

TS-Memory: Plug-and-Play Memory for Time Series Foundation Models

Sisuo Lyu*

The Hong Kong University of Science
and Technology (Guangzhou)
Guangzhou, China
sisuolyu@outlook.com

Siru Zhong*

The Hong Kong University of Science
and Technology (Guangzhou)
Guangzhou, China
siruzhong@outlook.com

Tiegang Chen

Tencent
Shenzhen, China
steelchen@tencent.com

Weilin Ruan

The Hong Kong University of Science
and Technology (Guangzhou)
Guangzhou, China
rwlinno@gmail.com

Qingxiang Liu

The Hong Kong University of Science
and Technology (Guangzhou)
Guangzhou, China
qingxiangliu737@gmail.com

Taiqiang Lv

Tencent
Shenzhen, China
felixlv@tencent.com

Qingsong Wen

Squirrel Ai Learning
Bellevue, Washington, USA
qingsongedu@gmail.com

Raymond Chi-Wing Wong[†]

The Hong Kong University of Science
and Technology
Hong Kong, China
raywong@cse.ust.hk

Yuxuan Liang[†]

The Hong Kong University of Science
and Technology (Guangzhou)
Guangzhou, China
yuxliang@outlook.com

Abstract

Time Series Foundation Models (TSFMs) achieve strong zero-shot forecasting through large-scale pre-training, but adapting them to downstream domains under distribution shift remains challenging. Existing solutions face a trade-off: *Parametric Adaptation* can cause catastrophic forgetting and requires costly multi-domain maintenance, while *Non-Parametric Retrieval* improves forecasts but incurs high inference latency due to datastore search. We propose *Parametric Memory Distillation* and implement it as **TS-Memory**, a lightweight memory adapter that augments frozen TSFMs. TS-Memory is trained in two stages. First, we construct an offline, retrieval-leakage-safe *k*NN teacher that synthesizes confidence-aware quantile targets from retrieved futures. Second, we distill this retrieval-induced distributional correction into a lightweight memory adapter via confidence-gated supervision. During inference, TS-Memory fuses memory and backbone predictions with constant-time overhead, enabling retrieval-free deployment. Experiments across diverse TSFMs and benchmarks demonstrate consistent improvements in both point and probabilistic forecasting over representative adaptation methods, with efficiency comparable to the frozen backbone. Code: <https://github.com/sisuolv/TS-Memory>.

CCS Concepts

• **Mathematics of computing** → **Time series analysis**; • **Computing methodologies** → **Machine learning**.

*Equal Contribution.

[†]Corresponding Author.



This work is licensed under a Creative Commons Attribution 4.0 International License.
KDD '26, Jeju Island, Republic of Korea
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2259-2/2026/08
<https://doi.org/10.1145/3770855.3817614>

Keywords

Time Series Forecasting, Time Series Foundation Models, Retrieval-Free Inference, Knowledge Distillation, Plug-and-Play Memory

ACM Reference Format:

Sisuo Lyu, Siru Zhong, Tiegang Chen, Weilin Ruan, Qingxiang Liu, Taiqiang Lv, Qingsong Wen, Raymond Chi-Wing Wong, and Yuxuan Liang. 2026. TS-Memory: Plug-and-Play Memory for Time Series Foundation Models. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26)*, August 09–13, 2026, Jeju Island, Republic of Korea. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3770855.3817614>

1 Introduction

Time series forecasting is a cornerstone of decision-making in critical domains such as energy, healthcare, and supply chains [20, 28, 32, 41, 52, 59, 63]. Recently, the field has witnessed a paradigm shift from training task-specific models from scratch [37, 42, 68, 69, 79, 80] to *Time Series Foundation Models* (TSFMs) [5, 11, 12, 39, 40, 70], which are pre-trained on diverse corpora to learn transferable temporal representations. However, a critical gap [17, 18, 24, 65, 74, 76, 77] remains between pre-training and deployment: real-world domain shifts (e.g., scale changes, seasonal discrepancies, sensor variations) can severely degrade the performance of frozen TSFMs.

To bridge this gap, existing adaptation strategies primarily fall into two paradigms (As shown in Figure 1):

- (1) **Parametric Adaptation:** Methods like full or parameter efficient fine-tuning adapt model weights to the target domain [50, 53, 75]. Despite their effectiveness, this approach requires maintaining a distinct model copy for each domain, leading to exorbitant storage and maintenance costs. Moreover, continuous updates to frozen backbones risk catastrophic forgetting [4, 8, 73] of general pre-training knowledge [26, 31, 47, 72].

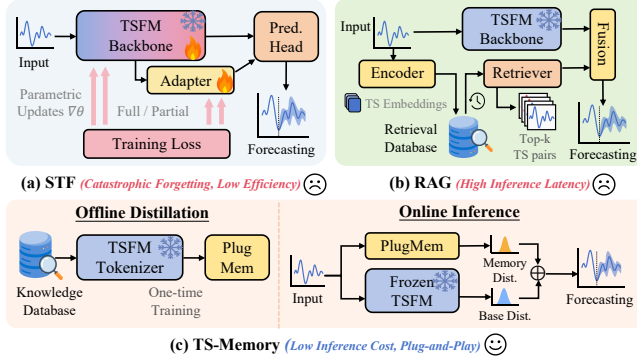


Figure 1: Comparison of TSFM adaptation paradigms: (a) Parametric Adaptation; (b) Non-Parametric Retrieval; (c) Parametric Memory Distillation (Ours).

(2) **Non-Parametric Retrieval:** To preserve the frozen backbone, methods like Metric Matching [57, 60, 64] and Retrieval Augmented Generation (RAG) retrieve historical instances from an external database during inference [16, 25, 34, 35, 54, 71]. While this avoids weight updates, it places heavy index maintenance and kNN queries on the *critical inference path*. The latency scales linearly with the size of the external knowledge base, making such methods prohibitive for real-time or edge deployment [3, 7, 22, 33, 46, 82] under tight latency constraints.

Existing methods face a dilemma between **maintenance** (per-domain model adaptation) and **inference** efficiency (runtime search cost). This raises a key question: *Can we capture the robust adaptability of retrieval-based methods while maintaining the constant-time inference and deployment simplicity of parametric adapters?*

We propose a third paradigm: **Parametric Memory Distillation** (Table 1). Our core insight is that the "knowledge" provided by online retrieval, namely the predictive distribution implied by similar inputs, can be compiled offline into a compact neural module. By shifting retrieval cost to training, we decouple retrieval benefits from runtime costs. Concretely, we introduce **TS-Memory**, a plug-and-play adapter for frozen TSFMs. Unlike RAG, which queries a datastore at every inference step, TS-Memory learns to approximate the teacher distribution from an offline kNN retriever. During inference, it serves as an internalized memory of domain patterns that fuses seamlessly with the TSFM's predictions.

This approach satisfies three critical deployment requirements: (1) **Zero Inference Search:** It eliminates dependencies on external vector databases during runtime; (2) **Constant Time Complexity:** Inference latency is $O(1)$, independent of the historical data size; (3) **Non-Destructive Adaptation:** It bypasses the risk of catastrophic forgetting by keeping the backbone frozen. As visualized in Figure 1, TS-Memory bridges the gap between parametric adaptation and non-parametric retrieval. Rather than storing or querying retrieved instances at test time, we distill the *distributional corrections* that retrieval induces on the frozen TSFM's base forecast into a lightweight neural memory module. This preserves the frozen backbone to prevent forgetting, adapts to local domains as RAG does, and operates with the speed of a standard forward pass.

Table 1: Comparison of TSFM adaptation paradigms. TS-Memory combines low latency with low storage cost.

Paradigm	Inference Search?	Deployment Cost	Latency Scaling	Backbone Status	Plug and Play?
Parametric Adaptation	No	High (Model Copies)	$O(1)$	Updated	No
Non-Parametric Retrieval	Yes	High (Vector DB)	$O(\mathcal{D})$	Frozen	Yes
Parametric Distillation	No	Low (Tiny Module)	$O(1)$	Frozen	Yes

Technically, TS-Memory operates in two phases. First, during an offline stage, we construct a *privileged supervision* by retrieving k -nearest neighbors for training instances and aggregating their future trajectories into *quantile targets*. This process captures the non-parametric uncertainty inherent in the data. Second, we train a lightweight parametric memory module to predict these teacher quantiles directly from the input context. To ensure robustness, we employ a *confidence-gated memory distillation* that selectively transfers knowledge only when retrieval provides reliable signals. During inference, the memory module functions as a plug-and-play addition to the frozen TSFM, fused via linear interpolation to enhance adaptability without incurring any search latency.

Our contributions are summarized as follows:

- **Paradigm Shift:** We propose the Parametric Memory Distillation framework for TSFMs, transforming expensive online retrieval into a learnable, offline memory mechanism.
- **TS-Memory:** We design a lightweight module that distills non-parametric kNN distributions into parametric representations, featuring a novel distribution fusion interface for frozen TSFMs.
- **Efficiency & Performance:** Extensive experiments demonstrate that TS-Memory outperforms representative methods from both paradigms in accuracy while significantly reducing inference latency, offering a superior trade-off for practical deployment.

2 Related Work

Time Series Foundation Models (TSFMs). Research has moved from task-specific training to pre-training TSFMs on large corpora for broad transferability. Early work [9, 51] has advanced to elaborate architectures with patch-based modeling [5, 6, 11, 12, 38–40, 70], sparse mixture-of-experts [36, 56], and discretization-based probabilistic objectives [1, 2, 67]. Concurrently, adapting general-purpose large models (LLMs/VLMs) to time series via reprogramming or prompting remains active [23, 42, 48, 58, 78, 81]. While TSFMs exhibit strong generalization, purely parametric variants perform poorly under severe distribution shifts in downstream domains, demanding effective adaptation strategies [17, 18, 24].

Adaptation Strategies for TSFMs. To tackle domain shifts, existing work falls into two paradigms: (1) *Parametric adaptation* updates model with target-domain data, ranging from full fine-tuning of all backbone weights, to partial fine-tuning of a small subset of parameters, and parameter-efficient add-ons that keep the backbone frozen while training lightweight modules such as adapters, LoRA-style low-rank updates, or prompt/prefix tuning [14, 15, 50, 75]. While these approaches reduce per-domain training cost, supporting many domains still incurs substantial maintenance overhead [31, 47, 55]. (2) *Non-parametric retrieval* augments frozen models with external datastore evidence, with lightweight implementations relying

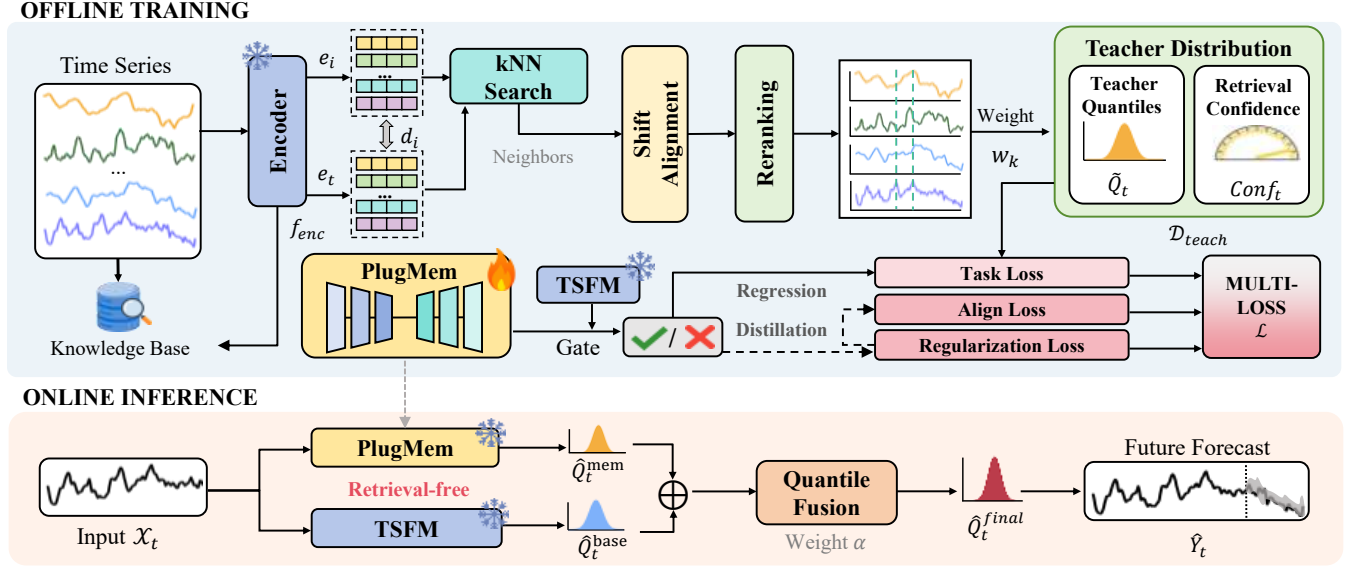


Figure 2: TS-Memory framework.

on metric-based matching, where query contexts are compared to support embeddings via learned similarity functions for prediction [29, 49, 57, 60, 64]. For time series, it is commonly realized as subsequence retrieval (via DTW or other distance metrics) followed by aggregation over matched historical windows [43–45, 61]. Inspired by NLP’s RAG [34], retrieval augmentation has been adapted to forecasting via relational retrieval, diffusion guidance, LLM-based retrieval and general frameworks [16, 25, 35, 71], with TSFM systems retrieving context-horizon pairs and fusing retrieved horizons for zero-shot forecasting [46]. Despite their flexibility, these methods introduce heavy search operations into the critical inference path, leading to latency that scales linearly with database size [7].

We thus propose (3) *Parametric Memory Distillation*. Inspired by memory-augmented networks [13, 66] and NLP retrieval-free methods [21, 62], we distill offline retrieval’s distributional knowledge into a lightweight offline module, enabling TS-Memory to achieve retrieval-based adaptability while retaining the constant inference speed and single-model efficiency of frozen TSFMs.

3 Problem Definition

Let $\mathbf{x}_{1:T} \in \mathbb{R}^{T \times C}$ denote a multivariate time series with C channels. At each time step t , the *lookback context* $\mathbf{X}_t = \mathbf{x}_{t-L+1:t} \in \mathbb{R}^{L \times C}$ and *forecast horizon* $\mathbf{Y}_t = \mathbf{x}_{t+1:t+H} \in \mathbb{R}^{H \times C}$ are defined accordingly. We target probabilistic forecasting via a set of quantile levels $\mathcal{Q} = \{q_k\}_{k=1}^Q$ with $q_k \in (0, 1)$. For brevity, we use the multi-index $u = (h, c)$ where $u \in \mathcal{U} = \{1, \dots, H\} \times \{1, \dots, C\}$, so that $Y_{t,u}$ denotes the scalar value at time $t + h$ for channel c .

Frozen Backbone. Let f_θ be a pre-trained TSFM with parameters θ . Given input $\mathbf{X}_t$, it outputs base quantile predictions:

$$\hat{\mathbf{Q}}_t^{\text{base}} = f_\theta(\mathbf{X}_t) \in \mathbb{R}^{Q \times H \times C}. \quad (1)$$

Crucially, to preserve general knowledge and avoid storage overhead, θ remains frozen throughout the adaptation phase.

Parametric Memory Module. We introduce **PlugMem**, a lightweight, learnable module g_ϕ , parameterized by ϕ . It maps the same context $\mathbf{X}_t$ to a complementary set of quantile estimates:

$$\hat{\mathbf{Q}}_t^{\text{mem}} = g_\phi(\mathbf{X}_t) \in \mathbb{R}^{Q \times H \times C}. \quad (2)$$

Our objective is to optimize ϕ to distill retrieval-based distributional knowledge into $\hat{\mathbf{Q}}_t^{\text{mem}}$, thereby achieving domain adaptation without accessing external databases during inference.

4 Methodology

We propose **TS-Memory**, a lightweight retrieval-free memory module for adapting frozen TSFMs to target domains. While online retrieval effectively mitigates distribution shifts, it introduces heavy inference-time search overhead. Our core insight is that retrieval not only provides point estimates but also induces a non-parametric conditional distribution over plausible futures. Dispersion among retrieved neighbors inherently encodes context-dependent uncertainty, difficult to learn from a single realized trajectory. We treat this retrieval-induced distribution as privileged offline supervision and distill it into a parametric module, preserving retrieval-based adaptivity while enabling constant-time inference. Since constructing such targets is leakage-sensitive and too costly for the online path, we adopt a two-stage design as shown in Figure 2.

- **Stage I: Privileged Supervision Construction.** To prevent test-time leakage, we construct a distributional teacher exclusively from the training set. We retrieve K nearest neighbors in the frozen embedding space, synthesize their future trajectories into empirical quantiles, and thereby mine a privileged supervision signal ($\mathcal{D}_{\text{teach}}$) that encodes local domain patterns and uncertainty structures without relying on a run-time index.
- **Stage II: Confidence-Gated Memory Distillation.** We train the memory module to directly predict teacher quantiles from

raw context. To mitigate noisy retrieval, we design a *confidence-gated* dual objective: the module aligns with ground truth via standard regression, and distills the retrieval-induced distribution only when the teacher provides a reliable high-confidence improvement signal. At inference, PlugMem runs in parallel with the frozen backbone for robust constant-time forecasting.

4.1 Privileged Supervision Construction

TS-Memory treats retrieval as training-time privileged supervision used only offline to build an auxiliary teacher dataset:

$$\mathcal{D}_{\text{teach}} = \{(\mathbf{X}_t, \mathbf{Y}_t, \tilde{\mathbf{Q}}_t, \text{Conf}_t)\}, \quad (3)$$

Concretely, a leakage-safe kNN teacher has privileged access to a knowledge base of past contexts and their realized futures, which is unavailable to the student at test time. This access allows the teacher to form a retrieval-conditioned predictive distribution, summarized as quantile targets $\tilde{\mathbf{Q}}_t \in \mathbb{R}^{Q \times H \times C}$ with reliability score $\text{Conf}_t \in [0, 1]$. The construction process involves three steps:

Knowledge Base Construction. We first build a leakage-safe index utilizing the training split only: $\mathcal{K} = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{N_{\text{train}}}$, ensuring that both the context window and forecast horizon are fully contained within the training segment. Then, we compute embeddings using a frozen encoder f_{enc} (e.g., the encoder of the TSFM):

$$\mathbf{e}_t = f_{\text{enc}}(\mathbf{X}_t), \quad \mathbf{e}_i = f_{\text{enc}}(\mathbf{X}^{(i)}). \quad (4)$$

Candidates are retrieved via Euclidean distance $d_i = \|\mathbf{e}_t - \mathbf{e}_i\|_2$. We explicitly exclude the index-matched window to prevent trivial self-retrieval. Here, $\mathbf{e}_t$ is the embedding of the current input $\mathbf{X}_t$, and $\mathbf{e}_i$ is the embedding of the i -th candidate context. We use “retrieval-leakage-safe” to denote that all context–future pairs used by the teacher and retrieval baselines are constructed strictly from the training split; this does not assume that every TSFM pretraining corpus is fully auditable.

Shift Alignment and Re-ranking. Distribution shift often manifests as level offsets, meaning that neighbors close in embedding space may be mis-aligned in absolute scale. We align each candidate using a trailing mean shift over the last m steps, computed channel-wise:

$$\mathbf{s}_i = \text{mean}(\mathbf{X}_{t, L-m+1:L}) - \text{mean}(\mathbf{X}_{L-m+1:L}^{(i)}) \in \mathbb{R}^C. \quad (5)$$

This is applied to the context and future of the retrieved candidate:

$$\mathbf{X}_{\text{align}}^{(i)} = \mathbf{X}^{(i)} + \mathbf{s}_i, \quad \mathbf{Y}_{\text{align}}^{(i)} = \mathbf{Y}^{(i)} + \mathbf{s}_i. \quad (6)$$

Candidates are then re-ranked by the ℓ_1 distance (i.e., $\text{score}_i = \|\mathbf{X}_t - \mathbf{X}_{\text{align}}^{(i)}\|_1$) between the query context $\mathbf{X}_t$ and the aligned candidate context $\mathbf{X}_{\text{align}}^{(i)}$, keeping the top K neighbors.

Teacher Aggregation and Confidence Scoring. Given the top K neighbors with distances $\{d_k\}_{k=1}^K$, we compute softmax weights:

$$w_k = \frac{\exp(-\psi(d_k)/\tau_{\text{ret}})}{\sum_{j=1}^K \exp(-\psi(d_j)/\tau_{\text{ret}})}, \quad \sum_{k=1}^K w_k = 1, \quad (7)$$

where $\psi(\cdot)$ is a monotone distance transform (identity by default) and τ_{ret} controls the weight sharpness. For each $u \in \mathcal{U}$, set $v_k = Y_{\text{align}, u}^{(k)}$ and let π sort $\{v_k\}$ in ascending order. Define cumulative

weights $S_m = \sum_{r=1}^m w_{\pi(r)}$. Then, the weighted empirical quantile is:

$$\tilde{Q}_{t, j, u} = v_{\pi(m^*)}, \quad m^* = \min\{m : S_m \geq q_j\}. \quad (8)$$

Here, v_k is the aligned future value from the k -th neighbor, and $\pi(r)$ denotes the index of the r -th smallest value. We define retrieval confidence as the concentration of retrieval weights:

$$\text{Conf}_t = \max_{1 \leq k \leq K} w_k. \quad (9)$$

The complete procedure is detailed in Algorithm 1 (Appendix A).

4.2 Confidence-Gated Memory Distillation

We parameterize the memory module g_ϕ as a lightweight encoder-decoder Transformer. The input $\mathbf{X}_t$ is first normalized via Instance Normalization (Eq. 10), partitioned into patches of length p , and projected to d -dimensional tokens. A Transformer encoder produces a memory representation, which is attended to by H learnable horizon queries in the decoder. A final quantile head maps these features to Q quantile values, after which the normalization is inverted. Crucially, g_ϕ operates independently of the backbone internals, ensuring plug-and-play compatibility.

Instance Normalization is defined as:

$$\tilde{\mathbf{X}}_t = (\mathbf{X}_t - \mu(\mathbf{X}_t)) / (\sigma(\mathbf{X}_t) + \epsilon). \quad (10)$$

To facilitate the memory module in learning transferable effective knowledge, we introduce three losses for joint optimization:

Task Supervision $\mathcal{L}_{\text{task}}$. Supervise PlugMem via standard quantile regression on ground-truth future values:

$$\mathcal{L}_{\text{task}} = \mathbb{E}_t \left[\text{Pinball}_Q(\tilde{\mathbf{Q}}_t^{\text{mem}}, \mathbf{Y}_t) \right], \quad (11)$$

where $\text{Pinball}_Q(\cdot, \cdot)$ denotes the standard pinball loss for quantile regression, aggregated over the forecast horizon and variables.

Confidence-Gated Distillation $\mathcal{L}_{\text{align}}$. Distill the teacher’s adaptive improvement into the memory module via confidence-gated distillation (two steps: error evaluation and incremental alignment), activated only when retrieval provides a reliable signal:

First, we evaluate the teacher’s absolute prediction quality via median absolute error (median index $j^* = \arg \min_j |q_j - 0.5|$) to screen valid distillation windows, defining median absolute error for the retrieval teacher (err_t^T) and frozen backbone ($\text{err}_t^{\text{base}}$) as:

$$\text{err}_t^* = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} |Y_{t, u} - Q_{t, j^*, u}^*|, \quad \star \in \{T, \text{base}\}, \quad (12)$$

where err_t^T and $\text{err}_t^{\text{base}}$ quantify the absolute deviation of the teacher and backbone predictions from the ground truth $Y_{t, u}$, respectively.

We then gate distillation with a margin ϵ_{gate} to retain only windows where the teacher outperforms the backbone:

$$\chi_t = \mathbb{I}(\text{err}_t^T + \epsilon_{\text{gate}} < \text{err}_t^{\text{base}}), \quad \omega_t = \chi_t \cdot \text{Conf}_t^\gamma, \quad (13)$$

where χ_t is the binary gating indicator (1 for valid windows), ω_t the confidence-weighted gating coefficient, Conf_t the retrieval confidence score, and γ the confidence scaling hyperparameter.

Second, we perform incremental alignment to distill the teacher’s adaptive improvement (not global bias) into the memory module, aligning quantile predictions via the robust Huber loss $\ell_\kappa(\cdot, \cdot)$:

$$\mathcal{D}_Q(t) = \frac{1}{Q|\mathcal{U}|} \sum_{j=1}^Q \sum_{u \in \mathcal{U}} \ell_\kappa(\tilde{\mathbf{Q}}_{t, j, u}^{\text{mem}}, \tilde{\mathbf{Q}}_{t, j, u}). \quad (14)$$

Table 2: Long-term forecasting results of TS-Memory across frozen TSFM backbones. Results are averaged over forecasting horizons $H \in \{96, 192, 336, 720\}$. Lower values indicate better performance. Best results are highlighted in bold, and second best results are underlined. Values in parentheses denote relative improvement (%). Full results are in Table 10 of Appendix D

TSFM	ChronosBolt				Chronos2				Sundial				TimesFM			
Dataset	Origin		TS-Memory		Origin		TS-Memory		Origin		TS-Memory		Origin		TS-Memory	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	<u>0.448</u>	<u>0.419</u>	0.421 (-6.0%)	0.414 (-1.2%)	<u>0.442</u>	<u>0.412</u>	0.420 (-5.0%)	0.407 (-1.2%)	<u>0.400</u>	<u>0.412</u>	0.396 (-0.9%)	0.410 (-0.5%)	<u>0.479</u>	<u>0.442</u>	0.448 (-6.5%)	0.431 (-2.5%)
ETTh2	<u>0.367</u>	<u>0.380</u>	0.354 (-3.4%)	0.375 (-1.1%)	<u>0.376</u>	<u>0.383</u>	0.350 (-6.9%)	0.378 (-1.5%)	<u>0.344</u>	<u>0.380</u>	0.340 (-1.1%)	0.378 (-0.6%)	<u>0.402</u>	<u>0.409</u>	0.364 (-9.4%)	0.398 (-2.9%)
ETTm1	<u>0.421</u>	<u>0.383</u>	0.381 (-9.4%)	0.371 (-3.1%)	<u>0.433</u>	<u>0.381</u>	0.393 (-9.2%)	0.372 (-2.3%)	<u>0.369</u>	<u>0.369</u>	0.356 (-3.6%)	0.364 (-1.5%)	<u>0.429</u>	<u>0.416</u>	0.382 (-11.0%)	0.396 (-4.8%)
ETTm2	<u>0.291</u>	<u>0.317</u>	0.267 (-8.0%)	0.311 (-2.1%)	<u>0.295</u>	<u>0.315</u>	0.270 (-8.4%)	0.309 (-2.0%)	<u>0.276</u>	<u>0.317</u>	0.266 (-3.6%)	0.312 (-1.6%)	<u>0.332</u>	<u>0.341</u>	0.279 (-16.0%)	0.324 (-5.0%)
Electricity	<u>0.159</u>	<u>0.244</u>	0.155 (-2.5%)	0.241 (-1.0%)	<u>0.163</u>	<u>0.244</u>	0.159 (-2.1%)	0.241 (-1.1%)	<u>0.148</u>	<u>0.242</u>	0.144 (-2.7%)	0.239 (-1.4%)	<u>0.154</u>	<u>0.244</u>	0.149 (-2.9%)	0.241 (-1.1%)
Exchange	<u>0.371</u>	<u>0.412</u>	0.365 (-1.8%)	0.406 (-1.5%)	<u>0.399</u>	<u>0.421</u>	0.366 (-8.3%)	0.408 (-3.1%)	<u>0.553</u>	<u>0.494</u>	0.471 (-14.8%)	0.464 (-6.0%)	<u>0.433</u>	<u>0.446</u>	0.425 (-1.9%)	0.433 (-2.7%)
Traffic	<u>0.435</u>	<u>0.263</u>	0.425 (-2.3%)	0.258 (-1.6%)	<u>0.394</u>	<u>0.237</u>	0.389 (-1.4%)	0.235 (-0.6%)	<u>0.461</u>	<u>0.286</u>	0.436 (-5.3%)	0.278 (-3.0%)	<u>0.370</u>	<u>0.244</u>	0.367 (-0.6%)	0.241 (-1.1%)
Weather	<u>0.263</u>	<u>0.276</u>	0.239 (-9.1%)	0.267 (-3.5%)	<u>0.274</u>	<u>0.267</u>	0.241 (-12.0%)	0.260 (-2.6%)	<u>0.244</u>	<u>0.269</u>	0.238 (-2.5%)	0.266 (-1.2%)	<u>0.222</u>	<u>0.237</u>	0.205 (-7.6%)	0.230 (-2.9%)

To mitigate sensitivity to global central bias of the frozen backbone, we further align the incremental median correction between the teacher and memory module (relative to the backbone):

$$\Delta_{t,u}^{\text{mem}} = \widehat{Q}_{t,j^*,u}^{\text{mem}} - \widehat{Q}_{t,j^*,u}^{\text{base}}, \quad \Delta_{t,u}^T = \widetilde{Q}_{t,j^*,u} - \widehat{Q}_{t,j^*,u}^{\text{base}}, \quad (15)$$

where $\Delta_{t,u}^{\text{mem}}$ and $\Delta_{t,u}^T$ denote the memory module and teacher's incremental median correction relative to the frozen backbone, respectively. The alignment loss for these corrections is:

$$\mathcal{D}_\Delta(t) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \ell_\kappa(\Delta_{t,u}^{\text{mem}}, \Delta_{t,u}^T). \quad (16)$$

The final confidence-gated alignment loss is:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_t[\omega_t \cdot (\mathcal{D}_Q(t) + \eta \mathcal{D}_\Delta(t))]. \quad (17)$$

Stability Regularization $\mathcal{L}_{\text{reg}}$. Constrain the memory module to conservative behavior for uncertain retrieval (small ω_t , weak $\mathcal{L}_{\text{align}}$ supervision), avoiding over-adjustment of the frozen backbone. We first anchor its median to the backbone with weight $(1 - \omega_t)$:

$$\mathcal{L}_{\text{anchor}} = \mathbb{E}_t \left[(1 - \omega_t) \cdot \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \ell_\kappa(\widehat{Q}_{t,j^*,u}^{\text{mem}}, \widehat{Q}_{t,j^*,u}^{\text{base}}) \right]. \quad (18)$$

We additionally impose a universal regularization to prevent quantile crossing in memory predictions, ensuring monotonic outputs:

$$\mathcal{L}_{\text{cross}} = \mathbb{E}_t \left[\frac{1}{(Q-1)|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{j=1}^{Q-1} \max(0, \widehat{Q}_{t,j,u}^{\text{mem}} - \widehat{Q}_{t,j+1,u}^{\text{mem}}) \right]. \quad (19)$$

The final stability regularization loss is: $\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{anchor}} + \lambda_{\text{cross}} \mathcal{L}_{\text{cross}}$.

Training objective. We train the memory module on $\mathcal{D}_{\text{teach}}$ using a composite loss that balances ground-truth regression with distributional distillation:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}. \quad (20)$$

4.3 Inference via Adaptive Fusion

At test time, TS-Memory performs strictly retrieval-free inference:

$$\widehat{Q}_t^{\text{base}} = f_\theta(\mathbf{X}_t), \quad \widehat{Q}_t^{\text{mem}} = g_\phi(\mathbf{X}_t). \quad (21)$$

We fuse the two quantile forecasts by quantile-wise interpolation:

$$\widehat{Q}_t^{\text{final}} = (1 - \alpha) \widehat{Q}_t^{\text{base}} + \alpha \widehat{Q}_t^{\text{mem}}, \quad \alpha \in [0, 1]. \quad (22)$$

We tune α on a validation split. Inference requires only two forward passes: one through the frozen backbone and one through the memory module. No retrieval index needs to be maintained and no nearest-neighbor search is performed at serving time.

Inference-time complexity. TS-Memory adds constant overhead: one lightweight extra forward pass and quantile-wise fusion. Run-time and memory do not depend on the knowledge base since retrieval is offline. In contrast, online retrieval requires query-time search and aggregation that scale with the datastore.

5 Experiment

We evaluate TS-Memory on long-horizon multivariate forecasting with four research questions (**RQs**):

- **RQ1:** Can TS-Memory consistently improve various TSFMs across diverse datasets and different backbones? $\rightarrow$ Sec. 5.2
- **RQ2:** How does TS-Memory compare with other adaptation strategies in performance and inference latency? $\rightarrow$ Sec. 5.3
- **RQ3:** How does TS-Memory perform under backbone scaling and cross-model/cross-dataset transfer? $\rightarrow$ Sec. 5.4
- **RQ4:** What are the key design factors of TS-Memory, how do parameter scaling and sensitivity impact its performance, and what qualitative results does it achieve? $\rightarrow$ Sec. 5.5

5.1 Experimental Setup

Datasets & Evaluation Metrics. We experiment on eight widely used multivariate long-term forecasting benchmarks (ETTh1/2, ETTm1/2, Electricity, Exchange-rate, Traffic, Weather) [30, 79]. Following the standard long-horizon protocol, we evaluate across multiple forecasting horizons and report the average performance to characterize overall long-range forecasting ability [68]. For point forecasting, we report *MSE* and *MAE* [19]; for probabilistic forecasting, we additionally present *CRPS* to assess prediction quality [10]. **Baselines & Backbones.** We adopt four representative TSFMs: *ChronosBolt* [2], *Chronos2* [1], *Sundial* [39] and *TimesFM* [5], with backbone parameters frozen unless specified otherwise. We compare against three baselines: (1) *Origin*, the zero-shot deployment of frozen backbones; (2) Online retrieval augmentation baselines (*RAFT* [16], *TS-RAG* [46]), which perform test-time *kNN* retrieval

Table 3: Long-term forecasting results comparing TS-Memory with online retrieval baselines on ChronosBolt. We use the same protocol as in Table 2. Full results are in Table 11 of Appendix D.

Dataset	Origin			RAFT			TS-RAG			LoRA			TS-Memory		
Metric	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
ETTh1	0.448	0.419	0.437	0.437(-2.5%)	0.418(-0.2%)	0.432(-1.1%)	0.442(-1.3%)	0.421(+0.7%)	0.440(+0.8%)	0.428(-4.4%)	0.416(-0.6%)	0.428(-2.0%)	0.421(-6.0%)	0.414(-1.2%)	0.407(-6.8%)
ETTh2	0.367	0.380	0.246	0.357(-2.5%)	0.378(-0.5%)	0.243(-1.0%)	0.362(-1.4%)	0.385(+1.5%)	0.250(+1.6%)	0.357(-2.7%)	0.377(-0.6%)	0.252(+2.6%)	0.354(-3.4%)	0.375(-1.1%)	0.238(-3.1%)
ETTh1	0.421	0.383	0.409	0.398(-5.5%)	0.377(-1.6%)	0.400(-2.3%)	0.406(-3.6%)	0.381(-0.4%)	0.406(-0.9%)	0.392(-6.9%)	0.376(-1.7%)	0.397(-2.9%)	0.381(-9.4%)	0.371(-1.1%)	0.370(-9.7%)
ETTh2	0.291	0.317	0.205	0.281(-3.3%)	0.315(-0.9%)	0.203(-1.2%)	0.283(-2.6%)	0.318(+0.3%)	0.205(-0.2%)	0.278(-4.5%)	0.314(-1.2%)	0.199(-3.2%)	0.267(-8.0%)	0.311(-2.1%)	0.190(-7.6%)
Electricity	0.159	0.244	0.239	0.156(-1.5%)	0.244(+0.2%)	0.240(+0.2%)	0.157(-1.4%)	0.245(+0.2%)	0.240(+0.3%)	0.156(-1.8%)	0.244(-0.1%)	0.244(+1.9%)	0.155(-2.5%)	0.241(-1.0%)	0.232(-3.2%)
Exchange	0.371	0.412	0.268	0.369(-0.7%)	0.409(-0.9%)	0.257(-4.2%)	0.367(-1.2%)	0.408(+1.0%)	0.253(-5.6%)	0.366(-1.4%)	0.407(-1.2%)	0.266(+0.8%)	0.365(-1.8%)	0.406(-1.5%)	0.250(-6.9%)
Traffic	0.435	0.263	0.273	0.426(-2.0%)	0.261(-0.8%)	0.271(-0.9%)	0.427(-1.9%)	0.261(-0.7%)	0.271(-0.8%)	0.425(-2.2%)	0.260(-1.1%)	0.268(-2.0%)	0.425(-2.3%)	0.258(-1.6%)	0.267(-2.4%)
Weather	0.263	0.276	0.404	0.254(-3.5%)	0.268(-3.0%)	0.390(-3.5%)	0.262(-0.6%)	0.279(+1.2%)	0.408(+1.0%)	0.247(-6.1%)	0.268(-2.8%)	0.378(-6.4%)	0.239(-9.1%)	0.267(-3.5%)	0.370(-8.4%)

Table 4: Inference latency comparison (ms/query) for Origin, online retrieval baselines, LoRA, and TS-Memory on ChronosBolt. We report retrieval time (Retr), forward time (Fwd), total time (Total), and retrieval fraction (Frac), following the same protocol as Table 2. Full results are in Table 12 of Appendix D.

Method	Origin				RAFT				TS-RAG				LoRA				TS-Memory			
Metric	Retr	Fwd	Total	Frac	Retr	Fwd	Total	Frac	Retr	Fwd	Total	Frac	Retr	Fwd	Total	Frac	Retr	Fwd	Total	Frac
ETTh1	0	3.53	3.53	0%	0.84	3.84(+8.8%)	4.68(+32.6%)	20.0%	3.53	3.80(+7.8%)	7.33(+107.9%)	51.1%	0	4.28(+21.4%)	4.28(+21.4%)	0%	0	3.69(+4.7%)	3.69(+4.7%)	0%
ETTh2	0	3.52	3.52	0%	0.84	3.82(+8.4%)	4.65(+32.1%)	19.7%	3.50	3.83(+8.7%)	7.33(+108.2%)	50.6%	0	4.29(+21.7%)	4.29(+21.7%)	0%	0	3.66(+4.0%)	3.66(+4.0%)	0%
ETTh1	0	3.52	3.52	0%	2.62	3.79(+7.5%)	6.41(+81.9%)	43.9%	6.46	3.78(+7.4%)	10.24(+190.6%)	65.7%	0	4.24(+20.4%)	4.24(+20.4%)	0%	0	3.67(+4.1%)	3.67(+4.1%)	0%
ETTh2	0	3.52	3.52	0%	2.52	3.79(+7.5%)	6.30(+79.0%)	43.4%	6.44	3.80(+7.8%)	10.23(+190.5%)	65.6%	0	4.24(+20.4%)	4.24(+20.4%)	0%	0	3.66(+3.9%)	3.66(+3.9%)	0%
Electricity	0	3.51	3.51	0%	1.49	3.79(+8.0%)	5.28(+50.4%)	31.2%	4.62	3.79(+8.2%)	8.41(+139.7%)	57.9%	0	3.89(+11.0%)	3.89(+11.0%)	0%	0	3.64(+3.8%)	3.64(+3.8%)	0%
Exchange	0	3.53	3.53	0%	0.51	3.79(+7.5%)	4.30(+21.9%)	13.3%	3.07	3.79(+7.6%)	6.87(+94.7%)	47.7%	0	4.29(+21.7%)	4.29(+21.7%)	0%	0	3.66(+3.8%)	3.66(+3.8%)	0%
Traffic	0	3.51	3.51	0%	1.04	3.79(+8.0%)	4.82(+37.5%)	23.8%	3.90	3.79(+8.1%)	7.69(+119.2%)	53.7%	0	3.88(+10.7%)	3.88(+10.7%)	0%	0	3.64(+3.8%)	3.64(+3.8%)	0%
Weather	0	3.52	3.52	0%	2.71	3.79(+7.7%)	6.49(+84.6%)	44.7%	6.67	3.79(+7.8%)	10.47(+197.6%)	66.4%	0	3.96(+12.6%)	3.96(+12.6%)	0%	0	3.65(+3.9%)	3.65(+3.9%)	0%

from a leakage-safe datastore and fuse retrieved futures for forecasting; (3) The parameter-efficient fine-tuning baseline *LoRA*, which injects low-rank adapters into attention projection layers and only trains the adapters, with trainable parameters matched to PlugMem. **Implementation Details.** Following prior work [68], we use prediction horizons $H \in \{96, 192, 336, 720\}$ and a fixed look-back window of $L = 512$. All models are optimized via Adam [27]. For baseline methods on backbone models, we strictly adhere to their original hyperparameter settings to ensure fair comparison. All experiments are implemented in PyTorch and run on NVIDIA A800 80GB GPUs. Additional details are provided in Appendix C.

5.2 TS-Memory Performance Across Backbones

Table 2 presents long-horizon forecasting results for TS-Memory deployed on four frozen TSFM backbones across eight benchmarks. TS-Memory enhances all backbones on every dataset, confirming that retrieval-driven gains can be distilled into a compact memory module without backbone parameter update. Averaged across all dataset-backbone pairs, TS-Memory reduces MSE by 5.8% and MAE by 2.1%, with peak reductions of 16.0% (MSE) and 6.0% (MAE). Gains are most pronounced on benchmarks with severe distribution shifts and rich temporal structures (e.g., Weather and ETT minute-level datasets), where double-digit MSE reductions are common, demonstrating improved robustness to long-horizon error accumulation. For regular benchmarks (e.g., Electricity), improvements are modest yet consistent. It matches the desired plug-and-play

adapter behavior: substantial gains for poorly performing zero-shot backbones, and stability for well-aligned ones. Consistent MAE reductions further confirm broad error mitigation across time steps and variables, rather than sporadic gains in isolated cases.

Pretraining-overlap check. Our retrieval-leakage-safe construction rules out test-window leakage for the offline teacher and online retrieval baselines, but does not assume fully auditable TSFM pretraining corpora. We therefore exclude confirmable backbone-dataset overlap pairs: TimesFM on Electricity, Traffic, and Weather; Sundial on Traffic; and Chronos2 on Electricity. On the 27 pairs, TS-Memory reduces MSE by 6.22% and MAE by 2.18%, slightly stronger than the full 32-pair reductions of 5.8% and 2.1%. This suggests our gains are not driven by known pretraining overlap.

5.3 Comparison with Adaptation Baselines

Performance comparison. Table 3 compares TS-Memory with online retrieval baselines (RAFT and TS-RAG) and the parameter-efficient fine-tuning baseline *LoRA* on ChronosBolt, reporting point (MSE/MAE) and probabilistic (CRPS) metrics. Our method is the only method that consistently reduces MSE, MAE, and CRPS across all eight datasets, indicating that retrieval-distilled corrections can improve not only the mean prediction but also the reliability of the predictive distribution. In contrast, *LoRA* yields competitive reductions in MSE/MAE on some datasets, though its CRPS behavior is inconsistent and can even degrade on certain benchmarks, suggesting that low-rank parameter update may overfit point objectives

Table 5: Scaling study of TS-Memory across ChronosBolt model sizes. We use the same protocol as in Table 2. Full results are in Table 14 of Appendix D.

Model	ChronosBolt Base (205M)				ChronosBolt Small (48M)				ChronosBolt Mini (21M)				ChronosBolt Tiny (9M)			
Dataset	Origin		TS-Memory		Origin		TS-Memory		Origin		TS-Memory		Origin		TS-Memory	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	<u>0.448</u>	<u>0.419</u>	0.421 (-6.0%)	0.414 (-1.2%)	<u>0.463</u>	<u>0.422</u>	0.437 (-5.5%)	0.417 (-1.2%)	<u>0.445</u>	<u>0.421</u>	0.424 (-4.8%)	0.417 (-1.0%)	<u>0.447</u>	<u>0.422</u>	0.427 (-4.6%)	0.418 (-1.0%)
ETTh2	<u>0.367</u>	<u>0.380</u>	0.354 (-3.4%)	0.375 (-1.1%)	<u>0.362</u>	<u>0.382</u>	0.357 (-1.5%)	0.379 (-0.7%)	<u>0.370</u>	<u>0.384</u>	0.358 (-3.2%)	0.380 (-1.0%)	<u>0.367</u>	<u>0.385</u>	0.361 (-1.7%)	0.383 (-0.7%)
ETTm1	<u>0.421</u>	<u>0.383</u>	0.381 (-9.4%)	0.371 (-3.1%)	<u>0.420</u>	<u>0.385</u>	0.377 (-10.3%)	0.373 (-3.1%)	<u>0.417</u>	<u>0.385</u>	0.374 (-10.3%)	0.372 (-3.2%)	<u>0.407</u>	<u>0.385</u>	0.373 (-8.4%)	0.375 (-2.7%)
ETTm2	<u>0.291</u>	<u>0.317</u>	0.267 (-8.0%)	0.311 (-2.1%)	<u>0.284</u>	<u>0.317</u>	0.266 (-6.4%)	0.311 (-1.7%)	<u>0.290</u>	<u>0.319</u>	0.266 (-8.2%)	0.312 (-2.2%)	<u>0.284</u>	<u>0.317</u>	0.264 (-7.2%)	0.312 (-1.8%)
Electricity	<u>0.159</u>	<u>0.244</u>	0.155 (-2.5%)	0.241 (-1.0%)	<u>0.163</u>	<u>0.249</u>	0.159 (-2.5%)	0.246 (-1.1%)	<u>0.167</u>	<u>0.253</u>	0.163 (-2.5%)	0.251 (-0.8%)	<u>0.172</u>	<u>0.259</u>	0.166 (-3.1%)	0.256 (-1.2%)
Traffic	<u>0.435</u>	<u>0.263</u>	0.425 (-2.3%)	0.258 (-1.6%)	<u>0.412</u>	<u>0.264</u>	0.408 (-1.1%)	0.260 (-1.4%)	<u>0.417</u>	<u>0.269</u>	0.411 (-1.2%)	0.265 (-1.3%)	<u>0.419</u>	<u>0.276</u>	0.414 (-1.2%)	0.271 (-2.1%)
Weather	<u>0.263</u>	<u>0.276</u>	0.239 (-9.1%)	0.267 (-3.5%)	<u>0.256</u>	<u>0.270</u>	0.239 (-6.5%)	0.266 (-1.8%)	<u>0.268</u>	<u>0.283</u>	0.240 (-10.6%)	0.273 (-3.4%)	<u>0.271</u>	<u>0.286</u>	0.239 (-11.7%)	0.275 (-3.8%)

Table 6: Cross-model transfer results of TS-Memory across retrieval teachers and frozen TSFM backbones. We use the same protocol as in Table 2. Full results are in Table 15 of Appendix D.

Model	Chronos2-PlugMem												Sundial-PlugMem											
	ChronosBolt (base)				Sundial (base)				TimesFM (base)				ChronosBolt (base)				Chronos2 (base)				TimesFM (base)			
Dataset	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory	Origin	TS-Memory
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	<u>0.448</u>	<u>0.419</u>	0.420	0.416	<u>0.400</u>	<u>0.412</u>	0.395	0.410	<u>0.479</u>	<u>0.442</u>	0.436	0.430	<u>0.448</u>	<u>0.419</u>	0.409	0.416	<u>0.442</u>	0.412	0.409	<u>0.412</u>	<u>0.479</u>	<u>0.442</u>	0.424	0.427
ETTh2	<u>0.366</u>	0.380	0.342	<u>0.383</u>	<u>0.344</u>	<u>0.380</u>	0.338	0.379	<u>0.402</u>	<u>0.409</u>	0.358	0.402	<u>0.366</u>	0.380	0.339	<u>0.382</u>	<u>0.376</u>	0.383	0.347	<u>0.384</u>	<u>0.402</u>	<u>0.409</u>	0.356	0.399
ETTm1	<u>0.421</u>	<u>0.383</u>	0.375	0.375	<u>0.369</u>	<u>0.369</u>	0.356	0.366	<u>0.429</u>	<u>0.416</u>	0.375	0.390	<u>0.421</u>	<u>0.383</u>	0.367	0.375	<u>0.433</u>	<u>0.381</u>	0.370	0.378	<u>0.429</u>	<u>0.416</u>	0.372	0.390
ETTm2	<u>0.290</u>	<u>0.317</u>	0.262	0.314	<u>0.276</u>	<u>0.317</u>	0.262	0.314	<u>0.332</u>	<u>0.341</u>	0.267	0.322	<u>0.290</u>	<u>0.317</u>	0.265	0.314	<u>0.295</u>	<u>0.315</u>	0.264	0.313	<u>0.332</u>	<u>0.341</u>	0.273	0.325
Electricity	<u>0.159</u>	<u>0.244</u>	0.154	0.242	<u>0.148</u>	<u>0.242</u>	0.144	0.238	<u>0.154</u>	<u>0.244</u>	0.151	0.242	<u>0.159</u>	<u>0.244</u>	0.154	0.243	<u>0.163</u>	<u>0.244</u>	0.157	0.242	<u>0.154</u>	<u>0.244</u>	0.151	0.243
Exchange	<u>0.371</u>	<u>0.412</u>	0.355	0.405	<u>0.553</u>	<u>0.494</u>	0.408	0.444	<u>0.433</u>	<u>0.446</u>	0.383	0.427	<u>0.371</u>	<u>0.412</u>	0.365	0.407	<u>0.399</u>	<u>0.421</u>	0.392	0.417	<u>0.433</u>	<u>0.446</u>	0.420	0.438
Traffic	<u>0.435</u>	<u>0.263</u>	0.417	0.260	<u>0.461</u>	<u>0.286</u>	0.427	0.275	<u>0.370</u>	<u>0.244</u>	0.366	0.243	<u>0.435</u>	<u>0.263</u>	0.418	0.261	<u>0.394</u>	0.237	0.388	<u>0.240</u>	<u>0.370</u>	<u>0.244</u>	0.367	0.243
Weather	<u>0.263</u>	<u>0.276</u>	0.235	0.269	<u>0.244</u>	<u>0.269</u>	0.235	0.269	<u>0.222</u>	<u>0.237</u>	0.198	0.232	<u>0.263</u>	<u>0.276</u>	0.235	0.273	<u>0.274</u>	0.267	0.233	<u>0.268</u>	<u>0.222</u>	<u>0.237</u>	0.198	0.233

or disturb the model’s uncertainty calibration. Overall, these results support the core motivation of TS-Memory: distilling retrieval guidance into a lightweight module can preserve the retrieval style robustness while maintaining stable probabilistic forecasts.

Inference latency comparison. Table 4 reports inference latency for all methods under the same evaluation protocol. Online retrieval introduces substantial query-dependent overhead: TS-RAG leads to nearly two-fold end-to-end slowdown, with retrieval dominating total latency, while RAFT is more efficient but still incurs noticeable overhead, especially when the datastore grows. LoRA avoids retrieval during inference, but still adds non-trivial constant latency due to layer-wise low-rank projections. TS-Memory removes retrieval entirely and introduces only a small constant overhead via a single lightweight forward pass, making its runtime close to the frozen-backbone baseline. This efficiency profile is important in deployment settings where retrieval infrastructure can become the dominant bottleneck due to index maintenance costs, latency variance, and poor scaling with horizon or datastore size.

5.4 Transfer generality

Scaling across model sizes. Table 5 shows that a TS-Memory module trained once on ChronosBolt-Base (205M) can be reused as-is on smaller ChronosBolt variants (down to 9M parameters). Across all eight benchmarks, it consistently improves MSE/MAE

Table 7: Train→Test dataset pairs under each setting. $\emptyset$: Zero Shot; ETT_{pool}: pooled four ETT datasets.

Setting	Train Dataset → Test Dataset			
Origin	$\emptyset \rightarrow$ ETTh1	$\emptyset \rightarrow$ ETTh2	$\emptyset \rightarrow$ ETTm1	$\emptyset \rightarrow$ ETTm2
Cross-Dom	Weather → ETTh1	Weather → ETTh2	Weather → ETTm1	Weather → ETTm2
Dom-Shift	ETTh2 → ETTh1	ETTh1 → ETTh2	ETTm2 → ETTm1	ETTm1 → ETTm2
Multi-Dom	ETT _{pool} → ETTh1	ETT _{pool} → ETTh2	ETT _{pool} → ETTm1	ETT _{pool} → ETTm2
In-Dom	ETTh1 → ETTh1	ETTh2 → ETTh2	ETTm1 → ETTm1	ETTm2 → ETTm2

without re-optimization, implying that the distilled correction signal is not tied to a specific backbone capacity. The average MSE gain remains around 5%, and improvements are often larger on challenging datasets (e.g., Weather and minute-level ETT), where smaller frozen backbones are more prone to long-horizon drift. It suggests that TS-Memory can act as a constant-cost adapter that upgrades compute-constrained backbones without retrieval infrastructure.

Cross-backbone transfer. Table 6 evaluates transfer across TSFM families by distilling PlugMem from a single retrieval teacher and applying it to different frozen target backbones. We observe strong cross-model transfer: PlugMem distilled from Chronos2 improves ChronosBolt, Sundial, and TimesFM, and PlugMem distilled from Sundial similarly benefits ChronosBolt, Chronos2, and TimesFM. Average MSE reductions fall in the 6%–10% range depending on the target backbone, supporting that TS-Memory captures generalizable retrieval-driven correction patterns. MAE gains are sometimes

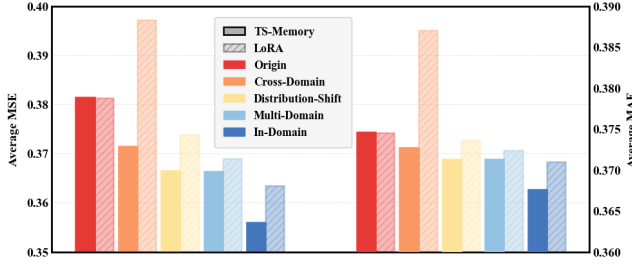


Figure 3: TS-Memory vs. LoRA under different train-test domains. Full per-dataset results are provided in Table 16.

smaller and can be near neutral for some teacher–target pairs, which is expected because different architectures may exhibit different output calibration and loss sensitivity. However, the overall trend remains consistently positive across targets.

Cross-backbone transfer is enabled by three design choices. First, PlugMem consumes the raw context window rather than backbone hidden states, making it independent of TSFM internals. Second, it predicts quantiles in the common forecasting output space, so the learned memory can be fused with different probabilistic backbones. Third, the Δ -alignment objective distills retrieval-induced corrections relative to the frozen backbone, encouraging PlugMem to learn domain-level temporal correction patterns rather than model-specific representations.

Generalization under domain shift. Figure 3 evaluates our method under domain shift on the ChronosBolt backbone. Table 7 summarizes the train→test domain relations, where *train* denotes the domain used to construct offline retrieval supervision and *test* the target domain for evaluation. Even with mismatched supervision, TS-Memory provides non-trivial gains: cross-domain training reduces average MSE by 2.7%, while more related transfer settings yield larger and more stable improvements, reaching 3.5% MSE reduction under distribution-shift transfer and 4.0% under multi-domain transfer. In-domain supervision performs best, achieving 6.8% lower MSE and 1.9% lower MAE on average, confirming that retrieval supervision is most effective when the knowledge base aligns with the target distribution. This smooth scaling with alignment suggests that TS-Memory does not rely on brittle dataset-specific shortcuts, but instead distills transferable temporal motifs. Meanwhile, LoRA exhibits weaker cross-domain robustness and can underperform the frozen backbone under domain mismatch.

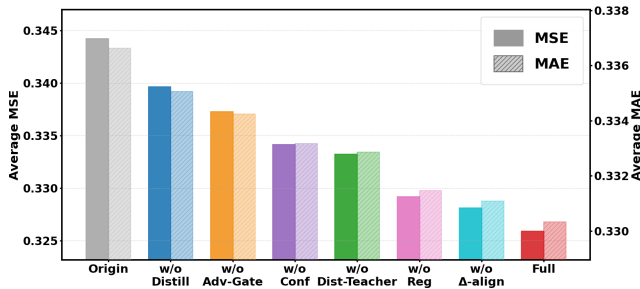


Figure 4: Ablation study of TS-Memory components.

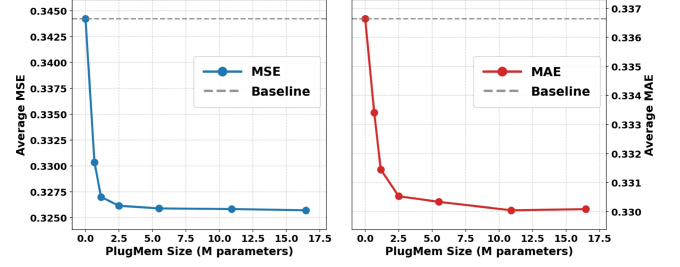


Figure 5: Scaling Analysis of PlugMem Capacity.

5.5 Model Analysis

Ablation study. Figure 4 reports component ablations averaged across datasets and horizons, where the full TS-Memory achieves the best overall performance (5.3% lower MSE and 1.9% lower MAE compared to the frozen backbone). When retrieval distillation is removed, the gains largely vanish, indicating that the offline retrieval teacher is the primary source of transferable corrective signal. Moreover, disabling either the advantage gate or the confidence-aware weighting consistently degrades performance, suggesting that *selective* and *confidence-conditioned* distillation is crucial: it filters unreliable retrieval targets and prevents negative transfer when retrieved futures are noisy or mismatched. Finally, auxiliary stability terms (e.g., Reg and Δ -align) yield modest yet consistent improvements, primarily by stabilizing training and discouraging over-correction so that the learned memory behaves as a conservative correction module on top of the strong frozen prior.

Scaling PlugMem capacity. Figure 5 studies the effect of PlugMem capacity under a fixed training recipe and frozen backbone, focusing on long-horizon forecasting. Even a small PlugMem recovers most of the gain over the frozen TSFM, suggesting that the retrieval teacher mainly provides low-complexity correction patterns rather than requiring large task-specific parameterization. As capacity increases beyond a moderate size, improvements saturate and can slightly regress on some datasets, indicating over-correction or idiosyncratic artifacts from the teacher. Overall, a moderately sized PlugMem offers the best accuracy–efficiency trade-off for plug-and-play deployment. Accordingly, we design three PlugMem variants of different sizes (see Appendix D, Table 17).

Sensitivity to the fusion weight. Figure 6 analyzes the fusion weight α for combining backbone and PlugMem quantile forecasts. Performance improves as α increases from 0, confirming that

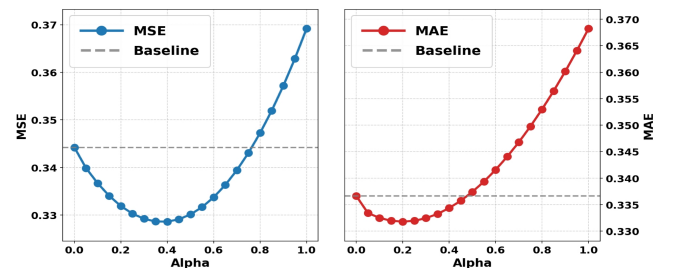


Figure 6: Sensitivity Analysis of the Fusion Weight α .

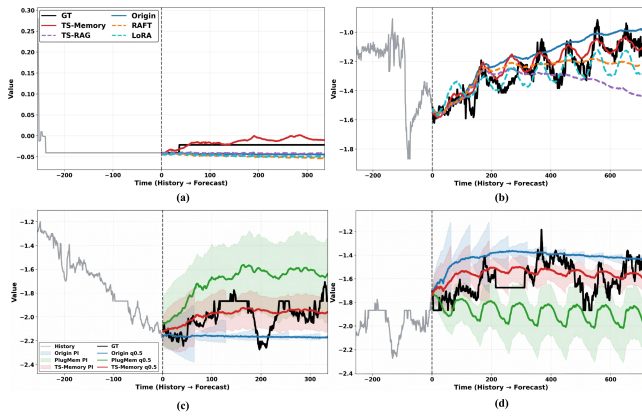
Table 8: Gate activation statistics during confidence-gated memory distillation.

Dataset	Act. ($\chi_t=1$)	Mean ω_t	Mean Conf.	Adv. Margin
ETT Avg	87.7%	0.479	0.562	0.173
Weather	96.7%	0.680	0.701	0.245
Traffic	68.1%	0.067	0.100	0.025
Electricity	50.8%	0.062	0.113	0.013
Exchange	99.4%	0.102	0.102	0.184

the memory branch provides complementary corrections rather than redundant predictions. However, excessively large α degrades MSE/MAE, as over-reliance on memory overrides the strong priors learned by the backbone and amplifies residual bias; in the extreme case of $\alpha = 1$, the backbone is discarded entirely in favor of PlugMem alone. Notably, the curve exhibits an optimum and remains flat near it, enabling stable, lightweight validation-based tuning.

Gate statistics. Table 8 summarizes how the advantage gate behaves during memory distillation. Here, $\chi_t=1$ indicates that the retrieval teacher outperforms the frozen backbone on a training window, ω_t is the confidence-weighted gate coefficient, and Adv. Margin measures the teacher’s improvement over the backbone. Across all eight datasets, the teacher outperforms the backbone on 83.5% of training windows, but the gate is not always-on. Activation and ω_t are lower on well-calibrated benchmarks such as Electricity and Traffic, while activation is much higher on shifting domains such as Weather and Exchange. This indicates that the gate acts as a domain-adaptive filter, selectively distilling useful retrieval signals while suppressing weak or uncertain teacher corrections.

Qualitative visualizations. Figure 7 presents representative cases highlighting behavioral differences among methods. In (a), the frozen backbone and several baselines remain nearly flat after the forecast start (dashed line), failing to track the target’s subsequent level shift, whereas TS-Memory produces a clear corrective trend that is better aligned with the ground truth, consistent with learning compact bias corrections from retrieval distillation. In (b), TS-RAG exhibits noticeable long-horizon drift, and LoRA/RAFT

**Figure 7: Qualitative visualizations comparing TS-Memory with the frozen backbone and adaptation baselines.**

introduce phase or smoothness artifacts, while TS-Memory stays closer to the ground-truth trajectory with reduced accumulated error, demonstrating that distilling retrieval offline into a memory module preserves adaptability without the instability inherent in online retrieval. In (c,d), probabilistic forecast comparisons illustrate why selective, confidence-aware training matters: PlugMem alone can become overly aggressive, producing a mis-centered median and excessively wide or oscillatory prediction intervals, while the frozen backbone tends to be conservative but biased; TS-Memory balances both, yielding well-calibrated interval width and a more accurately centered median. Overall, these plots corroborate the ablation findings: advantage-gated, confidence-weighted distillation enables TS-Memory to absorb useful retrieval signals while suppressing unreliable teacher artifacts, producing a robust correction mechanism rather than a fragile replacement predictor.

6 Conclusion and Future Work

We introduce **Parametric Memory Distillation**, a retrieval-to-memory paradigm that compiles retrieval-conditioned predictive distributions into a lightweight, plug-and-play module for adapting frozen Time Series Foundation Models. Realized as **TS-Memory**, our approach distills confidence-aware quantile corrections from an offline, leakage-safe k NN teacher and enables retrieval-free inference through simple fusion with the frozen backbone. Experiments across multiple TSFM backbones and long-horizon benchmarks demonstrate consistent improvements in both point and probabilistic forecasting with negligible inference overhead. Future directions include continual memory updates under evolving domains and richer teacher constructions with improved reliability estimation.

Limitations

TS-Memory focuses on long-horizon probabilistic forecasting; extending it to imputation, anomaly detection, and decision-focused forecasting remains future work. Its shift alignment mainly handles additive channel-wise offsets, so temporal warping, frequency shifts, or abrupt regime changes may reduce reliability. When the retrieval corpus poorly matches the deployment domain, teacher targets can be noisy; the advantage gate, anchoring loss, and backbone fusion mitigate but cannot eliminate this risk. We use a validation-tuned fusion weight for each dataset-backbone pair at test time, leaving input-adaptive fusion for future study.

Acknowledgement

We thank the reviewers for their valuable comments and efforts to improve this manuscript. This work is mainly supported by the Guangdong Basic and Applied Basic Research Foundation (No. 2025A1515011994). This work is also supported by the National Natural Science Foundation of China (No. 62402414), Guangdong Provincial Project 2025D03J0014, Guangzhou Municipal Science and Technology Project (No. 2023A03J0011), the Guangzhou Industrial Information and Intelligent Key Laboratory Project (No. 2024A03J0628), and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No. 2023B1212010007). Raymond Chi-Wing Wong was supported by the fund PRP/004/25FX.

References

- [1] Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. 2025. Chronos-2: From univariate to universal forecasting.
- [2] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series.
- [3] Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hananeh Hajishirzi, and Wen-tau Yih. 2024. Reliable, adaptable, and attributable language models with retrieval.
- [4] Maximilian Beichter, Nils Friederich, Janik Pinter, Dorina Werling, Kaleb Phipps, Sebastian Beichter, Oliver Neumann, Ralf Mikut, Veit Hagenmeyer, and Benedikt Heidrich. 2025. Decision-focused fine-tuning of time series foundation models for dispatchable feeder optimization. 100533 pages.
- [5] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting.
- [6] Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam Nguyen, Wesley M Gifford, Chandra Reddy, and Jayant Kalagnanam. 2024. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems* 37 (2024), 74147–74181.
- [7] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. 6491–6501 pages.
- [8] Xinghong Fu, Masanori Hirano, and Kentaro Imajo. 2025. Financial Fine-tuning a Large Time Series Model.
- [9] Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. 2023. TimeGPT-1.
- [10] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association* 102, 477 (2007), 359–378.
- [11] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. Moment: A family of open time-series foundation models.
- [12] Lars Graf, Thomas Ortner, Stanisł. Wołński, Angeliki Pantazi, et al. 2025. Flow-state: Sampling rate invariant time series forecasting.
- [13] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines.
- [14] Divij Gupta, Anubhav Bhatti, and Surajsinh Parmar. 2024. Beyond LoRA: Exploring Efficient Fine-Tuning Techniques for Time Series Foundational Models.
- [15] Divij Gupta, Anubhav Bhatti, Suraj Parmar, Chen Dan, Yuwei Liu, Bingjie Shen, and San Lee. 2024. Low-rank adaptation of time series foundational models for out-of-domain modality forecasting. 382–386 pages.
- [16] Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. 2025. Retrieval augmented time series forecasting.
- [17] Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. 2023. Domain adaptation for time series under feature and label shifts. 12746–12774 pages.
- [18] Fabian Hinder, Valerie Vaquet, and Barbara Hammer. 2024. One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: detecting concept drift. *Frontiers in Artificial Intelligence* 7 (2024), 1330257.
- [19] Rob J Hyndman and Anne B Koehler. 2006. Another look at measures of forecast accuracy. *International journal of forecasting* 22, 4 (2006), 679–688.
- [20] Sheikh Mohammad Idrees, M Afshar Alam, and Parul Agarwal. 2019. A prediction approach for stock market volatility based on time series data. *Ieee Access* 7 (2019), 17287–17298.
- [21] Gautier Izacard and Edouard Grave. 2021. Distilling Knowledge from Reader to Retriever for Question Answering.
- [22] Chao Jin, Zili Zhang, Xuanlin Jiang, Fangyue Liu, Shufan Liu, Xuanzhe Liu, and Xin Jin. 2025. Ragcache: Efficient knowledge caching for retrieval-augmented generation. *ACM Transactions on Computer Systems* 44, 1 (2025), 1–27.
- [23] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models.
- [24] Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. 2022. Domain adaptation for time series forecasting via attention sharing. 10280–10297 pages.
- [25] Baoyu Jing, Si Zhang, Yada Zhu, Bin Peng, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. 2022. Retrieval based time series forecasting.
- [26] Nouha Karauli, Denis Coquenot, Elisa Fromont, Martial Mermillod, and Marina Reyboz. 2025. Are Time Series Foundation Models Susceptible to Catastrophic Forgetting?
- [27] Diederik P Kingma. 2014. Adam: A method for stochastic optimization.
- [28] Dani Kiyasseh, Tingting Zhu, and David A Clifton. 2021. Clocs: Contrastive learning of cardiac signals across space, time, and patients. 5606–5615 pages.
- [29] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. 30 pages.
- [30] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. 95–104 pages.
- [31] Thomas L Lee, William Toner, Rajkarn Singh, Artjom Joosen, and Martin Asenov. 2025. Lightweight Online Adaption for Time Series Foundation Model Forecasts.
- [32] Songxin Lei, Chunming Ma, Haomin Wen, Yexin Li, Lizhenghe Chen, Qianyu Yang, Fugue Tsung, Lei Chen, Sijie Ruan, and Yuxuan Liang. 2026. Hierarchical Reinforcement Learning for Cooperative Air-Ground Delivery in Urban System.
- [33] Songxin Lei, Qiongyan Wang, Yanchen Zhu, Hanyu Yao, Sijie Ruan, Weilin Ruan, Yuyu Luo, Huaming Wu, and Yuxuan Liang. 2025. A Game-Theoretic Spatio-Temporal Reinforcement Learning Framework for Collaborative Public Resource Allocation.
- [34] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.
- [35] Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. 2024. Retrieval-augmented diffusion models for time series forecasting. *Advances in Neural Information Processing Systems* 37 (2024), 2766–2786.
- [36] Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. 2024. Moirai-moe: Empowering time series foundation models with sparse mixture of experts.
- [37] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting.
- [38] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024. Timer-xl: Long-context transformers for unified time series forecasting.
- [39] Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2025. Sundial: A family of highly capable time series foundation models.
- [40] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024. Timer: Generative pre-trained transformers are large time series models.
- [41] Yongzheng Liu, Siru Zhong, Gefeng Luo, Weilin Ruan, and Yuxuan Liang. 2025. Towards Multi-Scenario Forecasting of Building Electricity Loads with Multimodal Data. 2188–2196 pages.
- [42] Sisuo Lyu, Siru Zhong, Weilin Ruan, Qingxiang Liu, Qingsong Wen, Hui Xiong, and Yuxuan Liang. 2025. OccamVTS: Distilling Vision Models to 1% Parameters for Time Series Forecasting.
- [43] Francisco Martínez, María P Frías, Francisco Charte, and Antonio J Rivera. 2019. Time Series Forecasting with KNN in R: the tsfkn Package.
- [44] Francisco Martínez, María Pilar Frías, María Dolores Pérez, and Antonio Jesús Rivera. 2019. A methodology for applying k-nearest neighbor to time series forecasting. 2019–2037 pages.
- [45] Francisco Martínez, María Pilar Frías, María Dolores Pérez-Godoy, and Antonio Jesús Rivera. 2018. Dealing with seasonality by narrowing the training set in time series forecasting with kNN. *Expert systems with applications* 103 (2018), 38–48.
- [46] Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Yiming Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. 2025. Tsr-ag: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster.
- [47] Lin Ning, Harsh Lara, Meiqi Guo, and Abhinav Rastogi. 2025. Mode: Effective multi-task parameter efficient fine-tuning with a mixture of dyadic experts. 8233–8246 pages.
- [48] Wenzhe Niu, Zongxia Xie, Yanru Sun, Wei He, Man Xu, and Chao Hao. 2025. LangTime: A Language-Guided Unified Model for Time Series Forecasting with Proximal Policy Optimization.
- [49] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning.
- [50] Zhongzheng Qiao, Chenghao Liu, Yiming Zhang, Ming Jin, Quang Pham, Qingsong Wen, PN Suganthan, Xudong Jiang, and Savitha Ramasamy. 2025. Multi-Scale Finetuning for Encoder-based Time Series Foundation Models.
- [51] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, et al. 2023. Lag-llama: Towards foundation models for time series forecasting.
- [52] Hadi Rezaei, Hamidreza Faaljou, and Gholamreza Mansourfar. 2021. Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications* 169 (2021), 114332.
- [53] Weilin Ruan, Wei Chen, Xilin Dang, Jianxiang Zhou, Weichuang Li, Xu Liu, and Yuxuan Liang. 2025. ST-LoRA: Low-rank adaptation for spatio-temporal forecasting. 345–361 pages.
- [54] Weilin Ruan, Xilin Dang, Ziyu Zhou, Sisuo Lyu, and Yuxuan Liang. 2025. RAST: A Retrieval Augmented Spatio-Temporal Framework for Traffic Prediction. arXiv:2508.16623 [cs.LG] <https://arxiv.org/abs/2508.16623>
- [55] Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, et al. 2023. S-lora: Serving thousands of concurrent lora adapters.

- [56] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. 2024. Time-moe: Billion-scale time series foundation models with mixture of experts.
- [57] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning.
- [58] Yanru Sun, Emadeldeen Eldele, Zongxia Xie, Yucheng Wang, Wenzhe Niu, Qinghua Hu, Chee Keong Kwoh, and Min Wu. 2025. Adapting llms to time series forecasting via temporal heterogeneity modeling and semantic alignment.
- [59] Yanru Sun, Zongxia Xie, Yanhong Chen, and Qinghua Hu. 2022. Accurate solar wind speed prediction with multimodality information.
- [60] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. 1199–1208 pages.
- [61] Samya Tajmouati, Bouazza EL Wahbi, Adel Bedoui, Abdallah Abarda, and Mohamed Dakkon. 2024. Applying k-nearest neighbors to time series forecasting: Two new approaches. *Journal of Forecasting* 43, 5 (2024), 1559–1574.
- [62] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems* 35 (2022), 21831–21843.
- [63] Maria Tzelepi, Charalampos Symeonidis, Paraskevi Nousi, Efstratios Kakaletsis, Theodoros Manousis, Pavlos Tosidis, Nikos Nikolaidis, and Anastasios Tefas. 2023. Deep learning for energy time-series analysis and forecasting.
- [64] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning.
- [65] Qingsong Wen, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, Tieniu Tan, et al. 2023. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. *Advances in Neural Information Processing Systems* 36 (2023), 69949–69980.
- [66] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks.
- [67] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified training of universal time series forecasting transformers.
- [68] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis.
- [69] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* 34 (2021), 22419–22430.
- [70] Congxi Xiao, Jingbo Zhou, Yixiong Xiao, Xinjiang Lu, Le Zhang, and Hui Xiong. 2025. TimeFound: A Foundation Model for Time Series Forecasting.
- [71] Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. 2025. Timerag: Boosting llm time series forecasting via retrieval-augmented generation. 5 pages.
- [72] Zhengmao Ye, Dengchun Li, Zetao Hu, Tingfeng Lan, Jian Sha, Sicong Zhang, Lei Duan, Jie Zuo, Hui Lu, Yuanchun Zhou, et al. 2023. mLoRA: Fine-Tuning LoRA Adapters via Highly-Efficient Pipeline Parallelism in Multiple GPUs.
- [73] Yongbo Yu, Jari Peepkorn, Johannes De Smedt, and Jochen De Weerd. 2025. Time Series Foundation Models for Process Model Forecasting.
- [74] Lifan Zhao and Yanyan Shen. 2025. Proactive model adaptation against concept drift for online time series forecasting. 2020–2031 pages.
- [75] Lifan Zhao, Yanyan Shen, Zhaoyang Liu, Xue Wang, and Jiaji Deng. 2025. Less is More: Unlocking Specialization of Time Series Foundation Models via Structured Pruning.
- [76] Siru Zhong, Junjie Qiu, Yangyu Wu, Yiqiu Liu, Yuanpeng He, Zhongwen Rao, Bin Yang, Chenjuan Guo, Hao Xu, and Yuxuan Liang. 2026. Learning to Factorize and Adapt: A Versatile Approach Toward Universal Spatio-Temporal Foundation Models.
- [77] Siru Zhong, Junjie Qiu, Yangyu Wu, Xingchen Zou, Zhongwen Rao, Bin Yang, Chenjuan Guo, Hao Xu, and Yuxuan Liang. 2025. Learning to Factorize Spatio-Temporal Foundation Models.
- [78] Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. 2025. Time-VLM: Exploring Multimodal Vision-Language Models for Augmented Time Series Forecasting.
- [79] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. 11106–11115 pages.
- [80] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. 27268–27286 pages.
- [81] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* 36 (2023), 43322–43355.
- [82] Yun Zhu, Jia-Chen Gu, Caitlin Sikora, Ho Ko, Yinxiao Liu, Chu-Cheng Lin, Lei Shu, Liangchen Luo, Lei Meng, Bang Liu, et al. 2024. Accelerating inference of retrieval-augmented generation via sparse context selection.