Towards Principled Test-Time Adaptation for Time Series Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Test-time adaptation (TTA) has recently emerged as a promising approach for improving time series forecasting (TSF) under distribution shift. Existing TSF-TTA methods utilize revealed targets in different ways, leading to heterogeneous adaptation protocols that lack a clearly unified formulation. To address this issue, we revisit TSF-TTA through the lens of protocol cleanliness and introduce a principled adaptation protocol based solely on matured ground truth. Under this protocol, we analyze existing adapters in the frequency domain and reveal that their prediction corrections typically exhibit limited and weakly structured spectral modifications. Driven by this observation, we propose Frequency-Aware Calibration (FAC), a lightweight calibration method that directly parameterizes prediction corrections in the frequency domain. Across diverse datasets, forecasting horizons, and source forecasters, FAC achieves competitive and consistent performance while requiring substantially fewer trainable parameters than the compared TSF-TTA adapters.

1 Introduction

Time series forecasting (TSF) plays a critical role in a wide range of domains, including weather (Wu et al., 2023; Verma et al., 2024), energy (Maleki et al., 2024), traffic (Yin et al., 2022; Zhang et al., 2024), and finance (Tsay, 2010; Koa et al., 2026). Recent advances in deep learning-based architectures have substantially enhanced predictive performance across diverse time series forecasting settings (Zeng et al., 2023; Nie et al., 2023; Liu et al., 2024).

Despite this progress, deploying time series forecasters to real-world applications remains challenging due to the complex and evolving nature of temporal data. In particular, distribution shifts induced by non-stationarity pose a major obstacle to reliable forecasting (Granger, 2003; Du et al., 2021), especially in long-horizon settings. To address this issue, prior studies have mainly pursued two paradigms: introducing learnable normalization modules (Kim et al., 2022; Fan et al., 2023; Liu et al., 2023), which primarily operate during pre-training, or training forecasting models in an online manner (Pham et al., 2023; Zhang et al., 2023; Liang et al., 2024; Lau et al., 2025; Huang et al., 2026). While these approaches can partially mitigate the effects of distribution shift, they either focus largely on the training phase or require training an online forecaster from scratch. In light of these limitations, applying test-time adaptation (TTA) (Kim et al., 2025; Medeiros et al., 2025; Grover & Etemad, 2025; Im & Kwon, 2026) to a source forecaster has emerged as a promising alternative.

Existing TSF-TTA methods differ in how they utilize revealed targets, with some (Kim et al., 2025; Medeiros et al., 2025; Grover & Etemad, 2025) incorporating partially-observed ground truth (POGT) together with full ground truth, which we refer to as **matured ground truth** throughout this paper, to enable more proactive adaptation. At the mini-batch level, a past mini-batch is considered as matured when the full target horizon for every sample in that mini-batch has been completely observed. Intuitively, with respect to the current batch, matured supervision lies entirely to the left of the end of the first look-back window in the current mini-batch. In contrast, COSA (Im & Kwon, 2026) is presented under a streaming formulation based on revealed ground truth, where an output adapter is updated on the current mini-batch using targets revealed for that mini-batch and lightweight context constructed from previously observed targets, without

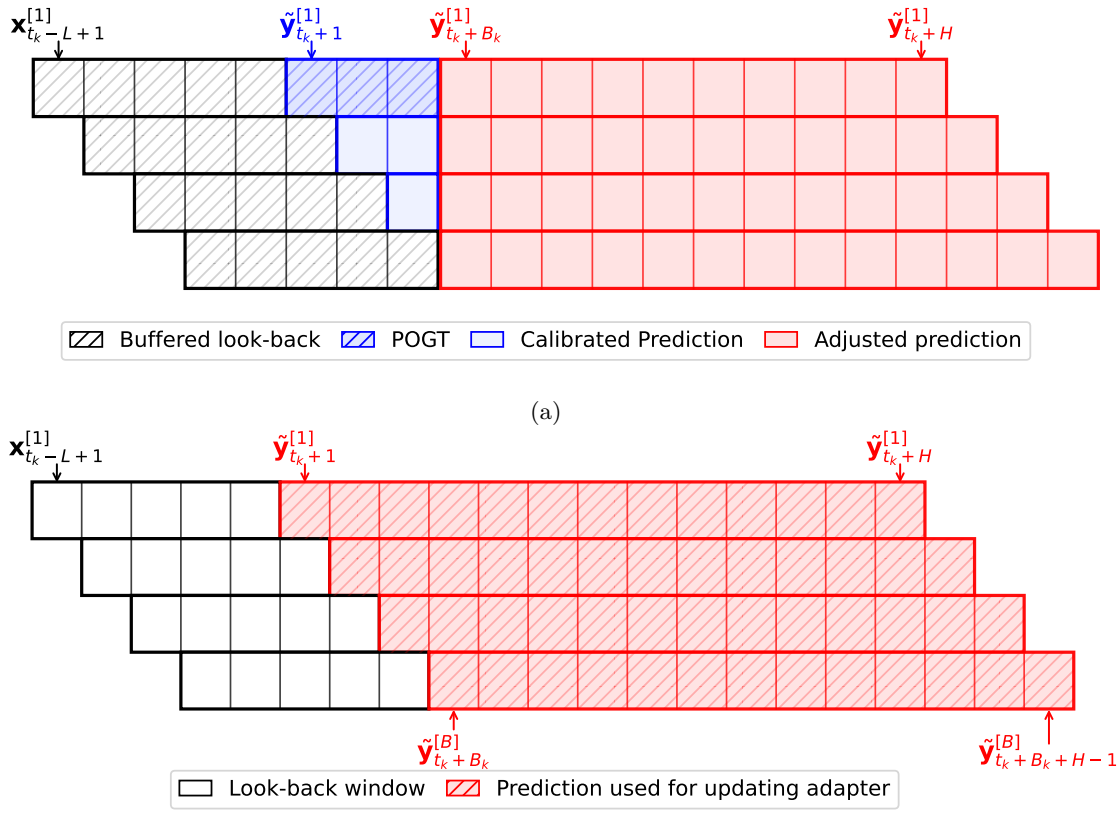

Figure 1: Comparison of adaptation supervision protocols in TSF-TTA. (a) Mixed supervision uses POGT from the current mini-batch, shown in blue, together with matured ground truth from a past mini-batch; the latter is omitted for visual clarity. (b) Streaming adaptation updates the adapter using revealed targets associated with the current mini-batch, indicated by the red shaded region.

being explicitly framed in terms of matured ground truth. As illustrated in Figure 1, the key distinction between the two families lies in the timing and manner in which target observations are used for adaptation.

Despite these design differences, the protocol-level role of revealed targets in TSF-TTA remains insufficiently clarified. Our analysis demonstrates that existing TSF-TTA protocols lack a clearly unified account of how revealed targets should be used for adaptation. This ambiguity motivates us to revisit TSF-TTA from the perspective of protocol cleanliness. In particular, we advocate using matured ground truth as a cleaner and more principled adaptation signal in rolling forecasting settings. We further examine the prediction corrections of existing TSF-TTA methods in the frequency domain, finding that they often exhibit relatively smooth and weakly structured spectral patterns despite their different calibration designs. Based on this observation, we propose Frequency-Aware Calibration (FAC), which achieves competitive and consistent performance under the proposed protocol across diverse forecasting settings.

Our contributions are summarized as follows:

- We revisit how revealed targets are used in TSF-TTA and show that a cleaner protocol based on matured ground truth provides a more principled setting for adaptation.
- We provide a frequency-domain diagnosis of existing TSF-TTA methods, showing that their realized corrections often exhibit limited and weakly structured spectral modifications.
- We propose Frequency-Aware Calibration (FAC), a lightweight adapter that explicitly models structured spectral residuals in the frequency domain and achieves competitive and consistent performance across diverse forecasting settings.

2 Related Work

2.1 Time Series Forecasting under Distribution Shift

A representative line of work addresses distribution shift in time series forecasting through learnable normalization or related distribution calibration mechanisms. RevIN (Kim et al., 2022) mitigates temporal distribution shift via reversible instance normalization and denormalization with learnable affine transformation. Dish-TS (Fan et al., 2023) characterizes both intra-space shift and inter-space shift, alleviating them through dual distribution modeling for the input and output spaces. SAN (Liu et al., 2023) further models non-stationarity by capturing statistical variation at the temporal-slice level. Although effective, these methods primarily improve robustness through training-time or architecture-level treatment of non-stationarity, rather than adapting a deployed source forecaster at test time.

Another line of work (Pham et al., 2023; Zhang et al., 2023; Liang et al., 2024; Lau et al., 2025; Huang et al., 2026) considers online time series forecasting (OTSF), where the forecasting model is continuously updated as new observations arrive. Recent studies (Liang et al., 2024; Lau et al., 2025) have pointed out that conventional OTSF protocols may suffer from evaluation leakage or unrealistic forecasting setups. In particular, Act-Now (Liang et al., 2024) and DSOF (Lau et al., 2025) revisit online forecasting protocols and advocate cleaner evaluation settings that avoid evaluating predictions on steps already used for model updates. Within this line of work, Act-Now (Liang et al., 2024) introduces a label decomposition model to tackle distribution shift, while DSOF (Lau et al., 2025) adopts a dual-stream teacher-student framework for online forecasting. ADAPT-Z (Huang et al., 2026) further studies the challenge of delayed-feedback online prediction and argues that distribution shift may be better addressed through feature-level adjustment. These approaches are related to TSF-TTA in spirit, but the primary objectives differ: OTSF focuses on training the online forecaster itself, while TSF-TTA adapts a frozen source forecaster at test time.

2.2 Test-Time Adaptation for Time Series Forecasting

Test-time adaptation (TTA) (Wang et al., 2021; 2022; Gong et al., 2023; Liang et al., 2025; Kim et al., 2026) has been widely studied in domains such as computer vision, where models are adapted at test time to improve robustness under distribution shift. Unlike conventional TTA methods (Wang et al., 2021; 2022; Gong et al., 2023; Kim et al., 2026), which typically rely on unlabeled test-time data, TSF-TTA can exploit forecasting targets that become sequentially revealed after prediction. As the pioneering work on TSF-TTA, TAFAS (Kim et al., 2025) incorporates both matured ground truth and partially observed ground-truth (POGT), aiming to enable earlier and more proactive adaptation. PETSA (Medeiros et al., 2025) further enhances parameter efficiency by introducing low-rank adapters together with a specialized adaptation loss. DynaTTA (Grover & Etemad, 2025) employs dynamic adaptation rates and shift-conditioned gating based on real-time estimation of distribution shift. In contrast, COSA (Im & Kwon, 2026) is presented under a streaming formulation based on revealed ground truth, where a single output adapter corrects the current mini-batch prediction using lightweight context from previously revealed targets and is updated directly against the targets of the current mini-batch. These methods therefore differ not only in adapter design, but also in when and how revealed targets are incorporated into adaptation.

3 Towards Cleaner Adaptation Protocols for TSF-TTA

3.1 Preliminaries: Rolling TSF-TTA Formulation

Existing TSF-TTA (Kim et al., 2025; Medeiros et al., 2025; Grover & Etemad, 2025; Im & Kwon, 2026) methods adopt a rolling test-time adaptation setting, in which a pre-trained source forecaster is deployed on temporally ordered test data under distribution shift. At test time, rolling windows are grouped into mini-batches, and predictions are produced for all samples within each mini-batch. Let L and H denote the look-back length and forecasting horizon, respectively. We index rolling mini-batches by k, and let B_k denote the size of the k-th mini-batch. We further let t_k denote the global time index immediately preceding the first forecasted target of the k-th mini-batch. Since rolling windows are grouped consecutively, the next mini-batch starts after B_k shifts, yielding $t_{k+1} = t_k + B_k$. For the j-th sample of the k-th mini-batch, where

$j \in \{1, \ldots, B_k\}$, the input window and the corresponding forecast are given by

$$\mathbf{X}_k^{[j]} = [\mathbf{x}_{t_k+j-L}, \ldots, \mathbf{x}_{t_k+j-1}], \quad \hat{\mathbf{Y}}_k^{[j]} = [\hat{\mathbf{y}}_{t_k+j}^{[j]}, \ldots, \hat{\mathbf{y}}_{t_k+j+H-1}^{[j]}] \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^C$ denotes the multivariate observation at time step t, and $\hat{\mathbf{y}}_t^{[j]} \in \mathbb{R}^C$ denotes the prediction of the j-th sample at time step t. Note that $\hat{\mathbf{y}}_t^{[j]}$ differs for different samples in the mini-batch.

Denote the corresponding target vector as

$$\mathbf{Y}_k^{[j]} = [\mathbf{y}_{t_k+j}, \ldots, \mathbf{y}_{t_k+j+H-1}], \tag{2}$$

then we can denote the forecasting target span of the k-th mini-batch as the union of all sample-level targets within the mini-batch, which is given by

$$\mathfrak{Y}_k = \bigcup_{j=1}^{B_k} \mathbf{Y}_k^{[j]} = \{\mathbf{y}_{t_k+1}, \ldots, \mathbf{y}_{t_k+H+B_k-1}\}. \tag{3}$$

3.2 A Closer Look at Existing Supervision Protocols

3.2.1 Mixed Supervision with POGT and Matured Ground Truth

A representative line of TSF-TTA methods (Kim et al., 2025; Medeiros et al., 2025; Grover & Etemad, 2025) adopts a mixed supervision protocol that combines POGT from the current mini-batch with supervision from the most recent matured mini-batch, as illustrated in Figure 1a. In these methods, the POGT length p_k is determined by the periodicity-aware adaptation scheduling (PAAS), with $p_k = B_k - 1$. For notational consistency, we will use $B_k - 1$ to denote the POGT length below.

Let $\mathcal{C}_{k-1}^{\text{in}}(\cdot)$ and $\mathcal{C}_{k-1}^{\text{out}}(\cdot)$ denote the input and output calibration modules available before adapting on the k-th mini-batch. For the j-th sample in that mini-batch, the calibrated prediction is

$$\hat{\mathbf{Y}}_k^{\text{cali},[j]} = \mathcal{C}_{k-1}^{\text{out}}(\mathcal{F}(\mathcal{C}_{k-1}^{\text{in}}(\mathbf{X}_k^{[j]}))) = [\hat{\mathbf{y}}_{t_k+j}^{\text{cali},[j]}, \ldots, \hat{\mathbf{y}}_{t_k+j+H-1}^{\text{cali},[j]}], \tag{4}$$

where $\mathcal{F}(\cdot)$ denotes the frozen source forecaster. Let $m(k)$ denote the index of the most recent matured mini-batch. Then the mixed supervision protocol can be abstractly written as

$$\mathcal{L}_k^{\text{mixed}} = \mathcal{L}_k^{\text{POGT}} + \mathcal{L}_{m(k)}^{\text{Matured}}, \tag{5}$$

with

$$\mathcal{L}_k^{\text{POGT}} = \text{MSE}\Big(\{\hat{\mathbf{y}}_t^{\text{cali},[1]}\}_{t=t_k+1}^{t_k+B_k-1}, \{\mathbf{y}_t\}_{t=t_k+1}^{t_k+B_k-1}\Big). \tag{6}$$

and

$$\mathcal{L}_{m(k)}^{\text{Matured}} = \text{MSE}\Big(\{\hat{\mathbf{Y}}_{m(k)}^{\text{cali},[j]}\}_{j=1}^{B_{m(k)}}, \{\mathbf{Y}_{m(k)}^{[j]}\}_{j=1}^{B_{m(k)}}\Big). \tag{7}$$

In implementation, these two terms may be consumed in separate adaptation steps, but together they define the mixed supervision protocol. After adaptation, the calibration modules are updated to $\mathcal{C}_k^{\text{in}}(\cdot)$ and $\mathcal{C}_k^{\text{out}}(\cdot)$. The current mini-batch is then re-forecast using the updated calibration modules:

$$\hat{\mathbf{Y}}_k^{\text{adp},[j]} = \mathcal{C}_k^{\text{out}}\Big(\mathcal{F}\Big(\mathcal{C}_k^{\text{in}}(\mathbf{X}_k^{[j]})\Big)\Big). \tag{8}$$

The adjusted prediction keeps the already observed prefix from the original calibrated prediction and replaces the unobserved suffix with the re-forecasted one, which is given by

$$\tilde{\mathbf{y}}_{k,t}^{[j]} = \begin{cases} \hat{\mathbf{y}}_t^{\text{cali},[j]}, & t_k + j \le t \le t_k + B_k - 1, \\ \hat{\mathbf{y}}_t^{\text{adp},[j]}, & t_k + B_k \le t \le t_k + j + H - 1. \end{cases} \tag{9}$$

However, by the time POGT becomes available for adaptation, the same revealed values have already entered the look-back windows of later samples within the same mini-batch. **This raises a natural question: is**

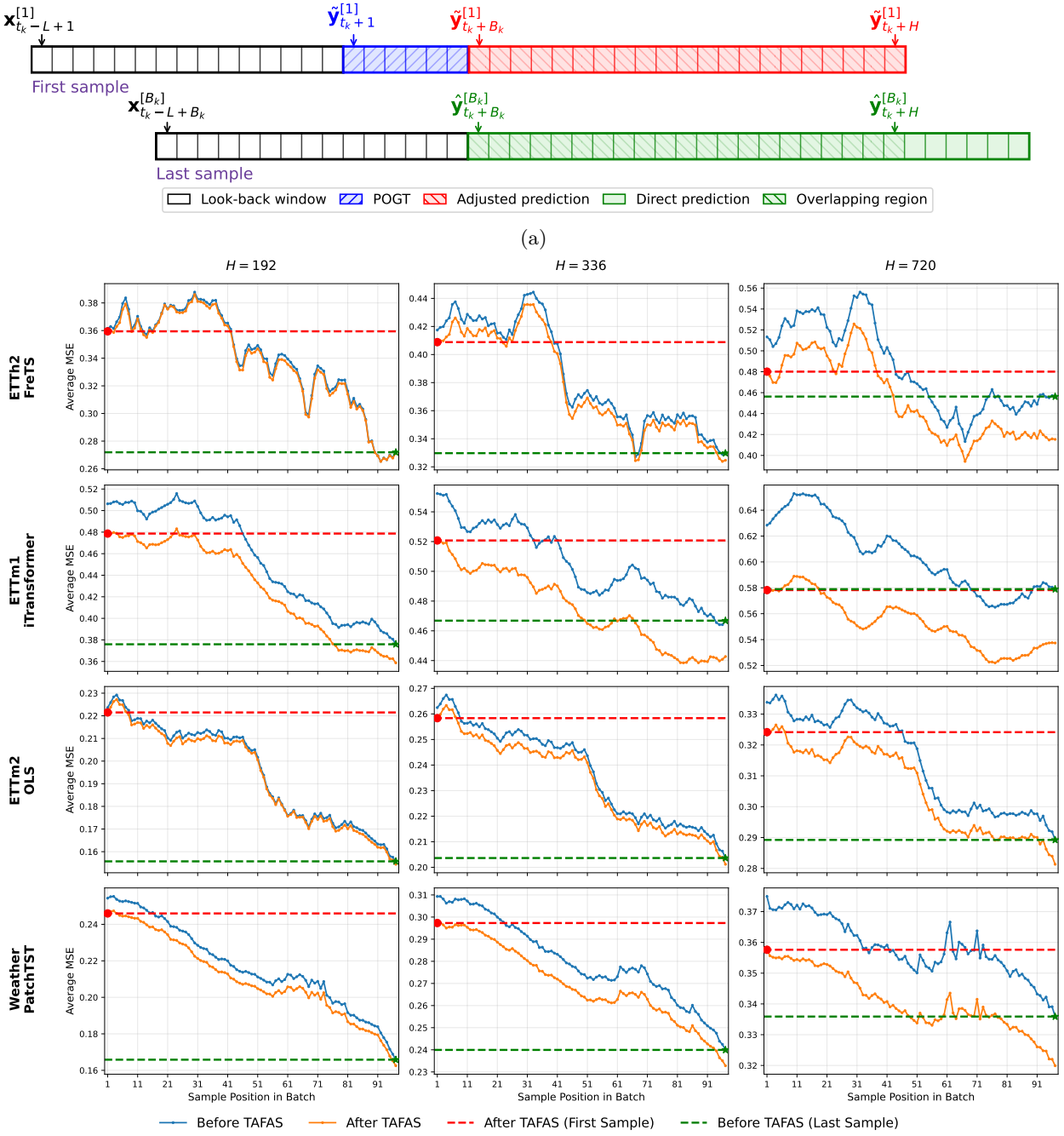

Figure 2: Comparison between the adjusted prediction of the first sample and later direct predictions within the same mini-batch. (a) Schematic illustration of the overlapping horizon shared by the first sample's adjusted prediction and the last sample's direct prediction. (b) Mean overlapping-region MSE at each sample position, averaged over mini-batches with $B_k = 97$ across representative datasets, horizons, and source forecasters. The blue and orange curves denote direct and adjusted predictions, respectively; the red dashed line marks the adjusted prediction of the first sample, and the green dashed line marks the direct prediction of the last sample. Additional results for $B_k = 25$ are provided in Appendix A.1.

it more beneficial to use these revealed values as supervision for adaptation, or simply as additional input context for later direct prediction? To examine this, we compare the MSE on the shared overlapping horizon between the adjusted prediction of the first sample and the direct predictions of later samples within the same mini-batch.

As illustrated in Figure 2a, the endpoint comparison provides the most direct way to assess whether using the revealed values as supervision offers a meaningful advantage over simply using the same values as input context in later direct prediction. At the same time, this endpoint view is only the most intuitive illustration of a broader phenomenon within the mini-batch. To examine this more comprehensively, we compute the mean overlapping-region MSE at each sample position over the mini-batches with $B_k = 97$, since this is the most common mini-batch size induced by PAAS in our experiments.

Let $\mathcal{I}_{\{B_k=B\}}$ denote the set of mini-batches satisfying $B_k = B$. For the direct prediction of the j-th sample, we define the mean overlapping-region MSE as

$$\hat{M}^{[j]}_{\mathcal{I}_{\{B_k=B\}}} = \frac{1}{\left|\mathcal{I}_{\{B_k=B\}}\right|} \sum_{k \in \mathcal{I}_{\{B_k=B\}}} \mathrm{MSE}\big([\hat{\mathbf{y}}^{[j]}_{t_k+B}, \ldots, \hat{\mathbf{y}}^{[j]}_{t_k+H}], [\mathbf{y}_{t_k+B}, \ldots, \mathbf{y}_{t_k+H}]\big). \tag{10}$$

The corresponding metric for the adjusted prediction, $\tilde{M}^{[j]}_{\mathcal{I}_{\{B_k=B\}}}$, is defined analogously by replacing $\hat{\mathbf{y}}^{[j]}$ with $\tilde{\mathbf{y}}^{[j]}$. Figure 2b further shows that this phenomenon is more widespread. Across multiple datasets, forecasting horizons and frozen source forecasters, the adjusted prediction of the first sample is generally inferior to the direct prediction of the last sample. In fact, the direct predictions from many intermediate sample positions already achieve comparable or lower overlapping-region MSE than the adjusted prediction of the first sample. These results indicate that using revealed values as supervision is not consistently more beneficial than simply using the same values as input context for later direct prediction.

3.2.2 Streaming Adaptation with Revealed Ground Truth

COSA (Im & Kwon, 2026) adopts a streaming adaptation protocol based on revealed ground truth, as illustrated in Figure 1b. Under this protocol, the current mini-batch is first predicted by the frozen source forecaster, after which an output adapter produces the final adapter-corrected prediction using a lightweight context vector constructed from previously revealed targets. Unlike the mixed-supervision protocol above, the output adapter is then updated directly by minimizing the MSE loss between these adapter-corrected predictions of the current mini-batch and the revealed ground truth of the current mini-batch.

Let $\mathcal{A}_{k-1}(\cdot,\cdot)$ denote the output adapter available before adapting on the k-th mini-batch, and let $\mathbf{c}_{k-1}$ denote the context vector constructed from previously revealed targets. For the j-th sample in the k-th mini-batch, the final streaming prediction is given by

$$\hat{\mathbf{Y}}^{\mathrm{stream},[j]}_k = \mathcal{A}_{k-1}(\mathcal{F}(\mathbf{X}^{[j]}_k), \mathbf{c}_{k-1}). \tag{11}$$

Accordingly, the streaming adaptation objective can be written as

$$\mathcal{L}^{\mathrm{stream}}_k = \frac{1}{B_k} \sum_{j=1}^{B_k} \mathrm{MSE}(\hat{\mathbf{Y}}^{\mathrm{stream},[j]}_k, \mathbf{Y}^{[j]}_k). \tag{12}$$

Although the protocol is presented in terms of revealed ground truth, such supervision is not necessarily matured relative to subsequent rolling predictions. Recall that the target span of the k-th mini-batch is $\mathfrak{Y}_k = \{\mathbf{y}_{t_k+1}, \ldots, \mathbf{y}_{t_k+H+B_k-1}\}$. For the $(k+1)$-th mini-batch, since $t_{k+1} = t_k + B_k$, the prediction span of the j-th sample is

$$\hat{\mathbf{Y}}^{\mathrm{stream},[j]}_{k+1} = [\hat{\mathbf{y}}^{\mathrm{stream},[j]}_{t_k+B_k+j}, \ldots, \hat{\mathbf{y}}^{\mathrm{stream},[j]}_{t_k+B_k+j+H-1}]. \tag{13}$$

When the forecasting horizon is sufficiently larger than the mini-batch size, the targets used to update on batch k may still overlap with the future prediction span of later mini-batches. For example, if $H \geq B_k + B_{k+1}$, then

$$\{t_k+1, \ldots, t_k+H+B_k-1\} \cap \{t_k+B_k+j, \ldots, t_k+B_k+j+H-1\} = \{t_k+B_k+j, \ldots, t_k+H+B_k-1\} \neq \emptyset. \tag{14}$$

This shows that the supervision used to update on batch k is not necessarily matured relative to subsequent rolling prediction spans.

3.3 A Cleaner Adaptation Protocol Based on Matured Ground Truth

The above examination of mixed supervision and streaming adaptation demonstrates the need for a cleaner protocol. In the mixed-supervision setting, the use of POGT naturally induces a comparison between using revealed values as supervision and using the same values as additional input context for later direct prediction. In the streaming setting, supervision based on revealed ground truth is not necessarily matured relative to subsequent rolling predictions. These observations motivate a cleaner protocol that restricts adaptation to matured ground truth only.

The cleaner protocol is defined by the temporal status of the supervision it permits, rather than by a unique implementation rule. For example, one may update based on any subset of matured ground truth. Formally, denote the index set of matured mini-batches available at step k as

$$\mathcal{M}_k := \{m < k : t_m + B_m + H - 1 \leq t_k\}. \tag{15}$$

Then a general adaptation objective based on matured ground truth may be written as

$$\mathcal{L}_k^{\text{clean}} = \sum_{m \in \mathcal{S}_k} \omega_{k,m} \, \mathcal{L}_m^{\text{matured}}, \qquad \mathcal{S}_k \subseteq \mathcal{M}_k, \tag{16}$$

where $\mathcal{S}_k$ denotes the subset of matured mini-batches selected for adaptation at step k, $\omega_{k,m} \geq 0$ are combination weights, and

$$\mathcal{L}_m^{\text{matured}} = \frac{1}{B_m} \sum_{j=1}^{B_m} \text{MSE}(\hat{\mathbf{Y}}_m^{[j]}, \mathbf{Y}_m^{[j]}). \tag{17}$$

Under our proposed protocol based solely on matured ground truth, predictions may still be organized in mini-batches, but the role of the mini-batch becomes different. Conceptually, the protocol no longer depends on supervision drawn from the current mini-batch; instead, adaptation is triggered only by previously matured targets. In this sense, the mini-batch mainly serves as an implementation-level grouping for prediction and evaluation. Moreover, although predictions within a mini-batch still overlap due to the rolling construction, the comparison between earlier and later samples in the overlapped region is no longer central to the protocol itself, since adaptation for each sample no longer relies on POGT from the current mini-batch. Rather, adaptation can already be triggered using matured past supervision before later samples in the current mini-batch are generated.

4 Frequency-Aware Calibration

Having established a cleaner protocol to specify which supervision signals should be used for adaptation, we now turn to the design of the adaptation module. Existing TSF-TTA adapters vary primarily in their parameterization strategies. TAFAS (Kim et al., 2025) calibrates predictions through temporal-domain transformations, whereas PETSA (Medeiros et al., 2025) incorporates spectral information by adding a frequency-domain loss to a low-rank temporal calibration module. Empirically, however, we observe that these formulations tend to produce relatively smooth and less selective spectral corrections, even when spectral regularization is explicitly imposed. This motivates Frequency-Aware Calibration (FAC), which directly parameterizes these corrections in the frequency domain.

FAC adopts a minimalist design, since a central requirement for test-time adaptation in TSF is to keep the adapter lightweight. Figure 3 provides an overview. Similar to TAFAS (Kim et al., 2025) and PETSA (Medeiros et al., 2025), FAC places calibration modules before and after the frozen source forecaster. Each module performs a gated residual correction to the signal being calibrated, rather than directly replacing it. In contrast to these designs, FAC directly parameterizes the calibration operation in the frequency domain through frequency-domain gated calibration modules (FreqGCMs). Specifically, FAC consists of an input and

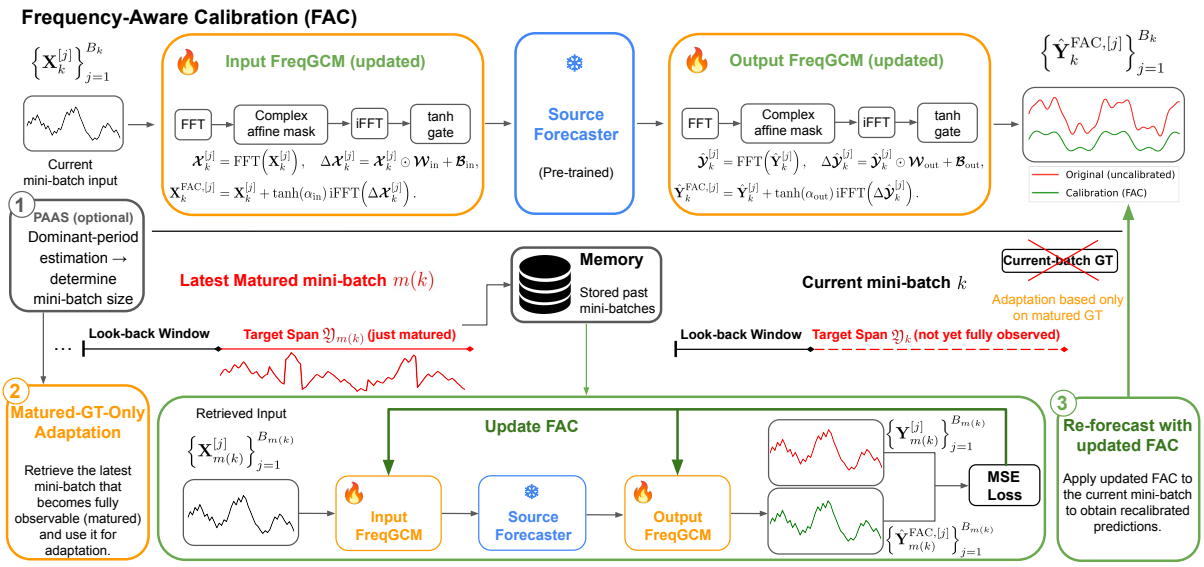

Figure 3: Overview of Frequency-Aware Calibration (FAC) under the proposed protocol. FAC uses input and output frequency-domain gated calibration modules (FreqGCMs), each applying a complex affine mask in the frequency domain followed by inverse FFT and gated residual correction. For simplicity, our implementation retrieves the most recent matured mini-batch for each update, after which the updated adapter is applied to re-forecast the current mini-batch. PAAS can optionally be used to determine the mini-batch size.

an output FreqGCM. Each FreqGCM first transforms the signal being calibrated into the frequency domain, applies a learnable complex affine mask, maps the frequency-domain correction back to the temporal domain through inverse FFT, and then controls the correction magnitude through a hyperbolic-tangent gate. On the input side, the input FreqGCM for the j-th sample in the k-th mini-batch is given by

$$\begin{aligned} \boldsymbol{\mathcal{X}}_k^{[j]} &= \mathrm{FFT}\left(\mathbf{X}_k^{[j]}\right), \\ \Delta\boldsymbol{\mathcal{X}}_k^{[j]} &= \boldsymbol{\mathcal{X}}_k^{[j]} \odot \boldsymbol{\mathcal{W}}_{\mathrm{in}} + \boldsymbol{\mathcal{B}}_{\mathrm{in}}, \\ \mathbf{X}_k^{\mathrm{FAC},[j]} &= \mathbf{X}_k^{[j]} + \tanh(\alpha_{\mathrm{in}})\,\mathrm{iFFT}\left(\Delta\boldsymbol{\mathcal{X}}_k^{[j]}\right). \end{aligned} \tag{18}$$

The calibrated input is then passed to the frozen source forecaster as $\hat{\mathbf{Y}}_k^{[j]} = \mathcal{F}(\mathbf{X}_k^{\mathrm{FAC},[j]})$. The output FreqGCM follows the same form on $\hat{\mathbf{Y}}_k^{[j]}$ with output-side parameters $(\boldsymbol{\mathcal{W}}_{\mathrm{out}}, \boldsymbol{\mathcal{B}}_{\mathrm{out}}, \alpha_{\mathrm{out}})$, yielding the final calibrated forecast $\hat{\mathbf{Y}}_k^{\mathrm{FAC},[j]}$. Here, $(\boldsymbol{\mathcal{W}}_{\mathrm{in}}, \boldsymbol{\mathcal{B}}_{\mathrm{in}})$ and $(\boldsymbol{\mathcal{W}}_{\mathrm{out}}, \boldsymbol{\mathcal{B}}_{\mathrm{out}})$ define element-wise complex affine transformations, while α_{in} and α_{out} are learnable gates. We write FFT/iFFT for notational simplicity, while using rFFT/irFFT in implementation for real-valued time series. The element-wise affine form keeps FAC lightweight and frequency-wise interpretable: the multiplicative term rescales and phase-shifts Fourier coefficients, while the additive term can modify weak or near-zero components that purely multiplicative scaling may miss. The residual form preserves the original signal when the learned correction is small, and the tanh gate controls the overall correction strength.

Our protocol allows adaptation on any subset of matured ground truth; for simplicity, FAC uses the most recent matured mini-batch:

$$\mathcal{L}_{m(k)}^{\mathrm{FAC}} = \mathrm{MSE}\left(\{\hat{\mathbf{Y}}_{m(k)}^{\mathrm{FAC},[j]}\}_{j=1}^{B_{m(k)}}, \{\mathbf{Y}_{m(k)}^{[j]}\}_{j=1}^{B_{m(k)}}\right). \tag{19}$$

Although additional regularization terms could be added to this objective, we keep the adaptation loss as a simple MSE on matured ground truth. This isolates the effect of frequency-domain parameterization and avoids conflating it with additional regularization choices. After the FAC modules are updated, the current mini-batch is re-forecast with the updated input and output FreqGCMs to obtain recalibrated predictions.

5 Experiments

5.1 Experimental Settings

We evaluate the proposed FAC on six multivariate TSF benchmark datasets: ETTh1, ETTh2, ETTm1, ETTm2, Weather, and Exchange. We use chronological train/validation/test splits for all datasets: the four ETT datasets use a ratio of $(0.6, 0.2, 0.2)$, while Weather and Exchange use $(0.7, 0.1, 0.2)$. The look-back length is set to $L = 96$, and the prediction horizon is chosen from $H \in \{96, 192, 336, 720\}$, except for Exchange, where we use $H \in \{96, 192, 336\}$ because the full target horizon for $H = 720$ does not mature before the end of the test sequence. Consistent with existing TSF-TTA studies, we evaluate five representative source forecasters: iTransformer (Liu et al., 2024), PatchTST (Nie et al., 2023), DLinear (Zeng et al., 2023), OLS (Toner & Darlow, 2024), and FreTS (Yi et al., 2023) [1].

We compare FAC with the frozen source forecaster without TTA, TAFAS (Kim et al., 2025), and PETSA (Medeiros et al., 2025) under the proposed protocol based solely on matured ground truth [2]. For fairness and simplicity, only the most recent matured mini-batch is used for adaptation across methods. Since the current mini-batch does not provide supervision under this protocol, the POGT-based partial prediction adjustment in the original TAFAS protocol is no longer used. Instead, after the adapter is updated on the most recent matured mini-batch, the current mini-batch is re-forecast and the full prediction horizon is evaluated using the updated adapter. In particular, for the j-th sample in the k-th mini-batch, we use

$$\tilde{\mathbf{Y}}_k^{[j]} = \hat{\mathbf{Y}}_k^{\mathrm{adp},[j]}, \quad j = 1, \ldots, B_k, \tag{20}$$

rather than a partial stitching rule between pre-update and post-update predictions [3]. Hyperparameters are taken from the officially released code whenever available. All experiments were conducted on an NVIDIA GeForce RTX 4070 Super 12GB and an NVIDIA GeForce RTX 4090 24GB.

5.2 Main Results

5.2.1 Forecasting Performance

Table 1 reports the MSE results across six datasets, five source forecasters, and four prediction horizons. All TSF-TTA methods are evaluated under the proposed protocol using only matured ground truth.

Overall, FAC achieves the best or second-best performance in most settings and often outperforms TAFAS and PETSA under the same protocol using only matured ground truth. In addition, all three adaptation methods generally improve over the frozen source forecaster, indicating that matured ground truth can serve as an effective adaptation signal for TSF-TTA.

5.2.2 Parameter Efficiency

Besides forecasting error, parameter efficiency is a central consideration when designing a test-time adapter. A direct way to measure parameter efficiency is to compare the number of trainable parameters in the calibration modules. Table 2 reports the trainable parameter counts of different adapters under our experimental settings. FAC consistently requires the fewest trainable parameters across datasets and horizons.

Table 3 summarizes how the adapter size scales with the look-back length L, prediction horizon H, number of variables C, and PETSA rank r. TAFAS scales quadratically with the sequence lengths due to its dense temporal calibration, whereas PETSA achieves rank-dependent linear scaling through low-rank adapters. FAC instead uses an element-wise frequency-domain affine parameterization, yielding linear scaling in C, L, and H without the rank factor.

[1] We omit MICN (Wang et al., 2023) from the main experiments because its architecture categorization is ambiguous in prior TSF-TTA comparisons. Although some studies group it with MLP-based forecasters, MICN is more accurately viewed as a multi-scale convolution-based model.

[2] We omit DynaTTA (Grover & Etemad, 2025) from the main experiments due to reproducibility issues with the officially released implementation; similar issues were also noted in COSA (Im & Kwon, 2026).

[3] We do not include a matured-only COSA (Im & Kwon, 2026) variant in the main experiments because COSA is designed around an output-space streaming adapter with context constructed from revealed target statistics. Adapting it to our proposed protocol would require nontrivial design choices.

Table 1: Forecasting performance under the proposed protocol using only matured ground truth. Lower MSE is better. The best result within each backbone and horizon is highlighted in red bold, and the second-best result is underlined.

Dataset	H	DLinear				FreTS				iTransformer				OLS				PatchTST			
		Base	TAFAS	PETSA	**FAC**	Base	TAFAS	PETSA	**FAC**	Base	TAFAS	PETSA	**FAC**	Base	TAFAS	PETSA	**FAC**	Base	TAFAS†	PETSA	**FAC**
ETTh1	96	0.4695	0.4617	0.4613	**0.4554**	0.4461	0.4391	0.4383	**0.4357**	0.4482	0.4421	**0.4394**	0.4400	0.4510	0.4428	0.4449	**0.4382**	0.4324	0.4270	0.4255	**0.4241**
	192	0.5213	0.5112	0.5123	**0.5089**	0.5022	0.4943	0.4960	**0.4934**	0.5091	0.5020	0.5014	**0.5010**	0.5045	0.4929	0.5007	**0.4921**	0.4900	0.4818	0.4818	**0.4797**
	336	0.5659	0.5595	0.5626	**0.5582**	0.5544	**0.5483**	0.5530	0.5498	0.5662	0.5624	0.5641	**0.5587**	0.5510	0.5427	0.5488	**0.5422**	0.5608	**0.5506**	0.5570	0.5525
	720	0.7117	0.6926	0.6938	**0.6891**	0.7181	**0.6857**	0.6908	0.7067	0.7007	0.6755	0.6763	**0.6687**	0.6996	**0.6704**	0.6805	0.6780	0.7112	**0.6807**	0.7019	0.7096
ETTh2	96	0.2323	0.2308	0.2312	**0.2288**	0.2385	0.2363	0.2366	**0.2338**	0.2642	0.2620	0.2602	**0.2561**	0.2306	0.2289	0.2293	**0.2274**	0.2385	0.2380	0.2383	**0.2374**
	192	0.2862	0.2830	0.2820	**0.2819**	0.2867	0.2830	0.2844	**0.2814**	0.3081	0.3004	0.3005	**0.2970**	0.2838	0.2821	0.2829	**0.2796**	0.2840	0.2752	0.2810	**0.2747**
	336	0.3252	0.3175	0.3190	**0.3144**	0.3318	0.3195	0.3250	**0.3172**	0.3402	0.3290	0.3312	**0.3284**	0.3257	0.3189	0.3240	**0.3137**	0.3233	0.3108	0.3190	**0.3079**
	720	0.4087	**0.3825**	0.3884	**0.3825**	0.4118	0.3814	0.3968	**0.3788**	0.4261	**0.3959**	0.4049	0.3982	0.4161	0.3859	0.3927	**0.3854**	0.4292	0.4065	0.4203	**0.4002**
ETTm1	96	0.3715	0.3504	0.3541	**0.3465**	0.3676	0.3559	0.3572	**0.3517**	0.3823	0.3680	0.3700	**0.3459**	0.3709	0.3542	0.3591	**0.3462**	0.4041	0.3826	0.3817	**0.3530**
	192	0.4438	0.4165	0.4192	**0.4139**	0.4325	0.4191	0.4189	**0.4183**	0.4407	0.4189	0.4189	**0.4087**	0.4437	0.4184	0.4252	**0.4140**	0.4515	0.4370	0.4405	**0.4213**
	336	0.5183	0.4801	0.4831	**0.4791**	0.5006	0.4810	**0.4793**	0.4836	0.5077	0.4800	0.4788	**0.4781**	0.5179	0.4811	0.4896	**0.4794**	0.5044	0.4864	0.4970	**0.4787**
	720	0.5929	0.5521	0.5600	**0.5494**	0.5703	**0.5472**	0.5523	0.5533	0.6021	0.5579	0.5615	**0.5460**	0.5923	0.5515	0.5594	**0.5497**	0.5608	0.5413	0.5479	**0.5399**
ETTm2	96	0.1598	0.1581	0.1583	**0.1560**	0.1582	0.1566	0.1569	**0.1563**	0.1647	0.1640	0.1640	**0.1613**	0.1602	0.1594	0.1595	**0.1563**	0.1581	0.1569	0.1569	**0.1558**
	192	0.1930	0.1917	0.1920	**0.1894**	0.1923	**0.1901**	0.1908	**0.1901**	0.2189	0.2164	0.2157	**0.2067**	0.1935	0.1924	0.1925	**0.1897**	0.2097	0.2058	0.2079	**0.1976**
	336	0.2324	0.2299	0.2303	**0.2286**	0.2320	**0.2291**	0.2312	0.2294	0.2808	0.2735	0.2653	**0.2623**	0.2331	0.2308	0.2316	**0.2289**	0.2482	0.2482	0.2473	**0.2422**
	720	0.3062	0.2993	**0.2984**	**0.2984**	0.3013	**0.2924**	0.2983	0.2955	0.3457	0.3303	**0.3263**	0.3290	0.3066	0.2996	0.3003	**0.2973**	0.3283	0.3218	0.3240	**0.3192**
Weather	96	0.1954	0.1803	0.1812	**0.1797**	0.1855	0.1757	0.1795	**0.1678**	0.1724	0.1662	0.1662	**0.1605**	0.1957	0.1825	0.1890	**0.1728**	0.1746	0.1700	0.1676	**0.1635**
	192	0.2403	0.2265	0.2308	**0.2231**	0.2309	0.2167	0.2229	**0.2114**	0.2239	0.2129	0.2165	**0.2103**	0.2406	0.2238	0.2293	**0.2171**	0.2188	0.2119	0.2120	**0.2050**
	336	0.2918	0.2726	0.2776	**0.2717**	0.2842	0.2661	0.2720	**0.2621**	0.2806	0.2629	0.2672	**0.2618**	0.2921	0.2729	0.2785	**0.2665**	0.2771	0.2659	0.2703	**0.2570**
	720	0.3643	0.3509	0.3557	**0.3426**	0.3599	0.3423	0.3550	**0.3376**	0.3563	0.3418	0.3470	**0.3357**	0.3645	0.3467	0.3598	**0.3401**	0.3555	0.3356	0.3529	**0.3341**
Exchange	96	0.0913	**0.0875**	0.0898	**0.0875**	0.0828	**0.0796**	0.0825	0.0810	0.0882	0.0873	0.0881	**0.0867**	0.0814	0.0799	0.0813	**0.0793**	0.0854	0.0816	0.0850	**0.0813**
	192	0.1827	**0.1732**	0.1787	0.1758	0.1734	**0.1654**	0.1729	0.1667	0.1841	0.1800	0.1820	**0.1773**	0.1727	0.1650	0.1719	**0.1648**	0.1774	**0.1666**	0.1765	0.1692
	336	0.3277	0.3052	0.3154	**0.2985**	0.3241	0.3180	0.3223	**0.3166**	0.3157	0.3014	0.3105	**0.2983**	0.3225	**0.2846**	0.3058	0.2924	0.3382	0.3048	0.3188	**0.3011**

Note. FAC denotes our proposed method. For PatchTST, TAFAS† follows the original PatchTST-specific implementation with single-channel calibration and no input calibration. PETSA uses multivariate calibration on PatchTST to match its frequency-domain and distributional adaptation objectives over multivariate forecasts.

Table 2: Trainable parameter counts of different test-time adapters. For TAFAS, we report both the standard count and the PatchTST-specific count, since TAFAS uses channel dimension 1 for PatchTST but the number of variables C for other backbones. The smallest count in each row is highlighted in red bold, and the second smallest is underlined.

Dataset	H	TAFAS	TAFAS (PatchTST)	PETSA	FAC (Ours)
ETT (h1/2, m1/2)	96	130,382	18,626	25,934	**2,758**
	192	324,590	46,370	38,894	**4,102**
	336	857,822	122,546	58,334	**6,118**
	720	3,699,038	528,434	110,174	**11,494**
Weather	96	391,146	18,626	71,658	**8,274**
	192	973,770	46,370	107,466	**12,306**
	336	2,573,466	122,546	161,178	**18,354**
	720	11,097,114	528,434	304,410	**34,482**
Exchange	96	149,008	18,626	29,200	**3,152**
	192	370,960	46,370	43,792	**4,688**
	336	980,368	122,546	65,680	**6,992**

Table 3: Parameter scaling of different test-time adapters.

Adapter	Parameter scaling
TAFAS	$O\big(C(L^2+H^2)\big)$
TAFAS (PatchTST)	$O\big(L^2+H^2\big)$
PETSA	$O(Cr(L+H))$
FAC (Ours)	$O(C(L+H))$

5.3 Analysis

We further analyze two aspects of our design: whether POGT is necessary for effective adaptation, and how different adapters modify predictions in the frequency domain. The first analysis examines the role of supervision timing, while the second characterizes the structure of adaptation-induced corrections.

5.3.1 Is POGT Necessary for Effective Adaptation?

We first compare the original mixed-supervision TAFAS protocol with a matured-only variant to assess whether POGT provides a consistent empirical benefit. As shown in Table 4, the original TAFAS protocol often achieves slightly lower MSE, but the variant using matured ground truth remains close across source forecasters and prediction horizons. These results show that POGT is not consistently necessary for effective adaptation. Together with the protocol-level motivation, this supports using the matured-GT variant as a cleaner protocol for our main experiments. Complete results for TAFAS are provided in Appendix A.2.

Table 4: Original TAFAS vs. matured-only TAFAS on ETTh1. Lower MSE is better. The better result in each pair is highlighted in red bold.

H	DLinear		FreTS		iTransformer		OLS		PatchTST	
	Orig.	Matured	Orig.	Matured	Orig.	Matured	Orig.	Matured	Orig.	Matured
96	0.4618	**0.4617**	0.4394	**0.4391**	**0.4398**	0.4421	**0.4419**	0.4428	**0.4261**	0.4270
192	0.5117	**0.5112**	**0.4940**	0.4943	**0.5017**	0.5020	**0.4924**	0.4929	**0.4809**	0.4818
336	0.5604	**0.5595**	**0.5475**	0.5483	0.5643	**0.5624**	**0.5415**	0.5427	**0.5501**	0.5506
720	**0.6820**	0.6926	0.6872	**0.6857**	**0.6591**	0.6755	**0.6654**	0.6704	0.6815	**0.6807**

5.3.2 Frequency-Domain Behavior of Prediction Corrections

We further examine how different adapters modify the forecast in the frequency domain. This analysis is not intended to measure forecasting error directly; instead, it characterizes the frequency-domain structure of the corrections induced by different adaptation methods. Specifically, we compute the difference between predictions before and after adaptation, apply rFFT along the forecasting horizon, and average the correction magnitude over samples and channels.

Figure 4 illustrates the resulting correction spectra for representative dataset–forecaster pairs at forecasting horizon $H = 720$, with more comprehensive results across horizons deferred to Appendix A.5. As shown in the figure, TAFAS and PETSA often produce relatively smooth frequency-domain corrections, with magnitudes changing gradually across frequencies. In contrast, FAC exhibits more localized peaks, showing that its corrections are more selective with respect to specific frequency components. This behavior is consistent with the motivation of FAC: by directly applying an element-wise complex affine mask in the frequency domain, FAC allows individual Fourier components to be adjusted separately, which can naturally lead to more localized spectral corrections. Therefore, the observed peaks should be interpreted as evidence of frequency-selective adaptation rather than as a direct measure of forecasting accuracy; a larger correction magnitude does not necessarily imply a more accurate forecast.

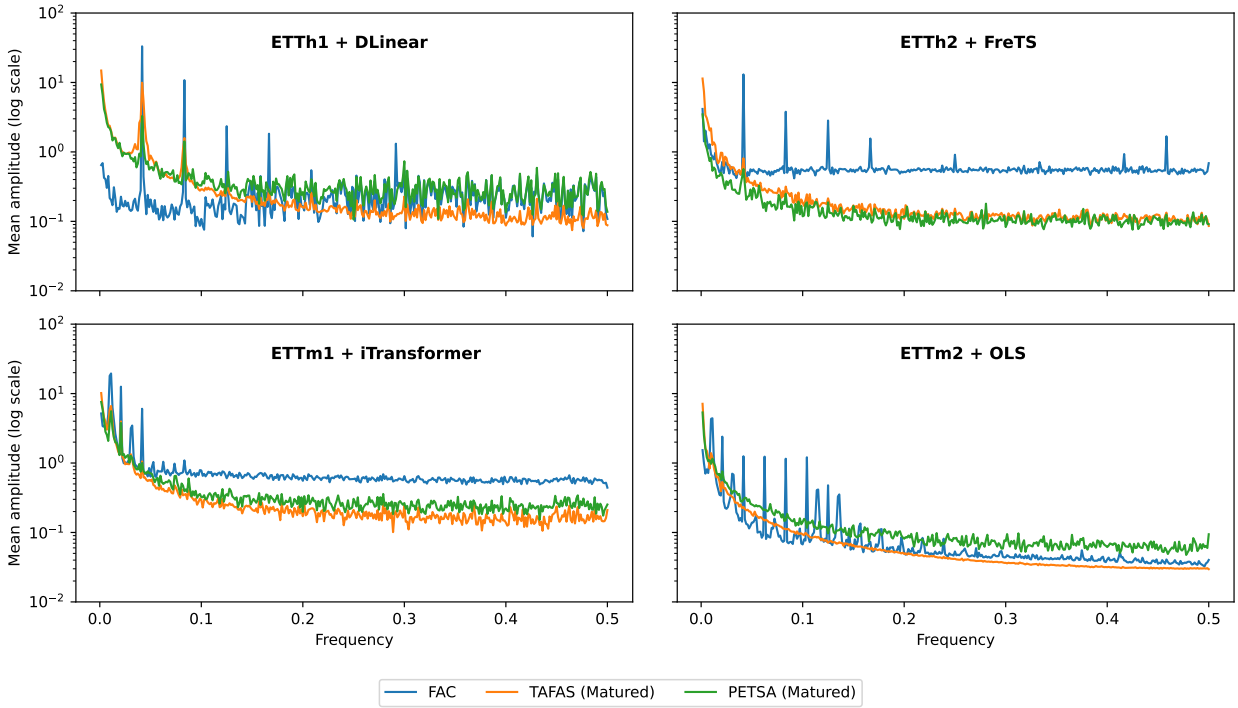

Figure 4: Frequency-domain magnitude of prediction corrections under the matured-only protocol for forecasting horizon $H = 720$. The plotted spectra are computed from the differences between predictions after and before adaptation. The x-axis denotes frequency, while the y-axis shows the correction magnitude on a logarithmic scale. The figure characterizes adaptation-induced corrections rather than forecasting error; a larger magnitude does not necessarily indicate a more accurate forecast.

6 Conclusions, Limitations, and Future Work

Conclusions. We re-examine the adaptation protocols employed by existing TSF-TTA methods. Specifically, we revisit the mixed-supervision protocol that incorporates both POGT and matured ground truth, as well as the streaming adaptation protocol based on revealed targets, and discuss their potential protocol-level limitations. To enhance protocol cleanliness, we propose a more principled adaptation protocol that uses only matured ground truth. We further diagnose existing adapters in the frequency domain and propose FAC, a lightweight adapter that directly parameterizes prediction corrections in the frequency domain. Experimental results demonstrate that FAC achieves competitive and consistent forecasting performance across diverse settings while requiring substantially fewer trainable parameters.

Limitations. Although FAC requires substantially fewer trainable parameters, this parameter efficiency does not always translate into lower wall-clock runtime under our current implementation. As detailed in Appendix A.4, when adaptation is performed using only the most recent matured mini-batch, rFFT/iFFT operations and additional tensor transformations can introduce computational overhead relative to TAFAS in some settings. Because a portion of this overhead stems from repeated frequency-domain transformations, more efficient caching or batched frequency-domain updates may speed up future implementations.

Future Work. Future work could explore more flexible matured-supervision implementations that do not require waiting for the entire mini-batch to mature before using already observable targets. Another direction is to use multiple available matured mini-batches, as allowed by Eq. 16, instead of only the most recent one. When multiple matured mini-batches are used, cached frequency-domain representations of historical matured mini-batches may help reduce repeated FFT computations and enhance the runtime efficiency of frequency-aware adaptation. Together, these directions could make adaptation based on matured ground truth more proactive and efficient while preserving the cleanliness of the proposed protocol.

References

Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 402–411, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482315. URL https://doi.org/10.1145/3459637.3482315.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-ts: A general paradigm for alleviating distribution shift in time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7522–7529, Jun. 2023. doi: 10.1609/aaai.v37i6.25914. URL https://ojs.aaai.org/index.php/AAAI/article/view/25914.

Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. SoTTA: Robust test-time adaptation on noisy data streams. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=3bdXag2rUd.

Clive W. J. Granger. Time series concepts for conditional distributions. *Oxford Bulletin of Economics and Statistics*, 65(s1):689–701, 2003. doi: 10.1046/j.0305-9049.2003.00094.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1046/j.0305-9049.2003.00094.x.

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*, 2025. URL https://openreview.net/forum?id=a399SmgWGl.

Xiannan Huang, Shuhan Qiu, Jiayuan Du, and Chao Yang. Online time series prediction using feature adjustment. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=s4U2FWEMTU.

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

Hyeongyu Kim, GeonHui Han, and Dosik Hwang. Buffer layers for test-time adaptation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=sSZ9OMO8KT.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (17):17868–17876, Apr. 2025. doi: 10.1609/aaai.v39i17.33965. URL https://ojs.aaai.org/index.php/AAAI/article/view/33965.

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

Kelvin J.L. Koa, Jan Chen, Yunshan Ma, Huanhuan Zheng, and Tat-Seng Chua. Reasoning on time-series for financial technical analysis. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PcjIe5xNaY.

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*, 2025. URL https://openreview.net/forum?id=I0n3EyogMi.

Daojun Liang, Haixia Zhang, Jing Wang, Dongfeng Yuan, and Minggao Zhang. Act now: A novel online forecasting framework for large-scale streaming data. *arXiv preprint arXiv:2412.00108*, 2024. URL https://arxiv.org/abs/2412.00108.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025. doi: 10.1007/s11263-024-02181-w. URL https://doi.org/10.1007/s11263-024-02181-w.

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

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

Neda Maleki, Oxana Lundström, Arslan Musaddiq, John Jeansson, Tobias Olsson, and Fredrik Ahlgren. Future energy insights: Time-series and deep learning models for city load forecasting. *Applied Energy*, 374: 124067, 2024. doi: 10.1016/j.apenergy.2024.124067. URL https://www.sciencedirect.com/science/article/pii/S0306261924014508.

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

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

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

William Toner and Luke Nicholas Darlow. An analysis of linear time series forecasting models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xl82CcbYaT.

Ruey S. Tsay. *Analysis of Financial Time Series*. John Wiley & Sons, 3 edition, 2010. ISBN 9780470414354. doi: 10.1002/9780470644560.

Yogesh Verma, Markus Heinonen, and Vikas Garg. ClimODE: Climate and weather forecasting with physics-informed neural ODEs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xuY33XhEGR.

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

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zt53IDUR1U.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7191–7201, 2022. doi: 10.1109/CVPR52688.2022.00706.

Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 5(6):602–611, 2023. doi: 10.1038/s42256-023-00667-9. URL https://doi.org/10.1038/s42256-023-00667-9.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=iif9mGCTfy.

Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 2022.

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, Jun. 2023. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Weijia Zhang, Le Zhang, Jindong Han, Hao Liu, Yanjie Fu, Jingbo Zhou, Yu Mei, and Hui Xiong. Irregular traffic time series forecasting based on asynchronous spatio-temporal graph convolutional networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 4302–4313, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671665. URL https://doi.org/10.1145/3637528.3671665.

Yi-Fan 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*, 2023. URL https://openreview.net/forum?id=Q25wMXsaeZ.

A Additional Experimental Results

A.1 Additional Early-versus-Late Comparison

Figure A.1 provides an additional early-versus-late comparison for $B_k = 25$, which is the second most common mini-batch size induced by PAAS in our experiments after $B_k = 97$. Compared with the $B_k = 97$ setting, this smaller mini-batch size gives later samples less additional revealed context, making the comparison relatively more favorable to earlier adjusted predictions. Nevertheless, later direct predictions still generally achieve comparable or lower MSE than the adjusted predictions of earlier samples. This further supports our argument that using revealed values as POGT supervision is not consistently more beneficial than using the same values as input context in later rolling windows.

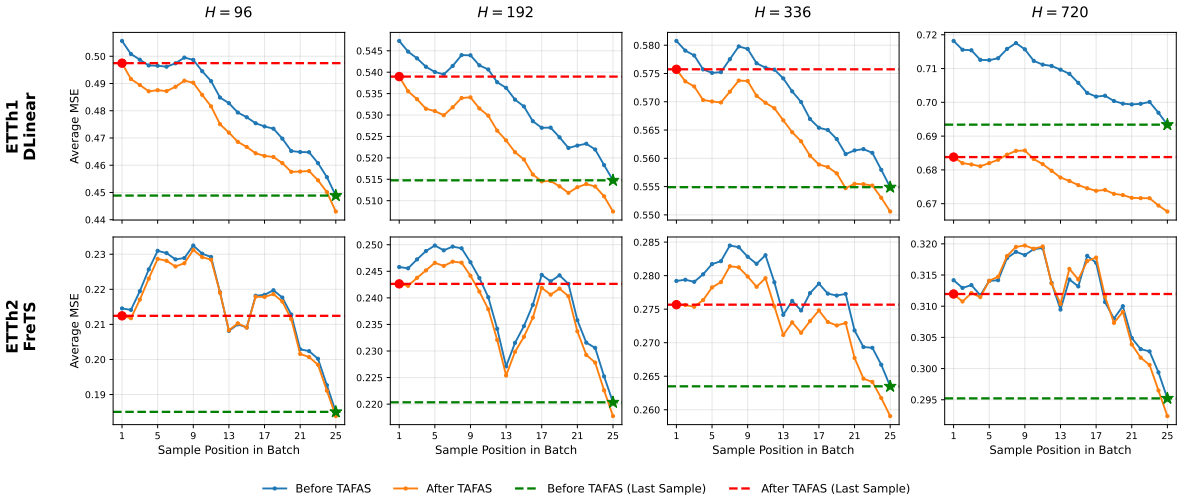

Figure A.1: Additional early-versus-late comparison with mini-batch size $B_k = 25$ on ETTh1 with DLinear and ETTh2 with FreTS. The curves show the average overlapping-region MSE across sample positions within a mini-batch. The dashed horizontal lines mark the adjusted prediction of the first sample and the direct prediction of the last sample as endpoint references.

A.2 Effect of Using Only Matured Ground Truth

Table A.1 provides the full comparison between the original TAFAS protocol and its variant using only matured ground truth. The original TAFAS protocol, which incorporates POGT through mixed supervision, often achieves slightly lower MSE. However, this advantage is not consistent across datasets, horizons, and source forecasters, and the matured-only variant remains close in most settings. These results complement the analysis in the main text and demonstrate that removing POGT-based supervision does not lead to substantial performance degradation, while yielding a cleaner adaptation protocol.

Table A.1: Original TAFAS versus its variant using only matured ground truth across datasets and source forecasters. Lower MSE is highlighted within each original/matured pair.

Dataset	Horizon	DLinear		FreTS		iTransformer		OLS		PatchTST	
		Orig.	Matured	Orig.	Matured	Orig.	Matured	Orig.	Matured	Orig.	Matured
ETTh1	96	0.4618	**0.4617**	0.4394	**0.4391**	**0.4398**	0.4421	**0.4419**	0.4428	**0.4261**	0.4270
	192	0.5117	**0.5112**	**0.4940**	0.4943	**0.5017**	0.5020	**0.4924**	0.4929	**0.4809**	0.4818
	336	0.5604	**0.5595**	**0.5475**	0.5483	0.5643	**0.5624**	**0.5415**	0.5427	**0.5501**	0.5506
	720	**0.6820**	0.6926	0.6872	**0.6857**	**0.6591**	0.6755	**0.6654**	0.6704	0.6815	**0.6807**
ETTh2	96	**0.2303**	0.2308	**0.2362**	0.2363	**0.2613**	0.2620	**0.2284**	0.2289	**0.2378**	0.2380
	192	0.2842	**0.2830**	**0.2828**	0.2830	**0.2984**	0.3004	**0.2817**	0.2821	**0.2743**	0.2752
	336	0.3185	**0.3175**	0.3204	**0.3195**	**0.3286**	0.3290	**0.3186**	0.3189	0.3119	**0.3108**
	720	0.3873	**0.3825**	0.3831	**0.3814**	0.3981	**0.3959**	0.3906	**0.3859**	**0.4023**	0.4065

Dataset	Horizon	DLinear		FreTS		iTransformer		OLS		PatchTST	
		Orig.	Matured	Orig.	Matured	Orig.	Matured	Orig.	Matured	Orig.	Matured
ETTm1	96	**0.3497**	0.3504	0.3574	**0.3559**	**0.3658**	0.3680	**0.3521**	0.3542	0.3912	**0.3826**
	192	0.4166	**0.4165**	0.4205	**0.4191**	**0.4182**	0.4189	**0.4164**	0.4184	**0.4366**	0.4370
	336	**0.4799**	0.4801	0.4816	**0.4810**	**0.4780**	0.4800	**0.4786**	0.4811	**0.4863**	0.4864
	720	**0.5488**	0.5521	0.5474	**0.5472**	**0.5529**	0.5579	**0.5479**	0.5515	0.5414	**0.5413**
ETTm2	96	0.1584	**0.1581**	0.1571	**0.1566**	**0.1639**	0.1640	**0.1592**	0.1594	0.1575	**0.1569**
	192	**0.1913**	0.1917	0.1907	**0.1901**	**0.2159**	0.2164	**0.1921**	0.1924	0.2063	**0.2058**
	336	**0.2289**	0.2299	0.2292	**0.2291**	0.2751	**0.2735**	**0.2299**	0.2308	0.2487	**0.2482**
	720	**0.2968**	0.2993	**0.2918**	0.2924	**0.3278**	0.3303	**0.2982**	0.2996	0.3223	**0.3218**
Weather	96	**0.1796**	0.1803	**0.1744**	0.1757	**0.1647**	0.1662	**0.1810**	0.1825	0.1715	**0.1700**
	192	**0.2244**	0.2265	**0.2151**	0.2167	**0.2119**	0.2129	**0.2221**	0.2238	0.2130	**0.2119**
	336	**0.2709**	0.2726	**0.2640**	0.2661	**0.2615**	0.2629	**0.2707**	0.2729	0.2660	**0.2659**
	720	**0.3500**	0.3509	**0.3400**	0.3423	**0.3410**	0.3418	**0.3443**	0.3467	0.3365	**0.3356**
Exchange	96	0.0885	**0.0875**	**0.0790**	0.0796	**0.0867**	0.0873	**0.0795**	0.0799	**0.0813**	0.0816
	192	0.1760	**0.1732**	**0.1639**	0.1654	**0.1760**	0.1800	**0.1643**	0.1650	0.1673	**0.1666**
	336	**0.2941**	0.3052	**0.2945**	0.3180	**0.2937**	0.3014	**0.2835**	0.2846	**0.2807**	0.3048

A.3 Is Output-Only Calibration Sufficient?

We compare the standard input-output calibration with an output-only variant across datasets, horizons, and source forecasters. The goal is to examine whether calibrating the look-back input window provides additional benefit beyond calibrating only the forecast output. Lower MSE is highlighted within each input-output versus output-only pair.

As shown in Table A.2, input-output calibration generally outperforms the output-only variant across TAFAS, PETSA, and FAC. This indicates that output-side calibration alone may not be sufficient in many test-time adaptation settings, even though it is a simpler design choice. Importantly, the additional parameter cost of input-side calibration is modest, especially for FAC.

Table A.3 reports the corresponding trainable parameter counts. For each dataset and method, the parameter difference between the input-output and output-only variants corresponds to the input adapter. Since the parameter count of the input adapter is determined by the look-back window rather than the prediction horizon, its relative cost becomes smaller as H increases.

Table A.2: Input-output versus output-only calibration across source forecasters. Lower MSE is highlighted within each input-output/output-only pair.

Dataset	Horizon	DLinear						FreTS					
		TAFAS		PETSA		FAC		TAFAS		PETSA		FAC	
		In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out
ETTh1	96	**0.4617**	0.4626	**0.4613**	0.4616	**0.4554**	0.4558	**0.4391**	**0.4391**	**0.4383**	**0.4383**	**0.4357**	0.4358
	192	**0.5112**	0.5115	**0.5123**	0.5128	0.5089	**0.5071**	**0.4943**	0.4945	**0.4960**	0.4961	0.4934	**0.4922**
	336	**0.5595**	0.5600	0.5626	**0.5622**	0.5582	**0.5555**	**0.5483**	0.5485	**0.5530**	**0.5530**	0.5498	**0.5468**
	720	**0.6926**	0.6970	**0.6938**	0.7003	**0.6891**	0.6918	0.6857	**0.6855**	**0.6908**	0.6937	0.7067	**0.7028**
ETTh2	96	**0.2308**	0.2315	**0.2312**	0.2316	**0.2288**	0.2302	**0.2363**	0.2366	**0.2366**	0.2370	**0.2338**	0.2341
	192	0.2830	**0.2818**	**0.2820**	0.2834	**0.2819**	0.2832	**0.2830**	0.2831	**0.2844**	0.2846	0.2814	**0.2810**
	336	**0.3175**	0.3180	**0.3190**	0.3202	**0.3144**	0.3165	**0.3195**	**0.3195**	**0.3250**	0.3252	**0.3172**	0.3180
	720	**0.3825**	0.3836	**0.3884**	0.3909	**0.3825**	0.3868	**0.3814**	**0.3814**	**0.3968**	0.3971	**0.3788**	0.3862
ETTm1	96	**0.3504**	0.3528	**0.3541**	0.3580	**0.3465**	0.3483	**0.3559**	0.3591	**0.3572**	0.3593	**0.3517**	0.3557
	192	**0.4165**	0.4172	**0.4192**	0.4226	**0.4139**	0.4160	**0.4191**	0.4200	**0.4189**	0.4201	**0.4183**	0.4187
	336	**0.4801**	0.4805	**0.4831**	0.4863	**0.4791**	0.4832	**0.4810**	0.4816	**0.4793**	0.4802	0.4836	**0.4826**
	720	**0.5521**	0.5526	**0.5600**	0.5618	**0.5494**	0.5517	**0.5472**	0.5475	**0.5523**	0.5526	0.5533	**0.5496**
ETTm2	96	**0.1581**	0.1586	**0.1583**	0.1587	**0.1560**	0.1569	**0.1566**	0.1571	**0.1569**	0.1575	**0.1563**	0.1565
	192	**0.1917**	0.1919	**0.1920**	0.1922	**0.1894**	0.1901	**0.1901**	0.1904	**0.1908**	0.1912	**0.1901**	0.1903
	336	**0.2299**	0.2301	**0.2303**	0.2304	0.2286	**0.2285**	**0.2291**	0.2293	**0.2312**	0.2314	0.2294	**0.2287**

Dataset	Horizon	DLinear						FreTS					
		TAFAS		PETSA		FAC		TAFAS		PETSA		FAC	
		In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out
	720	0.2993	**0.2987**	0.2984	**0.2968**	0.2984	**0.2980**	**0.2924**	0.2926	**0.2983**	0.2984	0.2955	**0.2934**
Weather	96	**0.1803**	0.1839	**0.1812**	0.1876	**0.1797**	0.1881	**0.1757**	0.1787	**0.1795**	0.1814	**0.1678**	0.1764
	192	**0.2265**	0.2282	**0.2308**	0.2313	**0.2231**	0.2298	**0.2167**	0.2186	**0.2229**	0.2240	**0.2114**	0.2175
	336	**0.2726**	0.2737	**0.2776**	0.2778	**0.2717**	0.2777	**0.2661**	0.2669	**0.2720**	**0.2720**	**0.2621**	0.2661
	720	**0.3509**	0.3542	**0.3557**	0.3565	**0.3426**	0.3482	**0.3423**	0.3426	**0.3550**	0.3554	**0.3376**	0.3392
Exchange	96	**0.0875**	0.0892	**0.0898**	0.0904	**0.0875**	0.0887	**0.0796**	0.0809	**0.0825**	0.0827	**0.0810**	0.0828
	192	**0.1732**	0.1758	**0.1787**	0.1802	0.1758	**0.1740**	**0.1654**	0.1673	**0.1729**	0.1731	0.1667	**0.1635**
	336	**0.3052**	0.3083	**0.3154**	0.3188	**0.2985**	0.3125	**0.3180**	0.3197	**0.3223**	0.3230	0.3166	**0.3017**

Dataset	Horizon	iTransformer						OLS						PatchTST					
		TAFAS		PETSA		FAC		TAFAS		PETSA		FAC		TAFAS (PatchTST)		PETSA		FAC	
		In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out
ETTh1	96	0.4421	**0.4420**	**0.4394**	0.4398	0.4400	**0.4378**	**0.4428**	0.4454	**0.4449**	0.4459	**0.4382**	0.4390	0.4270	**0.4260**	0.4255	**0.4253**	0.4241	**0.4238**
	192	0.5020	**0.5016**	**0.5014**	0.5017	0.5010	**0.4987**	**0.4929**	0.4951	**0.5007**	0.5016	**0.4921**	0.4926	0.4818	**0.4811**	**0.4818**	**0.4818**	0.4797	**0.4790**
	336	**0.5624**	0.5625	0.5641	**0.5639**	0.5587	**0.5561**	**0.5427**	0.5443	**0.5488**	0.5492	0.5422	**0.5405**	0.5506	**0.5505**	**0.5570**	0.5572	0.5525	**0.5495**
	720	**0.6755**	0.6772	**0.6763**	0.6783	**0.6687**	0.6718	**0.6704**	0.6747	**0.6805**	0.6835	**0.6780**	0.6832	0.6807	**0.6802**	0.7019	**0.7012**	0.7096	**0.6964**
ETTh2	96	**0.2620**	0.2623	0.2602	**0.2593**	**0.2561**	**0.2561**	**0.2289**	0.2296	**0.2293**	0.2299	**0.2274**	0.2280	0.2380	**0.2378**	**0.2383**	**0.2383**	**0.2374**	**0.2374**
	192	**0.3004**	0.3011	**0.3005**	0.3008	0.2970	**0.2955**	**0.2821**	0.2826	**0.2829**	0.2833	**0.2796**	0.2804	**0.2752**	0.2753	**0.2810**	0.2817	0.2747	**0.2737**
	336	0.3290	**0.3286**	0.3312	**0.3310**	0.3284	**0.3266**	**0.3189**	0.3201	**0.3240**	0.3245	**0.3137**	0.3163	**0.3108**	**0.3108**	**0.3190**	0.3200	**0.3079**	0.3103
	720	**0.3959**	0.3961	**0.4049**	0.4059	0.3982	**0.3952**	**0.3859**	0.3870	**0.3927**	0.3955	**0.3854**	0.3901	**0.4065**	0.4073	**0.4203**	0.4215	**0.4002**	0.4043
ETTm1	96	**0.3680**	0.3753	**0.3700**	0.3750	**0.3459**	0.3619	**0.3542**	0.3565	**0.3591**	0.3601	**0.3462**	0.3489	**0.3826**	0.3904	**0.3817**	0.3872	**0.3530**	0.3804
	192	**0.4189**	0.4238	**0.4189**	0.4238	**0.4087**	0.4159	**0.4184**	0.4203	**0.4252**	0.4266	**0.4140**	0.4171	0.4370	**0.4367**	0.4405	**0.4403**	**0.4213**	0.4269
	336	**0.4800**	0.4833	**0.4788**	0.4822	**0.4781**	0.4820	**0.4811**	0.4827	**0.4896**	0.4909	**0.4794**	0.4847	**0.4864**	0.4867	**0.4970**	0.4975	**0.4787**	0.4805
	720	**0.5579**	0.5595	**0.5615**	0.5622	**0.5460**	0.5506	**0.5515**	0.5524	**0.5594**	0.5603	**0.5497**	0.5535	0.5413	**0.5402**	**0.5479**	0.5485	**0.5399**	0.5420
ETTm2	96	**0.1640**	0.1642	**0.1640**	0.1642	0.1613	**0.1606**	**0.1594**	0.1595	**0.1595**	**0.1595**	**0.1563**	0.1573	0.1569	0.1571	**0.1569**	0.1573	**0.1558**	0.1562
	192	0.2164	**0.2163**	**0.2157**	0.2164	**0.2067**	0.2094	**0.1924**	0.1925	**0.1925**	0.1927	**0.1897**	0.1907	**0.2058**	0.2063	**0.2079**	0.2084	**0.1976**	0.2018
	336	**0.2735**	0.2758	**0.2653**	0.2748	**0.2623**	0.2700	**0.2308**	0.2309	**0.2316**	0.2318	**0.2289**	0.2293	**0.2482**	**0.2482**	**0.2473**	0.2476	**0.2422**	0.2458
	720	0.3303	**0.3295**	**0.3263**	0.3264	0.3290	**0.3251**	**0.2996**	**0.2996**	**0.3003**	**0.3003**	**0.2973**	0.2985	**0.3218**	0.3223	**0.3240**	0.3242	**0.3192**	0.3196
Weather	96	**0.1662**	0.1674	**0.1662**	0.1686	**0.1605**	0.1635	**0.1825**	0.1855	**0.1890**	0.1914	**0.1728**	0.1843	**0.1700**	0.1713	**0.1676**	0.1721	**0.1635**	0.1682
	192	**0.2129**	0.2136	**0.2165**	0.2171	**0.2103**	0.2113	**0.2238**	0.2255	**0.2293**	0.2304	**0.2171**	0.2249	**0.2119**	0.2130	**0.2120**	0.2137	**0.2050**	0.2088
	336	**0.2629**	0.2632	**0.2672**	0.2675	0.2618	**0.2599**	**0.2729**	0.2740	**0.2785**	**0.2785**	**0.2665**	0.2721	**0.2659**	0.2667	**0.2703**	0.2716	**0.2570**	0.2614
	720	**0.3418**	0.3422	**0.3470**	0.3474	**0.3357**	0.3365	**0.3467**	0.3475	**0.3598**	0.3604	**0.3401**	0.3424	0.3356	**0.3348**	**0.3529**	0.3543	**0.3341**	0.3357
Exchange	96	**0.0873**	0.0877	**0.0881**	**0.0881**	0.0867	**0.0863**	**0.0799**	0.0805	**0.0813**	**0.0813**	**0.0793**	0.0794	**0.0816**	0.0825	**0.0850**	0.0853	**0.0813**	0.0821
	192	**0.1800**	0.1820	**0.1820**	0.1836	0.1773	**0.1756**	**0.1650**	0.1669	**0.1719**	0.1722	**0.1648**	0.1677	0.1666	**0.1659**	**0.1765**	0.1769	**0.1692**	0.1718
	336	**0.3014**	0.3023	**0.3105**	0.3110	**0.2983**	0.3026	0.2846	**0.2823**	**0.3058**	0.3093	**0.2924**	0.3009	0.3048	**0.2763**	**0.3188**	0.3261	**0.3011**	0.3149

Table A.3: Trainable parameter counts for input-output and output-only calibration. For each method, "In+Out" denotes the standard input-output variant and "Out" denotes the output-only variant. The ETT datasets share the same parameter counts. For TAFAS, the PatchTST-specific implementation is listed separately.

Dataset	H	**TAFAS**		**TAFAS (PatchTST)**		**PETSA**		**FAC (Ours)**	
		In+Out	Out	In+Out	Out	In+Out	Out	In+Out	Out
ETT (h1/2, m1/2)	96	130,382	65,191	18,626	9,313	25,934	12,967	**2,758**	**1,379**
	192	324,590	259,399	46,370	37,057	38,894	25,927	**4,102**	**2,723**
	336	857,822	792,631	122,546	113,233	58,334	45,367	**6,118**	**4,739**
	720	3,699,038	3,633,847	528,434	519,121	110,174	97,207	**11,494**	**10,115**
Weather	96	391,146	195,573	18,626	9,313	71,658	35,829	**8,274**	**4,137**
	192	973,770	778,197	46,370	37,057	107,466	71,637	**12,306**	**8,169**
	336	2,573,466	2,377,893	122,546	113,233	161,178	125,349	**18,354**	**14,217**
	720	11,097,114	10,901,541	528,434	519,121	304,410	268,581	**34,482**	**30,345**
Exchange	96	149,008	74,504	18,626	9,313	29,200	14,600	**3,152**	**1,576**
	192	370,960	296,456	46,370	37,057	43,792	29,192	**4,688**	**3,112**
	336	980,368	905,864	122,546	113,233	65,680	51,080	**6,992**	**5,416**

A.4 Runtime and Parameter Efficiency

Table A.4: Runtime and parameter comparison on Weather with the longest forecasting horizon $H = 720$, a relatively demanding setting in our benchmark. Adaptation time, reported in milliseconds, includes calibration, loss computation, backward update, and period estimation, excluding the source forecaster forward pass. Overall runtime is reported in seconds. Results are averaged over 10 repeated runs.

Source Forecaster	Method	Trainable Parameters	MSE	Adaptation Time (ms)	Overall Runtime (s)
DLinear	TAFAS	11,097,114	0.3509	3.60 ± 0.25	3.49 ± 0.14
	PETSA	304,410	0.3557	6.88 ± 0.63	3.85 ± 0.23
	FAC (Ours)	**34,482**	**0.3426**	4.25 ± 0.41	3.55 ± 0.15
FreTS	TAFAS	11,097,114	0.3423	5.81 ± 0.18 **ms**	7.57 ± 0.18 s
	PETSA	304,410	0.3550	21.35 ± 0.56 ms	8.38 ± 0.24 s
	FAC (Ours)	**34,482**	**0.3376**	6.01 ± 0.39 ms	7.27 ± 0.24 **s**
iTransformer	TAFAS	11,097,114	0.3418	7.29 ± 1.15 **ms**	5.56 ± 0.12 **s**
	PETSA	304,410	0.3470	10.32 ± 0.58 ms	6.67 ± 1.50 s
	FAC (Ours)	**34,482**	**0.3357**	7.92 ± 1.35 ms	5.56 ± 0.22 s
OLS	TAFAS	11,097,114	0.3467	3.10 ± 0.31 **ms**	3.67 ± 0.29 **s**
	PETSA	304,410	0.3598	6.79 ± 0.46 ms	4.07 ± 0.15 s
	FAC (Ours)	**34,482**	**0.3401**	3.54 ± 0.28 ms	3.79 ± 0.23 s
PatchTST	TAFAS (PatchTST)	528,434	0.3356	2.97 ± 0.12 **ms**	20.76 ± 0.29 **s**
	PETSA	304,410	0.3529	95.70 ± 1.98 ms	27.78 ± 0.40 s
	FAC (Ours)	**34,482**	**0.3341**	18.16 ± 0.29 ms	27.21 ± 0.20 s

Note. TAFAS and PETSA are evaluated under the same protocol as FAC, where adaptation uses only matured ground truth. For PatchTST, TAFAS follows its original backbone-specific implementation, which uses a lightweight single-channel calibration module and skips input calibration. For frequency-domain calibration methods, including FAC and our PETSA variant, we use multivariate calibration across variables.

Runtime efficiency is important in practice, but reducing the number of trainable parameters does not necessarily yield proportional wall-clock speedups. According to Table A.4, under the same adaptation protocol, TAFAS generally achieves lower runtime overhead, whereas FAC introduces frequency awareness through a compact set of trainable parameters. Since frequency-domain adaptation involves FFT/iFFT operations and additional tensor transformations, FAC and PETSA may incur higher runtime overhead despite using fewer trainable parameters. Compared with PETSA, whose complex adaptation objective includes frequency-domain, distributional, and correlation-based terms, FAC directly parameterizes a lightweight frequency-domain calibration module. As a result, FAC achieves better forecasting accuracy with substantially fewer trainable parameters, while maintaining runtime close to TAFAS and lower than PETSA in most settings, except for PatchTST where TAFAS uses a particularly lightweight single-channel calibration module.

A.5 Additional Frequency-Domain Correction Spectra

We provide additional frequency-domain correction spectra across datasets and forecasting horizons to complement the analysis in Figure 4. Recall that these spectra are not intended to measure forecasting error. Instead, they characterize the magnitude and structure of the prediction changes induced by adaptation in the frequency domain.

For each adaptation method $q \in \{\text{FAC}, \text{TAFAS}, \text{PETSA}\}$, we first compute the correction for each collected prediction window. Specifically, we denote the correction as

$$\Delta\hat{\mathbf{Y}}^{q,[n]} = \hat{\mathbf{Y}}^{q,[n]} - \hat{\mathbf{Y}}^{[n]}, \qquad n = 1, \ldots, N,$$

where $\hat{\mathbf{Y}}^{[n]}$ denotes the prediction before adaptation and $\hat{\mathbf{Y}}^{q,[n]}$ denotes the prediction after applying method q to the n-th prediction window. We then concatenate the corrections from all prediction windows along the prediction-window dimension, yielding

$$\Delta\hat{\mathbf{Y}}^{q} \in \mathbb{R}^{N \times H \times C},$$

where N is the total number of prediction windows, H is the forecasting horizon, and C is the number of channels. Next, we apply rFFT along the forecasting-horizon dimension and take the magnitude as

$$\mathbf{A}^{q} = \left| \mathrm{rFFT}_{H}\left(\Delta\hat{\mathbf{Y}}^{q}\right) \right| \in \mathbb{R}^{N \times F \times C},$$

where $F = \lfloor H/2 \rfloor + 1$ is the number of rFFT frequency components. The plotted curve is obtained by averaging over the prediction windows and channels as

$$\bar{\mathbf{a}}^q = \mathrm{Avg}_{N,C}\left(\mathbf{A}^q_{f>0}\right) \in \mathbb{R}^{F-1}.$$

Here, $f > 0$ indicates that the DC component is omitted.

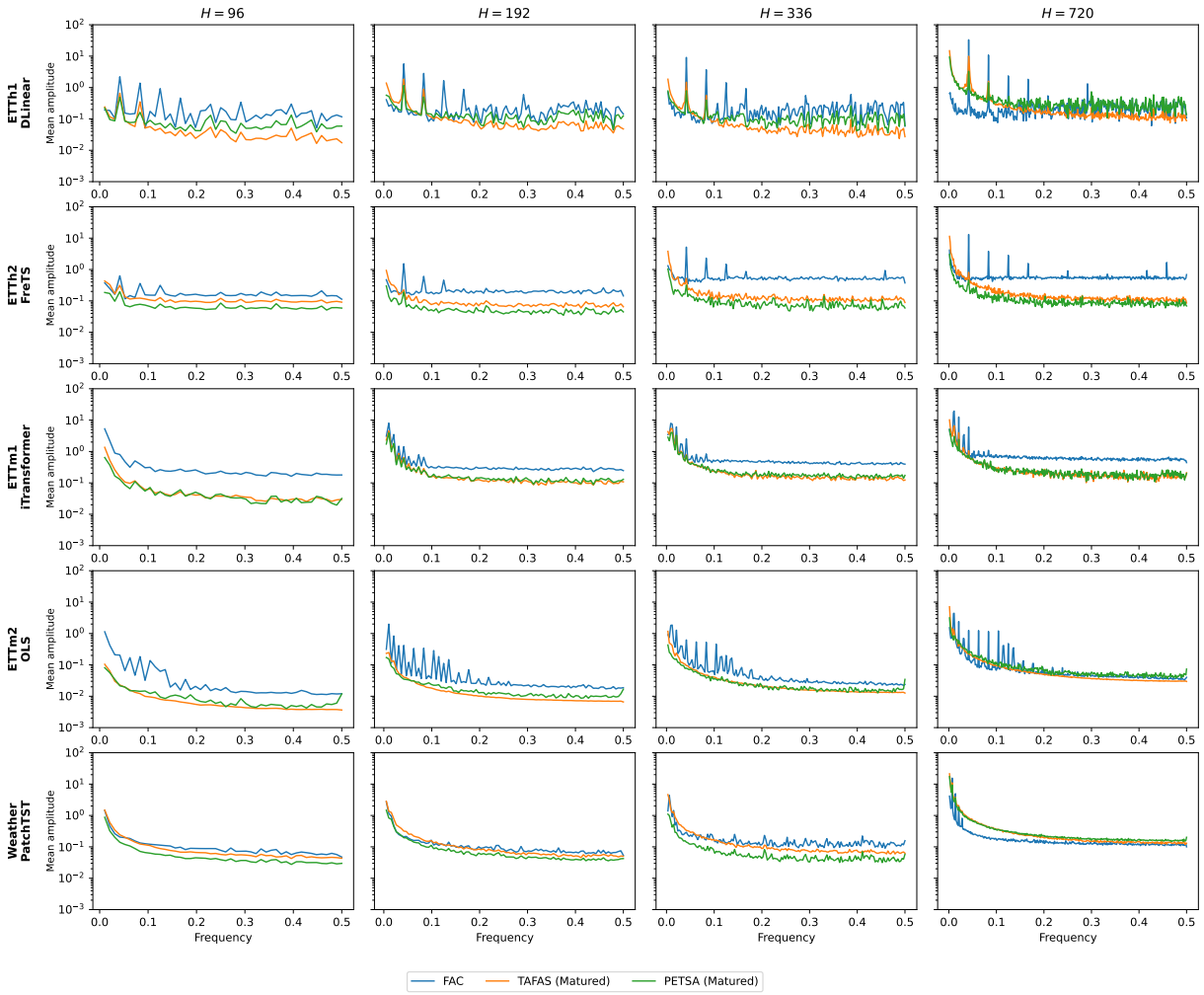

Figure A.2: Additional frequency-domain correction spectra across forecasting horizons. For each adaptation method, we compute the correction, apply rFFT to this correction along the forecasting horizon, and plot the spectral magnitude averaged over prediction windows and channels. The DC component is omitted. Columns correspond to $H \in \{96, 192, 336, 720\}$, and rows correspond to dataset–forecaster pairs ordered alphabetically by dataset name. The y-axis is shown on a logarithmic scale. We omit Exchange because no matured mini-batch is available under the $H = 720$ setting in our protocol.

Figure A.2 shows that the qualitative behavior observed in the main text is not limited to a single forecasting horizon. Across diverse settings, FAC produces more localized spectral corrections than TAFAS and PETSA, whose corrections often vary more smoothly across frequencies. At the same time, some cases, such as Weather with PatchTST, show smaller differences among the methods, indicating that the spectral behavior also depends on the source forecaster and dataset.