilities, we freeze their pretrained parameters and finetune only the PFRP-specific parameters. Additionally, we also compare PFRP with two RAG-based prediction methods: RATD (Liu et al. 2024a) and RAFT (Han et al. 2025).

Implementation Details **First stage:** When training the lookback window encoder with PCL, we set the batch size to 256, the temperature to 0.05, and the learning rate to 0.001. To identify positive and negative samples within the training batch, we exclude those with significant temporal overlap ($>$ 48 same timestamps) with the anchor. For constructing the GMB, the lookback window length is set to 96, and the prediction horizon to 720. As a result, the GMB does not need to be reconstructed for each prediction horizon. For example, when the prediction horizon is 96/192/336, we simply extract the first 96/192/336 time steps from the stored prediction horizon sequence as the retrieved sequence. **Second stage:** When training four baseline models and our PFRP, we use the Adam optimizer (Kingma and Ba 2014) with an L2 loss function and an initial learning rate of 0.0001. For different local prediction models, we follow the batch size, training epochs, and hyperparameter settings from their official implementations. Each experiment is repeated three times and we report the average results.

Main Result

Table 2 presents the forecasting results of baseline models and PFRP across seven datasets. We observe that PFRP consistently enhances the performance of these local prediction models. The most significant improvements are observed in DLinear and SparseTSF, two simple MLP-based models, with average gains of 7.1% and 8.4% across all datasets. For more complex models like PatchTST and TimesNet, the performance boost from PFRP is slightly smaller. On datasets with strong regularities, such as Traffic and Electricity, PFRP improves the forecasting accuracy of all local prediction models by an average of 17.4% and 10.1%, respectively. This suggests that in datasets with well-defined

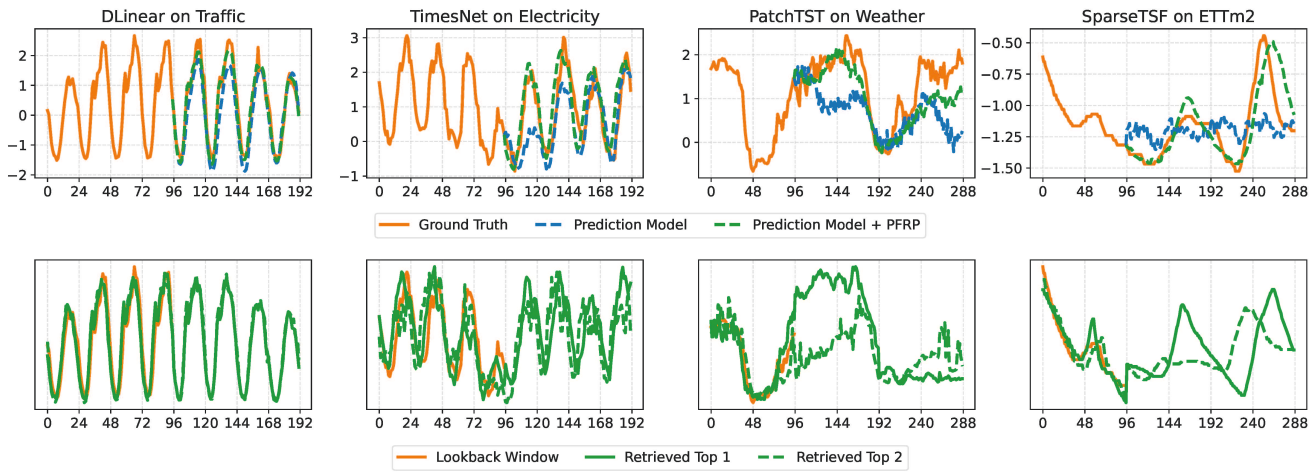

Figure 4: *Top:* Visualization of the prediction results for four baseline models, both with and without PFRP. *Bottom:* Visualization of the top 2 most relevant historical sequences retrieved by PFRP.

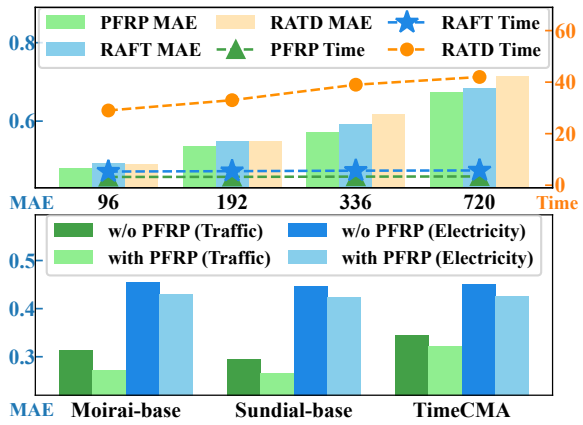

Figure 5: *Top:* Performance and efficiency comparison with two RAG-based methods. *Bottom:* Performance comparison of three large time-series models (with/without PFRP).

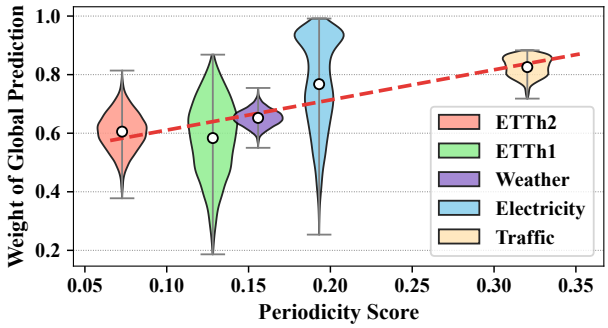

Figure 6: The x-axis represents the datasets' periodicity scores, while the y-axis represents the weights w_1 of the global prediction from the Dynamic Fusion module.

periodic patterns, similar sequences frequently recur, making our retrieval-based prediction approach particularly effective. Figure 4 visualizes the predictions generated by the baselines and PFRP. Figure 5 (top) further compares PFRP (+SparseTSF) with two other RAG-based prediction methods. PFRP consistently outperforms both alternatives across varying prediction horizons. PFRP also exhibits superior test-time inference speed by retrieving solely from the GMB, unlike RAFT, which requires exhaustive traversal of the entire training set. Conversely, RATD shows the lowest efficiency due to its multi-step diffusion sampling. In Figure 5 (bottom), we find that incorporating large time series models (LLM-based models or foundation models) into our PFRP framework can also enhance performance.

More Analysis

Can PFRP effectively retrieve relevant sequences to improve prediction performance? The bottom row in Figure 4 visualizes the top two relevant historical sequences retrieved by PFRP. On the relatively simpler Traffic and Electricity datasets, the retrieved historical sequences compensate for peaks, refining the local predictions. On more challenging datasets such as Weather and ETT, while precise forecasting remains difficult, the retrieved historical sequences still help align predictions more closely with the actual future values. Besides, retrieved past temporal patterns provides interpretability to the current forecasting process.

Does PFRP demonstrate performance improvements on datasets with weaker periodicity? To quantify the degree of periodicity in time series datasets, we combine autocorrelation at predefined lags (e.g., daily, weekly) with the inverse of normalized entropy. The resulting score, ranging from 0 to 1, reflects the strength of the dataset's periodic structure. As Figure 6 illustrates, the Traffic and Electricity datasets exhibit the strongest periodicity, while the four ETT datasets show the weakest. Although improvements on the ETT datasets are less pronounced in Table 2, PFRP still achieves average gains of 3.4%, 3.1%, 1.6%, and 2.2%, respectively. We further observe that the global prediction

		Traffic		Electricity	
		MSE	MAE	MSE	MAE
Retrieval Criterion	Feature	**0.1919**	**0.2710**	**0.3561**	**0.4214**
	MSE	0.1928	0.2752	0.3716	0.4296
	DTW	0.1922	0.2758	0.3659	0.4248
	PCC	0.2020	0.2809	0.3789	0.4339
Encoder Type	MLP	0.1919	0.2710	0.3561	0.4214
	PatchTST	0.1985	0.2760	**0.3492**	**0.4175**
	TimesNet	**0.1754**	**0.2599**	0.3754	0.4319
Training Strategy	PCL	**0.1919**	**0.2710**	**0.3561**	**0.4214**
	CL	0.2101	0.2828	0.3833	0.4350
	PL	0.2250	0.3012	0.3762	0.4325

Table 3: Ablations about GMB on two datasets.

weight (w_1) is directly proportional to the dataset's periodicity: for more periodic data, the final prediction relies more heavily on retrieval-based global prediction, shifting dependence away from the local prediction model's output.

Ablations about GMB We explore various aspects of the GMB, including retrieval criterion, encoder types, and encoder training strategies. (1) **Retrieval Criterion**: We use cosine similarity of encoded features as the retrieval criterion and compare it with non-feature-based approaches that directly measure raw lookback window sequence similarities. For instance, TimeRAG (Yang et al. 2024) employs DTW, while we also test MSE and the Pearson Correlation Coefficient (PCC). However, these methods prove less effective than our feature-based cosine similarity retrieval. (2) **Lookback Window Encoder**: We replace the default MLP encoder with two widely used time series encoders, PatchTST and TimesNet. Results show that the optimal encoder may vary by dataset and needs to be determined experimentally: TimesNet achieves the highest accuracy on the Traffic dataset, while PatchTST excels on Electricity. However, PFRP improves local prediction regardless of the encoder used. For efficiency in building the GMB, we select MLP as the default encoder. (3) **Encoder Training Strategy**: We compare three training strategies. Predictive Learning (PL) attaches a prediction head to the encoder and optimizes it with a standard forecasting task, as used in RATD (Liu et al. 2024a). Contrastive Learning (CL) selects positive samples based on lookback window sequence similarity. In contrast, our Predictive Contrastive Learning (PCL) selects positives based on the similarity of their prediction horizon sequences. Compared to the other two methods, PCL aligns better with the retrieval-based forecasting objective, enabling it to identify historical samples with more similar future behaviors.

Ablations about PFRP We conduct ablation experiments to investigate the impact of three learnable components in PFRP: the confidence gate, the output gate, and the local prediction model. The results in Table 4 show that removing either of the two gates, or both, leads to a decline in forecasting performance. Furthermore, we observe that even when the local prediction model is removed, leaving only the

	Traffic		Electricity	
	MSE	MAE	MSE	MAE
SparseTSF	0.2404	0.3082	0.4968	0.5207
SparseTSF+PFRP	0.1919	0.2710	**0.3561**	**0.4214**
w/o confidence gate	0.2385	0.3058	0.3960	0.4486
w/o output gate	0.2130	0.2989	0.5140	0.5296
w/o both gates	0.2128	0.2987	0.5763	0.5643
w/o prediction model	**0.1686**	**0.2476**	0.3952	0.4494

Table 4: Ablations about PFRP on two datasets.

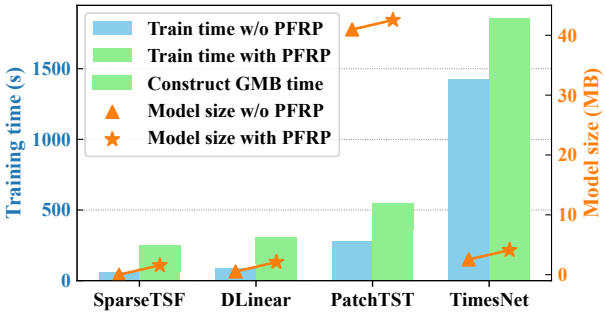

Figure 7: Total consumed time and model sizes of four baseline models with and without PFRP.

global prediction results, our method still outperforms the baseline. On the Electricity dataset, dynamically combining global and local predictions yields better results. Conversely, on the Traffic dataset, relying solely on global predictions can achieve even better performance.

Efficiency Analysis Figure 7 compares the model size and training time of four baseline models with and without PFRP, using the Electricity dataset (prediction horizon is 720). For a fair comparison, all methods were trained for 10 epochs. While GMB construction adds a fixed 186 seconds (134 seconds for PCL, 52 seconds for K-medoids clustering), PFRP's overall impact on efficiency is minimal. It consistently increases model size by only 1.57 MB and causes only a slight rise in training duration, underscoring PFRP's practicality and efficiency.

Conclusion

We introduce PFRP, a retrieval-enhanced univariate time series forecasting framework that explicitly incorporates historical patterns to enhance prediction accuracy. Unlike conventional local prediction models that rely solely on a fixed-length lookback window, our approach leverages a Global Memory Bank (GMB) to store and retrieve relevant historical sequences, seamlessly integrating them into the forecasting process. Our method dynamically combines global retrieval-based predictions with local model outputs, leading to superior performance. Extensive experiments validate the effectiveness of PFRP, demonstrating its ability to capture global correlations and enhance forecasting accuracy across different local prediction models.

Acknowledgments

This research was supported by fundings from the Hong Kong RGC General Research Fund (152169/22E, 152228/23E, 162161/24E, 162116/25E), Research Impact Fund (No. R5060-19, No. R5011-23), Collaborative Research Fund (No. C1042-23GF), NSFC/RGC Collaborative Research Scheme (Grant No. 62461160332 & CRS_HKUST602/24), Areas of Excellence Scheme (AoE/E-601/22-R), and the InnoHK (HKGAI).

References

Chang, Y.-Y.; Sun, F.-Y.; Wu, Y.-H.; and Lin, S.-D. 2018. A memory-network based solution for multivariate time-series forecasting. *arXiv preprint arXiv:1809.02105*.

Chen, K.; Han, T.; Gong, J.; Bai, L.; Ling, F.; Luo, J.-J.; Chen, X.; Ma, L.; Zhang, T.; Su, R.; et al. 2023. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Du, D.; Su, B.; and Wei, Z. 2023. Preformer: predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, 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*.

Jiang, W.; and Luo, J. 2022. Graph neural network for traffic forecasting: A survey. *Expert systems with applications*, 207: 117921.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; and Wen, Q. 2024. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

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.

Lim, B.; and Zohren, S. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194): 20200209.

Lin, S.; Lin, W.; Wu, W.; Chen, H.; and Yang, J. 2024. SparseTSF: Modeling Long-term Time Series Forecasting with 1k Parameters. In *Forty-first International Conference on Machine Learning*.

Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2025a. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 18, 18780–18788.

Liu, J.; Yang, L.; Li, H.; and Hong, S. 2024a. Retrieval-Augmented Diffusion Models for Time Series Forecasting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024b. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*.

Liu, Y.; Qin, G.; Shi, Z.; Chen, Z.; Yang, C.; Huang, X.; Wang, J.; and Long, M. 2025b. Sundial: A Family of Highly Capable Time Series Foundation Models. In *Forty-second International Conference on Machine Learning*.

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

Olivares, K. G.; Challu, C.; Marcjasz, G.; Weron, R.; and Dubrawski, A. 2023. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39(2): 884–900.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*.

Park, H.-S.; and Jun, C.-H. 2009. A simple and fast algorithm for K-medoids clustering. *Expert systems with applications*, 36(2): 3336–3341.

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.

Sonkavde, G.; Dharrao, D. S.; Bongale, A. M.; Deokate, S. T.; Doreswamy, D.; and Bhat, S. K. 2023. Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications. *International Journal of Financial Studies*, 11(3): 94.

Tire, K.; Taga, E. O.; Ildiz, M. E.; and Oymak, S. 2024. Retrieval Augmented Time Series Forecasting. *arXiv preprint arXiv:2411.08249*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2023. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*.

Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and ZHOU, J. 2024a. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *International Conference on Learning Representations*.

Wang, Y.; Wu, H.; Dong, J.; Liu, Y.; Qiu, Y.; Zhang, H.; Wang, J.; and Long, M. 2024b. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*.

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.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

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.

Yang, S.; Wang, D.; Zheng, H.; and Jin, R. 2024. TimeRAG: BOOSTING LLM Time Series Forecasting via Retrieval-Augmented Generation. *arXiv preprint arXiv:2412.16643*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, H.; Xu, C.; Zhang, Y.-F.; Zhang, Z.; Wang, L.; Bian, J.; and Tan, T. 2024. TimeRAF: Retrieval-Augmented Foundation model for Zero-shot Time Series Forecasting. *arXiv preprint arXiv:2412.20810*.

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, 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*, 27268–27286. PMLR.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36: 43322–43355.