Learning the Context of Errors: Black-Box Online Adaptation of Time Series Foundation Models

Xilin Dai[1,2], **Yiding Liu**[1], **Hongjie Xia**[1], **Yifan Hu**[1],
Zewei Dong[1,*], **Jiang-Ming Yang**[1], **Qiang Xu**[2]
[1] Ant International [2] The Chinese University of Hong Kong
{xldai26, qxu}@cse.cuhk.edu.hk
zewei.dong@ant-intl.com

Abstract

The rapid evolution of Time Series Foundation Models (TSFMs) has advanced zero-shot forecasting across diverse domains. Inspired by the current form of Large Language Models, future TSFMs may be offered as commercialized, closed-source application programming interface (API) services. However, many existing online adaptation methods still rely on white-box access for parameter fine-tuning or grey-box access for gradient backpropagation. This paradigm mismatch raises a question: *In black-box online adaptation for TSFMs, what should we learn?* We answer this with an insight: the predictive errors of the base model are conditioned on both the input and output of the base model (i.e., the context of errors). To validate this insight, we propose **ORCA** (Online Residual Contextual Adaptation). We conduct extensive experiments across 5 state-of-the-art TSFMs and 8 datasets to demonstrate the effectiveness of our approach. Furthermore, through ablation studies, we quantitatively analyze the impact of different adapter learning hypotheses on the final adaptation performance in black-box online adaptation. Code available at https://github.com/Fifthky/ORCA.

1 Introduction

Time series forecasting is a cornerstone task across diverse domains, including energy management, traffic planning, and meteorology [Miller et al., 2024, Liang et al., 2024, Dai et al., 2026b, Huang et al., 2026a]. Traditionally, time series forecasting has transitioned from classical statistical methods [Gardner Jr., 1985, Piccolo, 1990] to deep learning architectures [Zhou et al., 2021, Wu et al., 2022, Zeng et al., 2023]. Recently, inspired by the success of Large Language Models (LLMs), the forecasting paradigm is undergoing a shift towards Time Series Foundation Models (TSFMs) [Liang et al., 2024, Liu et al., 2026b]. Pre-trained on vast corpora of time series data spanning multiple domains, TSFMs demonstrate remarkable zero-shot capabilities when performing forecasting on unseen datasets[Aksu et al., 2024, Xu et al., 2026c, Chen et al., 2026], including models such as the Chronos series [Ansari et al., 2024, 2025] and the Moirai series [Woo et al., 2024, Liu et al., 2026a, 2025a].

For TSFMs, online adaptation is particularly crucial: streaming data naturally suffers from temporal concept drift, and there remains a knowledge gap between the general pre-trained TSFMs and the specific dynamics of application scenarios [Zhang et al., 2024, Benechehab et al., 2025, Dai et al., 2026a, Lee et al., 2026]. Depending on the required access level to the base model, existing online adaptation can be categorized into three paradigms (as illustrated in Figure 1): (1) *White-Box (Parameter Finetuning)*, exemplified by SOLID [Chen et al., 2024] and DSOF [Lau et al., 2024], requires a local computation graph to explicitly modify internal weights or specific layers of the backbone network; (2) *Grey-Box*, which includes TAFAS [Kim et al., 2025], δ-Adapter [Liang et al., 2025], and ADAPT-Z [Huang et al., 2025], demands gradient backpropagation through the frozen

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

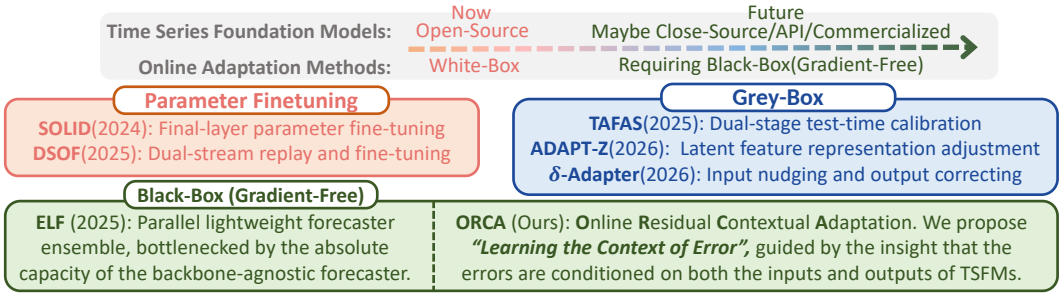

Figure 1: **Categorization of Time Series Online Adaptation Methods.** Depending on the access level to the base model, existing approaches are grouped into three paradigms. In the coming era of commercialized TSFM APIs, only the Black-Box paradigm could provide a feasible solution.

foundation model to update input nudging or latent representations; (3) *Black-Box*, such as ELF [Lee et al., 2025] and the Ada-Y variant of δ-Adapter Liang et al. [2025], operates exclusively on the external interface.

Inspired by the success and commercial value of LLMs, future TSFMs may increasingly be offered as commercialized, closed-source API services [Xu et al., 2026b]. Under this paradigm, users will only have black-box inference access. The strict API constraints render both White-Box and Grey-Box methods unfeasible. Consequently, the Black-Box paradigm emerges as the only viable solution. However, current explorations in this area are limited. ELF [Lee et al., 2025] performs a parallel lightweight forecaster ensemble. However, its performance is bottlenecked by the absolute capacity of this backbone-agnostic forecaster, and its contribution becomes marginal when the base model consistently dominates. Meanwhile, Ada-Y, an output-side variant of the δ-Adapter [Liang et al., 2025], solely relies on the base model's output, only capturing *what* errors the model makes, without the input context necessary to understand *when* those errors occur.

This brings us to a question: ***In black-box online adaptation for TSFMs, what should we learn?*** We argue that we should learn both what the errors are and, more crucially, when the base model makes these errors (i.e., **learning the context of errors**). Errors are not isolated noise; rather, they are conditioned on the inputs and outputs of the base model. Specifically, we should model the conditional distribution of the errors given both the input sequences and the base model's predictions.

To actualize this insight, we propose **ORCA** (Online Residual Contextual Adaptation), a plug-and-play black-box adaptation framework for streaming TSFM API inference of time series. Given the input and the output of the base model, ORCA learns the context-aware error representation. To mitigate overfitting on noisy residuals, ORCA utilizes a *Linear Adapter* with strong structural bias as its core component. To facilitate continuous learning, we design a buffer training mechanism with historical forgetting decay, alongside a predictive-space Bayesian loss. Furthermore, to utilize historical errors, we introduce a *Boltzmann Router*. The router treats historical errors from both the base and the adapted predictions as Boltzmann energy states, dynamically deriving a confidence value to fuse them into a final combined output. The contributions of our work are summarized as follows:

- **Black-Box Online Adaptation for TSFMs:** We conduct black-box online adaptation analyses utilizing the latest generation of TSFMs, establishing a timely foundation for future research in commercialized API forecasting.

- **The ORCA Framework:** We introduce **ORCA**, a black-box online adaptation framework for TSFMs. ORCA utilizes a Linear Adapter with structural bias to provide context-aware residual corrections, integrating a buffer training mechanism with historical forgetting decay, a predictive-space Bayesian loss, and a dynamic Boltzmann Router.

- **Contextual Error Modeling and "What to Learn" :** We propose that adapters should learn the context of errors, both *what* the errors are and *when* the base model makes them. Through ablation studies on the adapter's input configurations, we quantitatively analyze the impact of different learning hypotheses.

2 Related works

2.1 Deep learning for time series forecasting

Historically, the evolution of time series forecasting has transitioned from classical statistical methods [Gardner Jr., 1985, Piccolo, 1990] to deep learning architectures, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) [Connor et al., 1994, Hochreiter and Schmidhuber, 1997, Lai et al., 2018, Dai et al., 2025b], and subsequently, advanced structures like Transformers [Zhou et al., 2021, Wu et al., 2022, Liu et al., 2022, Nie et al., 2022, Liu et al., 2023, Wang et al., 2024] and state-space models [Ahamed and Cheng, 2024, Ma et al., 2024]. Researchers have also questioned the necessity of complex attention mechanisms, showing that simple linear models [Zeng et al., 2023, Xu et al., 2023, Dai et al., 2025a] can achieve competitive results.

2.2 Time series foundation models (TSFMs)

Driven by the success of pre-training in natural language and vision, the community has begun to focus on Time Series Foundation Models (TSFMs) [Liang et al., 2024, Meyer et al., 2025, Miller et al., 2024], which aim to provide universal, zero-shot forecasting capabilities. Popular paradigms involve reprogramming existing LLMs for time series or pre-training transformers from scratch with cross-domain time series data, such as Time-LLM [Jin et al., 2023], Chronos family [Ansari et al., 2024, 2025], Lag-Llama [Rasul et al., 2024], and other architectures [Das et al., 2024, Zhou et al., 2023, Chen et al., 2025]. Concurrent developments have also introduced TimeGPT-1 Garza et al. [2024], the Moirai series (Moirai 1.0, 2.0, and Moirai-MoE) Woo et al. [2024], Liu et al. [2026a, 2025a], the Timer family (Timer, Timer-S1) Liu et al. [2024, 2026c], Sundial Liu et al. [2025b], TiRex Auer et al. [2025], and lightweight models like Tiny Time Mixers (TTM) Ekambaram et al. [2024]. The static nature of these models leaves them vulnerable to temporal concept drifts and the knowledge gap between pre-training data and real-world applications.

2.3 Online learning in time series

Real-world time series data is intrinsically non-stationary, frequently experiencing concept drift [Besnard and Ragot, 2024, Zhang et al., 2024]. To mitigate catastrophic forgetting [Kirkpatrick et al., 2017] and maintain plasticity [Dohare et al., 2024, Ao and Fayek, 2023, Verwimp et al., 2023, Behrouz et al., 2025], continual and online learning methods update models sequentially as new data arrives. Classical deep online forecasting methods include FSNet [Pham et al., 2022] and OneNet [Zhang et al., 2023], as independent online evolving forecasters. When discussing time series online adapters, depending on the required access level to the base model, existing approaches can be categorized into three paradigms: (1) *White-Box (Parameter Finetuning)*, such as SOLID [Chen et al., 2024] and DSOF [Lau et al., 2024]; (2) *Grey-Box*, which includes TAFAS [Kim et al., 2025], δ-Adapter [Liang et al., 2025], ADAPT-Z [Huang et al., 2025], PETSA [Medeiros et al., 2025] and DynaTTA [Grover and Etemad, 2025]; (3) *Black-Box*, such as ELF [Lee et al., 2025] and the Ada-Y variant of δ-Adapter [Liang et al., 2025]. Additionally, broader explorations in other domains like traffic and spatial-temporal forecasting have introduced methods like FORESEE [Huang et al., 2026b], and ADCSD [Guo et al., 2025]. However, current literature specifically exploring online adaptation for TSFMs mainly includes TAFAS Kim et al. [2025], ELF Lee et al. [2025], δ-Adapter Liang et al. [2025], and remain insufficiently explored.

2.4 Comparison with online learning in natural language processing

While fine-tuning and online learning for LLMs such as Hu et al. [2021], Biderman et al. [2024], Houlsby et al. [2019], Pfeiffer et al. [2020], Li and Liang [2021], Xu et al. [2026a], Hu et al. [2026] have been widely analyzed, the online adaptation of TSFMs presents a different paradigm. The primary divergence lies in the availability of feedback during online inference. In typical natural language processing (NLP) applications, exact ground truth is rarely immediately accessible during test-time generation. In contrast, in time series forecasting, the ground truth reveals itself as time progresses, and this distinction motivates our **ORCA** methodology.

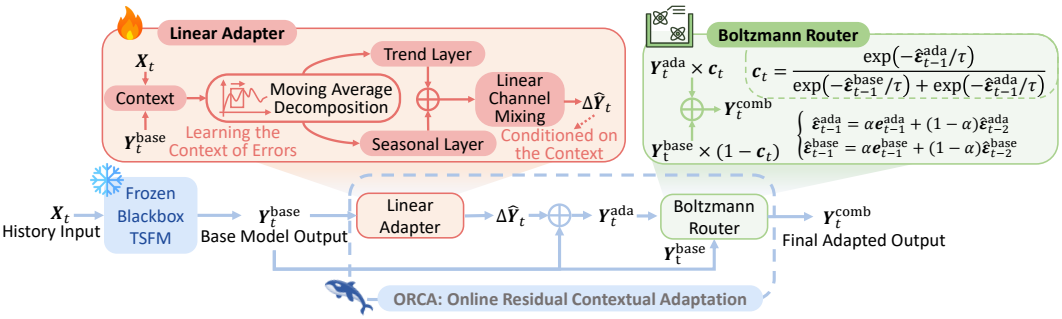

Figure 2: The overall architecture of ORCA. The history input $\boldsymbol{X}_t$ is fed into a frozen black-box TSFM to obtain the base output $\boldsymbol{Y}_t^{\text{base}}$. The Linear Adapter learns to generate the residual via Moving Average Decomposition, followed by Trend and Seasonal Layers. The Boltzmann Router dynamically calculates the confidence score $\boldsymbol{c}_t$ to mix $\boldsymbol{Y}_t^{\text{ada}}$ and $\boldsymbol{Y}_t^{\text{base}}$ into the final adapted output $\boldsymbol{Y}_t^{\text{comb}}$.

3 Methodology

3.1 Problem formulation

Consider a streaming time series forecasting scenario where data arrives sequentially. At each time step t, we observe a historical input matrix $\mathbf{X}_t \in \mathbb{R}^{D \times L}$, where D is the number of channels (variates) and L is the look-back window length. The objective is to forecast the future values over a horizon H. A pre-trained, frozen black-box TSFM takes $\mathbf{X}_t$ as input and generates a base prediction $\mathbf{Y}_t^{\text{base}} \in \mathbb{R}^{D \times H}$. Due to the non-stationarity of real-world environments and the knowledge gap between the pre-trained TSFM and the application scenario, the base model's prediction deviates from the ground truth $\mathbf{Y}_t^{\text{GT}} \in \mathbb{R}^{D \times H}$. We define the true prediction error as $\mathbf{E}_{\text{obs},t} = \mathbf{Y}_t^{\text{GT}} - \mathbf{Y}_t^{\text{base}}$. Because the base TSFM is treated as a black box (e.g., accessed via an API without access to internal gradients), we can only apply output-side corrections. Our goal is to learn an adapter function f_θ with parameters θ that predicts the residual $\Delta\hat{\mathbf{Y}}_t$ to refine the base prediction, ultimately yielding an adapted output $\mathbf{Y}_t^{\text{ada}} = \mathbf{Y}_t^{\text{base}} + \Delta\hat{\mathbf{Y}}_t$.

3.2 Context conditioned learning

ORCA aims to precisely capture the context in which base model errors occur. Conceptually, we aim to model the conditional error distribution $P(\mathbf{E}_t \mid \mathbf{X}_t, \mathbf{Y}_t^{\text{base}})$, where $\mathbf{E}_t$ denotes the true residual matrix at time t. Instead of relying on explicit probabilistic generative modeling, we pull this concept back into a deterministic forecasting paradigm. Let the contextual condition be denoted as $\mathbf{C}_t = [\mathbf{X}_t, \mathbf{Y}_t^{\text{base}}]$, and our Linear Adapter as a deterministic mapping $\Delta\hat{\mathbf{Y}}_t = f(\mathbf{C}_t)$. By optimizing the Mean Squared Error (MSE) loss during adaptation, we are minimizing the expected prediction risk. As derived in Appendix B.1, minimizing this risk mathematically dictates that the optimal deterministic mapping $f^*(\mathbf{C}_t)$ is exactly the conditional expectation:

$$f^*(\mathbf{C}_t) = \mathbb{E}[\mathbf{E}_t \mid \mathbf{X}_t, \mathbf{Y}_t^{\text{base}}] \tag{1}$$

This standard property of MSE optimization formalizes our objective: it allows us to rigorously translate the probabilistic modeling of $P(\mathbf{E}_t \mid \mathbf{X}_t, \mathbf{Y}_t^{\text{base}})$ into a purely deterministic residual regression task. Guided by this, our Linear Adapter takes the concatenation of the historical input $\mathbf{X}_t$ and the base prediction $\mathbf{Y}_t^{\text{base}}$ as its context, ensuring the model conditions on the relevant variables to estimate this expected residual.

3.3 Linear adapter

As illustrated in Figure 2, the Linear Adapter utilizes a linear architecture. Because the residual signal is highly noisy, we deliberately choose a linear architecture with strong inductive bias to provide the regularization and avoid overfitting. Specifically, the contextual input $\mathbf{C}_t$ is first processed through a Moving Average Decomposition block, which separates the temporal dynamics into a trend

component and a seasonal component. These components are independently processed by a Trend Layer and a Seasonal Layer. Finally, Linear Channel Mixing layers across the channel dimension capture cross-variate dependencies, outputting the predicted residual. The adapter output is then added to the base prediction to form the adapted output: $\mathbf{Y}_t^{\text{ada}} = \mathbf{Y}_t^{\text{base}} + \Delta\hat{\mathbf{Y}}_t$.

3.4 Boltzmann routing mechanism

To reduce the risk of online adaptation underperforming the base model, we introduce the **Boltzmann Router**. Drawing inspiration from statistical mechanics, where the Boltzmann distribution defines the probability of a system being in a specific state as a function of its energy, we analogously treat the smoothed predictive error as the energy of a model state. A lower error corresponds to a lower energy state, thus yielding a higher routing probability (confidence score). Let $\mathbf{e}_t^{\text{base}}$ and $\mathbf{e}_t^{\text{ada}}$ denote the instantaneous absolute error vectors (across D channels) for the base model and the adapter, respectively. We track their exponential moving averages (EMA) to estimate the smoothed errors (energies):

$$\begin{cases} \hat{\varepsilon}_{t-1}^{\text{ada}} = \alpha \mathbf{e}_{t-1}^{\text{ada}} + (1-\alpha)\hat{\varepsilon}_{t-2}^{\text{ada}} \\ \hat{\varepsilon}_{t-1}^{\text{base}} = \alpha \mathbf{e}_{t-1}^{\text{base}} + (1-\alpha)\hat{\varepsilon}_{t-2}^{\text{base}} \end{cases} \tag{2}$$

where $\alpha \in (0, 1)$ is the momentum coefficient. The channel-wise routing confidence vector $\mathbf{c}_t \in \mathbb{R}^D$ is then calculated via a Boltzmann softmax function with temperature τ:

$$\mathbf{c}_t = \frac{\exp(-\hat{\varepsilon}_{t-1}^{\text{ada}}/\tau)}{\exp(-\hat{\varepsilon}_{t-1}^{\text{base}}/\tau) + \exp(-\hat{\varepsilon}_{t-1}^{\text{ada}}/\tau)} \tag{3}$$

During inference, this confidence vector $\mathbf{c}_t$ gates the integration of the adapter's correction:

$$\mathbf{Y}_t^{\text{comb}} = \mathbf{Y}_t^{\text{base}} \odot (\mathbf{1} - \mathbf{c}_t) + \mathbf{Y}_t^{\text{ada}} \odot \mathbf{c}_t \tag{4}$$

where $\odot$ denotes element-wise multiplication broadcasted along the sequence dimension. From the perspective of online learning, this weighting mechanism connects to Prediction with Expert Advice and the Multiplicative Weights Update (MWU) framework [Freund and Schapire, 1997], acting to track the best expert in non-stationary environments [Herbster and Warmuth, 1998]. During online training, the channel-mean of this confidence vector, $\bar{c}_t = \text{mean}(\mathbf{c}_t)$, naturally acts as the prior regularization weight in our subsequent Bayesian loss function. A detailed theoretical analysis, including the regret bound of this Boltzmann routing mechanism, is provided in the Appendix B.3.

3.5 Predictive-space Bayesian update

A pivotal challenge in online learning is the plasticity-stability dilemma. Instead of imposing complex regularization on model parameters θ, we formulate the optimization as a Bayesian update directly in the predictive space. We define the belief over the conditional error distribution $P(\mathbf{E}_t \mid \mathbf{C}_t)$. At step t, we assume the observation likelihood follows an isotropic Gaussian distribution centered at the true error with noise variance σ_{obs}^2, denoted as $P(\mathbf{E}_{\text{obs},t} \mid \mathbf{E}_t, \mathbf{C}_t) \sim \mathcal{N}(\mathbf{E}_t, \sigma_{\text{obs}}^2\mathbf{I})$, where the observed error is $\mathbf{E}_{\text{obs},t} = \mathbf{Y}_t^{\text{GT}} - \mathbf{Y}_t^{\text{base}}$. To prevent catastrophic forgetting, we construct a prior distribution based on the prediction of the previous cycle's model, θ_{prior}. The prior is defined as $P(\mathbf{E}_t \mid \mathbf{C}_t, \mathcal{H}_{t-1}) \sim \mathcal{N}(\mathbf{E}_{\text{prior},t}, \sigma_{\text{prior}}^2\mathbf{I})$, where $\mathcal{H}_{t-1}$ denotes historical information, $\mathbf{E}_{\text{prior},t} = f_{\theta_{\text{prior}}}(\mathbf{C}_t)$ denotes the expected error predicted by the historical model, and σ_{prior}^2 represents the variance of this prior belief.

By applying Bayes' theorem and performing maximum a posteriori (MAP) estimation, minimizing the negative log-posterior is mathematically equivalent to minimizing the L_2 distance to both the observation and the prior, weighted by the precision ratio $\lambda_t = \sigma_{\text{obs}}^2/\sigma_{\text{prior}}^2$. As expanded in Appendix B.2, by transforming the errors back into the target time series space, we arrive at the Bayesian-inspired loss function defined on a training snapshot $s_t = \{\mathbf{X}_t, \mathbf{Y}_t^{\text{base}}, \mathbf{Y}_t^{\text{GT}}\}$:

$$\mathcal{L}(s_t, \theta) = \frac{1}{H \times D}\left(\|\mathbf{Y}_t^{\text{GT}} - \mathbf{Y}_t^{\text{ada}}\|_F^2 + \bar{c}_t\|\mathbf{Y}_t^{\text{ada}} - \mathbf{Y}_t^{\text{prior}}\|_F^2\right) \tag{5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathbf{Y}_t^{\text{prior}} = \mathbf{Y}_t^{\text{base}} + \mathbf{E}_{\text{prior},t}$, and the Linear Adapter output constructs $\mathbf{Y}_t^{\text{ada}} = \mathbf{Y}_t^{\text{base}} + \Delta\hat{\mathbf{Y}}_t$. The prior prediction is a stop-gradient target generated by the frozen adapter snapshot from the previous training cycle; gradients update only the current adapter.

We substitute the precision ratio λ_t with $\bar{c}_t = \text{mean}(\mathbf{c}_t)$, the scalar channel-mean derived from the Boltzmann Router. Solving for the optimal adapter output $\Delta\hat{\mathbf{Y}}_t^*$ gives:

$$\Delta\hat{\mathbf{Y}}_t^* = \frac{1}{1+\bar{c}_t}\mathbf{E}_{\text{obs},t} + \frac{\bar{c}_t}{1+\bar{c}_t}\mathbf{E}_{\text{prior},t} \tag{6}$$

This formulation corresponds exactly to the analytical mean of the product of two Gaussian distributions, as derived in Appendix B.2. As prior variance is computationally impractical and boundless, $\bar{c}_t$ serves as a stable surrogate. Practically, the exponential moving average of historical errors serves as a first-order proxy for localized predictive variance, and the Boltzmann Softmax smoothly maps this into a relative precision score. When the adapter performs well ($\bar{c}_t \to 1$), the prior precision is high, anchoring the model to historical beliefs. Conversely, the increase of adapter errors causes $\bar{c}_t \to 0$ to drop the prior, forcing the adapter to quickly digest new patterns.

3.6 Inference and online training pipeline

ORCA operates through a step-by-step inference and periodic cycle-training pipeline, utilizing a First-In-First-Out (FIFO) replay buffer $\mathcal{B}$ (Figure 3). During continuous inference at each time step t, the Linear Adapter conditions on the observable context $\mathbf{C}_t$ to generate the residual correction $\Delta\hat{\mathbf{Y}}_t$. Once the true horizon becomes fully observable, we construct a training snapshot s_t and push it into the FIFO buffer $\mathcal{B}$. ORCA executes cycle training periodically every horizon H steps. During each training cycle, we draw batches from the buffer using random sampling with an exponential decay probability. This mechanism assigns a higher selection likelihood to more recent snapshots. Consequently, it guarantees that the Linear Adapter maintains high plasticity towards recent distribution shifts, while retaining a sufficient proportion of historical anchor points to stabilize the Bayesian prior. Once a training cycle is complete, the current adapter parameters θ are frozen to update the delayed prior model θ_{prior}, preparing the Bayesian loss anchor for the subsequent cycle.

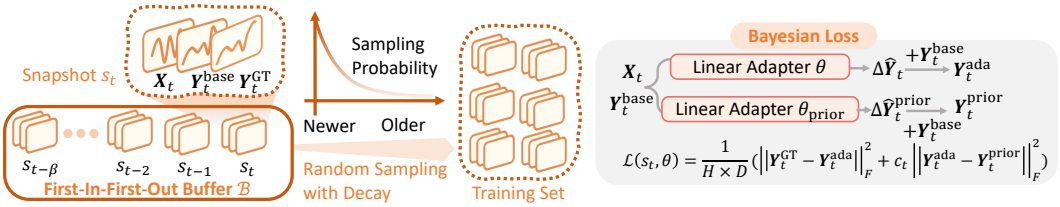

Figure 3: The online training pipeline of ORCA. Snapshots s_t containing $\{\boldsymbol{X}_t, \boldsymbol{Y}_t^{\text{base}}, \boldsymbol{Y}_t^{\text{GT}}\}$ are stored in a FIFO Buffer. Random sampling with an exponential decay probability prioritizes recent samples. The Bayesian Loss simultaneously aligns $\boldsymbol{Y}_t^{\text{ada}}$ with the ground truth $\boldsymbol{Y}_t^{\text{GT}}$ and anchors it to the prior prediction $\boldsymbol{Y}_t^{\text{prior}}$ generated by the frozen, stop-gradient model copy θ_{prior}.

4 Experiments

4.1 Experimental setup

Datasets. We evaluate our proposed ORCA framework on eight widely used real-world time series benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Electricity [Zhou et al., 2021], Exchange [Lai et al., 2018], Weather, and Traffic [Wu et al., 2021]. The full set of eight datasets is employed for the main experiments to compare ORCA against various baselines, as in previous work [Liang et al., 2025]. However, given the computational cost associated with online evaluation across all datasets for all base models, our ablation studies and hyperparameter sensitivity analyses are conducted on a subset of six datasets. This subset excludes the high-dimensional Traffic and Electricity datasets, which aligns with the evaluation protocols adopted by prior works [Kim et al., 2025].

Base Models. Existing online adaptation works have primarily conducted experiments on first-generation time series foundation models Kim et al. [2025], Lee et al. [2025], Liang et al. [2025]. Consequently, there remains a lack of evaluations on the latest base models that currently dominate public leaderboards such as GIFT-Eval [Aksu et al., 2024] and fev-bench [Shchur et al., 2026]. To bridge this gap, we select five recent (released between 2025 and 2026), popular, and high-performing

Table 1: Main experimental results comparing the vanilla zero-shot performance of five base TSFMs and their ORCA-refined counterparts. The results are averaged over three forecasting horizons ($H \in \{30, 96, 336\}$), with each horizon provided in Appendix C.2. A negative percentage indicates a reduction in MSE, meaning positive performance improvement.

Model		Chronos-2		Moirai-2		TiRex		TimesFM-2.5		Sundial		Avg.
ETTh1	Vanilla	0.2756		0.2653		0.2671		0.2771		0.2381		
	Refined	0.2605	-5.5%	0.2579	-2.8%	0.2579	-3.5%	0.2634	-4.9%	0.2460	3.3%	-2.7%
ETTh2	Vanilla	0.0373		0.0350		0.0357		0.0369		0.0358		
	Refined	0.0366	-1.8%	0.0355	1.3%	0.0355	-0.5%	0.0366	-0.9%	0.0360	0.6%	-0.3%
ETTm1	Vanilla	0.2197		0.2538		0.2564		0.2229		0.2103		
	Refined	0.2038	-7.2%	0.2164	-14.7%	0.2149	-16.2%	0.2053	-7.9%	0.2014	-4.2%	-10.1%
ETTm2	Vanilla	0.0245		0.0274		0.0263		0.0260		0.0248		
	Refined	0.0224	-8.7%	0.0232	-15.3%	0.0233	-11.4%	0.0231	-10.9%	0.0232	-6.5%	-10.6%
Exchange	Vanilla	0.0031		0.0033		0.0035		0.0033		0.0042		
	Refined	0.0031	-0.0%	0.0034	2.0%	0.0032	-8.5%	0.0032	-3.1%	0.0037	-10.4%	-4.0%
Weather	Vanilla	0.0477		0.0565		0.0542		0.0497		0.0415		
	Refined	0.0435	-8.9%	0.0446	-21.2%	0.0463	-14.5%	0.0441	-11.3%	0.0408	-1.7%	-11.5%
Electricity	Vanilla	0.0571		0.0576		0.0549		0.0579		0.0484		
	Refined	0.0517	-9.5%	0.0522	-9.5%	0.0503	-8.5%	0.0525	-9.4%	0.0453	-6.4%	-8.7%
Traffic	Vanilla	0.2239		0.2246		0.2462		0.2219		0.2564		
	Refined	0.2202	-1.7%	0.2217	-1.3%	0.2367	-3.8%	0.2201	-0.8%	0.2425	-5.4%	-2.6%
Models Avg.		-5.4%		-7.7%		-8.4%		-6.1%		-3.8%		**-6.3%**

base models for our evaluation: Chronos-2 Ansari et al. [2025], Moirai 2.0 Liu et al. [2026a], TiRex Auer et al. [2025], TimesFM-2.5 (the version released in September 2025) Das et al. [2024], and Sundial Liu et al. [2025b]. Across all selected models, the input look-back window length is uniformly set to $L = 520$, and the testing horizons are set to $H \in \{30, 96, 336\}$ [Lee et al., 2025]. This setting follows ELF and provides a common input/output interface across heterogeneous frozen TSFMs, covering short-, medium-, and long-range prediction. Unlike full-shot forecasting, where a separate model is trained on each dataset, this protocol controls the comparison across frozen backbones. Unless specified otherwise, results in this paper represent the average performance across these three horizons, a common practice in prior work [Liang et al., 2025]. While TSFMs output probabilistic forecasts, our framework focuses on operating on the median of the quantile or sampled outputs.

Other Settings. The primary objective of this paper is to investigate the potential for performance enhancement when the foundation model operates strictly as a black box (e.g., accessed via a cloud API). However, because the most recognized and highly adopted TSFMs are currently still in the open-source stage, we utilize these open-source base models to simulate the API deployment environment by strictly disabling gradient backpropagation. Other experimental details are provided in Appendix A, such as the comprehensive ORCA settings (decomposition kernel, layer size, and optimizer), the online training/evaluation data framework, and settings of baselines.

4.2 Main results

We evaluate the overall performance of ORCA against the vanilla zero-shot TSFMs across a comprehensive matrix of 120 experimental configurations (8 datasets $\times$ 5 base models $\times$ 3 forecasting horizons). The averaged MSE results and the relative performance improvements are summarized in Table 1. As illustrated, ORCA consistently reduces the forecasting error across the vast majority of scenarios. Across the 40 aggregated cases (8 datasets $\times$ 5 TSFMs), ORCA successfully decreases the base models' forecasting errors in 90% of the evaluations. Furthermore, regulated by the Boltzmann Router, the performance degradation in the few cases remains bounded, with the maximum error increase limited to 3.3%, whereas the maximum error reduction achieves up to 21.2%.

Results across three random seeds are provided in Appendix D.1.

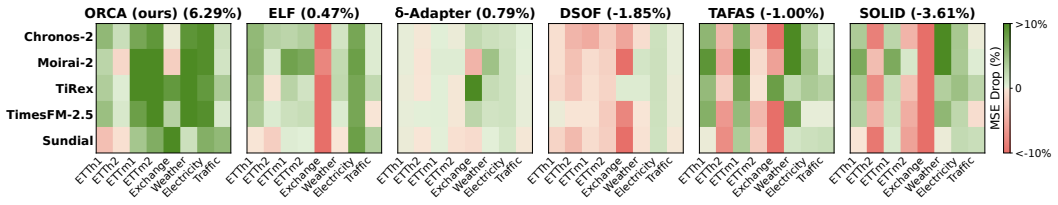

Figure 4: Heatmaps illustrating the relative MSE drop (%) achieved by various online adaptation methods compared to the vanilla zero-shot TSFMs across 8 datasets and 5 base models. Green cells indicate a reduction in MSE (improvement), while red cells indicate an increase in MSE (degradation).

4.3 Comparison with baselines

To demonstrate the effectiveness of our proposed framework, we compare ORCA against comprehensive baselines, as in Figure 4. **(1) First,** we compare SOTA (state-of-the-art) black-box online adaptation methods, including ELF [Lee et al., 2025] and δ-Adapter (Ada-Y variant) [Liang et al., 2025]. **(2) However,** because general black-box online adaptation is still insufficiently explored, we have to introduce more baselines by adapting SOTA white-box and grey-box methods into the black-box paradigm (DSOF Lau et al. [2024], TAFAS Kim et al. [2025], and SOLID Chen et al. [2024]), **only as a supplement**, detailed in Appendix A.4. Importantly, they are not traditional comparisons, but rather to underscore the **research urgency and non-trivality** for more black-box TSFM adaptation methods, by proving that direct modification from white-box and grey-box methods fails. **(3) Finally,** for a fairer comparison than insufficient black-box methods and adapted white-box and grey-box methods, we added statistical baselines including Ridge Regression [Hoerl and Kennard, 1970] and error–trend–seasonality (ETS) exponential smoothing [Hyndman et al., 2008], with settings and results detailed in Appendix C.3. Without proper structural design, as in ORCA or ELF, statistical methods overfit to noise; despite occasional reductions in base model errors, they can increase errors by hundreds of percent, excluded in the heatmap for readability. ETS predicts future residuals from past residuals, as it requires identical input-output sequences. Notably, Ridge Regression conditioned on context outperforms ETS, further supporting our hypothesis of learning the context of errors.

Additional forecasting comparisons are provided in Appendix D.2.

4.4 Quantitative analysis of different learning hypotheses

A question in online adaptation for black-box models is: *What should the adapter learn?* To quantitatively analyze different learning hypotheses, we conduct a comprehensive ablation study on the adapter's input of the proposed ORCA. In Figure 5, different input configuration of the adapter is supported by different learning hypotheses. Attempting to directly learn the historical error sequence itself yields highly suboptimal results, with an average performance drop of -0.7%. In several datasets, relying solely on past errors increases the forecasting error. Conversely, when we condition the adapter on either the Input or the Prediction, the performance improves. Notably, our proposed configuration, which utilizes both to obtain a comprehensive error context, achieves the optimal average improvement of 6.5%. Interestingly, incorporating the past error into this optimal set slightly degrades the average performance to 6.1%. The decline from adding past errors shows that richer inputs do not monotonically improve performance. Current inputs and base predictions are observable at prediction time, whereas the current error is only revealed with future ground truth. These findings corroborate our core insight: errors follow a conditional distribution $P(E|\boldsymbol{X}_t, \boldsymbol{Y}_t^{\text{base}})$, and therefore we should learn from the context of errors.

4.5 Ablation and sensitivity analysis

As depicted in Figure 6, removing any of the proposed components leads to a performance decline, especially the Boltzmann Router. This validates the effectiveness of the Boltzmann routing mechanism in safely harnessing the adaptation. Without the router, the framework still retains about 80% of the full improvement, indicating that routing complements an effective learned correction. Furthermore,

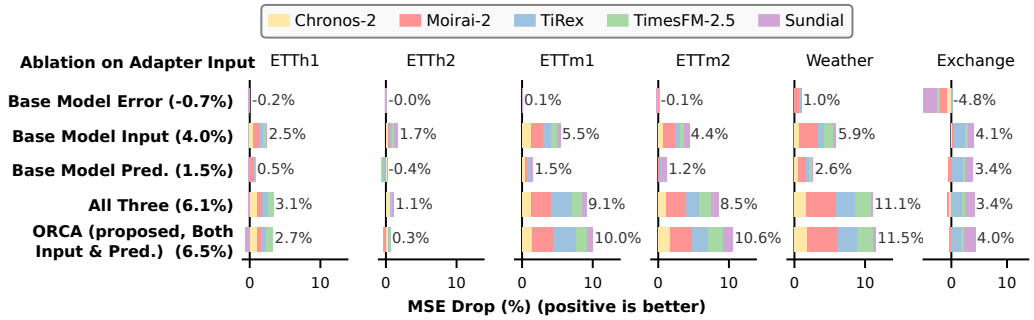

Figure 5: Ablation study on the adapter's input combinations across six datasets and five TSFMs, averaged over $H \in \{30, 96, 336\}$. The bars represent the MSE drop ratio (%). Positive values indicate an improvement (error reduction) over the vanilla base model. Our proposed input (Base Model Input & Prediction) achieves the highest average improvement of 6.5%.

the Linear Channel Mixing layers and the Bayesian prior loss also provide contributions. We also examine the role of the exponential decay FIFO buffer by testing a variant without a decay buffer, with a use-and-discard buffer (size 3000, batch size 256). It waits until 3000 samples are collected, triggers training, completely flushes all memory, and then waits to fill again. To further validate the necessity of the proposed Boltzmann routing, we tested a router variant, a hard binary router, detailed in Appendix C.1. With the same EMA error mechanism, the binary variant only achieves a MSE reduction of 3.80%, indicating the effectiveness of our Boltzmann router. Comparing the structural ablations with the input ablations from Section 4.4, the wrong learning hypotheses reduce the performance more severely. This contrast proves our insight: in black-box online adaptation, *what* the model learns is rather impactful.

Local regularization costs and a directed-prior comparison are examined in Appendix D.3.

In the ORCA framework, the hyperparameters primarily involve the Boltzmann Router and the online training configuration. For the Boltzmann Router, the routing temperature τ and the Exponential Moving Average (EMA) momentum α govern the confidence allocation. Guided by our theoretical analysis (detailed in Appendix B.3.2), we set the default values to $\tau = 0.1$ and $\alpha = 0.2$. Empirical evaluations, as illustrated in Figure 7, demonstrate that the relative MSE drop across five base TSFMs and 6 datasets, as in Figure 7, remains stable and insensitive under varying combinations of τ and α. Beyond the router, controlling the data instances the adapter encounters during each training cycle is crucial for balancing plasticity and stability. The two primary parameters governing this mechanism are the length of the decay FIFO buffer and the random sampling batch size. As further illustrated in Figure 7, varying the buffer length (from 2000 to 4000) and the batch size (from 128 to 512) also results in insignificant fluctuations in performance. Consequently, our default configuration of a buffer length of 3000 and a batch size of 256 was chosen directly based on device capacity.

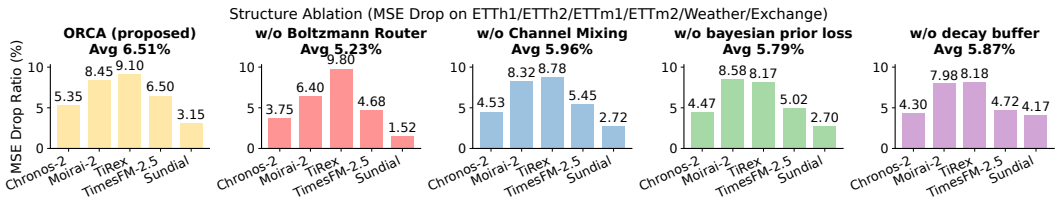

Figure 6: Structural ablation study across six datasets and five TSFMs, averaged over $H \in \{30, 96, 336\}$. The average MSE drop ratio falls from 6.51% to 5.23% without the Boltzmann Router, highlighting its critical role in harnessing online stability.

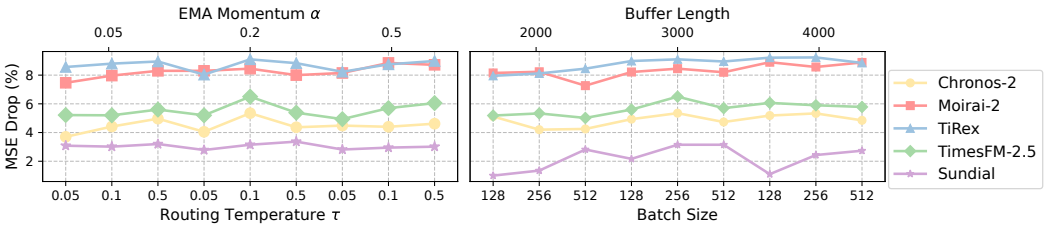

Figure 7: Hyperparameter sensitivity analysis of ORCA across different base models in 6 datasets. The relative MSE drop (%) remains generally stable under varying routing temperatures τ, EMA momentum α, FIFO buffer lengths, and sampling batch sizes.

Table 2: Computational efficiency of ORCA on ETTh1 and Electricity datasets with Chronos-2 and Moirai-2, evaluated on a single NVIDIA B200 GPU with a forecasting horizon of $H = 96$.

Dataset	**Base Inference Time (ms per step)**	**ORCA Inference (per step)**			**ORCA Training (per cycle)**	
		Time (ms)	FLOPs	GPU Usage (MB)	Time (ms)	GPU Usage (MB)
			Chronos-2			
ETTh1	121.45	16.74	2.5×10^6	353.9	2049.2	463.0
Electricity	167.30	12.61	1.1×10^8	4236.3	2115.9	6216.5
			Moirai-2			
ETTh1	152.90	23.82	2.5×10^6	161.3	2115.7	270.7
Electricity	386.57	25.68	1.1×10^8	4041.6	3520.9	6017.8

4.6 Performance and efficiency analysis

We analyze the efficiency of ORCA on two datasets: ETTh1 (7 channels) and Electricity (321 channels). We profile Chronos-2 and Moirai-2. The evaluation is conducted on a single NVIDIA B200 graphics processing unit (GPU) with a forecasting horizon of $H = 96$. As shown in Table 2, we consider a non-overlapping evaluation setting where the online cycle training is executed once every 96 steps. Across Chronos-2 and Moirai-2, the amortized ORCA overhead is 34.65–62.36 ms per step, excluding base-model inference. Amortized overhead is the adapter inference time plus one training-cycle time divided by 96. This leaves substantial headroom for minute-level forecasting, although synchronous training can introduce a 2.05–3.52 s pause once per cycle. Asynchronous training could make second-level deployment feasible. ORCA requires 2.48×10^6 and 1.14×10^8 floating-point operations (FLOPs) per inference step on ETTh1 and Electricity, respectively. Furthermore, it is important to note that the reported base model inference time reflects a local hardware deployment. While actual closed-source TSFM API latencies may fluctuate, the client-side overhead introduced by ORCA remains lightweight.

5 Conclusion and limitations

Conclusion. We addressed the critical challenge of adapting frozen, black-box TSFMs to streaming data. We introduced ORCA, an online adaptation framework that enhances TSFM predictions without requiring access to internal model parameters. Extensive evaluations across diverse datasets and base models demonstrate that ORCA consistently reduces errors while maintaining low latency.

Limitations. As a post-hoc corrector, ORCA's performance relies on the base model providing reasonable forecasts. Furthermore, while we have established evaluations on standard benchmark datasets with deterministic metrics, future work will explore its deployment in more complex data featuring abrupt structural shifts and with probabilistic metrics. Local smoothing costs remain a consideration; targeted dynamic metrics and directed regularization are promising directions beyond aggregate MSE.

Acknowledgments and Disclosure of Funding

This work was supported in part by the CUHK Research Matching Scheme under Grant No. 7106937 and 8601130.

References

Md Atik Ahamed and Qiang Cheng. TimeMachine: A Time Series is Worth 4 Mambas for Long-term Forecasting. *arXiv preprint arXiv:2403.09898*, August 2024. doi: 10.48550/arXiv.2403.09898.

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*, May 2024. ISSN 2835-8856.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From Univariate to Universal Forecasting. *arXiv preprint arXiv:2510.15821*, October 2025. doi: 10.48550/arXiv.2510.15821.

Sio-Iong Ao and Haytham Fayek. Continual Deep Learning for Time Series Modeling. *Sensors*, 23(16):7167, January 2023. ISSN 1424-8220. doi: 10.3390/s23167167.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-Shot Forecasting Across Long and Short Horizons with Enhanced In-Context Learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, October 2025.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested Learning: The Illusion of Deep Learning Architectures. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, October 2025.

Abdelhakim Benechehab, Vasilii Feofanov, Giuseppe Paolo, Albert Thomas, Maurizio Filippone, and Balázs Kégl. AdaPTS: Adapting Univariate Foundation Models to Probabilistic Multivariate Time Series Forecasting. In *Forty-Second International Conference on Machine Learning*, June 2025.

Quentin Besnard and Nicolas Ragot. Continual Learning for Time Series Forecasting: A First Survey. *Engineering Proceedings*, 68(1):49, 2024. ISSN 2673-4591. doi: 10.3390/engproc2024068049.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA Learns Less and Forgets Less. *Transactions on Machine Learning Research*, May 2024.

Mouxiang Chen, Lefei Shen, Han Fu, Zhuo Li, Jianling Sun, and Chenghao Liu. Calibration of Time-Series Forecasting: Detecting and Adapting Context-Driven Distribution Shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages 341–352, New York, NY, USA, August 2024. Association for Computing Machinery. doi: 10.1145/3637528.3671926.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. VisionTS: Visual Masked Autoencoders Are Free-Lunch Zero-Shot Time Series Forecasters. In *Forty-Second International Conference on Machine Learning*, June 2025.

Peng Chen, Zhihao Zhuang, Hongzhou Chen, Junhao Huang, Aiping Yang, Mengsen Wu, Yiding Liu, Xilin Dai, and Zewei Dong. Beyond numerical time series: A unified benchmark for multimodal forecasting with heterogeneous context. *arXiv preprint arXiv:2609.15087*, 2026.

J.T. Connor, R.D. Martin, and L.E. Atlas. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5(2):240–254, March 1994. ISSN 1941-0093. doi: 10.1109/72.279188.

Xilin Dai, Zhijian Xu, Wanxu Cai, and Qiang Xu. From Samples to Scenarios: A New Paradigm for Probabilistic Forecasting. In *The Fourteenth International Conference on Learning Representations*, October 2025a.

Xilin Dai, Ruidi Zhou, Jinhao Zhang, Keyi He, Fanfan Lin, and Hao Ma. SocNet: A Physics-Guided Neural Network for Battery State-of-Charge Estimation Robust to Temperature Variations and Sensor Noises. *IEEE Transactions on Transportation Electrification*, 11(5):11165–11176, October 2025b.

Xilin Dai, Wanxu Cai, Zhijian Xu, and Qiang Xu. Position: Universal time series foundation models rest on a category error. *arXiv preprint arXiv:2602.05287*, 2026a. URL `https://arxiv.org/abs/2602.05287`.

Xilin Dai, Ruidi Zhou, Jinhao Zhang, Fanfan Lin, Weifeng Zhang, and Hao Ma. Socgate: Physics-gated neural network for stable multicycle estimation of battery state-of-charge. *IEEE Transactions on Industrial Electronics*, 73(4):5518–5529, 2026b.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 10148–10167, Vienna, Austria, July 2024. JMLR.org.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Maintaining Plasticity in Deep Continual Learning. *arXiv preprint arXiv:2306.13812*, April 2024. doi: 10.48550/arXiv.2306.13812.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny Time Mixers (TTMs): Fast Pre-trained Models for Enhanced Zero/Few-Shot Forecasting of Multivariate Time Series. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

Yoav Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

Everette S. Gardner Jr. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1):1–28, 1985. ISSN 1099-131X. doi: 10.1002/for.3980040103.

Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. TimeGPT-1. *arXiv preprint arXiv:2310.03589*, May 2024. doi: 10.48550/arXiv.2310.03589.

Shivam Grover and Ali Etemad. Shift-Aware Test Time Adaptation and Benchmarking for Time-Series Forecasting. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! At ICML 2025*, July 2025.

Pengxin Guo, Pengrong Jin, Ziyue Li, Lei Bai, and Yu Zhang. Online Test-Time Adaptation of Spatial–Temporal Traffic Flow Forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 26(10):15323–15333, October 2025.

Mark Herbster and Manfred K. Warmuth. Tracking the Best Expert. *Machine Learning*, 32(2):151–178, August 1998.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799. PMLR, May 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, October 2021.

Yifan Hu, Xilin Dai, Zhiyuan Qu, Yiding Liu, Zewei Dong, Jiang ming Yang, and Qiang Xu. When tomorrow becomes today: Self-evolving policies for agentic time-series forecasting. *arXiv preprint arXiv:2609.24862*, 2026. URL `https://arxiv.org/abs/2609.24862`.

Xiannan Huang, Shuhan Qiu, Jiayuan Du, and Chao Yang. Online time series prediction using feature adjustment. In *The Fourteenth International Conference on Learning Representations*, October 2025.

Xiannan Huang, Shen Fang, Shuhan Qiu, Chengcheng Yu, Jiayuan Du, and Chao Yang. TEFL: Prediction-Residual-Guided Rolling Forecasting for Multi-Horizon Time Series. *arXiv preprint arXiv:2602.22520*, February 2026a. doi: 10.48550/arXiv.2602.22520.

Xiannan Huang, Quan Yuan, and Chao Yang. Learning from Yesterday's Error: An Efficient Online Learning Method for Traffic Demand Prediction. *arXiv preprint arXiv:2602.21757*, February 2026b. doi: 10.48550/arXiv.2602.21757.

Rob Hyndman, Anne Koehler, Keith Ord, and Ralph Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer Series in Statistics. Springer, Berlin, Heidelberg, 2008. ISBN 978-3-540-71916-8 978-3-540-71918-2. doi: 10.1007/978-3-540-71918-2.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *The Twelfth International Conference on Learning Representations*, October 2023.

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 Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, volume 39 of *AAAI'25*, pages 17868–17876. AAAI Press, February 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i17.33965.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. doi: 10.1073/pnas.1611835114.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 95–104, New York, NY, USA, June 2018. Association for Computing Machinery. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3210006.

Ying-yee Ava Lau, Zhiwen Shao, and Dit-Yan Yeung. Fast and Slow Streams for Online Time Series Forecasting Without Information Leakage. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Thomas L. Lee, William Toner, Rajkarn Singh, Artjom Joosen, and Martin Asenov. Lightweight Online Adaption for Time Series Foundation Model Forecasts. In *Forty-Second International Conference on Machine Learning*, June 2025.

Thomas L. Lee, Edoardo M. Ponti, and Amos Storkey. Adapting Time Series Foundation Models through Data Mixtures. *arXiv preprint arXiv:2603.02840*, March 2026. doi: 10.48550/arXiv.2603.02840.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353.

Daojun Liang, Qi Li, Yinglong Wang, Jing Chen, Hu Zhang, Xiaoxiao Cui, Qizheng Wang, and Shuo Li. The Forecast After the Forecast: A Post-Processing Shift in Time Series. In *The Fourteenth International Conference on Learning Representations*, October 2025.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation Models for Time Series Analysis: A Tutorial and Survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages 6555–6565, New York, NY, USA, August 2024. Association for Computing Machinery. doi: 10.1145/3637528.3671451.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When Less Is More for Time Series Forecasting. *arXiv preprint arXiv:2511.11698*, February 2026a. doi: 10.48550/arXiv.2511.11698.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-MoE: Empowering Time Series Foundation Models with Sparse Mixture of Experts. In *Forty-second International Conference on Machine Learning*, June 2025a.

Yiding Liu, Yifan Hu, Hongjie Xia, Peiyuan Liu, Hongzhou Chen, Xilin Dai, Zewei Dong, and Jiang-Ming Yang. Falcon-x: A time series foundation model for heterogeneous multivariate modeling. *arXiv preprint arXiv:2605.27286*, 2026b. URL `https://arxiv.org/abs/2605.27286`.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 9881–9893, Red Hook, NY, USA, November 2022. Curran Associates Inc. ISBN 978-1-7138-7108-8.

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*, October 2023.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 32369–32399, Vienna, Austria, July 2024. JMLR.org.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A Family of Highly Capable Time Series Foundation Models. In *Forty-Second International Conference on Machine Learning*, June 2025b.

Yong Liu, Xingjian Su, Shiyu Wang, Haoran Zhang, Haixuan Liu, Yuxuan Wang, Zhou Ye, Yang Xiang, Jianmin Wang, and Mingsheng Long. Timer-S1: A Billion-Scale Time Series Foundation Model with Serial Scaling. *arXiv preprint arXiv:2603.04791*, April 2026c. doi: 10.48550/arXiv.2603.04791.

Haoyu Ma, Yushu Chen, Wenlai Zhao, Jinzhe Yang, Yingsheng Ji, Xinghua Xu, Xiaozhu Liu, Hao Jing, Shengzhuo Liu, and Guangwen Yang. A Mamba Foundation Model for Time Series Forecasting. *arXiv preprint arXiv:2411.02941*, November 2024. doi: 10.48550/arXiv.2411.02941.

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*, July 2025.

Marcel Meyer, Sascha Kaltenpoth, Kevin Zalipski, and Oliver Müller. Time Series Foundation Models: Benchmarking Challenges and Requirements. *arXiv preprint arXiv:2510.13654*, October 2025. doi: 10.48550/arXiv.2510.13654.

John A. Miller, Mohammed Aldosari, Farah Saeed, Nasid Habib Barna, Subas Rana, I. Budak Arpinar, and Ninghao Liu. A Survey of Deep Learning and Foundation Models for Time Series Forecasting. *arXiv preprint arXiv:2401.13912*, January 2024. doi: 10.48550/arXiv.2401.13912.

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*, September 2022.

Kwangryeol Park, Jaeho Kim, and Seulki Lee. AliO: Output alignment matters in long-term time series forecasting. In *Thirty-ninth Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=AuOZDp4gy7`.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.617.

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*, September 2022.

Domenico Piccolo. A Distance Measure for Classifying Arima Models. *Journal of Time Series Analysis*, 11(2): 153–164, 1990. ISSN 1467-9892. doi: 10.1111/j.1467-9892.1990.tb00048.x.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting. *arXiv preprint arXiv:2310.08278*, February 2024. doi: 10.48550/arXiv.2310.08278.

Oleksandr Shchur, Abdul Fatir Ansari, Caner Turkmen, Lorenzo Stella, Nick Erickson, Pablo Guerron, Michael Bohlke-Schneider, and Yuyang Wang. Fev-bench: A Realistic Benchmark for Time Series Forecasting. *arXiv preprint arXiv:2509.26468*, February 2026. doi: 10.48550/arXiv.2509.26468.

THUML. Time Series Library (TSLib). GitHub repository, 2023. URL https://github.com/thuml/Time-Series-Library. Accessed October 2, 2026.

Eli Verwimp, Rahaf Aljundi, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L. Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H. Lampert, Martin Mundt, Razvan Pascanu, Adrian Popescu, Andreas S. Tolias, Joost van de Weijer, Bing Liu, Vincenzo Lomonaco, Tinne Tuytelaars, and Gido M. van de Ven. Continual Learning: Applications and the Road Forward. *Transactions on Machine Learning Research*, November 2023.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 53140–53164, Vienna, Austria, July 2024. JMLR.org.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*, November 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*, September 2022.

Binyan Xu, Xilin Dai, and Kehuan Zhang. Contextual agentic memory is a memo, not true memory. *arXiv preprint arXiv:2604.27707*, 2026a. URL https://arxiv.org/abs/2604.27707.

Binyan Xu, Fan Yang, Xilin Dai, Di Tang, and Kehuan Zhang. From internal diagnosis to external auditing: A vlm-driven paradigm for online test-time backdoor defense. *arXiv preprint arXiv:2601.19448*, 2026b. URL https://arxiv.org/abs/2601.19448.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling Time Series with $10k$ Parameters. In *The Twelfth International Conference on Learning Representations*, October 2023.

Zhijian Xu, Wanxu Cai, Xilin Dai, Zhaorong Deng, and Qiang Xu. Fidel-ts: A high-fidelity multimodal benchmark for time series forecasting. *arXiv preprint arXiv:2509.24789*, 2026c. URL https://arxiv.org/abs/2509.24789.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, June 2023.

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*, November 2023.

YiFan Zhang, Weiqi Chen, Zhaoyang Zhu, Dalin Qin, Liang Sun, Xue Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. Addressing Concept Shift in Online Time Series Forecasting: Detect-then-Adapt. *arXiv preprint arXiv:2403.14949*, March 2024. doi: 10.48550/arXiv.2403.14949.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, May 2021.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One Fits All: Power General Time Series Analysis by Pretrained LM. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

Appendix

A Experiment details . . . 16

A.1 Online training and evaluation framework . . . 16

A.2 Model configuration . . . 17

A.3 Evaluation metrics . . . 17

A.4 Adaptation of white-box and grey-box baselines to black-box settings . . . 17

B Theoretical analysis . . . 18

B.1 Context conditioned learning. . . . 18

B.2 Predictive-space Bayesian update . . . 19

B.2.1 Maximum a posteriori (MAP) derivation . . . 19

B.2.2 Theoretical justification of the precision surrogate . . . 19

B.3 Boltzmann routing . . . 20

B.3.1 Regret bound of Boltzmann routing . . . 20

B.3.2 Hyperparameter sensitivity analysis of Boltzmann routing . . . 21

C Supplementary experiment results . . . 22

C.1 A router variant compared to the Boltzmann router . . . 22

C.2 Horizon-wise performance breakdown . . . 22

C.3 Statistical baselines: setting and results . . . 23

D Additional empirical analyses . . . 26

D.1 Multi-seed reliability . . . 26

D.2 Full-shot forecasting baselines . . . 26

D.3 Bayesian regularization diagnostics . . . 27

A Experiment details

In addition, we split each dataset chronologically into training, validation, and testing sets following the standard 7:1:2 ratio. Since the evaluated TSFMs are zero-shot, no actual pre-training or fine-tuning of the base models is required. Our proposed ORCA adapter can seamlessly modify the base model's output immediately after a brief warmup phase (i.e., the first time the FIFO decay buffer has acquired enough data as buffer length). However, to ensure a fair and aligned comparison with existing literature and baselines, all reported metrics are strictly evaluated on the testing set portion of the data. Mean Squared Error (MSE) serves as our primary evaluation metric. Detailed formulas for the metrics are provided in this appendix.

A.1 Online training and evaluation framework

In the streaming time series forecasting scenario, it is critical to prevent future data leakage. Our online training and evaluation framework adheres to chronological boundaries. At time step t, the base model and the adapter only have access to the historical look-back window $\mathbf{X}_t$ to predict the future horizon $\mathbf{Y}_t$. The ground truth for this prediction, $\mathbf{Y}_t^{\mathrm{GT}}$, spans from t to $t + H - 1$. To avoid data leakage, this ground truth is not revealed to the model immediately. Instead, the framework waits until step $t + H$, when the true values of the entire horizon become fully observable. Only then is the complete snapshot paired and pushed into the FIFO replay buffer for cycle training. This mechanism ensures that the online adaptation relies solely on retrospective data, simulating real-world streaming deployments.

A.2 Model configuration

The complete ORCA model uses a standardized set of hyperparameters across all base models. Specifically, the Linear Adapter consists of 2 blocks with a hidden dimension of 128. The refiner context input concatenates the historical look-back $\mathbf{X}_t$ and the base model prediction $\mathbf{Y}_t^{\text{base}}$. During the streaming cycle training, we maintain a FIFO replay buffer with a capacity of 3000 snapshots. At each training cycle, we draw batches of size 256. The optimizer is AdamW with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5}. Furthermore, to strictly align with the continuous online learning paradigm and maintain cycle efficiency, the adapter undergoes a fixed number of parameter update steps during each training cycle without employing any early stopping mechanisms. To enforce sparsity, the Bayesian loss regularization weight via the L1 channel mixer is set to 1×10^{-3}. The core kernel size parameter for the moving average decomposition is dynamically determined by the prediction length H. Consistent with prior works, we set the kernel size to 25 when $H > 30$, and to 7 when $H \leq 30$. This configuration is specifically designed to effectively capture structural temporal patterns whenever sufficient contextual information is available. When a new base model begins its streaming inference, the adapter undergoes an initial warmup phase (first buffer settled) of 50 epochs on the buffer, after which it performs 10 gradient update steps per cycle. For the Boltzmann Router, the routing temperature is set to $\tau = 0.1$, and the exponential moving average (EMA) momentum for tracking the errors is $\alpha = 0.2$. The update rule is configured to the Bayesian mode to anchor the adapter output to the historical prior.

A.3 Evaluation metrics

To evaluate the forecasting performance, we employ Mean Squared Error (MSE). Because different time series foundation models employ distinct internal normalization techniques that are embedded as black-box operations, their direct loss magnitudes are mathematically incompatible. To establish a fair comparison, we implement a global scalar normalization. For each dataset, we compute a single dataset-level scalar σ_{global} representing the overall scale of the time series. Both the base models' predictions and the ORCA adapter's refined predictions are first evaluated against the ground truth in the raw, unnormalized domain. Subsequently, the calculated errors are divided by this dataset-specific global scalar σ_{global} (or its squared value for MSE). The globally normalized metrics are defined as follows:

$$\text{MSE} = \frac{1}{\sigma_{\text{global}}^2} \frac{1}{H \times D} \sum_{i=1}^{H} \sum_{j=1}^{D} (Y_{i,j}^{\text{GT}} - Y_{i,j}^{\text{pred}})^2 \tag{7}$$

Here Y^{pred} denotes either the base or refined forecast, with i indexing forecast steps and j indexing channels. This approach ensures that the improvements achieved by ORCA are consistently comparable across all combinations of datasets and base models. Furthermore, because this normalization applies a constant scalar multiplier to both the base model and the adapted model's errors, the relative MSE drop ratio remains strictly identical before and after normalization. This guarantees that the reported performance improvements are solely attributed to our adaptation framework and not an artifact of the normalization process.

A.4 Adaptation of white-box and grey-box baselines to black-box settings

To rigorously benchmark our proposed framework, we selected state-of-the-art Test-Time Adaptation (TTA) methods originally designed for white-box or grey-box settings and adapted them to the black-box contract. This contract strictly prohibits backpropagation through the base forecaster and avoids computationally expensive re-evaluations of the base model. Below, we detail the adaptations made for each baseline to ensure a fair comparison while preserving their core mechanisms.

TAFAS. The original TAFAS framework introduces Periodicity-Aware Adaptation Scheduling (PAAS) alongside Input and Output Gated Calibration Modules (GCMs). It calibrates both the input context and the output predictions by continuously backpropagating through the source forecaster using partially observed ground truths. Under the black-box setting, we cannot backpropagate through the base forecaster or re-evaluate it with recalibrated inputs. Therefore, our adaptation retains only the Output GCM as a post-hoc residual calibrator. The core mechanism of the Output GCM—a variable-wise residual transformation regulated by a tanh gating mechanism initialized near zero—is fully preserved. Instead of PAAS, which relies on partial ground truth, the adapted TAFAS uses

the same fully resolved ground truth training snapshots and warmup scheduling as our proposed framework, ensuring it operates strictly on observable retrospective data.

SOLID. The Sample-level Contextualized Adapter (SOLID) constructs a contextualized dataset for each test sample by selecting historical instances with similar periodic phases and high Euclidean similarity in their look-back windows. It then fine-tunes the top linear prediction layer of the base forecaster using this contextualized dataset. Under the black-box contract, modifying the internal prediction layer of the base forecaster is strictly forbidden. Consequently, we replace the internal prediction layer fine-tuning with a lightweight post-hoc Affine Head (comprising scale and bias parameters) applied directly to the frozen base predictions. Our adaptation fully preserves SOLID's core context-selection mechanism based on phase proximity and look-back similarity. For each test instance, we select the top-k nearest historical neighbors from an online history pool. We then clone the globally trained Affine Head, perform local gradient descent adaptation using these contextualized neighbors, and apply the locally adapted head to the current prediction.

DSOF. The Dual-Stream Framework for Online Time Series Forecasting (DSOF) introduces a teacher-student architecture operating on two distinct data streams. The fast stream employs temporal difference (TD) learning with pseudo-labels to rapidly adapt to recent data, while the slow stream stabilizes training via experience replay (ER) using fully observed ground truth sequences. In its original formulation, both the teacher and student models are fine-tuned. To align with the black-box contract, our adaptation strictly freezes the teacher model (i.e., the base forecaster) and relies entirely on a lightweight residual student multilayer perceptron (MLP) to calibrate predictions. The dual-stream mechanism is preserved in its entirety. The slow stream utilizes the same fully resolved ground truth training snapshots from our online FIFO buffer for experience replay. Concurrently, the fast stream immediately updates the student model using pseudo-labels, which are constructed by concatenating the most recent partial ground truth observations with the frozen teacher model's predictions.

B Theoretical analysis

B.1 Context conditioned learning

In Section 3, we assert that the optimal deterministic adapter should learn the conditional expectation of the errors given the context $\mathbf{C}_t = [\mathbf{X}_t, \mathbf{Y}_t^{\text{base}}]$. Here, we provide the detailed mathematical derivation.

We define the expected risk $\mathcal{R}(f)$ under the Mean Squared Error (MSE) loss as the expected Frobenius norm of the difference between the true error matrix $\mathbf{E}_t$ and our adapter's prediction $f(\mathbf{C}_t)$:

$$\mathcal{R}(f) = \mathbb{E}_{\mathbf{C}_t, \mathbf{E}_t} \left[\|\mathbf{E}_t - f(\mathbf{C}_t)\|_F^2 \right] \tag{8}$$

According to the law of total expectation, this risk can be decomposed by conditioning on the observable context $\mathbf{C}_t$:

$$\mathcal{R}(f) = \mathbb{E}_{\mathbf{C}_t} \left[\mathbb{E}_{\mathbf{E}_t|\mathbf{C}_t} \left[\|\mathbf{E}_t - f(\mathbf{C}_t)\|_F^2 \mid \mathbf{C}_t \right] \right] \tag{9}$$

To find the optimal mapping f^* that minimizes the global expected risk $\mathcal{R}(f)$, it is sufficient to minimize the inner conditional expectation for every realization of the context $\mathbf{C}_t$. Let $\mathcal{J}(f(\mathbf{C}_t))$ denote this inner objective, which can be expanded over the matrix elements:

$$\mathcal{J}(f(\mathbf{C}_t)) = \mathbb{E}_{\mathbf{E}_t|\mathbf{C}_t} \left[\sum_{i,j} \left(E_{t,ij} - f_{ij}(\mathbf{C}_t) \right)^2 \,\middle|\, \mathbf{C}_t \right] \tag{10}$$

Since $f(\mathbf{C}_t)$ is a deterministic function of $\mathbf{C}_t$, we can take the partial derivative of $\mathcal{J}$ with respect to each output element $f_{ij}(\mathbf{C}_t)$ and set it to zero:

$$\frac{\partial \mathcal{J}}{\partial f_{ij}(\mathbf{C}_t)} = -2\, \mathbb{E}_{\mathbf{E}_t|\mathbf{C}_t} \left[E_{t,ij} - f_{ij}(\mathbf{C}_t) \mid \mathbf{C}_t \right] = 0 \tag{11}$$

Solving this yields the optimal element-wise mapping:

$$f_{ij}^*(\mathbf{C}_t) = \mathbb{E}_{\mathbf{E}_t|\mathbf{C}_t} \left[E_{t,ij} \mid \mathbf{C}_t \right] \tag{12}$$

Reconstructing the matrix form, we obtain the rigorously optimal mapping function:

$$f^*(\mathbf{C}_t) = \mathbb{E}[\mathbf{E}_t \mid \mathbf{X}_t, \mathbf{Y}_t^{\text{base}}] \tag{13}$$

While it is a well-known mathematical property that minimizing MSE targets the conditional mean, this derivation serves a specific structural purpose for ORCA. It rigorously justifies our use of the conditional formulation $P(\mathbf{E}_t \mid \mathbf{X}_t, \mathbf{Y}_t^{\text{base}})$ within a deterministic framework, demonstrating that adapting via a deterministic mapping $f(\mathbf{C}_t)$ is mathematically equivalent to estimating the expected value of the probabilistic error distribution. Consequently, to effectively perform this residual regression, it is naturally appropriate to condition the adapter on the concatenated context of $\mathbf{X}_t$ and $\mathbf{Y}_t^{\text{base}}$.

B.2 Predictive-space Bayesian update

In Section 3.5, we formulated our online adaptation as a predictive-space Bayesian update and substituted the exact precision ratio λ_t with the Boltzmann routing confidence $\bar{c}_t$. This section provides the comprehensive mathematical proof and theoretical justification.

B.2.1 Maximum a posteriori (MAP) derivation

At step t, we seek to estimate the true residual $\mathbf{E}_t$ given the context $\mathbf{C}_t = [\mathbf{X}_t, \mathbf{Y}_t^{\text{base}}]$, the historical memory $\mathcal{H}_{t-1}$, and the current observation $\mathbf{E}_{\text{obs},t}$. According to Bayes' theorem, the posterior distribution is:

$$P(\mathbf{E}_t \mid \mathbf{C}_t, \mathbf{E}_{\text{obs},t}, \mathcal{H}_{t-1}) \propto P(\mathbf{E}_{\text{obs},t} \mid \mathbf{E}_t, \mathbf{C}_t) \cdot P(\mathbf{E}_t \mid \mathbf{C}_t, \mathcal{H}_{t-1}) \tag{14}$$

Assuming both the likelihood and the prior follow isotropic Gaussian distributions:

$$\text{Likelihood: } P(\mathbf{E}_{\text{obs},t} \mid \mathbf{E}_t, \mathbf{C}_t) \propto \exp\left(-\frac{1}{2\sigma_{\text{obs}}^2}\|\mathbf{E}_t - \mathbf{E}_{\text{obs},t}\|_F^2\right) \tag{15}$$

$$\text{Prior: } P(\mathbf{E}_t \mid \mathbf{C}_t, \mathcal{H}_{t-1}) \propto \exp\left(-\frac{1}{2\sigma_{\text{prior}}^2}\|\mathbf{E}_t - \mathbf{E}_{\text{prior},t}\|_F^2\right) \tag{16}$$

To perform MAP estimation, we substitute the adapter's deterministic prediction $\Delta\hat{\mathbf{Y}}_t$ for $\mathbf{E}_t$ and minimize the negative logarithm of the posterior:

$$-\log P \propto \frac{1}{2\sigma_{\text{obs}}^2}\|\Delta\hat{\mathbf{Y}}_t - \mathbf{E}_{\text{obs},t}\|_F^2 + \frac{1}{2\sigma_{\text{prior}}^2}\|\Delta\hat{\mathbf{Y}}_t - \mathbf{E}_{\text{prior},t}\|_F^2 \tag{17}$$

Since $\mathbf{E}_{\text{obs},t} = \mathbf{Y}_t^{\text{GT}} - \mathbf{Y}_t^{\text{base}}$, we have $\Delta\hat{\mathbf{Y}}_t - \mathbf{E}_{\text{obs},t} = (\mathbf{Y}_t^{\text{base}} + \Delta\hat{\mathbf{Y}}_t) - \mathbf{Y}_t^{\text{GT}} = \mathbf{Y}_t^{\text{ada}} - \mathbf{Y}_t^{\text{GT}}$. Similarly, for the prior term, $\Delta\hat{\mathbf{Y}}_t - \mathbf{E}_{\text{prior},t} = \mathbf{Y}_t^{\text{ada}} - \mathbf{Y}_t^{\text{prior}}$. By absorbing $2\sigma_{\text{obs}}^2$ into the learning rate and dividing by the dimensions $H \times D$, we define the theoretical precision ratio $\lambda_t = \sigma_{\text{obs}}^2/\sigma_{\text{prior}}^2$. This rigorously transforms the error-space optimization into the final predictive-space time series loss:

$$\mathcal{L}(s_t, \theta) \propto \frac{1}{H \times D}\left(\|\mathbf{Y}_t^{\text{GT}} - \mathbf{Y}_t^{\text{ada}}\|_F^2 + \lambda_t\|\mathbf{Y}_t^{\text{ada}} - \mathbf{Y}_t^{\text{prior}}\|_F^2\right) \tag{18}$$

B.2.2 Theoretical justification of the precision surrogate

In online time series forecasting, the precise prior variance σ_{prior}^2 is unknown and highly non-stationary due to concept drifts. While rigorous estimation of the prior precision can be mathematically possible, it often introduces unacceptable computational overhead and numerical instability for real-time streaming adaptation. To maintain both efficiency and stability, ORCA dynamically substitutes the theoretical precision ratio λ_t with an empirical surrogate $\bar{c}_t = \text{mean}(\mathbf{c}_t)$. We validate its structural optimality by solving for the minimum of the substituted loss function $\mathcal{L}^* = \|\Delta\hat{\mathbf{Y}}_t - \mathbf{E}_{\text{obs},t}\|_F^2 + \bar{c}_t\|\Delta\hat{\mathbf{Y}}_t - \mathbf{E}_{\text{prior},t}\|_F^2$.

Taking the derivative with respect to $\Delta\hat{\mathbf{Y}}_t$ and setting it to zero yields:

$$\frac{\partial\mathcal{L}^*}{\partial\Delta\hat{\mathbf{Y}}_t} = 2(\Delta\hat{\mathbf{Y}}_t - \mathbf{E}_{\text{obs},t}) + 2\bar{c}_t(\Delta\hat{\mathbf{Y}}_t - \mathbf{E}_{\text{prior},t}) = 0 \tag{19}$$

Solving for the optimal adapter output $\Delta\hat{\mathbf{Y}}_t^*$ gives Equation (6). Let μ_{post}, μ_{obs}, and μ_{prior} denote the posterior, observation, and prior means, respectively. This formulation corresponds exactly to the analytical mean of the product of two Gaussian distributions: $\mu_{\text{post}} = \frac{\sigma_{\text{prior}}^2 \mu_{\text{obs}} + \sigma_{\text{obs}}^2 \mu_{\text{prior}}}{\sigma_{\text{obs}}^2 + \sigma_{\text{prior}}^2}$. Setting $\sigma_{\text{obs}}^2 = 1$ and surrogate variance $\tilde{\sigma}_{\text{prior}}^2 = 1/\bar{c}_t$ mathematically bridges our empirical formulation with the exact Bayesian posterior mean structure.

Furthermore, utilizing the Boltzmann routing confidence as a surrogate provides significant advantages in online stability. Unlike the theoretical precision ratio $\lambda_t \in [0, \infty)$, which can become unbounded and trigger gradient explosion under extreme distribution shifts, $\bar{c}_t \in (0, 1)$ ensures a strictly bounded loss landscape. Additionally, the exponential moving average (EMA) of absolute errors acts as a robust proxy for predictive variance, offering superior resilience against heavy-tailed outliers compared to traditional squared variance estimation. Therefore, $\bar{c}_t$ dynamically behaves as a regularized pre-conditioner that enables adaptive scaling without the burden of explicit variance tracking.

B.3 Boltzmann routing

B.3.1 Regret bound of Boltzmann routing

Beyond training, the Boltzmann Router manages the inference-stage integration of $\mathbf{Y}_t^{\text{base}}$ and $\mathbf{Y}_t^{\text{ada}}$ into the final output $\mathbf{Y}_t^{\text{comb}}$. This mechanism can be formally analyzed as an online *Prediction with Expert Advice* problem [Freund and Schapire, 1997].

Let $N = 2$ be the number of experts, corresponding to the base model (Expert 1) and the adapter (Expert 2). At each step t, the router assigns a normalized weight $\mathbf{c}_t \in [0, 1]^D$ to the adapter and $(\mathbf{1} - \mathbf{c}_t)$ to the base model. To maintain plasticity in drifting environments, ORCA utilizes an Exponential Moving Average (EMA) of errors:

$$\begin{cases} \hat{\varepsilon}_{t-1}^{\text{ada}} = \alpha \mathbf{e}_{t-1}^{\text{ada}} + (1-\alpha)\hat{\varepsilon}_{t-2}^{\text{ada}} \\ \hat{\varepsilon}_{t-1}^{\text{base}} = \alpha \mathbf{e}_{t-1}^{\text{base}} + (1-\alpha)\hat{\varepsilon}_{t-2}^{\text{base}} \end{cases} \tag{20}$$

where $\alpha \in (0, 1)$ is the momentum coefficient. The channel-wise routing confidence vector $\mathbf{c}_t \in \mathbb{R}^D$ is then calculated via a Boltzmann softmax function with temperature τ:

$$\mathbf{c}_t = \frac{\exp(-\hat{\varepsilon}_{t-1}^{\text{ada}}/\tau)}{\exp(-\hat{\varepsilon}_{t-1}^{\text{base}}/\tau) + \exp(-\hat{\varepsilon}_{t-1}^{\text{ada}}/\tau)} \tag{21}$$

To derive the regret bound, we analyze the mechanism for a single channel, dropping the channel index for brevity. Let $e_t^k \in [0, M]$ denote the instantaneous bounded error for expert $k \in \{\text{base}, \text{ada}\}$. Because standard error metrics (such as MSE) are convex, the loss of the combined prediction $\mathbf{Y}_t^{\text{comb}} = \mathbf{c}_t \odot \mathbf{Y}_t^{\text{ada}} + (\mathbf{1} - \mathbf{c}_t) \odot \mathbf{Y}_t^{\text{base}}$ is upper-bounded by the expected loss under the routing distribution:

$$\mathcal{L}_t(\mathbf{Y}_t^{\text{comb}}) \leq c_t e_t^{\text{ada}} + (1 - c_t) e_t^{\text{base}} \tag{22}$$

By expanding the recursive EMA formulation, the smoothed error is equivalent to a discounted sum of all past errors: $\hat{\varepsilon}_{t-1}^k = \alpha \sum_{s=1}^{t-1} (1-\alpha)^{t-1-s} e_s^k$. Substituting this into the Boltzmann softmax reveals that the routing probability is exactly proportional to:

$$p_t^k \propto \exp\left(-\frac{\alpha}{\tau} \sum_{s=1}^{t-1} (1-\alpha)^{t-1-s} e_s^k \right) \tag{23}$$

This formulation proves that our EMA-based Boltzmann routing is mathematically equivalent to the *Discounted Exponential Weights* (DEW) algorithm [Herbster and Warmuth, 1998], operating with a discount factor $\gamma = 1 - \alpha$ and an effective learning rate $\eta = \alpha/\tau$.

Let R_T denote the static regret over T steps against the best single expert in hindsight. The discounting restricts the algorithm's effective memory to approximately $1/\alpha$ steps. According to the standard theoretical analysis of DEW, the regret consists of the standard Exponential Weights bound over the effective window plus a tracking penalty bias proportional to the discount rate. Using Hoeffding's

lemma for bounded losses, the regret is bounded by:

$$R_T = \sum_{t=1}^{T} \mathcal{L}_t(\mathbf{Y}_t^{\text{comb}}) - \min_{k \in \{\text{base,ada}\}} \sum_{t=1}^{T} e_t^k \leq \frac{\ln 2}{\eta} + \frac{\eta}{8} T M^2 + \alpha T M \tag{24}$$

Substituting $\eta = \alpha/\tau$ back into the inequality, we obtain the bound for the Boltzmann Router:

$$R_T \leq \frac{\tau \ln 2}{\alpha} + \frac{\alpha T M^2}{8\tau} + \alpha T M = \frac{\tau \ln 2}{\alpha} + \alpha T \left(\frac{M^2}{8\tau} + M \right) \tag{25}$$

To achieve a sublinear regret, we balance the terms with respect to T. By applying the arithmetic–geometric mean (AM–GM) inequality, the minimum of this upper bound is reached when the two terms are equal, yielding the optimal momentum coefficient α^*:

$$\alpha^* = \sqrt{\frac{8\tau^2 \ln 2}{T(M^2 + 8\tau M)}} = \mathcal{O}\left(\frac{1}{\sqrt{T}} \right) \tag{26}$$

Substituting α^* back into the inequality, the regret bound becomes sublinear:

$$R_T \leq 2\sqrt{\tau \ln 2 \cdot T \left(\frac{M^2}{8\tau} + M \right)} = \mathcal{O}(\sqrt{T}) \tag{27}$$

Given $N = 2$, the logarithmic term $\ln 2$ is a constant. This bound $\mathcal{O}(\sqrt{T})$ guarantees that the time-averaged regret R_T/T converges to zero. In streaming scenarios where the horizon T is not known, one can employ a time-varying momentum $\alpha_t \propto 1/\sqrt{t}$ to maintain this property. Consequently, if a distribution shift breaks the linear adapter ($\mathbf{e}_t^{\text{ada}} \gg \mathbf{e}_t^{\text{base}}$), the Boltzmann routing mechanism ensures that $\mathbf{c}_t \to \mathbf{0}$, falling back to the base TSFM.

B.3.2 Hyperparameter sensitivity analysis of Boltzmann routing

The derived regret bound explicitly reveals the theoretical trade-offs governed by the two primary hyperparameters of the Boltzmann Router: the temperature τ and the Exponential Moving Average (EMA) momentum α. By analyzing the inequality $R_T \leq \frac{\tau \ln 2}{\alpha} + \frac{\alpha T M^2}{8\tau} + \alpha T M$, we can quantitatively evaluate the sensitivity of the adaptation mechanism.

Sensitivity to Temperature (τ): The temperature parameter dictates the strictness of the routing distribution. In the regret bound, τ presents a clear dichotomy. A larger τ inflates the first term $\frac{\tau \ln 2}{\alpha}$, which represents the penalty of slow convergence toward the optimal expert. Physically, a high temperature leads to a more uniform mixing of $\mathbf{Y}_t^{\text{base}}$ and $\mathbf{Y}_t^{\text{ada}}$, providing stability but diluting the adapter's corrective potential. Conversely, a smaller τ minimizes this initialization penalty but heavily inflates the variance term $\frac{\alpha T M^2}{8\tau}$. A low temperature forces the router to act greedily, making it highly sensitive to instantaneous noise and prone to oscillating drastically between the base model and the adapter.

Sensitivity to EMA Momentum (α): The momentum parameter controls the effective memory length of the router, which is approximately proportional to $1/\alpha$. A small α implies a long memory, which effectively suppresses the tracking penalties $\frac{\alpha T M^2}{8\tau}$ and $\alpha T M$, resulting in a highly stable routing trajectory that is robust to outlier errors. However, a heavily smoothed error drastically increases the first term $\frac{\tau \ln 2}{\alpha}$, causing a delayed response. If a sudden concept drift occurs, a small α prevents the router from rapidly shedding its historical confidence in a failing adapter. On the other hand, a large α allows for swift adaptation to recent shifts but exposes the router to high variance, potentially causing it to overreact to stochastic noise rather than genuine distribution changes.

Consequently, achieving optimal online adaptation requires a delicate balance between τ and α. The mathematical boundary demonstrates that while the Boltzmann Router is fundamentally robust, adjusting these hyperparameters allows the adaptation to be perfectly tailored to the specific non-stationary dynamics and noise levels of the target streaming environment. Based on this analysis, we empirically set our default parameters to $\tau = 0.1$ and $\alpha = 0.2$. An EMA momentum of $\alpha = 0.2$ provides a half-life of approximately three to four steps, which is perfectly suited for tracking

Table 3: Ablation study on the routing mechanism: Relative MSE drop (%) using a naive Hard Router instead of the proposed Boltzmann Router ($H = 96$).

Model	Chronos-2	Moirai-2	TiRex	TimesFM-2.5	Sundial	Datasets Avg.
ETTh1	-4.40%	-2.00%	0.00%	0.70%	3.60%	-0.40%
ETTh2	-1.80%	0.60%	1.80%	-0.60%	-0.40%	-0.10%
ETTm1	-4.70%	-11.30%	-11.60%	-3.60%	0.70%	-6.10%
ETTm2	-1.40%	-8.30%	-11.40%	-8.80%	-3.40%	-6.70%
Exchange	-2.70%	1.00%	-7.20%	-2.20%	-9.50%	-4.10%
Weather	-2.30%	-12.90%	-7.80%	-5.50%	2.30%	-5.20%
Models Avg.	-2.90%	-5.50%	-6.10%	-3.30%	-1.10%	-3.80%

the high-frequency concept drifts typical in streaming time series while smoothing out immediate stochastic noise. Simultaneously, a routing temperature of $\tau = 0.1$ produces a sharp yet differentiable softmax distribution. This ensures that when the adapter's error significantly deviates from the base model's, the router swiftly shifts its confidence toward the superior forecaster, preventing negative optimization, while still allowing a probabilistic blend when their performances are comparable. Our numerical analysis in Figure 7 further corroborates that this optimal region is quite broad, making the model practically insensitive to minor deviations around these default values.

C Supplementary experiment results

C.1 A router variant compared to the Boltzmann router

To further validate the necessity of the proposed Boltzmann routing mechanism, we conduct an additional ablation study by replacing it with a naive Hard Router. Under the identical experimental configuration with a forecasting horizon of $H = 96$, the Hard Router retains the exact same Exponential Moving Average (EMA) tracking rules and hyperparameters, but strictly outputs either the base model's prediction or the adapted prediction based solely on whichever has the lower historical smoothed error. As demonstrated in Table 3, this rigid binary selection strategy yields an overall average MSE reduction of only 3.80%. This performance is markedly inferior to the results achieved by our proposed Boltzmann Router. The comparison highlights that a hard switching mechanism is overly sensitive to instantaneous noise and local distribution shifts. In contrast, the Boltzmann Router provides a soft, probabilistic mixing mechanism that more safely and effectively harnesses the corrective potential of the online adapter.

C.2 Horizon-wise performance breakdown

In this section, we present the detailed forecasting performance of ORCA and the zero-shot base models broken down by individual forecasting horizons: $H = 30$ (Table 4), $H = 96$ (Table 5), and $H = 336$ (Table 6). In the tables of the appendix, the following abbreviations are used: Van. for Vanilla, Ref. for Refined, Exc. for Exchange, Elc. for Electricity, and Wea. for Weather.

As demonstrated across the three tables, ORCA achieves consistent and robust MSE reductions regardless of the forecasting length. Importantly, there is no pronounced bias or trend indicating that the adaptation is disproportionately effective for only short or long horizons. This uniform stability can be directly attributed to our online cycle training scheme. Since the training cadence is dynamically linked to the horizon (i.e., triggered every H steps when the full ground truth becomes observable), the learning regime scales naturally. Short-horizon forecasts trigger frequent, rapid updates to capture fast-evolving concept drifts, whereas long-horizon forecasts accumulate broader contexts before executing more comprehensive updates. Consequently, the adapter maintains its plasticity and calibration capacity optimally tailored to the intrinsic frequency of the targeted horizon.

Table 4: Performance breakdown for forecasting horizon $H = 30$.

Model		Chronos-2		Moirai-2		TiRex		TimesFM-2.5		Sundial		Avg.
ETTh1	Van.	0.2086		0.21		0.2099		0.2201		0.1903		
	Ref.	0.2062	-1.20%	0.2066	-1.60%	0.2084	-0.70%	0.2143	-2.60%	0.1979	4.00%	-0.40%
ETTh2	Van	0.0228		0.022		0.0222		0.0233		0.022		
	Ref.	0.0225	-1.30%	0.022	0.00%	0.0223	0.50%	0.023	-1.30%	0.0221	0.50%	-0.30%
ETTm1	Van	0.1365		0.1675		0.1696		0.1445		0.1371		
	Ref.	0.1325	-2.90%	0.1495	-10.70%	0.1509	-11.00%	0.138	-4.50%	0.1344	-2.00%	-6.20%
ETTm2	Van	0.0131		0.0143		0.0141		0.0141		0.0134		
	Ref.	0.0127	-3.10%	0.0132	-7.70%	0.0132	-6.40%	0.0133	-5.70%	0.013	-3.00%	-5.20%
Exc.	Van	0.0007		0.0007		0.0007		0.0007		0.0008		
	Ref.	0.0007	0.00%	0.0007	0.00%	0.0007	0.00%	0.0007	0.00%	0.0008	0.00%	0.00%
Wea.	Van	0.039		0.0433		0.0448		0.0449		0.0353		
	Ref.	0.0365	-6.40%	0.0376	-13.20%	0.0395	-11.80%	0.0391	-12.90%	0.0351	-0.60%	-9.00%
Elc.	Van	0.0457		0.0469		0.0444		0.0475		0.0382		
	Ref.	0.041	-10.30%	0.0421	-10.30%	0.04	-9.80%	0.0428	-9.90%	0.0349	-8.60%	-9.80%
Traffic	Van	0.1982		0.2005		0.2206		0.2038		0.2294		
	Ref.	0.1947	-1.70%	0.1975	-1.50%	0.2112	-4.30%	0.2011	-1.30%	0.2162	-5.70%	-2.90%
Models Avg.		-3.40%		-5.60%		-5.40%		-4.80%		-1.90%		-4.20%

C.3 Statistical baselines: setting and results

ETS. The ETS adapter is a lightweight channel-wise Holt-style smoother that operates on the residual sequence, rather than on the input-output context, because the classic ETS often requires that its input and output be identical sequences. At each update, it first forms the residual by subtracting the median base forecast from the ground truth, then clips extreme impulses with a residual scale factor, and updates a level-trend state with smoothing coefficients. In our final configuration, the effective settings are level and trend smoothing coefficients $\alpha = 0.3$ and $\beta = 0.03$, damping factor 0.98, maximum residual history length 12000, gain sensitivity 1.0, residual clipping scale 3.0, minimum history for gating 5, and warm-up steps 3. The prediction stage extrapolates the residual state forward and adds it back to the base forecast, with a stability gate computed from recent residual variability. This design is intentionally simple and fast, but it also assumes that the residual process is approximately homogeneous over time. Table 7 shows that this assumption is too restrictive for black-box online adaptation: ETS is consistently weak on average, with a dataset-averaged score of 126.30%, and it degrades substantially on ETTh2, ETTm1, ETTm2, and Exchange. These results suggest that modeling residuals in isolation is not sufficient when the error dynamics are nonstationary.

Ridge Regression. The Ridge adapter follows a different philosophy: instead of modeling the residual sequence alone, it learns a correction from the concatenation of the past input window and the base model output along the feature dimension. In other words, the target is still the residual correction, but the predictor is conditioned on the XY context, which makes it a direct implementation of our learning hypothesis that adaptation should learn the context of errors rather than the residual process in isolation. The final implementation uses closed-form ridge regression with ridge penalty $\lambda = 10^{-3}$, collects 512 training windows before the first fit, refits once every horizon cycle, and keeps the same lightweight robustness mechanism as a safeguard, including residual-history gating, clipping, and warm-up control. Concretely, the runtime defaults are a maximum history length of 2048, gain sensitivity 1.0, residual clipping scale 3.0, minimum gating history 5, and warm-up steps 10. Table 8 shows that this XY-conditioned formulation is much stronger than ETS: Ridge reduces the dataset-averaged score to 68.78%, which is far better than ETS's 126.30%, and it achieves especially strong improvements on ETTh1 to ETTm2, where several backbones even obtain negative relative scores. The remaining failures on Exchange and Weather indicate that a purely linear model is still limited, but the overall pattern strongly supports the claim that conditioning on the main model's

Table 5: Performance breakdown for forecasting horizon $H = 96$.

Model		Chronos-2		Moirai-2		TiRex		TimesFM-2.5		Sundial		Avg.
ETTh1	Van.	0.2942		0.2789		0.2777		0.2913		0.2467		
	Ref.	0.2731	-7.20%	0.2703	-3.10%	0.2684	-3.30%	0.2741	-5.90%	0.2548	3.30%	-3.20%
ETTh2	Van.	0.0373		0.0345		0.0351		0.0368		0.0347		
	Ref.	0.0363	-2.70%	0.0345	0.00%	0.0351	0.00%	0.0358	-2.70%	0.035	0.90%	-0.90%
ETTm1	Van.	0.2123		0.2428		0.2461		0.2132		0.2051		
	Ref.	0.1982	-6.60%	0.207	-14.70%	0.2096	-14.80%	0.1987	-6.80%	0.1957	-4.60%	-9.50%
ETTm2	Van.	0.0224		0.025		0.0246		0.0242		0.0229		
	Ref.	0.0209	-6.70%	0.0217	-13.20%	0.0216	-12.20%	0.0217	-10.30%	0.0214	-6.60%	-9.80%
Exc.	Van.	0.0019		0.0022		0.0021		0.0021		0.0027		
	Ref.	0.002	5.30%	0.0023	4.50%	0.0019	-9.50%	0.0021	0.00%	0.0024	-11.10%	-2.20%
Wea.	Van.	0.0486		0.0603		0.0556		0.0522		0.042		
	Ref.	0.0445	-8.40%	0.0469	-22.20%	0.0479	-13.80%	0.0459	-12.10%	0.0417	-0.70%	-11.50%
Elc.	Van.	0.0564		0.0569		0.054		0.0575		0.0479		
	Ref.	0.0511	-9.50%	0.0519	-8.70%	0.0497	-8.00%	0.0523	-9.10%	0.0448	-6.40%	-8.30%
Traffic	Van.	0.2255		0.2273		0.2512		0.2257		0.2609		
	Ref.	0.2224	-1.30%	0.2255	-0.80%	0.2423	-3.50%	0.2248	-0.40%	0.247	-5.30%	-2.30%
Models Avg.		-4.60%		-7.30%		-8.20%		-5.90%		-3.80%		-6.00%

XY-aligned context is more informative than predicting residuals from residuals alone. We assume that the successful error reduction on the ETT series comes from the periodic physical nature of electricity transformers. When it comes to more complicated systems like Exchange (based on society and economics) and Weather (a complex natural system), the Ridge regression is incapable.

Table 6: Performance breakdown for forecasting horizon $H = 336$.

Model		Chronos-2		Moirai-2		TiRex		TimesFM-2.5		Sundial		Avg.
ETTh1	Van.	0.3239		0.3069		0.3138		0.32		0.2774		
	Ref.	0.3023	-6.70%	0.2969	-3.30%	0.2968	-5.40%	0.3019	-5.70%	0.2854	2.90%	-3.60%
ETTh2	Van.	0.0518		0.0485		0.0498		0.0507		0.0508		
	Ref.	0.0511	-1.40%	0.0499	2.90%	0.0492	-1.20%	0.051	0.60%	0.051	0.40%	0.30%
ETTm1	Van.	0.3103		0.351		0.3536		0.311		0.2887		
	Ref.	0.2808	-9.50%	0.2927	-16.60%	0.2841	-19.70%	0.2793	-10.20%	0.2741	-5.10%	-12.20%
ETTm2	Van.	0.038		0.0429		0.0401		0.0396		0.038		
	Ref.	0.0335	-11.80%	0.0347	-19.10%	0.035	-12.70%	0.0344	-13.10%	0.0351	-7.60%	-12.90%
Exc.	Van.	0.0067		0.0071		0.0078		0.007		0.009		
	Ref.	0.0066	-1.50%	0.0072	1.40%	0.0071	-9.00%	0.0067	-4.30%	0.008	-11.10%	-4.90%
Wea.	Van.	0.0556		0.066		0.0622		0.0521		0.0473		
	Ref.	0.0494	-11.20%	0.0492	-25.50%	0.0516	-17.00%	0.0474	-9.00%	0.0457	-3.40%	-13.20%
Elc.	Van.	0.0693		0.0692		0.0664		0.0687		0.0593		
	Ref.	0.0631	-8.90%	0.0625	-9.70%	0.061	-8.10%	0.0624	-9.20%	0.0563	-5.10%	-8.20%
Traffic	Van.	0.2479		0.246		0.2666		0.2361		0.2789		
	Ref.	0.2433	-1.90%	0.2421	-1.60%	0.2566	-3.80%	0.2346	-0.60%	0.2642	-5.30%	-2.60%
Models Avg.		-6.60%		-8.90%		-9.60%		-6.40%		-4.30%		-7.20%

Table 7: ETS results for residual-to-residual online adaptation across backbones and datasets. The results are averaged over three forecasting horizons ($H \in \{30, 96, 336\}$). A negative percentage indicates a reduction in MSE, meaning positive performance improvement.

Model	Chronos-2	Moirai-2	TiRex	TimesFM-2.5	Sundial	Datasets Avg.
ETTh1	109.30%	100.60%	104.60%	99.60%	97.20%	102.26%
ETTh2	138.60%	135.60%	112.60%	120.60%	94.30%	120.34%
ETTm1	156.60%	156.80%	148.60%	159.10%	142.40%	152.70%
ETTm2	174.70%	174.60%	145.70%	156.00%	122.10%	154.62%
Exchange	169.90%	201.00%	142.50%	167.30%	65.60%	149.26%
Weather	95.50%	78.90%	74.70%	69.10%	74.80%	78.60%
Models Avg.	140.77%	141.25%	121.45%	128.62%	99.40%	126.30%

Table 8: Ridge regression results for XY-conditioned residual correction across backbones and datasets. The results are averaged over three forecasting horizons ($H \in \{30, 96, 336\}$). A negative percentage indicates a reduction in MSE, meaning positive performance improvement.

Model	Chronos-2	Moirai-2	TiRex	TimesFM-2.5	Sundial	Datasets Avg.
ETTh1	-3.00%	-4.40%	-2.70%	-2.00%	-0.10%	-2.44%
ETTh2	-3.70%	-3.20%	-1.80%	-4.20%	0.60%	-2.46%
ETTm1	-4.70%	-7.80%	-5.80%	-3.50%	-2.90%	-4.94%
ETTm2	-5.70%	-9.70%	-6.70%	-6.00%	-2.70%	-6.16%
Exc.	137.80%	122.00%	62.00%	87.90%	162.70%	114.48%
Wea.	41.30%	371.30%	573.40%	382.60%	202.30%	314.18%
Models Avg.	27.00%	78.03%	103.07%	75.80%	59.98%	68.78%

D Additional empirical analyses

D.1 Multi-seed reliability

Table 9 summarizes seeds 42, 43, and 44 over three horizons, eight datasets, and five TSFMs (360 runs). After averaging horizons, we report the mean and standard deviation (SD) of relative MSE changes across seeds; negative means indicate improvements, and SD is measured in percentage points. Avg. entries average cell means and SDs separately. Variability is small except for larger fluctuations on Exchange.

Table 9: Relative MSE change across three seeds, expressed as $(\text{mean} \pm \text{SD})\%$. All entries use percentage units.

Dataset	Chronos-2	Moirai-2	TiRex	TimesFM-2.5	Sundial	Avg.
ETTh1	(-5.5 ± 0.05)	(-2.8 ± 0.04)	(-3.5 ± 0.05)	(-4.9 ± 0.06)	(3.3 ± 0.12)	(-2.68 ± 0.06)
ETTh2	(-1.8 ± 0.10)	(1.1 ± 0.21)	(-0.5 ± 0.11)	(-1.1 ± 0.36)	(0.6 ± 0.07)	(-0.34 ± 0.17)
ETTm1	(-7.2 ± 0.06)	(-14.6 ± 0.18)	(-16.2 ± 0.05)	(-7.9 ± 0.06)	(-4.1 ± 0.32)	(-10.00 ± 0.13)
ETTm2	(-8.6 ± 0.25)	(-15.4 ± 0.10)	(-11.3 ± 0.11)	(-10.9 ± 0.15)	(-6.5 ± 0.00)	(-10.54 ± 0.12)
Exchange	(-0.0 ± 0.00)	(1.5 ± 1.46)	(-9.1 ± 0.99)	(-4.2 ± 1.42)	(-9.9 ± 0.57)	(-4.34 ± 0.89)
Weather	(-8.9 ± 0.00)	(-21.2 ± 0.05)	(-14.5 ± 0.00)	(-11.3 ± 0.00)	(-1.7 ± 0.00)	(-11.52 ± 0.01)
Electricity	(-9.5 ± 0.00)	(-9.5 ± 0.00)	(-8.5 ± 0.05)	(-9.4 ± 0.00)	(-6.4 ± 0.05)	(-8.66 ± 0.02)
Traffic	(-1.7 ± 0.00)	(-1.3 ± 0.00)	(-3.8 ± 0.05)	(-0.8 ± 0.00)	(-5.4 ± 0.00)	(-2.60 ± 0.01)
Models Avg.	(-5.40 ± 0.06)	(-7.78 ± 0.26)	(-8.43 ± 0.18)	(-6.31 ± 0.26)	(-3.76 ± 0.14)	(-6.34 ± 0.18)

D.2 Full-shot forecasting baselines

DLinear [Zeng et al., 2023], PatchTST [Nie et al., 2022], and iTransformer [Liu et al., 2023] use model configurations and training hyperparameters from the Time Series Library (TSLib) [THUML, 2023]. We train and validate chronologically with context length 96, seed 42, and reversible instance normalization (RevIN); $H = 30$ uses the $H = 96$ hyperparameters. Testing follows ORCA's windows, normalization, and metrics. Table 10 reports the results. Their overall MSEs are 0.1509, 0.1268, and 0.1246, respectively; the five TSFMs average 0.1128 (range 0.1074–0.1180), or 0.1063 (0.1049–0.1085) with ORCA.

Table 10: Full-shot baseline MSE (seed 42). Exc., Wea., and Elc. denote Exchange, Weather, and Electricity. Horizon averages are computed before rounding.

Model	ETTh1	ETTh2	ETTm1	ETTm2	Exc.	Wea.	Elc.	Traffic
				$H = 30$				
DLinear	0.231	0.024	0.164	0.016	0.001	0.089	0.088	0.447
PatchTST	0.205	0.025	0.145	0.014	0.001	0.088	0.093	0.255
iTransformer	0.211	0.025	0.159	0.015	0.001	0.080	0.084	0.220
				$H = 96$				
DLinear	0.296	0.036	0.220	0.023	0.002	0.078	0.097	0.450
PatchTST	0.265	0.038	0.208	0.023	0.002	0.076	0.126	0.285
iTransformer	0.276	0.038	0.225	0.025	0.002	0.077	0.101	0.251
				$H = 336$				
DLinear	0.333	0.052	0.336	0.037	0.006	0.080	0.099	0.415
PatchTST	0.302	0.053	0.303	0.037	0.006	0.079	0.117	0.298
iTransformer	0.322	0.054	0.320	0.040	0.006	0.079	0.112	0.268
				Three-horizon average				
DLinear	0.2866	0.0376	0.2399	0.0254	0.0029	0.0823	0.0947	0.4374
PatchTST	0.2574	0.0385	0.2188	0.0247	0.0029	0.0809	0.1122	0.2792
iTransformer	0.2695	0.0388	0.2346	0.0267	0.0029	0.0784	0.0991	0.2463

D.3 Bayesian regularization diagnostics

Local adaptation cost. We compare Bayesian and prior-free updates at $H = 96$, seed 42, on six datasets and five TSFMs. High-change and level-anomaly points are the channel-wise top 10% of absolute ground-truth first differences and median-absolute-deviation-normalized departures from the median, respectively. Table 11 gives relative MSE changes from plain to Bayesian updates, computed after averaging MSE over the corresponding configurations; positive values indicate a cost.

Table 11: Bayesian regularization cost on selected difficult points ($H = 96$, seed 42). MSE changes are in percent.

Metric	ETTh1	ETTh2	ETTm1	ETTm2	Exchange	Weather	Overall
High-change MSE Δ	+0.20	-0.03	-0.51	+0.28	+0.89	+0.03	-0.09
Level-anomaly MSE Δ	+0.25	+0.33	-0.29	+1.36	+0.30	+0.06	+0.07

Average local effects are small but vary across datasets.

Directed-prior control. Following RegPull in AliO [Park et al., 2025], we apply the prior penalty only where the frozen prior predicts the observed target more accurately than the current adapter. Table 12 compares six datasets, five TSFMs, and $H \in \{30, 96, 336\}$ (90 configurations), with other settings unchanged. MSE is averaged over horizons before computing relative reductions.

Table 12: Relative MSE reduction (%) with the Bayesian and RegPull-directed priors over three horizons and six datasets. Larger values are better; differences are in percentage points (pp).

Method	ETTh1	ETTh2	ETTm1	ETTm2	Exchange	Weather	Overall
ORCA (Bayesian prior)	2.68	0.26	10.04	10.56	4.00	11.52	6.51
RegPull-prior	3.01	1.04	9.03	8.56	3.00	11.11	5.96
Difference (pp)	+0.33	+0.78	-1.01	-2.00	-1.00	-0.41	-0.55

The directed prior improves ETTh1 and ETTh2 but lowers the overall gain from 6.51% to 5.96%, supporting the original Bayesian prior. Aggregate MSE alone does not exclude all local over-smoothing effects.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize ORCA as a black-box online adaptation framework, and the stated contributions match the method, theory, and experiments reported in the main paper and appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes an explicit Limitations section, noting that ORCA depends on the base model producing reasonable forecasts and that future work should examine abrupt structural shifts and probabilistic metrics.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper provides full derivations and proofs in the appendix for the context-conditioned learning result, the predictive-space Bayesian update, and the Boltzmann routing regret bound.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The appendix discloses the datasets, base models, horizons, split ratio, optimizer, buffer settings, router hyperparameters, and evaluation protocol needed to reproduce the main experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is available at `https://github.com/Fifthky/ORCA`. The appendix describes the datasets, training, evaluation, and hyperparameter settings used for reproduction.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the chronological 7:1:2 split, look-back window, horizons, optimizer, learning rate, buffer size, batch size, warmup, and router settings, which are the key details needed to understand the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Table 9 reports means and standard deviations across seeds 42, 43, and 44 for three horizons, eight datasets, and five TSFMs (360 runs). Standard deviations quantify seed variability in percentage points; they are not confidence intervals. Appendix D.1 defines the aggregation.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper provides an efficiency table for a subset of settings, but it does not give complete compute-resource details for all experiments such as total runtime and hardware requirements across the full benchmark suite.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work studies online forecasting adaptation using public benchmark datasets and does not involve human-subject experiments or collection of personal data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on forecasting methodology and benchmark evaluation rather than a specific deployment, and does not provide a detailed analysis of potential societal benefits and harms in downstream applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models or data that require special misuse safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: The paper cites the datasets and base models used, but it does not explicitly list license names or terms of use for the existing assets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: ORCA's method and experimental settings are documented in the paper and appendix. The accompanying implementation is available at `https://github.com/Fifthky/ORCA`.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method does not use LLMs as an important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.