REMEMBER, DON'T DESCEND: ONLINE RETRIEVAL EPISODIC MEMORY FOR TEST-TIME ADAPTATION IN TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series forecasting models often suffer performance degradation after deployment as the data distribution evolves over time. Existing test-time adaptation (TTA) methods for time-series forecasting address this issue by continually updating lightweight adaptation parameters using newly observed values. However, such updates persist feedback from a particular time point in the model parameters, so an inappropriate update may adversely affect subsequent predictions that are unrelated to the context in which the update was made. This raises a fundamental question: how can a forecasting model effectively adapt to a changing environment by exploiting the large amount of data that accumulates at inference time? We answer this question with a substantially simpler approach based on an online memory bank, demonstrating that it can outperform the prevailing paradigm of parameter-update-based TTA without requiring continual parameter optimization. Our findings suggest that time-series TTA can move beyond adaptation centered on per-step parameter updates toward a memory-centric adaptation paradigm, in which experience accumulated over time is explicitly retained and leveraged to continually respond to evolving distributions. Across five forecasting backbones, six datasets, and four forecast horizons, our method, OREM (Online Retrieval Episodic Memory), reduces MSE and MAE by 5.43% and 3.37% on average, respectively. OREM improves MSE over the frozen predictor in all 120 experimental settings. OREM also substantially reduces harmful corrections to the frozen backbone's forecasts compared with existing TTA methods, resulting in more reliable forecast adaptation.

1 INTRODUCTION

Time-series data differ from conventional static data in that new observations are generated and accumulated sequentially over time. Accordingly, time-series forecasting is performed repeatedly as newly observed values are incorporated into subsequent input windows. As a result, forecasting models deployed in the real world continually encounter an evolving stream of data beyond the fixed dataset available during training. A key challenge is that the underlying data distribution can shift over time, causing deployment data to differ from the patterns observed during training. Seasonal patterns and long-term trends may change, and external factors can alter the statistical properties of the observed data. Consequently, patterns observed during training are not guaranteed to persist at inference time, and a forecasting model that performs well at training time may gradually lose accuracy as the deployment distribution evolves (Kim et al., 2022; Wu et al., 2021; Liu et al., 2022; 2023).

At the same time, time-series forecasting has an important property that makes post-deployment adaptation particularly appealing. Although the ground truth for a future prediction is unavailable at the time the prediction is made, the corresponding future values are naturally revealed as time progresses. Thus, as deployment continues, previous predictions and their realized futures continuously accumulate. Time-series forecasting can therefore be viewed as a setting in which the data distribution may evolve over time, while new feedback that can be used for adaptation also becomes continually available. To exploit this property, recent test-time adaptation (TTA) methods for time-series forecasting adapt a pretrained model using newly observed feedback during deployment rather

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

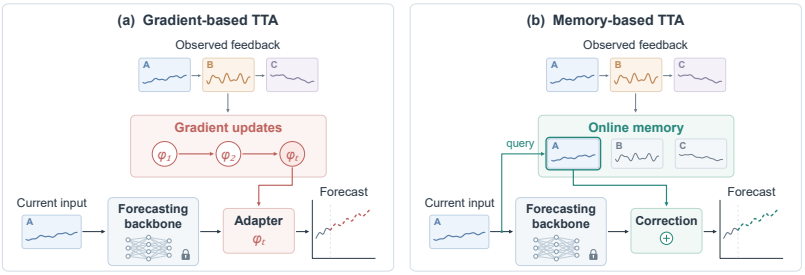


Figure 1: **Conceptual illustration of gradient-based and memory-based TTA.** Gradient-based TTA updates model parameters with observed feedback, whereas memory-based TTA retrieves relevant past feedback to correct predictions.

than keeping it entirely fixed (Kim et al., 2025; Medeiros et al., 2025). A common approach is to repeatedly update an adapter or a subset of model parameters online as new targets become available. In other words, the model state is incrementally modified using newly revealed ground-truth values, and the updated model is then used for subsequent predictions.

However, it is unclear whether continually updating parameters with newly observed feedback is always appropriate in an evolving time-series environment. Each update is based on recently observed values, yet its effect persists in the model parameters and continues to influence subsequent predictions. Moreover, adjacent time periods do not necessarily exhibit similar patterns or belong to the same regime, so an update that is useful for a particular period may become inappropriate under a later regime. As such updates repeatedly accumulate over time, continual parameter changes may instead destabilize predictions or increase forecasting error.

This limitation raises an essential question:

> Can a forecasting model effectively adapt to a changing environment by exploiting the large amount of data accumulated during deployment, without modifying its parameters?

We show that a simple online memory bank is sufficient to achieve effective adaptation without a single parameter update. Our proposed method, OREM (Online Retrieval Episodic Memory), performs test-time adaptation without updating any model parameters. As deployment proceeds, episodes whose future values have become fully observed are continuously stored in an evolving online memory. For each incoming query, OREM retrieves past episodes that are relevant to the current input and uses the information contained in these episodes to correct the prediction of a frozen backbone. To incorporate retrieved evidence more reliably, OREM adjusts the magnitude of the correction according to retrieval confidence and explicitly bounds its influence, preventing memory-based adaptation from excessively altering the backbone prediction.

We show that this simple memory-based online adaptation provides an effective alternative to parameter-update-based TTA. Across five forecasting backbones, six datasets, and four forecasting horizons, for a total of 120 settings, OREM reduces MSE and MAE by 5.43% and 3.37% on average, respectively, relative to the frozen predictor, and improves MSE in all 120 settings. Our ablation studies further show that the main gains arise from the continually evolving online memory together with the bounded correction mechanism. OREM also substantially reduces corrections that move the frozen forecast away from the ground truth compared with existing TTA methods.

These results offer a different perspective on test-time adaptation for time-series forecasting. Whereas existing TTA methods primarily focus on incorporating deployment feedback through parameter updates, we show that remembering accumulated experience and selectively retrieving it when needed can itself serve as an effective adaptation mechanism. This suggests a new direction for time-series TTA that moves beyond continual parameter optimization toward memory-based adaptation to evolving environments.

Our main contributions are as follows:

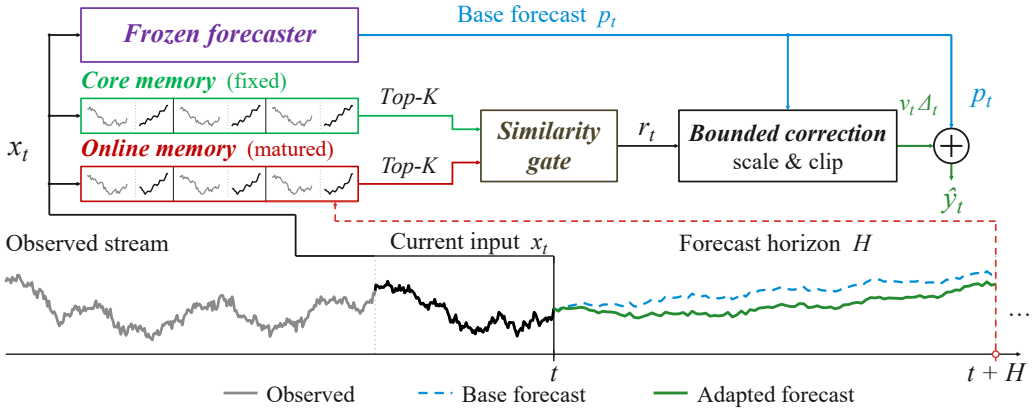


Figure 2: **Overview of OREM.** The frozen forecaster is corrected using query-relevant episodes retrieved from a fixed core bank and a causally updated online memory. Bank-specific retrievals are combined before a bounded correction is applied. An episode enters online memory only after its full forecast horizon has been observed.

- We identify a limitation of existing forecasting TTA methods that continually incorporate newly observed feedback through online parameter updates, and introduce a new adaptation paradigm that leverages online memory to adapt to an evolving test stream.
- We propose OREM to instantiate this paradigm. OREM keeps the forecasting backbone frozen, retrieves past experiences relevant to the current query, and uses them to reliably correct the frozen backbone's prediction.
- We extensively evaluate OREM across 120 settings spanning five backbones, six datasets, and four forecast horizons. OREM improves the MSE of the frozen backbone in every setting, with average improvements of 5.43% in MSE and 3.37% in MAE. Moreover, a bounded online memory retains most of the performance gain of the full bank, while additional analyses **show** more stable and effective adaptation than existing TTA methods.

2 Related Work

Forecasting under Distribution Shift and Online Adaptation. A variety of approaches have been explored to mitigate time-varying distribution shifts and non-stationarity in time-series forecasting. Early studies focused on reducing such distribution shifts during training through normalization or decomposition (Kim et al., 2022; Wu et al., 2021; Liu et al., 2022; 2023). Meanwhile, test-time adaptation (TTA), which has been largely developed in computer vision, introduced a paradigm for addressing distribution shift by directly adjusting model parameters using test data after deployment (Sun et al., 2020; Wang et al., 2021; Niu et al., 2022; Wang et al., 2022). This line of research has subsequently been extended to forecasting, where the fact that prediction targets become observable over time enables methods to continually update the forecaster or a lightweight adapter using newly observed feedback (Pham et al., 2023; Zhang et al., 2023b; Kim et al., 2025; Medeiros et al., 2025; Im & Kwon, 2026). In contrast, OREM exploits such deployment feedback through the **accumulation and retrieval of episodic memory rather than parameter updates**.

Retrieval- and Memory-Based Prediction. Retrieval-based prediction follows the intuition of classical k-nearest neighbors (k-NN), which exploits the outcomes of past examples that are similar to the current input. This idea of instance-based prediction has been extended to language modeling by combining frozen models with nearest-neighbour interpolation (Khandelwal et al., 2020), and to classification through predictions based on prototypes or nearest-neighbor information (Iwasawa & Matsuo, 2021; Zhang et al., 2023a; Jang et al., 2023; Chen et al., 2022; Karmanov et al., 2024). More recently, retrieval-augmented forecasting has been proposed for time-series forecasting, where historical windows similar to the current window are retrieved and their subsequent futures are used for prediction (Tire et al., 2026; Ning et al., 2025); however, RAFT and PFRP use a **fixed training corpus** as the retrieval source (Han et al., 2025; Du et al., 2026). OREM extends this k-NN-based

retrieval intuition to **online adaptation** by continually accumulating newly observed deployment episodes in memory and selectively leveraging experiences relevant to the current query, thereby adapting to an evolving distribution.

3 METHOD: ONLINE EPISODIC RETRIEVAL

3.1 PROBLEM FORMULATION

At each time step t, a frozen forecaster produces $p_t = f_{\theta_0}(x_t)$ from the latest input window $x_t \in \mathbb{R}^{L\times C}$. We refer to the current input window x_t as the retrieval query. At each subsequent time step, the window shifts forward by one step, forming a new query. Here L is the input length, C is the number of channels, and $p_t \in \mathbb{R}^{H\times C}$ predicts the next H observations. We denote the realized future by y_t and the adapted prediction by $\hat{y}_t$, both with the same shape as p_t. A past episode j is considered matured at time t once its full H-step target window is observable, which occurs when $j + H \le t$. Only matured episodes are eligible for retrieval, so $\hat{y}_t$ is produced without access to y_t, strictly preventing target leakage from the current forecast horizon. OREM keeps its model parameters θ_0 fixed and updates its episodic memory $\mathcal{M}_t$ and coefficient statistics from matured feedback without gradient optimization.

3.2 ACCUMULATING DEPLOYMENT EXPERIENCE

OREM maintains historical and deployment-time memories so that retrieval can leverage both stable past patterns and newly observed test-time experience. OREM uses two episodic memory banks, $B \in \{\text{core}, \text{online}\}$, that provide complementary retrieval candidates. The online bank stores each matured deployment episode as an input–future pair. It starts empty and receives newly matured episodes before each prediction, gradually accumulating patterns from the deployment distribution. This enables OREM to adapt its predictions to distribution shifts at test time by leveraging newly observed deployment patterns. The core bank is constructed from complete episodes extracted with a stride of one from the training and validation splits and remains fixed throughout deployment. It provides retrieval candidates when deployment feedback is scarce and preserves historical patterns throughout deployment.

To compare temporal patterns across different levels and scales, OREM normalizes each episode using its own input window. Because temporal patterns and scales can differ across channels, OREM processes each channel independently for normalization, retrieval, and correction. For channel $c \in \{1, \dots, C\}$, let $\mu_{t,c}$ and $\nu_{t,c} = \mathrm{std}(x_{t,c}) + \epsilon$ be the input mean and stabilized population standard deviation. A tilde denotes these normalized coordinates:

$$\widetilde{x}_{t,c} = \frac{x_{t,c} - \mu_{t,c}}{\nu_{t,c}}, \qquad \widetilde{y}_{t,c} = \frac{y_{t,c} - \mu_{t,c}}{\nu_{t,c}}, \qquad \widetilde{p}_{t,c} = \frac{p_{t,c} - \mu_{t,c}}{\nu_{t,c}}. \tag{1}$$

These normalized representations are used for memory retrieval: each stored episode provides a key $\tilde{x}_{j,c}$ and its corresponding future $\tilde{y}_{j,c}$, while the current query is represented by $\tilde{x}_{t,c}$. Its unknown future $\tilde{y}_{t,c}$ is not used.

3.3 QUERY-CONDITIONED RETRIEVAL

The relevance of stored episodes varies across queries. OREM therefore performs query-conditioned retrieval to select past episodes that best match the current temporal pattern. For either bank B, we score episode j by $a_{tj,c} = \langle \widetilde{x}_{t,c}, \widetilde{x}_{j,c} \rangle / L$, a stabilized Pearson correlation (Han et al., 2025). Let $\mathcal{N}^B_{t,c}$ contain the K highest-scoring episodes in that bank. Their weights are obtained by applying a softmax to the similarity scores, and the retrieved future $r^B_{t,c} \in \mathbb{R}^H$ is formed as their weighted sum:

$$w^B_{tj,c} = \frac{\exp(a_{tj,c}/T)}{\sum_{k\in\mathcal{N}^B_{t,c}} \exp(a_{tk,c}/T)}, \qquad r^B_{t,c} = \sum_{j\in\mathcal{N}^B_{t,c}} w^B_{tj,c} \widetilde{y}_{j,c}, \tag{2}$$

Here, $T > 0$ is the temperature.

To determine how much to rely on each bank, OREM compares how well its retrieved neighbours match the current query. We measure this using the mean neighbour similarity $s^B_{t,c} =$

$\frac{1}{K}\sum_{j\in\mathcal{N}^B_{t,c}} a_{tj,c}$. Once at least K online episodes have matured, OREM converts the difference in bank relevance into a gating weight and uses it to combine the two retrieved futures:

$$\begin{aligned} g_{t,c} &= \mathrm{sigmoid}\big(\kappa(s^{\mathrm{online}}_{t,c} - s^{\mathrm{core}}_{t,c})\big)\,, \\ r_{t,c} &= (1-g_{t,c})r^{\mathrm{core}}_{t,c} + g_{t,c}r^{\mathrm{online}}_{t,c}. \end{aligned} \tag{3}$$

Here, $g_{t,c} \in (0,1)$ determines the relative contribution of the two banks, while κ controls how sharply the gate responds to their similarity difference. The bank with closer matches receives more weight. Before the online bank is ready, OREM uses only the core retrieval, $r_{t,c} = r^{\mathrm{core}}_{t,c}$.

3.4 Correcting the frozen prediction

To ensure stable adaptation, OREM applies retrieved evidence to the frozen forecast with bounded influence. Let $d_{t,c} = r_{t,c} - \tilde{p}_{t,c} \in \mathbb{R}^H$ denote the correction suggested by retrieval relative to the frozen prediction. Applying this correction in full can be unreliable, because even similar past episodes may differ in how accurately their retrieved futures match the target. OREM therefore uses a scalar correction strength $\hat{\beta}_{t,c} \in [0,1]$ to control how much of $d_{t,c}$ is applied.

To estimate $\hat{\beta}_{t,c}$, OREM combines two complementary estimates from matured feedback. The first, $\beta^{\mathrm{ret}}_{t,c}$, is a local estimate obtained from the online neighbours used for retrieval. Each online episode stores a correction coefficient indicating how strongly its retrieval correction should have been applied. This coefficient is computed once the episode's target becomes available. OREM averages these stored coefficients over the retrieved neighbours, allowing the current query to reuse correction strengths that were appropriate for similar past episodes.

The second, $\beta^{\mathrm{bin}}_{t,c}$, provides a broader estimate from past episodes with similar neighbour similarity. OREM groups matured episodes by channel and neighbour-similarity level, and uses the corresponding group to estimate a typical correction strength for the current query. Thus, $\beta^{\mathrm{ret}}_{t,c}$ captures query-specific evidence from the nearest past episodes, while $\beta^{\mathrm{bin}}_{t,c}$ provides a more stable estimate from a broader set of past cases.

OREM combines the two estimates as

$$\hat{\beta}_{t,c} = \mathrm{clip}_{[0,1]}\left((1-\alpha_{t,c})\beta^{\mathrm{bin}}_{t,c} + \alpha_{t,c}\beta^{\mathrm{ret}}_{t,c}\right), \tag{4}$$

where $\alpha_{t,c}$ gives more weight to $\beta^{\mathrm{ret}}_{t,c}$ when the online neighbours closely match the current query. Appendix B provides the exact definitions and update order.

The final prediction is

$$\begin{aligned} \Delta_{t,c} &= \mathrm{clip}_{[-\tau,\tau]}\left(\hat{\beta}_{t,c}d_{t,c}\right), \\ \hat{y}_{t,c} &= p_{t,c} + \nu_{t,c}\Delta_{t,c}. \end{aligned} \tag{5}$$

Here, clipping limits the correction magnitude in normalized coordinates, and $\nu_{t,c}$ restores it to the forecaster's output scale.

4 Experiments

4.1 Setup

We evaluate six long-horizon benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, and Electricity. The input length is $L = 96$, and the forecast horizons are $H \in \{96, 192, 336, 720\}$. We report MSE and MAE. Following the official TSLib (Wang et al., 2026) setup, the ETT datasets use the standard 12/4/4-month train/validation/test split, while Weather and Electricity use a chronological $70/10/20$ split. We evaluate five forecasting backbones: DLinear (Zeng et al., 2023), OLS (Ordinary Least Squares), iTransformer (Liu et al., 2024), PatchTST (Nie et al., 2023), and TQNet (Lin et al., 2025), for $5 \times 6 \times 4 = 120$ settings.

We compare OREM with representative TTA methods for time-series forecasting, TAFAS (Kim et al., 2025), PETSA (Medeiros et al., 2025), and COSA (Im & Kwon, 2026). Baseline configurations and the expanded evaluation scope are described in Appendix B. We use the baseline implementations and evaluation procedures provided in the COSA repository (Im & Kwon, 2026). All methods start from the same frozen backbones. Appendix F provides the setting-level results.

Table 1: **Main results.** MSE/MAE averaged over horizons $\{96, 192, 336, 720\}$. Base denotes the frozen model, and bold marks the best adapted result in each column. OREM is gradient-free and adds no trainable parameters.

		ETTh1		ETTh2		ETTm1		ETTm2		Weather		ECL	
	Method	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
DLinear	Base	0.450	0.439	0.375	0.398	0.414	0.408	0.285	0.327	0.273	0.291	0.226	0.309
	OREM	**0.421**	**0.427**	**0.371**	**0.394**	**0.376**	**0.384**	**0.280**	**0.322**	0.260	**0.281**	**0.189**	**0.278**
	TAFAS	0.447	0.444	0.377	0.400	0.391	0.405	0.289	0.332	0.259	0.292	0.222	0.310
	PETSA	0.445	0.447	0.373	0.397	0.393	0.403	0.284	0.329	**0.258**	0.288	0.224	0.314
	COSA	0.450	0.441	0.382	0.401	0.399	0.404	0.312	0.338	0.267	0.293	0.225	0.309
OLS	Base	0.445	0.432	0.374	0.398	0.413	0.407	0.286	0.327	0.273	0.291	0.215	0.292
	OREM	**0.418**	**0.423**	**0.370**	**0.394**	**0.376**	**0.384**	**0.280**	**0.321**	0.260	**0.281**	**0.187**	**0.273**
	TAFAS	0.442	0.437	0.376	0.400	0.391	0.405	0.291	0.333	0.259	0.293	0.211	0.294
	PETSA	0.438	0.439	0.373	0.397	0.392	0.403	0.287	0.331	**0.259**	0.289	0.213	0.299
	COSA	0.444	0.434	0.381	0.402	0.398	0.403	0.313	0.339	0.267	0.293	0.215	0.292
iTransformer	Base	0.453	0.448	0.401	0.414	0.414	0.414	0.307	0.344	0.257	0.279	0.189	0.277
	OREM	**0.423**	**0.433**	**0.383**	**0.402**	**0.375**	**0.390**	**0.291**	**0.330**	0.250	**0.272**	**0.172**	**0.262**
	TAFAS	0.448	0.447	0.406	0.416	0.389	0.408	0.306	0.345	**0.247**	0.281	0.187	0.278
	PETSA	0.446	0.450	0.398	0.412	0.394	0.412	0.301	0.342	0.248	0.279	0.189	0.282
	COSA	0.450	0.448	0.410	0.418	0.394	0.409	0.331	0.352	0.254	0.281	0.188	0.277
PatchTST	Base	0.453	0.449	0.385	0.408	0.390	0.404	0.295	0.334	0.257	0.279	0.206	0.294
	OREM	**0.424**	**0.434**	**0.377**	**0.401**	**0.371**	**0.388**	**0.286**	**0.327**	0.251	**0.274**	**0.182**	**0.273**
	TAFAS	0.452	0.449	0.384	0.408	0.384	0.403	0.295	0.336	**0.249**	0.284	0.200	0.289
	PETSA	0.455	0.451	0.382	0.406	0.397	0.408	0.296	0.336	0.253	0.283	0.210	0.297
	COSA	0.449	0.450	0.392	0.410	0.387	0.404	0.318	0.343	0.253	0.281	0.205	0.294
TQNet	Base	0.441	0.433	0.378	0.402	0.377	0.393	0.278	0.323	0.243	0.269	0.164	0.258
	OREM	**0.417**	**0.424**	**0.374**	**0.398**	**0.362**	**0.382**	**0.275**	**0.319**	0.241	**0.267**	**0.158**	**0.252**
	TAFAS	0.437	0.437	0.378	0.403	0.371	0.393	0.281	0.326	**0.234**	0.271	0.164	0.260
	PETSA	0.437	0.441	0.375	0.400	0.373	0.396	0.281	0.326	0.236	0.271	0.165	0.262
	COSA	0.441	0.435	0.386	0.406	0.374	0.393	0.304	0.333	0.241	0.271	0.164	0.258

4.2 MAIN RESULTS

Table 1 reports the four-horizon average for each backbone and dataset. OREM reduces MSE in all 120 settings, with mean reductions of 5.43% in MSE and 3.37% in MAE. The MSE reduction ranges from 2.0% on ETTh2 to 11.0% on Electricity. OREM achieves stronger performance than the existing gradient-update-based TTA methods, indicating that it provides a more effective approach to adaptation on the test stream.

4.3 COMPONENT ABLATION

Which components make deployment experience useful? Table 2 evaluates variants of OREM by removing or replacing individual components on ETTm1, ETTm2, and Weather. The "w/o online bank" variant retains only the fixed historical core bank, while "w/o bounded corr." removes correction clipping. The global β variant replaces query-specific coefficient retrieval with a single global estimate, and binned β uses only the confidence-bin estimate. Removing the online bank increases MSE for every backbone–dataset pair, supporting the value of newly observed experience beyond the fixed historical core bank. Removing the correction bound also increases MSE, indicating that stable forecast correction is important. Replacing coefficient retrieval with global or binned estimates increases MSE overall; the smaller differences for the binned alternative indicate a refinement to correction strength. The full OREM configuration achieves the lowest MSE across all evaluated combinations.

Table 2: **Component ablation.** MSE/MAE averaged over four horizons on three representative datasets. Each row removes or replaces one component of the deployed configuration; Base is the frozen model. Bold marks the best result in each backbone column.

		DLinear		OLS		iTransformer		PatchTST		TQNet	
	Config	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTm1	Base	0.414	0.408	0.413	0.407	0.414	0.414	0.390	0.404	0.377	0.393
	OREM	**0.376**	**0.384**	**0.376**	**0.384**	**0.375**	**0.390**	**0.371**	**0.388**	**0.362**	**0.382**
	w/o online bank	0.392	0.392	0.392	0.392	0.392	0.399	0.386	0.396	0.376	0.389
	w/o bounded corr.	0.382	0.386	0.381	0.386	0.379	0.390	0.378	0.390	0.370	0.384
	global β	0.387	0.391	0.387	0.391	0.389	0.399	0.381	0.395	0.371	0.387
	binned β	0.382	0.388	0.382	0.387	0.381	0.393	0.375	0.390	0.366	0.384
ETTm2	Base	0.285	0.327	0.286	0.327	0.307	0.344	0.295	0.334	0.278	0.323
	OREM	**0.280**	**0.322**	**0.280**	**0.321**	**0.291**	**0.330**	**0.286**	**0.327**	**0.275**	**0.319**
	w/o online bank	0.282	0.324	0.282	0.324	0.296	0.334	0.290	0.331	0.277	0.322
	w/o bounded corr.	0.295	0.328	0.295	0.328	0.300	0.333	0.295	0.331	0.287	0.325
	global β	0.282	0.324	0.282	0.324	0.296	0.335	0.290	0.331	0.277	0.321
	binned β	0.282	0.323	0.282	0.323	0.294	0.332	0.289	0.329	0.276	0.320
Weather	Base	0.273	0.291	0.273	0.291	0.257	0.279	0.257	0.279	0.243	0.269
	OREM	**0.260**	**0.281**	**0.260**	**0.281**	**0.250**	**0.272**	**0.251**	**0.274**	**0.241**	0.267
	w/o online bank	0.263	0.283	0.264	0.283	0.252	0.274	0.253	0.275	0.242	0.267
	w/o bounded corr.	0.261	0.281	0.261	0.281	0.257	0.276	0.258	0.277	0.251	0.272
	global β	0.263	0.282	0.264	0.282	0.252	0.273	0.252	0.274	0.241	**0.266**
	binned β	0.262	0.282	0.262	0.282	0.251	0.273	0.252	0.274	0.241	0.266

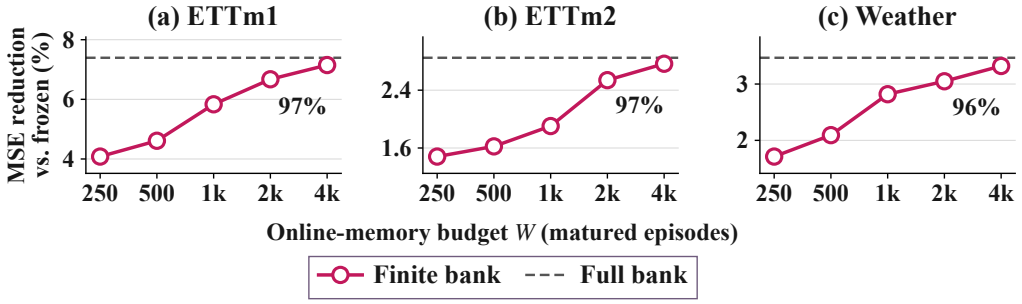


Figure 3: Memory-budget sweep on three representative datasets. Each point averages the MSE reduction over five backbones and four horizons. Dashed lines show the full-bank result; labels at $W = 4{,}000$ report the fraction of that gain retained by the finite bank.

4.4 Memory budget and retrieval efficiency

Does OREM need to retain the entire deployment history, and how does the online-memory budget affect retrieval cost? To examine this, we retain only the most recent W matured online episodes and vary the memory budget. As shown in Figure 3, the performance gain increases with W and gradually saturates; at $W = 4{,}000$, the finite bank retains 96.5% of the full-bank MSE reduction.

Table 3 reports the runtime and memory costs associated with OREM's retrieval procedure. With a bounded online memory of $W = 4{,}000$, the online-bank memory usage decreases from 58 to 21 MiB on ETTm2, from 160 to 62 MiB on Weather, and from 1,198 to 945 MiB on Electricity, while retrieval latency remains comparable to that of the full bank. Together with the 96.5% retained MSE reduction above, these results show that OREM does not need to preserve the entire deployment history: a bounded online memory retains most of the adaptation benefit while limiting the growth of the adaptive state.

Table 3: **Retrieval efficiency under different online-memory budgets.** Values are reported for $W = 1\text{K}, 4\text{K}$, and the full bank.

Dataset (C)	Retrieval Latency (ms/query) ↓			Online-Bank Memory (MiB) ↓			Retrieval Peak GPU (MiB) ↓		
	1K	4K	Full	1K	4K	Full	1K	4K	Full
ETTm2 (7)	1.61	1.81	1.64	5	21	58	130	138	158
Weather (21)	2.16	2.15	2.62	16	62	160	342	365	415
Electricity (321)	9.76	10.90	11.17	236	945	1,198	2,594	2,950	3,077

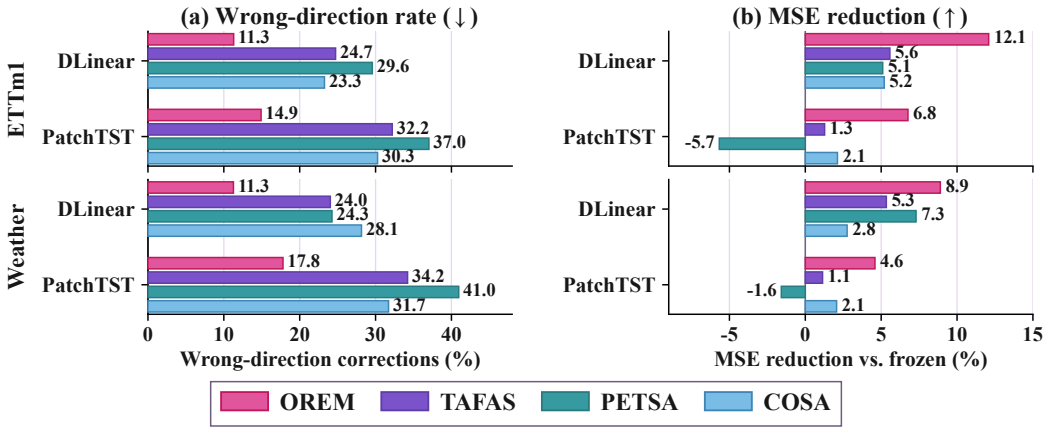


Figure 4: **Direction and outcome of prediction changes on ETTm1 and Weather** ($H = 96$)**.** DLinear and PatchTST represent linear and Transformer-based forecasting models, respectively. Panel (a) reports the fraction of prediction changes that worsen the frozen backbone prediction by moving away from the target, while panel (b) reports the MSE reduction over the full test stream relative to the frozen backbone.

4.5 Direction of Prediction Changes

We examine how OREM, TAFAS, PETSA, and COSA change the frozen prediction on ETTm1 and Weather. DLinear and PatchTST provide representative linear and Transformer backbones. For each test window, we compare the frozen prediction p_t and the adapted prediction $\hat{y}_t$ obtained under each method's evaluation protocol against the realized target y_t.

The frozen-model error $e_t = y_t - p_t$ points from the original prediction toward the target. The output displacement $u_t = \hat{y}_t - p_t$ records the change made by adaptation. We flatten the horizon and channel dimensions and compute the signed alignment $a_t = \langle u_t, e_t \rangle / (\|u_t\|_2 \|e_t\|_2)$ for $\|u_t\|_2 > 0$. A negative value means that the adapted output moved away from the target. Figure 4(a) reports the percentage of evaluated windows with $a_t < 0$. Figure 4(b) reports the percentage reduction in MSE over the complete test stream relative to the frozen prediction.

OREM's wrong-direction rate is 11.3–17.8% across the four backbone–dataset settings. The three baselines range from 23.3% to 41.0%. OREM also gives the largest MSE reduction in each setting. These results suggest that OREM's memory-based approach provides substantially more stable and effective adaptation to evolving test streams than parameter-update-based TTA methods.

4.6 Adaptation under Recurring Domain Shifts

We evaluate whether OREM adapts effectively in an extreme setting with repeated abrupt domain shifts. Using a PatchTST backbone trained on ETTm1, we construct four consecutive 30-day test periods that alternate between ETTm1 and ETTm2. The first period uses ETTm1. We use observations from the corresponding dates in each dataset to preserve chronological order throughout evaluation.

Table 4: **Prediction performance of PatchTST trained on ETTm1 under recurring domain shifts** (MSE / MAE). Each period spans 30 consecutive days, and all values use ETTm1 training-set normalization. Windows whose input or forecast horizon crosses a transition boundary are excluded from the reported scores. Bold indicates the lowest error in each period.

Method	Period 1 ETTm1	Period 2 ETTm2	Period 3 ETTm1	Period 4 ETTm2
Frozen	0.3059 / 0.3583	1.1456 / 0.7159	0.3951 / 0.4025	1.4136 / 0.7775
TAFAS	0.3045 / 0.3568	1.1530 / 0.7203	0.3916 / 0.4011	1.4232 / 0.7814
PETSA	0.3250 / 0.3697	1.1658 / 0.7265	0.4123 / 0.4105	1.4349 / 0.7857
COSA	0.3031 / 0.3558	1.2263 / 0.7444	0.3942 / 0.4019	1.3864 / 0.7802
OREM	**0.2930 / 0.3431**	**1.1332 / 0.7094**	**0.3714 / 0.3843**	**1.3722 / 0.7563**

In Table 4, existing TTA methods incur higher prediction errors than the frozen backbone in some periods. This suggests that parameter-update-based adaptation does not guarantee consistent improvements under recurring domain shifts and can instead increase prediction error in some periods.

OREM improves on the frozen backbone in all four periods and achieves the lowest MSE and MAE among the compared methods. The improvements persist both after a switch away from the training domain and upon returning to a previously encountered domain. These results support the effectiveness of online-memory-based forecast correction in providing consistent period-level improvements under recurring domain shifts.

4.7 Qualitative Examples

Figure 5 shows representative DLinear forecasts at $H = 96$ on ETTm2, Weather, and Electricity. In these examples, OREM corrects the frozen backbone prediction toward the ground truth, improving the forecast in terms of both level and amplitude. These examples provide a qualitative view of how the quantitative improvements reported in Table 1 appear in the forecast trajectories.

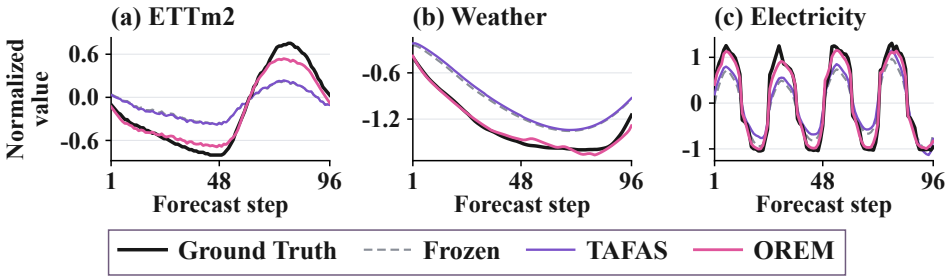


Figure 5: Qualitative DLinear forecasts at $H = 96$ on ETTm2, Weather, and Electricity.

5 Conclusion

This study investigates whether time-series forecasting models can effectively adapt to evolving environments by leveraging experience accumulated over the test stream, without requiring additional parameter updates. OREM selectively retrieves past episodes relevant to the current input and uses them to correct the prediction of a frozen backbone, achieving stronger adaptation performance than existing parameter-update-based TTA methods. These results show that continual parameter optimization is not essential for test-time adaptation in time-series forecasting, and suggest that memory-based test-time adaptation for time-series forecasting—where accumulated experience is retained and selectively reused—can serve as an important direction for future research. Future work may extend this principle to time-series foundation models, as well as to settings with abrupt distribution shifts and substantially larger memory banks.

Ethics and Reproducibility Statement

This work uses public forecasting benchmarks and involves no human subjects or personal data. Appendix B specifies the implementation and hyperparameters. Anonymized code is provided as a separate submission.

AI Use Statement

We used generative AI tools to assist with verifying method implementations and debugging, provide feedback on experimental design, review mathematical claims and proofs, check for potential errors in the interpretation of results, and translate and revise manuscript text. We also used these tools for literature searches. The analyses were conducted by the authors, and the interpretation of experimental results and conclusions were independently determined by the authors. All AI-assisted material was reviewed by the authors, who take full responsibility for the final work.

References

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive Test-Time Adaptation . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–305, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.00039. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00039.

Dazhao Du, Tao Han, and Song Guo. Predicting the future by retrieving the past. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20896–20904, 2026.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=GUDnecJdJU.

Jeonghwan Im and Hyuk-Yoon Kwon. COSA: Context-aware output-space adapter for test-time adaptation in time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=L7Z5wBMPrW.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=e_yvNqkJKAW.

Minguk Jang, Sae-Young Chung, and Hye Won Chung. Test-time adaptation via self-training with nearest neighbor information. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=EzLtB4M1SbM.

Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14162–14171. IEEE, 2024.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklBjCEKvH.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17868–17876, 2025.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=e24CueVty2.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=ucNDIDRNjjv.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5BqDSw8r5j.

Heitor Rapela Medeiros, Hossein Sharifi-Noghabi, Gabriel L. Oliveira, and Saghar Irandoust. Accurate parameter-efficient test-time adaptation for time series forecasting. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025. URL https://openreview.net/forum?id=uFj4EL4GTB.

E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964. doi: 10.1137/1109020. URL https://doi.org/10.1137/1109020.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Y. Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. TS-RAG: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=PymOnHw4Ty.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/niu22a.html.

Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven Hoi. Learning fast and slow for online time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=q-PbpHD3EOk.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Retrieval augmented time series forecasting. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7191–7201. IEEE, 2022.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Geoffrey S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372, 1964. ISSN 0581572X. URL http://www.jstor.org/stable/25049340.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=J4gRj6d5Qm.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Yifan Zhang, Xue Wang, Kexin Jin, Kun Yuan, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Adanpc: Exploring non-parametric classifier for test-time adaptation. In *International conference on machine learning*, pp. 41647–41676. PMLR, 2023a.

YiFan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=Q25wMXsaeZ.

TABLE OF CONTENTS

1 Introduction **1**

2 Related Work **3**

3 Method: Online Episodic Retrieval **4**

3.1 Problem formulation 4

3.2 Accumulating deployment experience 4

3.3 Query-conditioned retrieval 4

3.4 Correcting the frozen prediction 5

4 Experiments **5**

4.1 Setup 5

4.2 Main results 6

4.3 Component ablation 6

4.4 Memory budget and retrieval efficiency 7

4.5 Direction of prediction changes 8

4.6 Adaptation under recurring domain shifts 8

4.7 Qualitative Examples 9

5 Conclusion **9**

A Datasets and Evaluation **14**

B Implementation Details **14**

C Stability and Scope of Memory Adaptation **15**

C.1 Bounded prediction changes 15

C.2 Which past episodes influence a query? 15

D Additional Experimental Analyses **16**

D.1 Effect of memory sources 16

D.2 Hyperparameter sensitivity 16

D.3 Effect of output clipping 16

E Additional Qualitative Examples **18**

F Full Results **19**

A Datasets and Evaluation

Table 5 summarizes the six datasets and their chronological splits. ETT uses 12/4/4 months for training, validation, and testing, with 30 days per month; Weather and Electricity use a 70/10/20 split. Each variable is standardized using its training-set mean and standard deviation. Validation and test inputs may include preceding observations as context; their forecast targets remain within the corresponding split.

Table 5: **Dataset summary.** Split sizes count observations, excluding reused input context.

Dataset	Channels	Interval	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 min	34,560	11,520	11,520
ETTm2	7	15 min	34,560	11,520	11,520
Weather	21	10 min	36,887	5,270	10,539
Electricity	321	1 hour	18,412	2,632	5,260

Retrieval-efficiency profiling. We profile DLinear with $L = H = 96$ on an NVIDIA RTX 5070 Ti Laptop GPU. For each dataset and budget, we evaluate the final causal query after three warm-up retrievals and report the median of 20 synchronized exact-retrieval measurements. Peak GPU memory is the maximum allocated CUDA memory during retrieval. The profile excludes backbone inference and coefficient updates. The corresponding p90 latencies for $W = 1\mathrm{K}/4\mathrm{K}/\mathrm{Full}$ are 2.44/2.64/2.32 ms on ETTm2, 2.37/2.93/2.94 ms on Weather, and 10.02/11.64/11.62 ms on Electricity.

B Implementation Details

All 120 settings use $K = 10$ neighbours per bank and channel, temperature $T = 0.1$, bank-gate sharpness $\kappa = 10$, ridge $\lambda = 10^{-2}$, decay $\rho = 0.997$, four confidence bins, and correction limit $\tau = 0.5$ in normalized units. The standard-deviation stabilizer is $\epsilon = 10^{-6}$.

Baseline configurations. We use the baseline implementations provided in the COSA repository and refer to its experimental protocols (Im & Kwon, 2026). All methods use the same frozen forecasting backbones, data splits, and normalization statistics. All methods are evaluated on the same test windows and target coordinates, with MSE and MAE computed over the complete forecast horizon. COSA uses the official PAAS-enabled implementation with a context size of 10, three adaptation steps, an initial learning rate of 10^{-3}, and weight decay of 10^{-4}. The configurations used for each baseline are provided with the code.

Correction strength. When episode j matures, its normalized error $e_{j,c} = \widetilde{y}_{j,c} - \widetilde{p}_{j,c}$ becomes available. Using the gap $d_{j,c}$ saved when that forecast was issued, we compute

$$\beta^*_{j,c} = \mathrm{clip}_{[0,1]}\left(\frac{\langle d_{j,c}, e_{j,c}\rangle}{\|d_{j,c}\|_2^2 + \lambda}\right). \tag{6}$$

The inner product and squared norm sum over the H forecast steps. This ridge fit estimates how much of the past correction would have matched the observed error before output clipping. The coefficient is stored with the episode.

For the current query, we average these stored coefficients over the same online neighbours and with the same weights used for the retrieved future (Nadaraya, 1964; Watson, 1964). The local estimate and mixing weight in Eq. 4 are

$$\begin{aligned} \beta^{\mathrm{ret}}_{t,c} &= \sum_{j \in \mathcal{N}^{\mathrm{online}}_{t,c}} w^{\mathrm{online}}_{tj,c}\, \beta^*_{j,c}, \\ \alpha_{t,c} &= \mathrm{clip}_{[0,1]}(s^{\mathrm{online}}_{t,c}). \end{aligned} \tag{7}$$

With fewer than K online episodes, we use $\hat{\beta}_{t,c} = \mathrm{clip}_{[0,1]}(\beta^{\mathrm{bin}}_{t,c})$.

Pooled estimate. Past corrections are grouped by channel and retrieval similarity. With both banks available, let $q_{t,c} = \mathrm{clip}_{[0,1]}(\max(s^{\mathrm{core}}_{t,c}, s^{\mathrm{online}}_{t,c}))$; otherwise use $q_{t,c} = \mathrm{clip}_{[0,1]}(s^{\mathrm{core}}_{t,c})$. The bin is $b_{t,c} = \min(\lfloor 4q_{t,c} \rfloor, 3)$. Initialize $A_{c,b} = D_{c,b} = 0$ for each channel and bin. Each matured episode contributes once to the bin saved at its original query:

$$\begin{aligned} A_{c,b} &\leftarrow \rho A_{c,b} + \mathbf{1}\{b = b_{j,c}\}\langle d_{j,c}, e_{j,c}\rangle, \\ D_{c,b} &\leftarrow \rho D_{c,b} + \mathbf{1}\{b = b_{j,c}\}\|d_{j,c}\|_2^2, \qquad b = 0, 1, 2, 3. \end{aligned} \tag{8}$$

All bins decay at each update. A accumulates gap–error alignment; D accumulates squared gap magnitude. After all newly matured episodes are processed, $\beta^{\mathrm{bin}}_{t,c} = A_{c,b_{t,c}}/(D_{c,b_{t,c}} + \lambda)$. This ratio enters Eq. 4 before clipping; it can lie outside $[0, 1]$.

Update order. At each time t, first process newly matured episodes satisfying $j + H \le t$ in chronological order. Use their saved input statistics, frozen predictions, gaps, and bins to update Eq. 8 and compute Eq. 6, then add their normalized keys, futures, and coefficients to the online bank. Next, obtain the frozen prediction, retrieve eligible neighbours, and compute the gap and bin for the current query. Apply Eq. 4 and Eq. 5, and retain the query quantities until its full target is observed. Main-table runs retain all matured online episodes. Budgeted runs retrieve only from the most recent W matured episodes and the unchanged fixed core bank.

C Stability and Scope of Memory Adaptation

C.1 Bounded prediction changes

We bound the change from the frozen prediction in normalized coordinates.

Proposition 1 (Bounded, history-uniform displacement)**.** *Fix an input x_t. Under deployment history $\mathcal{H}$, define $\widetilde{\hat{y}}^{\mathcal{H}}_{t,c} = (\hat{y}^{\mathcal{H}}_{t,c} - \mu_{t,c})/\nu_{t,c}$. For any two histories $\mathcal{H}$ and $\mathcal{H}'$,*

$$\|\widetilde{\hat{y}}^{\mathcal{H}}_t - \widetilde{p}_t\|_\infty \le \tau, \qquad \|\widetilde{\hat{y}}^{\mathcal{H}}_t - \widetilde{\hat{y}}^{\mathcal{H}'}_t\|_\infty \le 2\tau. \tag{9}$$

Here $\|\cdot\|_\infty$ is the largest absolute coordinate. The RMS displacement from $\widetilde{p}_t$, over all horizon–channel coordinates, is also at most τ.

Proof. Equation 5 gives $\widetilde{\hat{y}}^{\mathcal{H}}_t = \widetilde{p}_t + \Delta^{\mathcal{H}}_t$, with every coordinate of $\Delta^{\mathcal{H}}_t$ in $[-\tau, \tau]$. Both its infinity norm and RMS are at most τ. The same input fixes the backbone prediction and normalization across histories, so the triangle inequality gives 2τ. □

Thus, storing more experience leaves the maximum normalized correction unchanged. In the forecaster's output units, channel c satisfies $\|\hat{y}_{t,c} - p_{t,c}\|_\infty \le \nu_{t,c}\tau$. Its scale follows the current input through $\nu_{t,c} = \mathrm{std}(x_{t,c}) + \epsilon$.

Comparison with accumulated parameter updates. For a fixed input, suppose $f_\theta(x_t)$ is L_θ-Lipschitz from the Euclidean parameter norm to the output infinity norm. After n updates,

$$\|f_{\theta_n}(x_t) - f_{\theta_0}(x_t)\|_\infty \le L_\theta\|\theta_n - \theta_0\|_2 \le L_\theta \sum_{s=1}^{n} \|\theta_s - \theta_{s-1}\|_2. \tag{10}$$

The right-hand side depends on the total update magnitude and can increase over deployment. OREM's normalized bound in Eq. 9 stays fixed at τ.

C.2 Which past episodes influence a query?

At a fixed query and memory state, the retrieved futures and neighbour-based coefficient use the selected neighbours (Eqs. 2 and 7); their mean similarities determine the bank gate. The query thus selects the experience used directly for retrieval. Other matured episodes can affect the pooled coefficient $\beta^{\mathrm{bin}}_{t,c}$. Earlier memory states also shape later gaps and their stored coefficients. These shared statistics and stored coefficients carry indirect historical influence.

D Additional Experimental Analyses

We evaluate memory sources and hyperparameter sensitivity on ETTm1 and Weather using DLinear, PatchTST, and TQNet. Each setting uses the same backbone as the main experiments.

D.1 Effect of memory sources

We examine how the source of stored experience affects adaptation. Full OREM combines a training-and-validation core with online memory. We compare it with an online-only variant that removes the fixed core and a training-core variant that excludes validation episodes while retaining online memory. All variants use the same correction settings. Table 6 shows that online-only adaptation improves both MSE and MAE over the frozen backbone in all six settings. Excluding validation episodes also yields results close to full OREM. These results show that deployment experience alone provides useful information for adaptation.

Table 6: **Effect of memory sources.** Results at $L = H = 96$. Online only removes the fixed core; train core + online excludes validation episodes. Bold marks the lowest error in each dataset–backbone setting, using unrounded values.

Configuration	DLinear		PatchTST		TQNet	
	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1						
Frozen	0.3533	0.3744	0.3270	0.3654	0.3111	0.3532
OREM	**0.3106**	**0.3435**	**0.3048**	**0.3453**	**0.2974**	**0.3404**
Online only	0.3197	0.3492	0.3081	0.3477	0.3016	0.3429
Train core + online	0.3118	0.3444	0.3065	0.3464	0.2987	0.3412
Weather						
Frozen	0.1952	0.2345	0.1742	0.2152	0.1576	0.2011
OREM	**0.1778**	0.2194	**0.1662**	**0.2078**	**0.1545**	0.1978
Online only	0.1799	0.2204	0.1673	0.2078	0.1549	**0.1970**
Train core + online	0.1780	**0.2193**	0.1665	0.2078	0.1547	0.1978

D.2 Hyperparameter sensitivity

We vary the number of neighbours, retrieval temperature, statistical decay, and clipping strength one at a time, keeping the remaining settings fixed. Tables 7 and 8 report the results. Across the tested neighbour counts, temperatures, and decay rates, all three backbones retain MSE improvements over the frozen predictor. Temperature and decay have relatively small effects. The effect of clipping varies across datasets and backbones; a tighter correction limit does not consistently yield lower forecast error. The default configuration is unchanged after this analysis.

D.3 Effect of output clipping

We compare forecast errors with and without output clipping for TAFAS and PETSA on DLinear, PatchTST, and TQNet. The clipped variants retain their adaptation procedures and limit the final displacement from the frozen prediction using OREM's correction threshold, $\tau = 0.5$. Table 9 reports MSE and MAE. Clipping has different effects across methods, datasets, and backbones.

Table 7: **Hyperparameter sensitivity on ETTm1.** Default: $K = 10$, $T = 0.1$, $\rho = 0.997$, and $\tau = 0.5$. Each row changes one setting. Bold marks the lowest error per backbone, using unrounded values.

Configuration	DLinear		PatchTST		TQNet	
	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1						
Default	0.3106	0.3435	0.3048	0.3453	0.2974	0.3404
$K = 5$	0.3151	0.3462	0.3090	0.3478	0.3005	0.3424
$K = 20$	0.3090	**0.3420**	**0.3039**	**0.3442**	**0.2964**	**0.3394**
$T = 0.05$	0.3107	0.3436	0.3049	0.3453	0.2974	0.3404
$T = 0.2$	0.3106	0.3435	0.3048	0.3453	0.2974	0.3404
$\rho = 0.99$	0.3108	0.3435	0.3051	0.3454	0.2977	0.3405
$\rho = 1$	0.3109	0.3437	0.3049	0.3454	0.2974	0.3405
$\tau = 0.25$	0.3192	0.3479	0.3081	0.3480	0.2991	0.3419
$\tau = 1$	**0.3052**	0.3427	0.3041	0.3449	0.2972	0.3400
No clipping	0.3170	0.3445	0.3126	0.3467	0.3065	0.3417

Table 8: **Hyperparameter sensitivity on Weather.** Default: $K = 10$, $T = 0.1$, $\rho = 0.997$, and $\tau = 0.5$. Each row changes one setting. Bold marks the lowest error per backbone, using unrounded values.

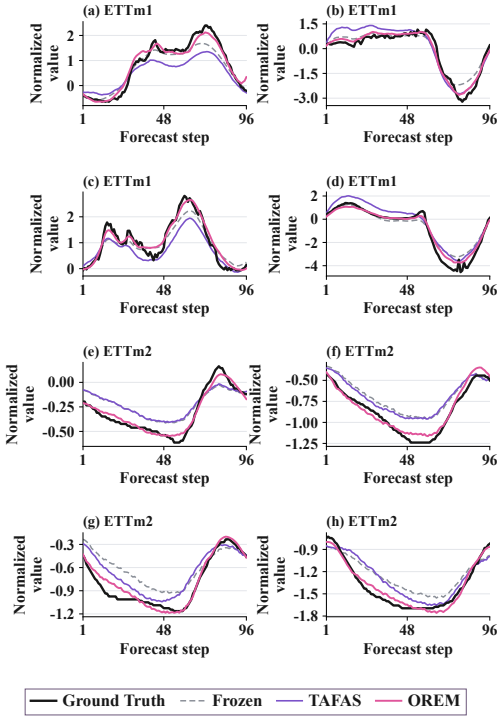
Configuration	DLinear		PatchTST		TQNet	
	MSE	MAE	MSE	MAE	MSE	MAE
Weather						
Default	0.1778	0.2194	0.1662	0.2078	0.1545	0.1978
$K = 5$	0.1789	0.2203	0.1671	0.2084	0.1550	0.1981
$K = 20$	0.1770	0.2188	0.1657	0.2075	**0.1542**	0.1978
$T = 0.05$	0.1780	0.2194	0.1663	0.2077	0.1546	**0.1978**
$T = 0.2$	0.1778	0.2194	0.1662	0.2078	0.1545	0.1979
$\rho = 0.99$	0.1779	0.2194	0.1664	0.2078	0.1547	0.1980
$\rho = 1$	0.1777	0.2192	0.1663	0.2077	0.1545	0.1978
$\tau = 0.25$	0.1839	0.2243	0.1683	0.2095	0.1548	0.1980
$\tau = 1$	0.1722	0.2146	**0.1649**	0.2066	0.1552	0.1984
No clipping	**0.1695**	**0.2104**	0.1660	**0.2057**	0.1582	0.1997

Table 9: **Effect of output clipping.** Unclipped scores follow the main evaluation; clipped scores are from additional runs with the same settings. Bold marks the lowest error per dataset and backbone, using unrounded values.

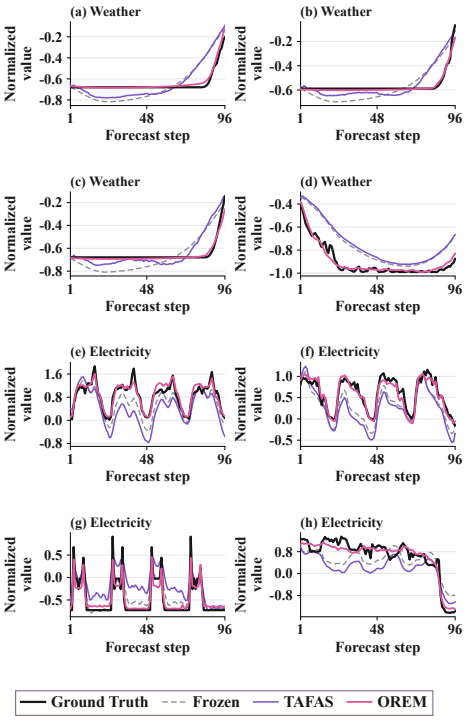
Method	DLinear		PatchTST		TQNet	
	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1						
Frozen	0.3533	0.3744	0.3270	0.3654	0.3111	0.3532
OREM	**0.3106**	**0.3435**	**0.3048**	**0.3453**	**0.2974**	**0.3404**
TAFAS	0.3336	0.3689	0.3228	0.3644	0.3049	0.3503
TAFAS + clip	0.3357	0.3675	0.3221	0.3627	0.3054	0.3504
PETSA	0.3353	0.3682	0.3455	0.3748	0.3087	0.3532
PETSA + clip	0.3371	0.3677	0.3330	0.3699	0.3088	0.3533
Weather						
Frozen	0.1952	0.2345	0.1742	0.2152	0.1576	0.2011
OREM	**0.1778**	**0.2194**	**0.1662**	**0.2078**	0.1545	**0.1978**
TAFAS	0.1848	0.2324	0.1723	0.2174	**0.1518**	0.2002
TAFAS + clip	0.1868	0.2315	0.1727	0.2167	0.1529	0.1994
PETSA	0.1810	0.2281	0.1770	0.2208	0.1537	0.2017
PETSA + clip	0.1855	0.2292	0.1748	0.2177	0.1542	0.2002

E ADDITIONAL QUALITATIVE EXAMPLES

Figure 6 provides additional qualitative examples.



(a) ETTm1 and ETTm2



(b) Weather and Electricity

Figure 6: (a) Additional qualitative examples on ETTm1 and ETTm2 (DLinear, $H = 96$). (b) Additional qualitative examples on Weather and Electricity (DLinear, $H = 96$).

F FULL RESULTS

Tables 10–14 report MSE and MAE for all 120 backbone–dataset–horizon settings. Averaging the four horizons using unrounded values reproduces Table 1. The accompanying artifact contains these results, ablation and memory-budget measurements, the temporal diagnostic, and verification scripts.

Table 10: **Full results: DLinear.** MSE and MAE for each dataset and forecast horizon H. Base is the frozen predictor. Bold marks the best adapted result for each metric, using unrounded values. These are the individual results averaged in Table 1.

Dataset	H	Base		OREM		TAFAS		PETSA		COSA	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.392	0.402	**0.369**	**0.392**	0.389	0.404	0.388	0.404	0.390	0.403
	192	0.443	0.431	**0.415**	**0.420**	0.436	0.434	0.435	0.438	0.441	0.432
	336	0.485	0.452	**0.450**	**0.439**	0.471	0.454	0.475	0.462	0.482	0.454
	720	0.481	0.472	**0.453**	**0.459**	0.493	0.484	0.483	0.484	0.486	0.476
ETTh2	96	0.290	0.338	**0.288**	**0.336**	0.290	0.339	0.290	0.339	0.295	0.342
	192	0.376	0.390	**0.370**	**0.388**	0.375	0.391	0.375	0.391	0.382	0.394
	336	0.414	0.425	**0.408**	**0.420**	0.413	0.424	0.414	0.423	0.410	0.423
	720	0.418	0.438	0.416	**0.434**	0.429	0.445	**0.413**	0.436	0.441	0.447
ETTm1	96	0.353	0.374	**0.311**	**0.344**	0.334	0.369	0.335	0.368	0.331	0.365
	192	0.391	0.393	**0.357**	**0.370**	0.371	0.389	0.373	0.387	0.372	0.386
	336	0.424	0.414	**0.388**	**0.393**	0.403	0.412	0.404	0.410	0.410	0.410
	720	0.487	0.449	**0.450**	**0.430**	0.456	0.449	0.459	0.449	0.482	0.454
ETTm2	96	0.182	0.265	**0.176**	**0.256**	0.181	0.266	0.182	0.266	0.185	0.267
	192	0.246	0.304	**0.241**	**0.299**	0.245	0.305	0.246	0.306	0.258	0.309
	336	0.307	0.342	**0.301**	**0.337**	0.309	0.346	0.308	0.345	0.335	0.352
	720	0.407	0.398	0.402	**0.395**	0.423	0.410	**0.401**	0.398	0.469	0.425
Weather	96	0.195	0.234	**0.178**	**0.219**	0.185	0.232	0.181	0.228	0.187	0.234
	192	0.240	0.270	0.225	**0.260**	**0.224**	0.268	0.225	0.266	0.232	0.271
	336	0.292	0.306	0.280	**0.299**	**0.271**	0.303	0.274	0.301	0.283	0.307
	720	0.365	0.353	0.356	**0.348**	0.356	0.365	**0.352**	0.357	0.366	0.360
ECL	96	0.208	0.291	**0.163**	**0.252**	0.201	0.290	0.203	0.291	0.207	0.291
	192	0.208	0.294	**0.172**	**0.262**	0.202	0.294	0.204	0.296	0.208	0.295
	336	0.223	0.309	**0.189**	**0.280**	0.218	0.310	0.220	0.314	0.222	0.309
	720	0.264	0.340	**0.234**	**0.318**	0.266	0.346	0.270	0.355	0.264	0.340

Table 11: **Full results: OLS.** MSE and MAE for each dataset and forecast horizon H. Base is the frozen predictor. Bold marks the best adapted result for each metric, using unrounded values. These are the individual results averaged in Table 1.

Dataset	H	Base		OREM		TAFAS		PETSA		COSA	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.384	0.392	**0.364**	**0.386**	0.380	0.395	0.380	0.394	0.382	0.392
	192	0.436	0.422	**0.411**	**0.415**	0.428	0.426	0.428	0.429	0.434	0.423
	336	0.479	0.444	**0.447**	**0.435**	0.466	0.448	0.467	0.454	0.475	0.447
	720	0.480	0.469	**0.451**	**0.457**	0.493	0.481	0.477	0.478	0.485	0.473
ETTh2	96	0.289	0.337	**0.286**	**0.334**	0.289	0.338	0.290	0.338	0.294	0.341
	192	0.374	0.390	**0.368**	**0.387**	0.373	0.391	0.376	0.391	0.381	0.394
	336	0.414	0.425	**0.407**	**0.419**	0.415	0.425	0.414	0.423	0.410	0.423
	720	0.420	0.439	0.417	**0.435**	0.426	0.445	**0.412**	0.436	0.439	0.449
ETTm1	96	0.353	0.374	**0.310**	**0.343**	0.335	0.370	0.336	0.368	0.331	0.365
	192	0.391	0.392	**0.356**	**0.369**	0.371	0.389	0.373	0.386	0.371	0.386
	336	0.423	0.414	**0.387**	**0.393**	0.402	0.412	0.403	0.409	0.409	0.410
	720	0.486	0.449	**0.450**	**0.430**	0.455	0.448	0.456	0.447	0.481	0.453
ETTm2	96	0.182	0.265	**0.175**	**0.255**	0.182	0.266	0.182	0.266	0.186	0.267
	192	0.246	0.305	**0.241**	**0.299**	0.246	0.306	0.247	0.307	0.258	0.309
	336	0.307	0.343	**0.301**	**0.337**	0.309	0.346	0.306	0.344	0.333	0.352
	720	0.407	0.398	**0.402**	**0.395**	0.428	0.414	0.414	0.405	0.473	0.426
Weather	96	0.195	0.234	**0.178**	**0.219**	0.185	0.233	0.180	0.228	0.188	0.234
	192	0.240	0.270	0.225	**0.260**	**0.224**	0.268	0.227	0.267	0.232	0.271
	336	0.292	0.306	0.280	**0.299**	**0.271**	0.304	0.275	0.303	0.284	0.307
	720	0.364	0.353	0.356	**0.348**	0.356	0.366	**0.353**	0.358	0.365	0.360
ECL	96	0.197	0.274	**0.161**	**0.247**	0.191	0.273	0.192	0.275	0.197	0.274
	192	0.198	0.277	**0.169**	**0.257**	0.192	0.277	0.193	0.280	0.197	0.277
	336	0.212	0.292	**0.186**	**0.275**	0.207	0.294	0.209	0.299	0.212	0.292
	720	0.254	0.325	**0.230**	**0.311**	0.254	0.333	0.259	0.343	0.254	0.325

Table 12: **Full results: iTransformer.** MSE and MAE for each dataset and forecast horizon H. Base is the frozen predictor. Bold marks the best adapted result for each metric, using unrounded values. These are the individual results averaged in Table 1.

Dataset	H	Base		OREM		TAFAS		PETSA		COSA	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.392	0.411	**0.369**	**0.396**	0.389	0.410	0.391	0.413	0.388	0.409
	192	0.447	0.441	**0.416**	**0.425**	0.440	0.437	0.439	0.442	0.444	0.440
	336	0.483	0.459	**0.451**	**0.444**	0.470	0.451	0.470	0.459	0.477	0.457
	720	0.490	0.482	**0.458**	**0.467**	0.494	0.489	0.484	0.487	0.491	0.484
ETTh2	96	0.338	0.365	**0.314**	**0.349**	0.338	0.365	0.336	0.364	0.342	0.367
	192	0.404	0.409	**0.378**	**0.393**	0.405	0.410	0.404	0.409	0.413	0.414
	336	0.437	0.439	**0.421**	**0.428**	0.433	0.437	0.435	0.437	0.431	0.436
	720	0.425	0.441	0.420	**0.437**	0.446	0.455	**0.419**	0.439	0.453	0.457
ETTm1	96	0.343	0.374	**0.308**	**0.348**	0.321	0.365	0.327	0.370	0.315	0.363
	192	0.385	0.397	**0.351**	**0.373**	0.367	0.390	0.369	0.392	0.364	0.389
	336	0.429	0.422	**0.386**	**0.398**	0.406	0.418	0.407	0.420	0.408	0.417
	720	0.500	0.464	**0.454**	**0.441**	0.463	0.460	0.473	0.467	0.489	0.465
ETTm2	96	0.189	0.269	**0.176**	**0.256**	0.182	0.266	0.183	0.267	0.187	0.268
	192	0.260	0.315	**0.248**	**0.304**	0.261	0.316	0.258	0.314	0.266	0.317
	336	0.341	0.367	**0.321**	**0.351**	0.335	0.367	0.333	0.366	0.360	0.373
	720	0.439	0.423	**0.420**	**0.409**	0.448	0.429	0.431	0.421	0.511	0.449
Weather	96	0.171	0.211	**0.162**	**0.203**	0.165	0.212	0.168	0.214	0.166	0.211
	192	0.222	0.257	0.213	**0.249**	**0.211**	0.257	0.213	0.256	0.216	0.257
	336	0.278	0.298	0.271	**0.292**	**0.259**	0.295	0.264	0.295	0.270	0.298
	720	0.358	0.349	0.353	**0.345**	0.352	0.360	**0.345**	0.351	0.361	0.356
ECL	96	0.161	0.252	**0.144**	**0.236**	0.157	0.251	0.161	0.255	0.161	0.252
	192	0.174	0.263	**0.159**	**0.250**	0.171	0.263	0.173	0.266	0.174	0.263
	336	0.189	0.279	**0.174**	**0.266**	0.187	0.281	0.189	0.284	0.189	0.279
	720	0.230	0.314	**0.212**	**0.298**	0.231	0.318	0.232	0.322	0.230	0.314

Table 13: **Full results: PatchTST.** MSE and MAE for each dataset and forecast horizon H. Base is the frozen predictor. Bold marks the best adapted result for each metric, using unrounded values. These are the individual results averaged in Table 1.

Dataset	H	Base		OREM		TAFAS		PETSA		COSA	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.386	0.405	**0.370**	**0.395**	0.385	0.404	0.393	0.409	0.387	0.405
	192	0.442	0.439	**0.415**	**0.423**	0.441	0.438	0.444	0.439	0.443	0.439
	336	0.499	0.469	**0.448**	**0.444**	0.494	0.469	0.495	0.466	0.475	0.465
	720	0.487	0.485	**0.464**	**0.473**	0.486	0.487	0.489	0.489	0.493	0.492
ETTh2	96	0.315	0.356	**0.300**	**0.346**	0.308	0.354	0.313	0.356	0.320	0.358
	192	0.371	0.399	0.365	**0.391**	**0.363**	0.395	0.365	0.394	0.380	0.400
	336	0.422	0.428	**0.415**	**0.422**	0.428	0.431	0.425	0.430	0.422	0.429
	720	0.433	0.449	0.427	**0.443**	0.435	0.451	**0.425**	0.444	0.447	0.454
ETTm1	96	0.327	0.365	**0.305**	**0.345**	0.323	0.364	0.346	0.375	0.319	0.363
	192	0.365	0.386	**0.349**	**0.372**	0.360	0.386	0.373	0.391	0.358	0.385
	336	0.410	0.417	**0.388**	**0.400**	0.401	0.415	0.410	0.418	0.403	0.415
	720	0.459	0.447	**0.442**	**0.435**	0.454	0.445	0.461	0.447	0.469	0.454
ETTm2	96	0.184	0.263	**0.178**	**0.257**	0.184	0.264	0.187	0.267	0.186	0.264
	192	0.255	0.312	**0.247**	**0.304**	0.255	0.313	0.256	0.314	0.260	0.313
	336	0.320	0.351	**0.311**	**0.344**	0.321	0.353	0.322	0.353	0.335	0.357
	720	0.421	0.411	**0.410**	**0.404**	0.419	0.415	0.420	0.413	0.491	0.438
Weather	96	0.174	0.215	**0.166**	**0.208**	0.172	0.217	0.177	0.221	0.170	0.215
	192	0.221	0.256	**0.215**	**0.251**	0.216	0.259	0.219	0.259	0.215	0.257
	336	0.274	0.296	0.271	**0.292**	**0.264**	0.300	0.268	0.298	0.269	0.296
	720	0.357	0.348	0.353	**0.345**	**0.344**	0.360	0.349	0.355	0.359	0.355
ECL	96	0.178	0.267	**0.153**	**0.244**	0.176	0.264	0.194	0.279	0.177	0.266
	192	0.189	0.280	**0.166**	**0.258**	0.184	0.274	0.195	0.282	0.189	0.279
	336	0.208	0.300	**0.184**	**0.277**	0.200	0.291	0.207	0.297	0.208	0.299
	720	0.248	0.329	**0.226**	**0.312**	0.240	0.326	0.246	0.331	0.248	0.329

Table 14: **Full results: TQNet.** MSE and MAE for each dataset and forecast horizon H. Base is the frozen predictor. Bold marks the best adapted result for each metric, using unrounded values. These are the individual results averaged in Table 1.

Dataset	H	Base		OREM		TAFAS		PETSA		COSA	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.371	0.393	**0.361**	**0.388**	0.371	0.393	0.371	0.394	0.372	0.393
	192	0.428	0.423	**0.408**	**0.417**	0.422	0.426	0.422	0.430	0.429	0.425
	336	0.476	0.446	**0.444**	**0.436**	0.461	0.447	0.468	0.458	0.471	0.448
	720	0.488	0.470	**0.453**	**0.457**	0.494	0.480	0.489	0.482	0.490	0.473
ETTh2	96	0.294	0.343	**0.290**	**0.338**	0.295	0.344	0.296	0.344	0.302	0.347
	192	0.368	0.393	**0.365**	**0.389**	0.369	0.393	0.368	0.392	0.374	0.395
	336	0.418	0.428	**0.412**	**0.422**	0.413	0.426	0.417	0.426	0.415	0.427
	720	0.434	0.446	0.429	0.442	0.435	0.448	**0.419**	**0.440**	0.454	0.457
ETTm1	96	0.311	0.353	**0.297**	**0.340**	0.305	0.350	0.309	0.353	0.302	0.350
	192	0.357	0.379	**0.344**	**0.367**	0.351	0.379	0.353	0.379	0.350	0.378
	336	0.389	0.401	**0.375**	**0.391**	0.383	0.401	0.384	0.403	0.386	0.400
	720	0.451	0.439	**0.434**	**0.430**	0.445	0.441	0.445	0.447	0.456	0.444
ETTm2	96	0.173	0.256	**0.170**	**0.252**	0.173	0.257	0.174	0.258	0.176	0.257
	192	0.243	0.300	**0.238**	**0.296**	0.241	0.301	0.245	0.303	0.249	0.302
	336	0.301	0.340	**0.296**	**0.335**	0.298	0.341	0.300	0.342	0.326	0.348
	720	0.396	0.396	**0.394**	**0.393**	0.412	0.407	0.404	0.402	0.465	0.425
Weather	96	0.158	0.201	0.155	**0.198**	**0.152**	0.200	0.154	0.202	0.154	0.201
	192	0.206	0.244	0.204	**0.242**	**0.195**	0.243	0.199	0.244	0.201	0.245
	336	0.263	0.288	0.263	0.287	**0.249**	**0.286**	0.253	0.288	0.261	0.290
	720	0.344	0.342	0.343	**0.340**	0.341	0.354	**0.340**	0.352	0.349	0.350
ECL	96	0.135	0.230	**0.130**	**0.224**	0.133	0.229	0.136	0.232	0.135	0.230
	192	0.152	0.246	**0.148**	**0.241**	0.151	0.246	0.153	0.248	0.152	0.246
	336	0.169	0.265	**0.163**	**0.258**	0.168	0.266	0.169	0.268	0.169	0.265
	720	0.198	0.292	**0.193**	**0.286**	0.203	0.297	0.204	0.300	0.198	0.292