

VRATF: Vision-based Retrieval-Augmented Time Series Forecasting

Zhun Xie

Chongqing Bosscom Science&Technology Co.,Ltd

Jing Ling

Chongqing University of Posts and Telecommunications

Zhihao Zhang

XZ2296048925@163.com

Shandong University

Guifeng Xia

Chongqing Bosscom Science&Technology Co.,Ltd

Research Article

Keywords: Time series forecasting, Retrieval-augmented, Multimodality, Visual time series

Posted Date: October 26th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11111645/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

VRATF: Vision-based Retrieval-Augmented Time Series Forecasting

Zhun Xie¹, Jing Ling², Zhihao Zhang^{3*}, Guifeng Xia⁴

¹ Chongqing Bosscom Science&Technology Co., Ltd., Chongqing, 400065, China.

² Chongqing University of Posts and Telecommunications, Chongqing, 400065, China.

^{3*} Shandong University, Shandong, 250100, China.

⁴Chongqing Bosscom Science&Technology Co., Ltd., Chongqing, China.

*Corresponding author(s). E-mail(s): XZ2296048925@163.com;
Contributing authors: zhunxie39@gmail.com;
d250201030@stu.cqupt.edu.cn; xiafeng919@163.com;

Abstract

Time series forecasting is essential for decision-making in energy, finance, meteorology, and other real-world applications. Recent retrieval-augmented forecasting methods improve prediction by retrieving similar historical segments as external references, but retrieval based directly on raw time series is often sensitive to high-frequency fluctuations, anomalous spikes, and local numerical variations. To address this limitation, this paper proposes VRATF, a vision-based retrieval-augmented time series forecasting framework. VRATF first extracts low-frequency trend components from historical sequences using frequency-domain filtering, renders the denoised trends as line-chart images, and encodes them with a frozen vision encoder to construct a visual retrieval database. Given a query sequence, visually similar historical trends are retrieved, and their corresponding future trajectories are aggregated through similarity-aware Gaussian weighting to produce a retrieval-based forecast. In parallel, a lightweight DLinear forecaster generates a base prediction from the current input sequence. The two prediction streams are then combined through a learnable prediction-level fusion module, followed by a calibration layer to obtain the final forecast. Experiments on six multivariate forecasting benchmarks show that VRATF achieves competitive and often superior performance compared with recent retrieval-augmented and representative forecasting baselines, particularly in several long-horizon settings, while keeping the trainable forecasting components lightweight.

Keywords: Time series forecasting, Retrieval-augmented, Multimodality, Visual time series

1 Introduction

Time series forecasting aims to predict future values over multiple time steps based on historical observations. It plays a fundamental role in critical applications such as electricity load dispatching [1], traffic flow management [2], financial risk control [3], and weather forecasting [4]—accurate forecasts provide decision-makers with sufficient lead time to optimize resource allocation and reduce operational risks.

Time series forecasting has evolved from recurrent and convolutional neural networks (RNNs[5] and CNNs[6]) to Transformer-based architectures [7], lightweight linear models such as DLinear [8] and N-BEATS [9], and, more recently, large-scale time series foundation models (e.g., MOIRAI [10], Chronos [11], and TimesFM [12]) that learn universal temporal representations through pretraining. Inspired by the success of retrieval-augmented generation (RAG) [13], recent studies have introduced retrieval augmentation into time series forecasting. For example, RAFT [14] enhances forecasting by retrieving similar historical segments, while DUET [15] addresses temporal distribution shift and channel dependencies through dual clustering on the temporal and channel dimensions.

However, existing retrieval-augmented forecasting methods still suffer from two fundamental limitations: the retrieval target and the fusion strategy. **First, at the retrieval stage, similarity is typically computed on raw time series, making retrieval highly susceptible to high-frequency noise.** Random fluctuations, sensor jitter, and anomalous spikes contribute equally to similarity computation despite carrying little predictive value, often leading to the retrieval of historically similar but temporally uninformative segments. By contrast, retrieving based on the underlying trend enables the model to identify historical segments with more consistent temporal dynamics, providing more informative references for future forecasting. This phenomenon is illustrated in Fig. 1, which compares the retrieval results of different retrieval paradigms under the same query sequence. Moreover, trend patterns are naturally represented as line charts, where high-level geometric characteristics such as slopes, curvatures, and turning points become visually distinguishable. Recent vision foundation models [16, 17] have demonstrated remarkable capability in learning such structural representations. Motivated by this observation, we perform retrieval in the visual embedding space rather than the original numerical space, enabling similarity matching based on global trend morphology instead of point-wise numerical correspondence. **Second, at the fusion stage, existing methods lack an explicit prediction-level mechanism to balance the contributions of the base predictor and retrieved references.** Fixed-weight or projection-based fusion applies a uniform combination strategy across different forecasting settings, while gating mechanisms typically operate at the local time-step level and cannot globally regulate retrieval participation. Moreover, because fusion is performed in latent space, the contributions of the two prediction sources remain difficult to interpret and control.

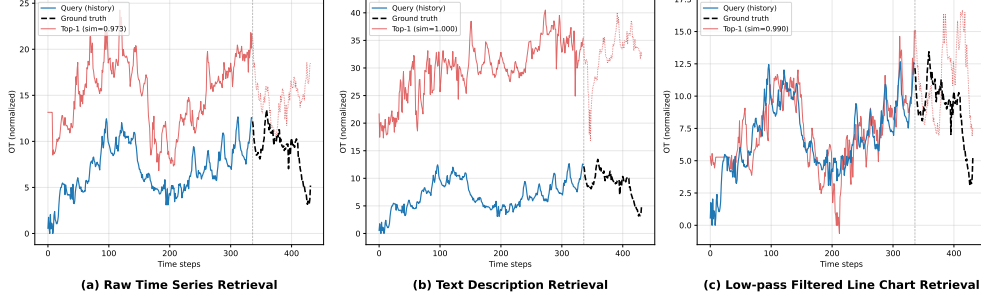


Fig. 1 Comparison of retrieval results under the identical query sequence. Subplots: (a) raw time series retrieval; (b) text description retrieval; (c) retrieval via low-pass filtered line charts.

To address these limitations, we propose **VRATF** (**V**ision-based **R**etrieval-Augmented **T**ime Series **F**orecasting). Instead of retrieving on raw sequences, VRATF performs retrieval on noise-filtered trend representations encoded by a frozen vision encoder, enabling more reliable retrieval of historically relevant patterns. Furthermore, VRATF introduces a lightweight learnable prediction-level fusion mechanism that explicitly balances the contributions of the base predictor and retrieved forecasts. Extensive experiments demonstrate that VRATF achieves competitive and often superior performance compared with existing retrieval-augmented forecasting methods while maintaining high parameter efficiency.

The main contributions of this paper are summarized as follows:

- We propose VRATF, a retrieval-augmented forecasting framework that performs retrieval on denoised visual trend representations rather than raw time series.
- We design a channel-independent retrieval mechanism and an interpretable learnable prediction-level fusion strategy to improve retrieval quality and explicitly balance retrieved and base predictions.
- Extensive experiments on six benchmark datasets demonstrate that VRATF achieves competitive forecasting performance with high parameter efficiency, with clear advantages on several long-horizon settings.

2 Related Work

2.1 Time Series Forecasting

Time series forecasting has been widely studied due to its importance in practical applications such as energy management, traffic planning, finance, and weather forecasting. Early deep learning approaches commonly model temporal dependencies using recurrent or convolutional architectures, such as LSTM [5], TCN [6], and LSTNet [18]. These models are effective for capturing local or sequential temporal patterns, but their ability to represent complex temporal structures is often limited by the design of the sequence encoder.

Transformer-based models have become an important direction for forecasting by improving the modeling of temporal dependencies and non-stationary patterns. Informer

[7] improves the efficiency of self-attention for long sequence modeling, Autoformer [19] introduces decomposition and auto-correlation mechanisms, and Non-stationary Transformer [20] addresses distribution shifts in non-stationary time series. PatchTST [21] segments time series into patches, while TimesNet [22] models temporal variations from a two-dimensional perspective. In parallel, lightweight architectures have also shown strong performance. DLinear [8] demonstrates that simple decomposition-based linear models can be highly competitive, N-BEATS [9] uses basis expansion for interpretable forecasting, and recent MLP-based models such as TimeMixer [23] and CycleNet [24] further exploit multi-scale or periodic temporal structures. More recently, time series foundation models, including MOIRAI [10], Chronos [11], and TimesFM [12], have explored large-scale pretraining for general-purpose forecasting. Despite these advances, most forecasting models generate predictions mainly from the current input sequence and learned model parameters, without explicitly retrieving and reusing similar historical precedents as external references.

2.2 Retrieval-Augmented Time Series Forecasting

Retrieval augmentation has achieved notable success in knowledge-intensive natural language processing tasks by incorporating external information into generation models [13]. Inspired by this paradigm, recent studies have introduced retrieval mechanisms into time series forecasting. Instead of relying solely on the current input window, retrieval-augmented forecasting methods search for historical segments that are similar to the query sequence and use their future trajectories to provide additional forecasting references.

RAFT [14] represents an early attempt to adapt retrieval augmentation to time series forecasting by retrieving similar historical samples and incorporating their future observations into the prediction process. DUET [15] further improves multivariate forecasting through enhanced representation learning and alignment, enabling more informative historical references to be exploited. SERAF [25] introduces semantic descriptions into retrieval-augmented forecasting, converting time series patterns into textual representations to enrich the retrieval process. These methods demonstrate the potential of retrieval augmentation for forecasting; however, their retrieval spaces are mainly based on raw numerical sequences, learned temporal embeddings, or generated textual descriptions. Raw-sequence retrieval can be sensitive to high-frequency fluctuations and local noise, while text-based descriptions may abstract away fine-grained geometric information such as slopes, curvatures, and turning points. In contrast, VRATF performs retrieval on denoised visual trend representations, aiming to match historical segments according to global trend morphology rather than point-wise numerical similarity or coarse textual semantics.

2.3 Multimodal Time Series Retrieval

Recent studies have begun to explore multimodal representations for time series by connecting numerical signals with textual or visual information. GPT4MTS [26] incorporates textual information with numerical time series for forecasting, showing that external modalities can provide useful contextual knowledge beyond the raw sequence.

TRACE [27] further studies multimodal time series retrieval by grounding time series embeddings in aligned textual context and supporting cross-modal retrieval between time series and text. SERAF [25] also follows this direction in the forecasting setting by retrieving from both numerical time series and self-generated textual descriptions.

These studies indicate that retrieval over multimodal representations can enrich time series modeling, but existing multimodal retrieval methods mainly focus on textual semantics or context descriptions. Textual representations are useful for capturing high-level attributes such as trend direction, seasonality, or volatility, but they may be less precise in preserving detailed shape information. For forecasting, however, visual characteristics of trend curves—including slopes, turning points, and curvature changes—can be directly relevant to identifying historical segments with similar future evolution. Vision foundation models such as Vision Transformer [16] and CLIP [17] provide general-purpose visual encoders that can be used to represent rendered time series charts without training a vision model from scratch. Different from text-centered multimodal retrieval, VRATF converts denoised trend sequences into line-chart images and performs retrieval in the visual embedding space. This design uses the visual modality to capture global trend morphology while reducing the influence of high-frequency noise in the original numerical sequences.

3 Preliminary

3.1 Problem Definition

Let $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^N$ denote an observed set containing N samples. Each sample consists of a historical observation sequence $\mathbf{X}_i \in \mathbb{R}^{L \times C}$ and its corresponding future sequence $\mathbf{Y}_i \in \mathbb{R}^{T \times C}$, where L denotes the look-back length, T denotes the prediction horizon, and C is the number of variables (channels).

Given a historical observation sequence $\mathbf{X}$, the objective of multivariate time series forecasting is to learn a mapping function $f: \mathbb{R}^{L \times C} \rightarrow \mathbb{R}^{T \times C}$, such that the predicted future sequence $\hat{\mathbf{Y}} = f(\mathbf{X})$ approximates the ground-truth future observations $\mathbf{Y}$ as accurately as possible.

3.2 Retrieval-Augmented Forecasting Paradigm

Retrieval-augmented forecasting extends the conventional forecasting paradigm by introducing an external retrieval mechanism. Given a query sequence $\mathbf{Q}$, the model retrieves the top- K most similar historical segments from a retrieval database $\mathcal{B}$, $\{\mathbf{X}_{j_1}, \dots, \mathbf{X}_{j_K}\}$, together with their corresponding future observations, $\{\mathbf{Y}_{j_1}, \dots, \mathbf{Y}_{j_K}\}$.

The retrieval prediction is then obtained by aggregating the retrieved future sequences with similarity-aware weights:

$$\hat{\mathbf{Y}}_r = \sum_{k=1}^K w_k \mathbf{Y}_{j_k}, \quad \sum_{k=1}^K w_k = 1, \quad (1)$$

where $\mathbf{Y}_{j_k}$ denotes the future sequence corresponding to the k -th retrieved historical segment, w_k is its normalized aggregation weight determined by the retrieval similarity,

and K is the number of retrieved neighbors. The retrieval prediction is combined with the prediction produced by a base forecasting model f_b , which estimates future values solely from the current input sequence.

This work focuses on two fundamental challenges in retrieval-augmented forecasting: (1) designing a retrieval mechanism that retrieves historical segments with genuinely similar trend structures rather than being distracted by high-frequency noise, and (2) developing a flexible and interpretable learnable fusion strategy to effectively combine the base prediction and the retrieval prediction.

4 Methodology

4.1 Overview

The overall architecture of VRATF is illustrated in Fig. 2. The framework consists of two complementary branches: an **Image Retriever** and a **Forecaster**. The Image Retriever retrieves historical segments with trend patterns similar to the input sequence and leverages their future trajectories as retrieval-based references. In parallel, the Forecaster adopts DLinear as the base forecasting model to extrapolate future values from the current observations. The predictions produced by the two branches are finally combined through a lightweight learnable fusion module to generate the final forecast.

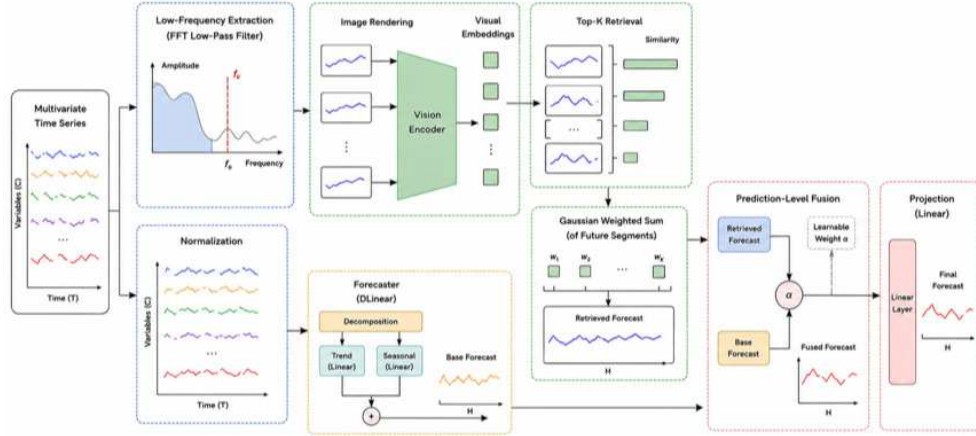


Fig. 2 Architecture of the proposed VRATF framework. Starting from the input multivariate time series, the retrieval branch first extracts low-frequency trend components via FFT-based low-pass filtering, converts them into line-chart images, and retrieves trend-consistent historical samples in the visual embedding space. Their future segments are aggregated using Gaussian weighting to produce the retrieved forecast. In parallel, the normalized input is fed into a DLinear forecaster to obtain the base forecast. The two prediction streams are finally combined through a learnable prediction-level fusion module, followed by a linear projection to generate the final forecasting output.

4.2 Image Retriever and Retrieval Knowledge Base

The Image Retriever aims to retrieve historical segments with trend patterns similar to the query sequence. Instead of directly comparing raw numerical sequences, VRATF converts trend curves into line chart images and performs retrieval in the visual embedding space, where high-level temporal morphology can be more effectively represented.

Given an input sequence $\mathbf{X} \in \mathbb{R}^{L \times C}$, we first normalize each channel by subtracting its last observation:

$$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{X}_{[-1,:]} \quad (2)$$

This operation removes the level offset and aligns the endpoints of different segments while preserving relative temporal variations. As a result, the vision encoder can focus on geometric characteristics such as slopes, curvatures, and turning points rather than absolute value levels. The removed offset is added back after forecasting to recover the original scale.

After preprocessing, the retrieval pipeline consists of four stages: (1) trend extraction via FFT-based low-pass filtering, (2) visual encoding of trend curves, (3) retrieval database construction, and (4) similarity-based retrieval with Gaussian kernel aggregation.

FFT-based Trend Extraction. For each channel $\tilde{\mathbf{x}}_c \in \mathbb{R}^L$, we first perform a one-dimensional Real Fast Fourier Transform (RFFT):

$$\mathbf{F}_c = \text{RFFT}(\tilde{\mathbf{x}}_c), \quad \mathbf{F}_c \in \mathbb{C}^{\lfloor L/2 \rfloor + 1}, \quad (3)$$

where $\mathbf{F}$ denotes the frequency-domain representation of the input sequence. To suppress high-frequency noise, only the lowest k_{freq} frequency components are retained:

$$\mathbf{F}_c[k_{\text{freq}}:] = 0. \quad (4)$$

The denoised trend is then reconstructed through the inverse transform:

$$\bar{\mathbf{x}}_c = \text{IRFFT}(\mathbf{F}_c, n = L). \quad (5)$$

The cutoff frequency k_{freq} is selected according to the sampling frequency of each dataset to preserve the dominant low-frequency components while filtering out high-frequency fluctuations. Consequently, the reconstructed sequence retains the overall temporal evolution of the original series while effectively suppressing noise.

Visual Encoding. Rather than performing retrieval directly in the numerical space, VRATF converts each denoised trend sequence into a line chart and leverages a pretrained vision encoder to extract visual representations. This design is motivated by the observation that line charts provide an intuitive representation of temporal trends, while pretrained vision models possess strong capabilities for capturing high-level geometric patterns such as slopes, curvatures, and turning points.

Specifically, each trend sequence $\bar{\mathbf{x}}_c$ is first normalized to the range $[0, 1]$ using Min-Max normalization and then rendered as a 224×224 RGB line chart. The rendered

image is fed into a frozen vision encoder to obtain its visual embedding:

$$\mathbf{e}_c = \text{ViT}(\text{img}_c) \in \mathbb{R}^{512}. \quad (6)$$

In this work, we employ the pretrained CLIP ViT-B/32 model as the vision encoder. Benefiting from large-scale vision-language pretraining, CLIP provides general-purpose visual representations that may capture geometric patterns such as slopes, curvatures, and turning points in rendered trend charts. Finally, the embedding is L2-normalized:

$$\hat{\mathbf{e}}_c = \frac{\mathbf{e}_c}{\|\mathbf{e}_c\|_2}, \quad (7)$$

so that similarity retrieval can be efficiently performed using cosine similarity. Compared with numerical similarity computed in the original sequence space, visual embeddings emphasize global geometric consistency while being less sensitive to local fluctuations, making them more suitable for retrieval under noisy forecasting settings.

Retrieval Knowledge Base. During preprocessing, an independent retrieval database is constructed for each variate channel. For every training sample, the normalized visual embedding and its corresponding future sequence are stored as a retrieval entry:

$$\mathcal{DB}_c = \{(\hat{\mathbf{e}}_i^c, \mathbf{Y}_i^c)\}_{i=1}^N. \quad (8)$$

Maintaining an independent database for each channel preserves channel-specific temporal semantics and prevents interference among heterogeneous variables. To avoid information leakage, training samples whose prediction windows overlap with the query sequence are excluded during retrieval.

Similarity Retrieval and Gaussian Aggregation. For a query sequence, the visual embedding of each channel is obtained following the same preprocessing pipeline. Cosine similarity is then computed between the query embedding and every entry in the corresponding retrieval database:

$$s_i = \hat{\mathbf{e}}_c^q \cdot \hat{\mathbf{e}}_i^c, \quad i = 1, \dots, N. \quad (9)$$

The Top- K most similar samples are selected as retrieval candidates. Instead of assigning equal weights, VRATF employs a Gaussian kernel to transform similarity scores into aggregation weights:

$$w_j = \exp\left(-\frac{(1 - s_j)^2}{2\tau^2}\right), \quad j \in \mathcal{S}_c, \quad (10)$$

where τ denotes the kernel bandwidth controlling the sensitivity to similarity differences. The normalized weights are used to aggregate the retrieved future sequences:

$$\hat{\mathbf{y}}_r^c = \sum_{j \in \mathcal{S}_c} \hat{w}_j \mathbf{Y}_j^c, \quad \hat{w}_j = \frac{w_j}{\sum_{k \in \mathcal{S}_c} w_k}. \quad (11)$$

Repeating this procedure for all channels yields the final retrieval prediction $\hat{\mathbf{Y}}_r \in \mathbb{R}^{T \times C}$.

4.3 Forecaster

The forecasting branch consists of a base forecasting model and a learnable prediction-level fusion module. We adopt DLinear as the base predictor due to its simplicity, efficiency, and explicit trend-seasonality decomposition, which naturally complements the trend-oriented retrieval paradigm of VRATF.

Base Forecasting Model. Given the preprocessed input sequence $\tilde{\mathbf{X}}$, DLinear first decomposes each channel into trend and seasonal components using a moving average operator:

$$\mathbf{X}_{\text{trend}} = \text{MA}(\tilde{\mathbf{X}}), \quad \mathbf{X}_{\text{seasonal}} = \tilde{\mathbf{X}} - \mathbf{X}_{\text{trend}}. \quad (12)$$

The two components are independently projected to the prediction horizon through linear layers:

$$\hat{\mathbf{Y}}_{\text{trend}} = \mathbf{X}_{\text{trend}} \mathbf{W}_t + \mathbf{b}_t, \quad \hat{\mathbf{Y}}_{\text{seasonal}} = \mathbf{X}_{\text{seasonal}} \mathbf{W}_s + \mathbf{b}_s, \quad (13)$$

where $\mathbf{W}_t, \mathbf{W}_s \in \mathbb{R}^{L \times T}$ and $\mathbf{b}_t, \mathbf{b}_s \in \mathbb{R}^T$. The base prediction is obtained by

$$\hat{\mathbf{Y}}_b = \hat{\mathbf{Y}}_{\text{trend}} + \hat{\mathbf{Y}}_{\text{seasonal}}. \quad (14)$$

Learnable Prediction-Level Fusion. Existing retrieval-augmented forecasting methods typically perform feature fusion in the latent space, making the contributions of the base predictor and retrieval branch difficult to interpret. In contrast, VRATF performs fusion directly at the prediction level, where the two prediction sources remain explicitly separable.

Specifically, a single learnable scalar parameter $w \in \mathbb{R}$ is introduced and mapped to the interval $(0, 1)$ through the sigmoid function:

$$\alpha = \sigma(w) = \frac{1}{1 + e^{-w}}. \quad (15)$$

The two prediction streams are first linearly combined at the prediction level:

$$\hat{\mathbf{Y}}_{\text{fused}} = \alpha \hat{\mathbf{Y}}_b + (1 - \alpha) \hat{\mathbf{Y}}_r. \quad (16)$$

The fused prediction is further refined through a lightweight channel-wise output projection. For each channel c , this projection is defined as

$$\hat{\mathbf{y}}^c = \mathbf{W}_o \hat{\mathbf{y}}_{\text{fused}}^c + \mathbf{b}_o, \quad c = 1, \dots, C, \quad (17)$$

where $\hat{\mathbf{y}}_{\text{fused}}^c \in \mathbb{R}^T$ denotes the fused prediction of the c -th channel, $\mathbf{W}_o \in \mathbb{R}^{T \times T}$, and $\mathbf{b}_o \in \mathbb{R}^T$. The calibrated channel-wise predictions are then stacked to form $\hat{\mathbf{Y}} \in \mathbb{R}^{T \times C}$. Finally, the removed offset is restored to recover the prediction in the original value space:

$$\hat{\mathbf{Y}}_{\text{final}} = \hat{\mathbf{Y}} + \mathbf{X}_{[-1, :]}. \quad (18)$$

The fusion weight α provides an interpretable measure of the relative contribution assigned to the two prediction branches before the final calibration layer. A larger α indicates greater reliance on the base forecasting model, whereas a smaller α places more emphasis on retrieved historical references. Since α is optimized end-to-end, the model learns an appropriate global balance between extrapolation and retrieval for each experimental setting.

4.4 Complexity Analysis

VRATF separates representation construction from forecast optimization.

During preprocessing, historical samples are low-pass filtered, rendered as line charts, encoded by the frozen vision encoder, and stored in the retrieval database. This offline stage is performed once before model optimization and therefore does not introduce additional trainable parameters or back-propagation cost.

During training, the vision encoder and the retrieval database remain fixed. Gradients are propagated only through the DLinear forecasting branch, the learnable fusion scalar, and the output projection layer. Retrieval references are obtained by similarity search over fixed embeddings and are then treated as non-trainable inputs to the forecasting module. Compared with retrieval-augmented forecasting methods that jointly optimize retrieval and forecasting modules, VRATF keeps the trainable component lightweight while still optimizing the final forecasting objective end-to-end.

During inference, the additional cost consists of query preprocessing, frozen visual encoding, nearest-neighbor search, and Gaussian aggregation. Given a query embedding, cosine similarity is computed against the retrieval database to identify the Top- K candidates, followed by Gaussian-weighted aggregation. Since the database embeddings are precomputed and have a fixed dimensionality, the search can be implemented efficiently with batched matrix multiplication and can be further accelerated using approximate nearest-neighbor indexing for large-scale deployment.

As a result, VRATF trades a one-time offline preprocessing cost for a lightweight trainable forecasting model and efficient retrieval-based inference.

4.5 Training Objective

The entire framework is trained end-to-end by minimizing the Mean Squared Error (MSE) between the predicted sequence and the ground-truth future observations:

$$\mathcal{L} = \frac{1}{TC} \|\hat{\mathbf{Y}}_{\text{final}} - \mathbf{Y}\|_2^2. \quad (19)$$

Since the retrieval database and the vision encoder are fixed throughout training, gradients are propagated only through the base forecasting model, the prediction-level fusion module, and the output projection layer.

Table 1 Statistics of the datasets.

Dataset	Variables	Frequency	Train	Validation	Test
ETTh1	7	1 Hour	8,640	2,880	2,880
ETTh2	7	1 Hour	8,640	2,880	2,880
ETTM1	7	15 Minutes	34,560	11,520	11,520
ETTM2	7	15 Minutes	34,560	11,520	11,520
Weather	21	10 Minutes	36,792	5,270	10,539
Exchange	8	1 Day	5,312	665	1,428

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate VRATF on six widely used multivariate forecasting benchmarks (ETTh1, ETTh2, ETTm1, ETTm2, Weather, and Exchange). The detailed statistics are summarized in Table 1.

The ETT datasets [7] record electricity transformer temperature, Weather [19] contains multivariate meteorological observations, Exchange [18] consists of daily exchange rates of eight countries. Following previous studies, the standard chronological split is adopted for multivariate datasets with prediction horizons of $\{96, 192, 336, 720\}$.

Baselines. We compare VRATF against ten representative forecasting methods, including the linear model DLinear[8], Transformer-based models (PatchTST[21], Autoformer[19], TimesNet[22], and Stationary[20]), MLP-based models (TimeMixer[23] and CycleNet[24]), and recent retrieval-augmented forecasting methods (RAFT[14], DUET[15], and SERAF[25]). In particular, RAFT, DUET, and SERAF are included to evaluate the proposed retrieval paradigm against recent retrieval-augmented forecasting methods.

All baseline results are reproduced using the official implementations with the hyperparameter settings recommended in their original papers. To ensure a fair comparison, all methods are evaluated under the same train/validation/test splits and the same prediction horizons. For the ETT datasets, the standard chronological split of 12/4/4 months is used. For Weather and Exchange, we follow the commonly used chronological splits summarized in Table 1. The random seed is fixed to 2021 for reproducibility.

Implementation Details. For all six multivariate forecasting datasets, the look-back length is set to $L = 720$, and the prediction horizon is selected from $T \in \{96, 192, 336, 720\}$. VRATF is trained using the Adam optimizer with early stopping, where the patience is set to 10. The learning rate is selected according to the dataset and prediction horizon: ETTh1 and ETTh2 use 10^{-3} for shorter horizons and 10^{-4} for longer horizons, Exchange uses 5×10^{-4} , and Weather uses 10^{-3} . We adopt the default type-1 learning-rate schedule, which decays the learning rate after each epoch. The batch size is set to 32 for most datasets, 8 for Exchange, and 16 for Weather. Each experiment is conducted once following the unified setting of `--itr = 1`. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

5.2 Experimental Results

Table 2 compares VRATF with representative retrieval-augmented forecasting methods, while Table 3 reports comparisons with widely used forecasting models.

Table 2 Multivariate forecasting results: comparison with retrieval-augmented methods. The best and the second-best results are highlighted in **bold** and underline, respectively.

Dataset	Horizon	VRATF		SERAF		RAFT		DUET	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	<u>0.371</u>	0.396	0.388	0.414	0.366	0.396	0.382	<u>0.407</u>
	192	<u>0.408</u>	0.418	0.420	0.434	0.403	<u>0.419</u>	0.421	0.439
	336	0.437	0.439	0.446	0.452	0.437	<u>0.443</u>	<u>0.44</u>	0.453
	720	0.451	0.469	0.476	0.489	<u>0.467</u>	<u>0.477</u>	0.479	0.49
	Avg	0.417	0.431	0.432	0.447	<u>0.418</u>	<u>0.434</u>	0.43	0.447
ETTh2	96	0.268	0.333	0.356	0.393	<u>0.278</u>	<u>0.345</u>	0.307	0.369
	192	0.335	0.376	0.438	0.430	<u>0.347</u>	<u>0.393</u>	0.352	0.397
	336	0.361	0.403	0.398	0.433	<u>0.374</u>	0.423	0.375	<u>0.416</u>
	720	0.396	<u>0.443</u>	0.408	0.449	0.432	0.472	<u>0.4</u>	0.438
	Avg	0.340	0.389	0.400	0.426	<u>0.358</u>	0.408	0.358	<u>0.405</u>
ETTh1	96	0.307	0.349	0.296	0.349	0.303	<u>0.350</u>	<u>0.297</u>	0.351
	192	0.337	<u>0.366</u>	0.350	0.380	0.327	0.365	<u>0.33</u>	0.371
	336	0.369	<u>0.387</u>	0.370	0.392	0.353	0.381	<u>0.358</u>	0.388
	720	0.421	0.415	0.439	0.441	0.406	0.412	<u>0.411</u>	<u>0.419</u>
	Avg	0.358	0.379	0.364	0.390	0.347	0.377	<u>0.349</u>	<u>0.382</u>
ETTh2	96	0.161	0.254	0.176	0.274	<u>0.164</u>	0.256	0.165	0.254
	192	0.216	0.293	0.243	0.317	<u>0.22</u>	0.296	0.223	<u>0.294</u>
	336	0.270	0.330	0.291	0.344	<u>0.275</u>	0.335	0.279	<u>0.331</u>
	720	<u>0.352</u>	<u>0.386</u>	0.431	0.445	<u>0.369</u>	0.398	0.351	0.378
	Avg	0.250	<u>0.316</u>	0.285	0.345	0.257	0.321	<u>0.254</u>	0.314
Exchange	96	0.083	0.199	<u>0.085</u>	<u>0.201</u>	0.09	0.207	0.098	0.224
	192	0.179	0.297	<u>0.199</u>	<u>0.317</u>	0.201	<u>0.317</u>	0.206	0.329
	336	0.324	0.406	0.908	0.666	0.377	<u>0.445</u>	<u>0.371</u>	0.446
	720	0.981	0.735	2.152	1.095	<u>1.129</u>	0.805	1.202	<u>0.783</u>
	Avg	0.392	0.409	0.836	0.570	<u>0.449</u>	<u>0.444</u>	0.469	0.446
Weather	96	0.166	0.221	<u>0.147</u>	<u>0.202</u>	0.165	0.222	0.146	0.191
	192	0.210	0.260	0.187	<u>0.243</u>	0.211	0.264	<u>0.188</u>	0.231
	336	0.256	0.296	<u>0.236</u>	<u>0.281</u>	0.26	0.302	0.234	0.268
	720	<u>0.319</u>	<u>0.342</u>	<u>0.325</u>	<u>0.355</u>	0.327	0.355	0.305	0.319
	Avg	0.238	0.280	<u>0.224</u>	<u>0.270</u>	0.240	0.285	0.218	0.252

Comparison with Retrieval-Augmented Methods. As shown in Table 2, VRATF achieves the best average MSE on four of the six datasets (ETTh1, ETTh2, ETTm2, and Exchange), supporting the effectiveness of the proposed visual trend-based retrieval paradigm. The performance gain is particularly evident on datasets with complex temporal dynamics, where suppressing high-frequency noise before

Table 3 Multivariate forecasting results: comparison with other forecasting methods. The best and the second-best results are highlighted in **bold** and underline, respectively.

Dataset	Horizon	VRATF		DLinear		TimeMixer		CycleNet		PatchTST	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.371	0.396	0.418	0.444	0.408	0.429	<u>0.396</u>	<u>0.418</u>	0.539	0.513
	192	0.408	0.418	0.461	0.471	<u>0.435</u>	<u>0.444</u>	<u>0.435</u>	<u>0.444</u>	0.611	0.551
	336	0.437	0.439	0.527	0.506	<u>0.464</u>	<u>0.462</u>	0.468	0.465	0.675	0.582
	720	0.451	0.469	0.68	0.612	<u>0.463</u>	<u>0.477</u>	0.473	0.487	0.716	0.616
	Avg	0.417	0.431	0.521	0.508	<u>0.442</u>	<u>0.453</u>	0.443	0.454	0.635	0.565
ETTh2	96	0.268	0.333	0.391	0.436	0.31	0.374	<u>0.283</u>	<u>0.344</u>	0.385	0.428
	192	0.335	0.376	0.447	0.469	0.353	0.398	<u>0.35</u>	<u>0.391</u>	0.408	0.443
	336	0.361	0.403	0.522	0.454	0.381	0.418	<u>0.377</u>	<u>0.415</u>	0.422	0.451
	720	0.396	0.443	0.519	0.527	0.433	0.456	<u>0.401</u>	<u>0.44</u>	0.434	0.465
	Avg	0.340	0.389	0.445	0.472	0.369	0.412	<u>0.353</u>	<u>0.398</u>	0.412	0.446
ETTm1	96	0.307	0.349	0.352	0.391	0.326	0.369	<u>0.323</u>	<u>0.363</u>	0.523	0.487
	192	0.337	0.366	0.375	0.406	0.358	0.389	<u>0.349</u>	<u>0.378</u>	0.479	0.453
	336	0.369	0.387	0.408	0.428	0.383	0.406	<u>0.378</u>	<u>0.395</u>	0.568	0.494
	720	0.421	0.415	0.464	0.463	0.474	0.453	<u>0.426</u>	<u>0.421</u>	0.624	0.532
	Avg	0.358	0.379	0.4	0.422	0.385	0.404	<u>0.369</u>	<u>0.39</u>	0.548	0.492
ETTm2	96	0.161	0.254	0.21	0.296	<u>0.169</u>	<u>0.261</u>	0.173	0.266	0.272	0.335
	192	0.216	0.293	0.259	0.326	0.231	0.304	<u>0.226</u>	<u>0.302</u>	0.308	0.353
	336	0.270	0.330	0.307	0.358	0.283	0.338	<u>0.276</u>	<u>0.335</u>	0.347	0.383
	720	0.352	0.386	0.385	0.405	<u>0.358</u>	<u>0.386</u>	0.362	0.39	0.465	0.452
	Avg	0.250	0.316	0.29	0.346	0.26	0.323	<u>0.259</u>	<u>0.323</u>	0.348	0.381
Exchange	96	0.083	0.199	<u>0.088</u>	<u>0.211</u>	0.092	0.219	0.091	0.212	0.137	0.271
	192	<u>0.179</u>	0.297	0.177	0.307	0.205	0.326	0.184	<u>0.305</u>	0.209	0.332
	336	0.324	0.406	0.369	0.456	0.37	0.442	0.361	<u>0.438</u>	<u>0.351</u>	0.44
	720	<u>0.981</u>	<u>0.735</u>	0.911	0.713	0.996	0.746	1.043	0.756	1.046	0.778
	Avg	<u>0.392</u>	0.409	0.386	<u>0.422</u>	0.416	0.433	0.42	0.428	0.436	0.455
Weather	96	<u>0.166</u>	0.221	0.195	0.252	0.163	0.209	0.17	<u>0.216</u>	0.186	0.227
	192	0.210	0.260	0.237	0.295	0.234	0.25	<u>0.222</u>	<u>0.259</u>	0.234	0.265
	336	0.256	<u>0.296</u>	0.282	0.331	0.284	0.287	<u>0.275</u>	<u>0.296</u>	0.284	0.301
	720	0.319	<u>0.342</u>	<u>0.345</u>	0.382	0.356	0.341	0.349	0.345	0.356	0.349
	Avg	0.238	0.280	0.264	0.315	0.259	0.271	<u>0.254</u>	<u>0.279</u>	0.265	0.285

retrieval helps the model identify historical segments with more consistent temporal evolution patterns. For example, on ETTh2, VRATF achieves the lowest MSE among all retrieval-augmented baselines across every prediction horizon, with an average MSE of 0.340, which represents a 5.0% improvement over the second-best method, RAFT (0.358). On Exchange, the advantage is even more pronounced: at the 336-step prediction horizon, VRATF achieves an MSE of 0.324, whereas SERAF degrades sharply

to 0.908. This observation suggests that retrieval based directly on raw sequences may become unreliable when high-frequency fluctuations dominate the similarity computation, whereas the proposed FFT-filtered visual representation can better preserve the underlying trend structure and support more reliable retrieval. By contrast, RAFT achieves the best overall performance on ETTm1, and DUET performs best on Weather, indicating that raw-sequence retrieval remains competitive when temporal patterns are highly regular and noise levels are relatively low. Nevertheless, VRATF achieves competitive performance across all datasets, suggesting that the proposed retrieval strategy generalizes reasonably well across the evaluated benchmarks.

Comparison with Other Forecasting Methods. Table 3 compares VRATF with representative forecasting models, including the linear model DLinear, the Transformer-based model PatchTST, and the MLP-based models TimeMixer and CycleNet. VRATF outperforms its backbone predictor DLinear on most datasets, indicating that the proposed retrieval branch provides complementary predictive information beyond what the forecasting model alone can capture. For example, on ETTh1, VRATF reduces the average MSE from 0.521 (DLinear) to 0.417, corresponding to a 20.0% improvement. Compared with the other forecasting baselines, VRATF achieves the best average performance on five of the six datasets. Although DLinear obtains a slightly lower average MSE than VRATF on Exchange (0.386 vs. 0.392), VRATF still achieves the best results at both the 96-step and 336-step prediction horizons. Overall, these results suggest that the proposed retrieval mechanism can be effectively integrated with a lightweight forecasting backbone to improve forecasting performance in most benchmark settings while maintaining a simple model architecture.

Performance Across Forecasting Horizons. A general trend observed in both tables is that VRATF remains particularly competitive at longer forecasting horizons. For instance, on ETTh1 with a prediction horizon of 720, VRATF achieves an MSE of 0.451, outperforming DLinear (0.680) by 33.7% and PatchTST (0.716) by 37.0%. Similar observations can be made on ETTh2, ETTm2, and Weather, where the retrieval branch provides increasingly valuable historical references as the forecasting horizon extends. This observation is consistent with the motivation of retrieval-augmented forecasting: as future uncertainty accumulates over longer horizons, historical precedents become increasingly valuable for providing complementary predictive guidance beyond the observed context.

5.3 Ablation Studies

To investigate the contribution of each component in VRATF, we conduct a series of ablation experiments under the same experimental settings as the main experiments. Unless otherwise specified, all other configurations remain unchanged.

Effect of Retrieval Augmentation (w/o Retrieval). To evaluate the contribution of the retrieval branch, we remove it entirely and retain only the DLinear forecaster with offset subtraction, reducing VRATF to its backbone model. As shown in Fig. 3, removing the retrieval branch leads to performance degradation across all prediction horizons on both datasets, suggesting that the retrieved references provide complementary predictive information to the backbone forecaster.

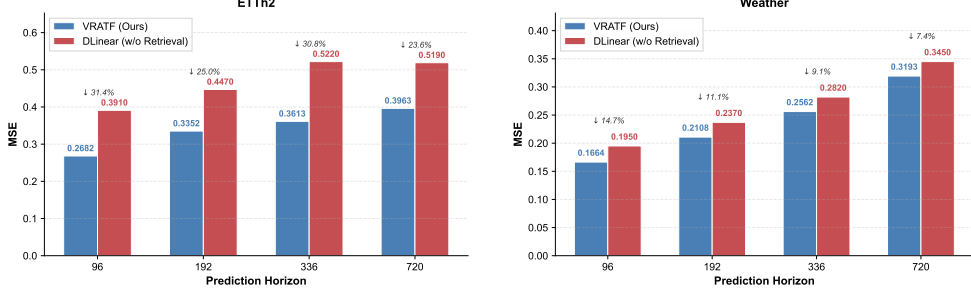


Fig. 3 Comparison of VRATF and DLinear (w/o Retrieval) on ETTh2 and Weather datasets across four prediction horizons.

Effect of Retrieval Representation (Vision vs. Text). To evaluate the effectiveness of representing temporal trends as visual line charts, we compare VRATF with SERAF, which performs retrieval based on textual descriptions generated from the same FFT-filtered trend sequences. Both methods share the same retrieval-augmented forecasting paradigm but differ only in the representation used for retrieval, enabling a direct comparison between visual and textual retrieval. As shown in Fig. 4, VRATF outperforms SERAF on five of the six benchmark datasets, with the largest improvement observed on Exchange (53.1%). These results indicate that, in the evaluated settings, visual representations can capture global trend morphology more effectively than textual descriptions, leading to more reliable retrieval and improved forecasting performance.

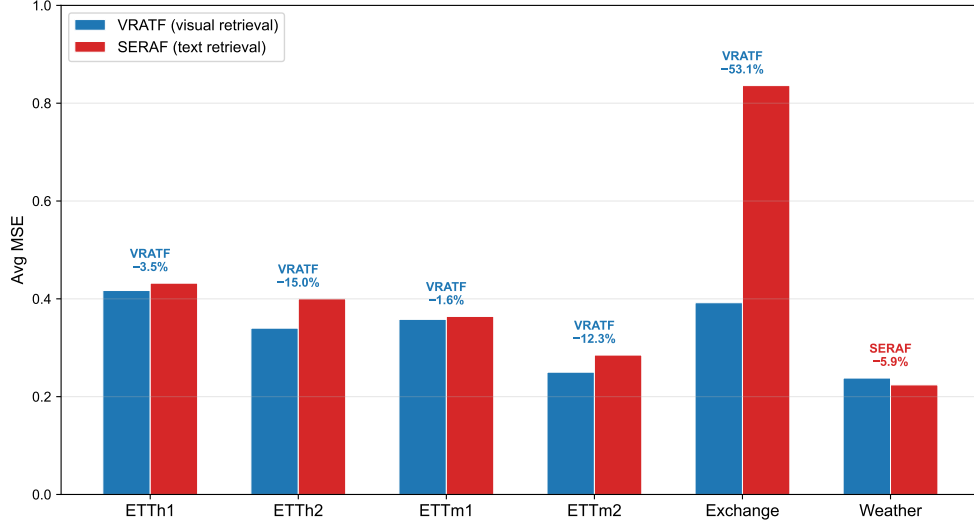


Fig. 4 Comparison of VRATF (visual retrieval) and SERAF (text retrieval) across six benchmark datasets in terms of average MSE.

Effect of Visual Trend Retrieval Components. To disentangle the contributions of low-pass filtering and visual representation, we compare VRATF against two numerical retrieval variants that replace the CLIP-based visual encoding with direct cosine similarity on time series vectors. *Raw numerical retrieval* performs similarity search on the original input sequences without any filtering or visual encoding, serving as a baseline for pure numerical retrieval. *Low-pass numerical retrieval* applies the same FFT-based low-pass filtering as VRATF before computing cosine similarity in the numerical trend space, isolating the effect of visual encoding from that of frequency-domain denoising, as summarized in Table 4. This experiment is conducted on ETTh1, ETTh2, and Exchange across four prediction horizons.

Table 4 Ablation design for disentangling the effects of low-pass filtering and visual trend representation.

Variant	Low-pass filtering	Visual encoding	Retrieval space
Raw numerical	×	×	Original sequence values
Low-pass numerical	✓	×	FFT-filtered numerical trends
VRATF	✓	✓	Image embeddings of trends

The results are presented in Table 5. Several findings emerge. First, **visual encoding consistently outperforms numerical retrieval** on ETTh1 and Exchange at most horizons. The advantage is most pronounced on Exchange at $H=96$, where VRATF achieves an MSE of 0.083 compared to 0.168 for both numerical variants—a $2\times$ improvement—indicating that visual representations are substantially more effective at capturing predictive patterns in noisy financial data where raw numerical similarity is unreliable. On ETTh1, the benefit of visual encoding grows with the prediction horizon: the gap increases from under 1% at $H=96$ to 8–11% at $H=720$, suggesting that visual trend representations are particularly valuable for long-range forecasting. Second, **FFT-based low-pass filtering contributes measurably to retrieval quality**. On ETTh2, Low-pass numerical with FFT consistently outperforms Raw numerical without FFT across all four horizons, with MSE reductions of 1.8–2.3% at $H=96$ and $H=192$, demonstrating that frequency-domain denoising removes high-frequency noise that would otherwise mislead numerical similarity matching. On ETTh1, the effect is horizon-dependent: FFT filtering improves short-horizon retrieval slightly, while at $H=336$ and $H=720$, the raw variant performs marginally better, suggesting that longer input sequences contain informative high-frequency details that aggressive low-pass filtering may discard. On Exchange, where data is dominated by irregular daily fluctuations, the two variants produce comparable results, as the absence of strong periodic structure limits the benefit of FFT-based denoising. These results indicate that while FFT filtering alone provides a modest improvement to numerical retrieval on datasets with regular periodicity, its primary contribution is realized in combination with visual encoding—where removing high-frequency noise produces cleaner line-chart renderings that allow the vision encoder to extract more discriminative trend features.

Table 5 MSE comparison of retrieval component ablation. Δ denote the relative change (%) of low-pass numerical and raw numerical with respect to VRATF.

Dataset	H	Low-pass Numerical		Raw Numerical		VRATF (full)
		MSE	Δ_8 (%)	MSE	Δ_7 (%)	
ETTh1	96	0.3742	+0.9	0.3727	+0.5	0.371
	192	0.4116	+0.9	0.4078	-0.0	0.408
	336	0.4510	+3.2	0.4419	+1.1	0.437
	720	0.5014	+11.2	0.4882	+8.2	0.451
ETTh2	96	0.2729	+1.8	0.2791	+4.1	0.268
	192	0.3355	+0.1	0.3418	+2.0	0.335
	336	0.3634	+0.7	0.3639	+0.8	0.361
	720	0.3802	-4.0	0.3760	-5.1	0.396
Exchange	96	0.1677	+102.0	0.1678	+102.2	0.083
	192	0.2098	+17.2	0.2101	+17.4	0.179
	336	0.2531	-21.9	0.2532	-21.9	0.324
	720	1.1266	+14.8	1.0869	+10.8	0.981

Effect of Learnable Fusion (Fixed Fusion). We replace the learnable fusion weight with a fixed coefficient of 0.5, assigning equal importance to the base forecast and the retrieved forecast. This experiment evaluates whether learnable prediction-level fusion is necessary for effectively balancing the two prediction streams. As shown in Fig. 5, the learnable fusion outperforms the fixed counterpart across all prediction horizons on ETTh2. The performance gap widens substantially as the prediction horizon increases, from 1.1% at $T = 96$ to 11.9% at $T = 720$, indicating that learning the fusion ratio is beneficial for multi-step forecasting in this setting.

5.4 Hyperparameter Analysis

To evaluate the sensitivity of VRATF to key hyperparameters, we investigate the influence of the number of retrieved neighbors, the Gaussian kernel bandwidth, and the FFT cutoff frequency. **Effect of Top- K and Kernel Bandwidth.** We evaluate the sensitivity of VRATF to the number of retrieved neighbors $K \in \{5, 10, 20, 50\}$ and the Gaussian kernel bandwidth $\tau \in \{0.05, 0.1, 0.2, 0.5\}$. This study is conducted on ETTh1 at the longest prediction horizon ($T = 720$), where the retrieval mechanism has the greatest influence on forecasting performance.

The results are presented in Fig. 6. VRATF exhibits low sensitivity to both hyperparameters in this experiment. The kernel bandwidth τ has a limited effect: for $\tau \in \{0.05, 0.1, 0.2\}$, the MSE curves across different K values are nearly overlapping, with differences below 0.0001. Only at the largest bandwidth ($\tau = 0.5$) does a slight deviation appear, and only for $K \in \{10, 20\}$, suggesting that an excessively wide kernel begins to blur the distinction between relevant and irrelevant neighbors. Regarding K , the optimal performance is achieved at $K = 20$ (MSE=0.4530), with $K = 10$ yielding a nearly identical result (0.4547). Reducing K to 5 leads to a modest degradation (MSE=0.4600, +1.5%), as a smaller neighborhood is more susceptible to variance from individual noisy retrievals. Conversely, increasing K to 50 also yields slightly

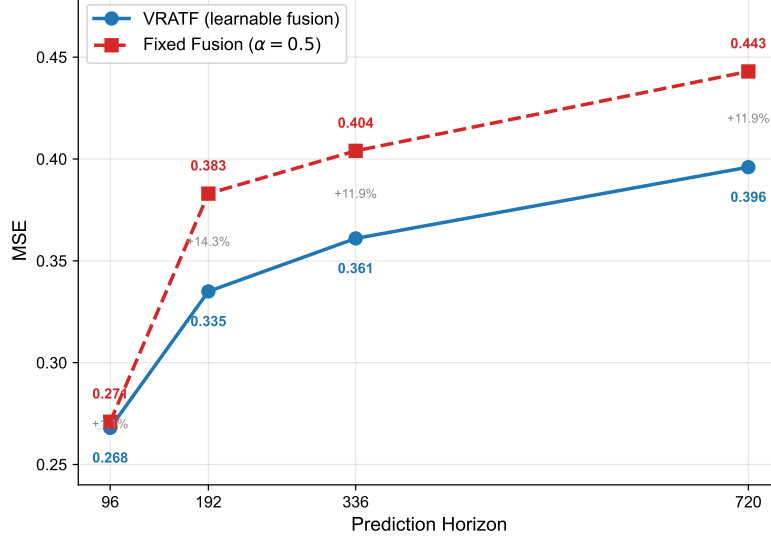


Fig. 5 Comparison of learnable fusion and fixed fusion ($\alpha = 0.5$) on ETTh2 across four prediction horizons.

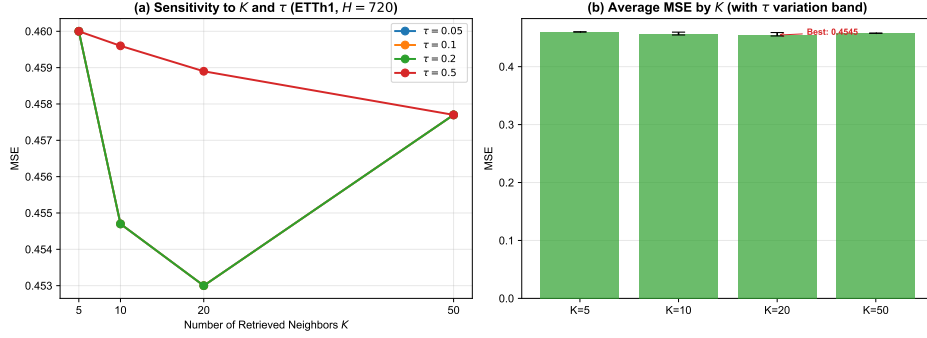


Fig. 6 Sensitivity of VRATF to the number of retrieved neighbors K and Gaussian kernel bandwidth τ on ETTh1 ($T = 720$). (a) MSE versus K under different τ values. (b) Average MSE for each K with the min-max band across all τ values showing the limited influence of kernel bandwidth in the tested range.

worse performance (MSE=0.4577, +1.0%), indicating that including too many distant neighbors dilutes the retrieval signal. Overall, the MSE varies within a narrow band of 0.007 across all 16 (K , τ) combinations. These results suggest that VRATF’s retrieval mechanism is relatively stable under the tested hyperparameter settings: the L2-normalized visual embeddings appear to produce sufficiently separated similarity scores, and the Gaussian kernel weighting suppresses the contribution of low-similarity neighbors across the tested bandwidth settings.

Effect of Cutoff Frequency. We further investigate the influence of the FFT cutoff frequency k_{freq} by varying the number of preserved low-frequency components

at the longest prediction horizon ($T = 720$). This experiment is conducted on ETTh1 (hourly, base cutoff $k_{\text{freq}} = 24$) and Exchange (daily, base cutoff $k_{\text{freq}} = 7$), with cutoff values tested in $\{5, 10, 30\}$ and $\{5, 10, 21\}$, respectively.

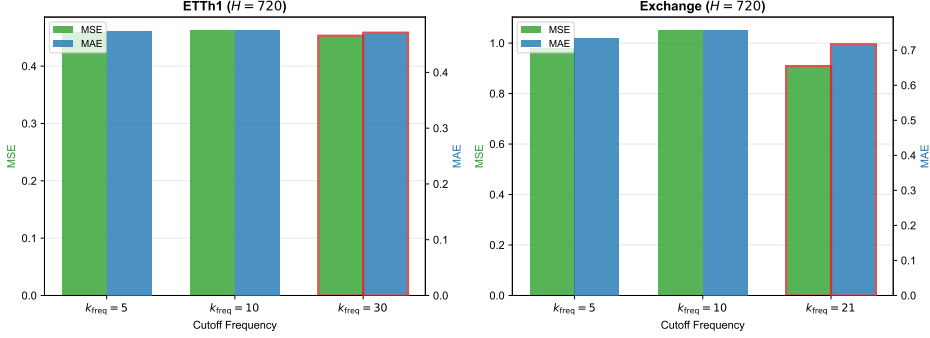


Fig. 7 Sensitivity of VRATF to the FFT cutoff frequency k_{freq} on ETTh1 and Exchange datasets ($T = 720$). The optimal cutoff for each dataset is highlighted with a red border.

The results are shown in Fig. 7. On ETTh1, the model exhibits low sensitivity to the cutoff frequency, with MSE varying within a narrow range of 0.009 across the three settings. The optimal cutoff is $k_{\text{freq}} = 30$ (MSE=0.453), indicating that for hourly data with strong diurnal patterns, preserving a broader set of low-frequency components better captures the multi-scale trend structure. On Exchange, the sensitivity is more pronounced: $k_{\text{freq}} = 21$ achieves the best performance (MSE=0.909), outperforming $k_{\text{freq}} = 5$ (0.982) by 7.4% and $k_{\text{freq}} = 10$ (1.051) by 13.5%. Contrary to the intuition that aggressive filtering benefits noisy financial data, retaining more frequency components proves advantageous for Exchange at long horizons. This suggests that daily exchange rate fluctuations contain informative low-frequency patterns spanning several weeks, and an overly aggressive cutoff discards predictive trend information that aids visual retrieval. The results highlight that the optimal cutoff frequency is dataset-dependent: data with strong periodic structure (e.g., hourly ETT) tolerates a wider range, while aperiodic data (e.g., daily Exchange) requires careful tuning to balance noise suppression against signal preservation.

5.5 Efficiency Analysis

Training Efficiency. We compare the training efficiency of VRATF against DLinear on ETTh1 (horizon $T = 96$) under identical hardware and software configurations. The vision encoder (CLIP ViT-B/32) is frozen and the retrieval database is constructed before model optimization. Therefore, the reported training time mainly reflects optimization of the lightweight forecasting network, the learnable fusion scalar, and the output projection layer. Once the fixed embeddings and retrieved references are available, the retrieval branch adds similarity lookup and weighted aggregation to each forward pass without back-propagation through the visual encoder. As summarized in Fig. 8, VRATF completes training in 342s over 15 epochs, while DLinear requires 672s

over 19 epochs—a $2\times$ reduction in total training time. More importantly, VRATF converges substantially faster in this setting: it reaches DLinear’s best test loss (0.376) by epoch 3 (approximately 20s of wall-clock time), and ultimately achieves a lower final test loss (0.371 vs. 0.376). This accelerated convergence suggests that retrieval-augmented references provide a useful inductive bias, allowing the forecasting network to exploit informative historical patterns from the earliest training iterations.

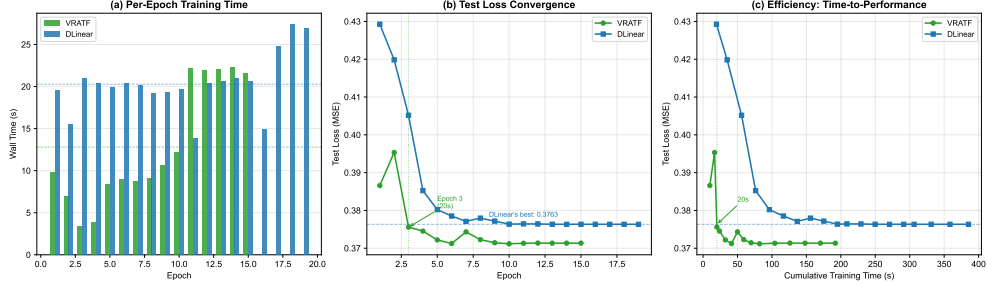


Fig. 8 Training efficiency comparison between VRATF and DLinear on ETTh1 ($T=96$). (a) Per-epoch wall-clock time. (b) Test loss convergence across epochs. (c) Test loss versus cumulative training time.

6 Conclusion

This paper presented VRATF, a vision-based retrieval-augmented framework for time series forecasting. The central idea is to perform retrieval on denoised visual trend representations rather than directly on raw numerical sequences. By extracting low-frequency trend components, rendering them as line-chart images, and encoding them with a frozen vision encoder, VRATF retrieves historical segments with similar global trend morphology and uses their future trajectories as complementary forecasting references. In addition, a learnable prediction-level fusion module explicitly balances the base forecast produced by a lightweight DLinear backbone and the retrieval-based forecast, improving interpretability while keeping the trainable components compact.

Experiments on six multivariate forecasting benchmarks show that VRATF achieves competitive performance against both recent retrieval-augmented forecasting methods and representative general forecasting baselines, with clear advantages in several long-horizon settings. The ablation and sensitivity analyses further suggest that the retrieval branch, visual trend representation, and learnable fusion strategy contribute to the overall forecasting performance. The efficiency analysis also indicates that separating offline visual representation construction from online forecast optimization provides a favorable trade-off between forecasting accuracy and training efficiency.

Future work will explore more scalable retrieval indexes for large-scale time series databases, adaptive strategies for selecting frequency components across different domains, and stronger visual encoders or time-series-specific visual pretraining objectives to further improve trend representation quality.

7 Acknowledgement

This work was supported by the Natural Science Foundation of Chongqing (Grant No. CSTB2022NSCQ-MSX0436).

References

- [1] Hong, T., Fan, S.: Probabilistic electric load forecasting: A tutorial review. *Int. J. Forecast.* **32**(3), 914–938 (2016)
- [2] Lana, I., Del Ser, J., Velez, M., Oregi, I.: Road traffic forecasting: Recent advances and new challenges. *IEEE Intell. Transp. Syst. Mag.* **10**(2), 93–109 (2018)
- [3] Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.* **90**, 106181 (2020)
- [4] Schultz, M.G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L.H., *et al.*: Can deep learning beat numerical weather prediction? *Philos. Trans. R. Soc. A* **379**(2194), 20200097 (2021)
- [5] Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
- [6] Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018)
- [7] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: *Proc. AAAI Conf. Artif. Intell.*, vol. 35, pp. 11106–11115 (2021)
- [8] Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: *Proc. AAAI Conf. Artif. Intell.*, vol. 37, pp. 11121–11128 (2023)
- [9] Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In: *Proc. Int. Conf. Learn. Represent. (ICLR)* (2020)
- [10] Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., Sahoo, D.: Unified training of universal time series forecasting transformers. In: *Proc. Int. Conf. Mach. Learn. (ICML)* (2024)
- [11] Ansari, A.F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S.S., Arango, S.P., Kapoor, S., *et al.*: Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815* (2024)
- [12] Das, A., Kong, W., Sen, R., Zhou, Y.: A decoder-only foundation model for

- time-series forecasting. In: Proc. Int. Conf. Mach. Learn. (ICML) (2024)
- [13] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, pp. 9459–9474 (2020)
 - [14] Han, S., Lee, S., Cha, M., Arik, S.O., Yoon, J.: Retrieval augmented time series forecasting. arXiv preprint arXiv:2505.04163 (2025)
 - [15] Qiu, X., Wu, X., Lin, Y., Guo, C., Hu, J., Yang, B.: DUET: Dual clustering enhanced multivariate time series forecasting. In: Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min. (KDD) (2025)
 - [16] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proc. Int. Conf. Learn. Represent. (ICLR) (2021)
 - [17] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Proc. Int. Conf. Mach. Learn. (ICML) (2021)
 - [18] Lai, G., Chang, W.-C., Yang, Y., Liu, H.: Modeling long- and short-term temporal patterns with deep neural networks. In: Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 95–104 (2018)
 - [19] Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 34, pp. 22419–22430 (2021)
 - [20] Liu, Y., Wu, H., Wang, J., Long, M.: Non-stationary transformers: Exploring the stationarity in time series forecasting. In: Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 35, pp. 9881–9893 (2022)
 - [21] Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: Proc. Int. Conf. Learn. Represent. (ICLR) (2023)
 - [22] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: TimesNet: Temporal 2D-variation modeling for general time series analysis. In: Proc. Int. Conf. Learn. Represent. (ICLR) (2023)
 - [23] Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J.Y., Zhou, J.: TimeMixer: Decomposable multiscale mixing for time series forecasting. In: Proc. Int. Conf. Learn. Represent. (ICLR) (2024)

- [24] Lin, S., Lin, W., Hu, X., Wu, W., Mo, R., Zhong, H.: CycleNet: Enhancing time series forecasting through modeling periodic patterns. In: Adv. Neural Inf. Process. Syst. (NeurIPS) (2024)
- [25] Zhou, S., Wu, Z., Schöner, H., Fouché, E., Wilson, I.A.G., Wang, S.: SERAF: Semantics-enhanced retrieval-augmented time series forecasting. arXiv preprint arXiv:2508.04888 (2025)
- [26] Jia, F., Wang, K., Zheng, Y., Cao, D., Liu, Y.: GPT4MTS: Prompt-based large language model for multimodal time-series forecasting. In: Proc. AAAI Conf. Artif. Intell., vol. 38, pp. 23343–23351 (2024). <https://doi.org/10.1609/aaai.v38i21.30383>
- [27] Chen, J., Zhao, Z., Nurbek, G., Feng, A., Maatouk, A., Tassiulas, L., Gao, Y., Ying, R.: TRACE: Grounding time series in context for multimodal embedding and retrieval. arXiv preprint arXiv:2506.09114 (2025)