

000 001 002 003 004 005 THE FORECAST AFTER THE FORECAST: A POST- 006 PROCESSING SHIFT IN TIME SERIES 007 008 009

010 **Anonymous authors**
 011 Paper under double-blind review
 012
 013
 014
 015
 016
 017
 018
 019
 020
 021
 022
 023
 024
 025
 026
 027

ABSTRACT

011 Time series forecasting has long been dominated by advances in model archi-
 012 tecture, with recent progress driven by deep learning and hybrid statistical tech-
 013 niques. However, as forecasting models approach diminishing returns in accu-
 014 racy, a critical yet underexplored opportunity emerges: the strategic use of post-
 015 processing. In this paper, we address the last-mile gap in time-series forecasting,
 016 which is to improve accuracy and uncertainty without retraining or modifying a
 017 deployed backbone. We propose δ -Adapter, a lightweight, architecture-agnostic
 018 way to boost deployed time series forecasters without retraining. δ -Adapter learns
 019 tiny, bounded modules at two interfaces: input nudging (soft edits to covariates)
 020 and output residual correction. We provide local descent guarantees, $O(\delta)$ drift
 021 bounds, and compositional stability for combined adapters. Meanwhile, it can act
 022 as a feature selector by learning a sparse, horizon-aware mask over inputs to se-
 023 lect important features, thereby improving interpretability. In addition, it can also
 024 be used as a distribution calibrator to measure uncertainty. Thus, we introduce
 025 a Quantile Calibrator and a Conformal Corrector that together deliver calibrated,
 026 personalized intervals with finite-sample coverage. Our experiments across di-
 027 verse backbones and datasets show that δ -Adapter improves accuracy and calibra-
 028 tion with negligible compute and no interface changes.

029 1 INTRODUCTION 030

031 Time Series Forecasting (TSF) powers decisions across energy Anderson (1976), finance Hyndman
 032 & Athanasopoulos (2018), retail Piccolo (1990), transportation Gardner Jr (1985), and the sciences
 033 Piccolo (1990); Gardner Jr (1985). Despite impressive gains from modern neural forecasters Ekam-
 034 baram et al. (2024); Hollmann et al. (2025); Liang (2025); Liu et al. (2025), ranging from temporal
 035 convolutions Lea et al. (2016); Wu et al. (2019; 2022); Li et al. (2023) and Transformers Zhou et al.
 036 (2021); Nie et al. (2022); Liu et al. (2022b); Nie et al. (2023); Wang et al. (2024a); Liu et al. (2023);
 037 Ye et al. (2024); Wang et al. (2024a:b) to hybrid statistical–neural models Liu et al. (2025); Ekam-
 038 baram et al. (2024), condition drift Baier et al. (2020) is still not alleviated. Conventional remedies,
 039 e.g., full fine-tuning, architectural changes, or ensembling, either demand substantial compute, risk
 040 destabilizing a hardened system, or complicate operations. To cope with this, testing-time adapta-
 041 tion (TTA) is introduced into TSF. The testing-time methods aim to mitigate test-time concept drift
 042 via selective layer retraining Chen et al. (2024), online linear adapter updates Kim et al. (2025),
 043 auxiliary loss Medeiros et al. (2025), dynamic gating Grover & Etemad (2025), parallel forecaster
 044 combines Lee et al. (2025), layer-wise adjustment and memory Pham et al. (2023), and dynamic
 045 model selection Wen et al. (2023). However, these methods rely, to varying degrees, on future labels
 046 for online model updates, thereby introducing label leakage, where future ground-truth labels are
 047 unavailable when actually applied, that causes model performance degradation Liang et al. (2024);
 048 yee Ava Lau et al. (2025). Furthermore, LoRA-style adapters Hu et al. (2022); Pfeiffer et al. (2020);
 049 Li & Liang (2021) in NLP tend to lead to high performance variance, since the output range is not
 fixed Biderman et al. (2024).

050 Thus, TSF in real deployments still faces the last-mile gap: 1) Conditions drift Baier et al. (2020),
 051 which refers to gradual changes in the data-generating process (e.g., seasonal regime shifts, covariate
 052 shifts in demand patterns) that occur after the model has been deployed, making full retraining
 053 costly; 2) High performance variance. Existing post-processing techniques are prone to have high
 performance variance due to unstable training; 3) Inefficient training/inference. Using complex

modules or frequent updates to absorb low-complexity residuals Vovk et al. (2017; 2018) makes models suffer from inefficient training/inference. Based on these, we ask a different question: *Can we really keep the strong forecaster intact and learn only a tiny, post-hoc module that makes small targeted corrections, so accuracy and reliability improve without heavy retraining?*

We answer “yes” with δ -Adapter, a lightweight, model-agnostic framework that augments a frozen forecaster F by learning a tiny adapter A in two minimal placements: input-side nudging (softly editing covariates before inference) and output-side correction (residual refinement after inference). Concretely, we instantiate additive or multiplicative forms for both placements, with a small trust-region parameter $\delta \in (0, 1)$ that bounds edits for safety and stability. Since A is a tiny network (e.g., shallow MLP or low-rank head) trained while F remains frozen, it produces consistent gains with negligible training time and zero changes to F ’s inference interface.

Further, a key instantiation of the input adapter is a feature-selector (mask) adapter that learns a sparse, nearly binary, horizon-aware mask $M \in [0, 1]^{L \times d}$ and applies it multiplicatively to the context $X' = X \odot M$. We train M end-to-end with sparsity, temporal-smoothness, and budget regularizers so that the adapter preserves the base model’s inductive biases while exposing the most consequential inputs for the frozen forecaster. This yields transparent selections, stable training, and strong empirical gains under tight compute budgets.

Beyond point accuracy, δ -Adapter also upgrades forecast uncertainty without modifying F . We present two distributional correctors: (i) a Quantile Calibrator that learns horizon-wise quantile functions as bounded offsets from the point forecast, with a monotonic parameterization and pinball-loss training augmented by reliability regularization; and (ii) a Conformal Calibrator that learns a scale function for normalized-residual conformal prediction, delivering finite-sample coverage with personalized, heteroscedastic intervals. Empirically, both calibrators achieve state-of-the-art coverage quality and produce tight, well-behaved intervals.

Through δ -Adapter, this “last-mile” adjustment consistently improves forecasting accuracy in our experiments across diverse backbones and datasets, with negligible training time and no change to inference interfaces. The main contributions are:

- We formalize δ -Adapter and instantiate two placements (input nudging and output residual correction) in additive/multiplicative forms, all drop-in and architecture-agnostic.
- We introduce a learnable, budgeted mask that identifies and preserves the most consequential inputs, improving transparency and stability.
- We propose quantile and conformal calibrators that deliver calibrated, heteroscedastic uncertainty with finite-sample coverage guarantees, all while keeping F frozen.
- Across diverse backbones and benchmarks, δ -Adapter improves accuracy and calibration; ablations illuminate the roles of δ , capacity, horizon features, and residual structure.

2 METHODOLOGY

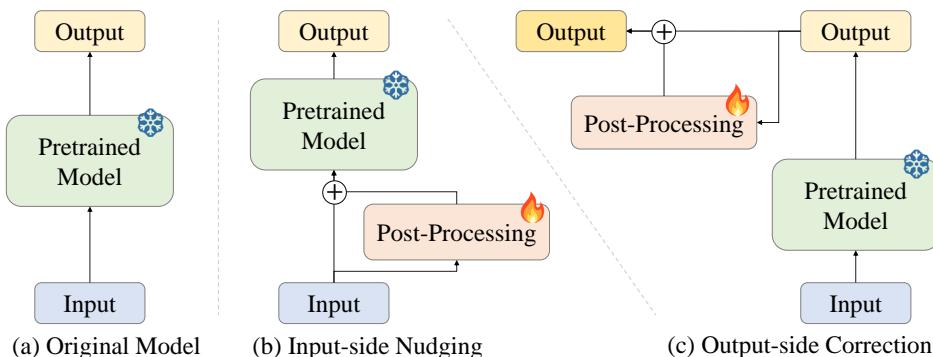


Figure 1: δ -Adapter performs input nudging and output correction on the frozen forecaster.

2.1 PROBLEM SETUP

Let $\mathcal{D} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^N$ denote training pairs of context windows $X \in \mathbb{R}^{L \times d}$ and future targets $Y \in \mathbb{R}^{H \times m}$ (history length L , horizon H , d covariates, m target dimensions). A pre-trained forecaster F maps X to predictions $\hat{Y} = F(X) \in \mathbb{R}^{H \times m}$. We keep all parameters of F fixed and introduce a lightweight, learnable adapter A_θ with parameters θ trained on \mathcal{D} . The adapter composes with F via two families of edits:

$$\textbf{Input-side nudging: } \tilde{X} = X + \delta A_\theta^{\text{in}}(X), \quad (\text{additive input}) \quad (1.1)$$

$$\tilde{X} = X \odot (1 + \delta A_\theta^{\text{in}}(X)), \quad (\text{multiplicative input}) \quad (1.2)$$

$$\textbf{Output-side correction: } \tilde{Y} = F(X) + \delta A_{\theta}^{\text{out}}(F(X), X), \quad (\text{additive output}) \quad (1.3)$$

$$\tilde{Y} = F(X) \odot \left(1 + \delta A_\theta^{\text{out}}(F(X), X)\right), \quad (\text{multiplicative output}) \quad (1.4)$$

The base risk of F under a loss ℓ is

$$\mathcal{R}(F) = \mathbb{E}_{(X,y) \sim \mathcal{D}} \left[\ell(\tilde{Y}, Y) \right]. \quad (2)$$

Here, we consider two adapters, trained by minimizing empirical risk over θ with F frozen, as shown in Eq. 1. The key questions are: (i) when does a small δ provably help; (ii) why do lightweight adapters suffice; and (iii) How do we choose δ and what is the stability of the adapter A ? Now, let's answer these questions.

2.2 OUTPUT-SIDE ADAPTERS AS SHRINKAGE RESIDUAL LEARNING

Here, we consider the additive adapter, as shown in Eq. 1.3: $\tilde{Y} = F(X) + \delta A_\theta^{\text{out}}(F(X), X)$. With slight modifications, the relevant analyses and theories also apply to multiplicative adapters.

Let $r(X) = Y - F(X)$ denote the residual process. For squared error $\ell(\hat{Y}, Y) = \frac{1}{2} \|\hat{Y} - Y\|_2^2$, the population risk of the output adapter with a fixed F equals

$$\mathcal{R}_{\text{out}}(\delta) = \frac{1}{2} \mathbb{E} \left[\|r(X) - \delta g(X)\|_2^2 \right], \quad g(X) := A_\theta^{\text{out}}(F(X), X). \quad (3)$$

Expanding.

$$\mathcal{R}_{\text{out}}(\delta) = \frac{1}{2}\mathbb{E}[\|r\|^2] - \delta \underbrace{\mathbb{E}[\langle r, g \rangle]}_{\text{signal alignment}} + \frac{1}{2}\delta^2\mathbb{E}[\|g\|^2]. \quad (4)$$

Proposition 2.1 (Small-step improvement). *If $\mathbb{E}[\langle r, q \rangle] > 0$, then for all*

$$0 < \delta < \frac{2\mathbb{E}[\langle r, g \rangle]}{\mathbb{E}[\|g\|^2]}, \quad (5)$$

we have $\mathcal{R}_{\text{out}}(\delta) < \mathcal{R}_{\text{out}}(0) = \frac{1}{2}\mathbb{E}[||r||^2]$. The quadratic in δ has negative derivative at 0 and a unique minimizer $\delta^* = \frac{\mathbb{E}[\langle r, g \rangle]}{\mathbb{E}[||a||^2]}$.

Remark. Improvement hinges on alignment between the learned correction g and the residual r . Even when A is tiny, if residuals have low-complexity structure (calendar offsets, horizon-dependent bias, scale drift), a small g can achieve positive alignment, and a shrunken step δ guarantees risk reduction. This is exactly the first step of boosting with shrinkage or a stacked residual learner with a conservative learning rate.

In practice, we learn g from finite data with a penalty $\Omega(\theta)$ (e.g., ℓ_2 , low rank, sparsity). The empirical objective

$$\min_{\theta} \frac{1}{2} \sum_i \|y_i - F(X_i) - \delta g_{\theta}(X_i)\|^2 + \lambda \Omega(\theta) \quad (6)$$

yields a shrunken projection of residuals onto the function class of A . With small δ and a low-capacity A , we target the dominant residual modes while avoiding variance blow up.

162 2.3 INPUT-SIDE ADAPTERS VIA FIRST-ORDER LINEARIZATION
163

164 For the input-nudging adapter, as shown in Eq. 1.1: $\tilde{X} = X + \delta A_\theta^{\text{in}}(X)$, apply a first-order expansion
165 of F around X :

$$166 F(X + \delta u(X)) \approx F(X) + \delta J_F(X)u(X), \quad (7)$$

167 where $u(X) := A(X, h)$ and $J_F(X) \in \mathbb{R}^{H \times d}$ is the Jacobian of F w.r.t. inputs. Under squared
168 loss, replacing g by $J_F u$ in the previous derivation yields

$$169 170 \mathcal{R}_{\text{in}}(\delta) \approx \frac{1}{2}\mathbb{E}[\|r\|^2] - \delta\mathbb{E}[\langle r, J_F u \rangle] + \frac{1}{2}\delta^2\mathbb{E}[\|J_F u\|^2]. \quad (8)$$

171 In general, for $\hat{y}_{\text{in}}(X; \delta) = F(X + \delta u(X))$, $\mathcal{R}_{\text{in}}(\delta) = \frac{1}{2}\mathbb{E}[\|y - \hat{y}_{\text{in}}(X; \delta)\|_2^2]$, we have

173 **Proposition 2.2** (General δ -step improvement). *If $\mathbb{E}[\langle r, J_F u \rangle] > 0$, then there exists $\delta > 0$ such
174 that $\mathcal{R}_{\text{in}}(\delta) < \mathcal{R}_{\text{in}}(0)$ for all $\delta \in (0, \delta]$. And, if F is affine in the near of X , Prop. 2.1 is also hold.*

175 The proof is given in Appendix B.2. This proposition states that for a differentiable loss ℓ ,
176 the loss gradient w.r.t. inputs satisfies $\nabla_x \ell(F(X), y) = J_F(X)^\top \nabla_y \ell$. Choosing $u(X) \approx
177 -B\nabla_x \ell(F(X), y)$ for a small, learned preconditioner B recovers a learned, damped gradient step
178 in input space; training A on data finds such steps implicitly without computing J_F^\top at test time.
179

180 3 THE STABILITY OF δ -ADAPTER
181183 3.1 PREDICTION STABILITY UNDER BOUNDED INPUT EDITS
184

185 Let $\tilde{X} = X + \delta A_\phi^{\text{in}}(X)$ (additive case). Then, we have

186 **Proposition 3.1** (Drift bound). *Assume the frozen forecaster F is L_F -Lipschitz, the change in pre-
187 diction is bounded by*

$$188 \|\tilde{Y} - \hat{Y}\| \leq \delta L_F \|A_\phi^{\text{in}}(X)\| \leq \delta L_F \sqrt{Ld}. \quad (9)$$

190 The proof is given in Appendix B.3. Further, let $\tilde{X} = X \odot \exp(\delta A_\phi^{\text{in}}(X))$, we have

191 **Corollary 1** (Multiplicative input edits). *If $\|X\|_\infty \leq B_X$, then*

$$193 \|\tilde{Y} - \hat{Y}\| \leq \delta e^\delta L_F B_X \|A_\phi^{\text{in}}(X)\|. \quad (10)$$

194 In particular, for $\delta \leq 1$, $\|\tilde{Y} - \hat{Y}\| = O(\delta)$.

196 The proof is given in Appendix B.4. Corollary 1 means that small δ yields Lipschitz-stable prediction
197 changes for input adapters.

199 3.2 LOSS STABILITY AND GUARANTEED LOCAL IMPROVEMENT
200

201 Let $\hat{Y} = F(X)$ and consider an output edit $\tilde{Y} = \hat{Y} + \delta d$ with $d := A_\phi^{\text{out}}(\hat{Y}, X)$, we have

$$203 204 \ell(\tilde{Y}, y) \leq \ell(\hat{Y}, y) + \delta \langle g, d \rangle + \frac{\beta}{2} \delta^2 \|d\|^2, \quad g := \nabla_u \ell(u, y)|_{u=\hat{Y}}. \quad (11)$$

205 If d aligns with $-g$, i.e. $\langle g, d \rangle \leq -\alpha \|g\| \|d\|$, we get

206 **Theorem 2** (Descent for output adapters). *If the per-sample prediction loss $\ell(\cdot, y)$ is β -smooth in
207 its first argument (e.g., MSE, Huber), for any sample,*

$$208 209 \ell(\tilde{Y}, y) - \ell(\hat{Y}, y) \leq -\delta \alpha \|g\| \|d\| + \frac{\beta}{2} \delta^2 \|d\|^2. \quad (12)$$

211 Hence, for any $\delta \in (0, \frac{2\alpha\|g\|}{\beta\|d\|})$, the loss strictly decreases. The optimal $\delta^* = \frac{\alpha\|g\|}{\beta\|d\|}$ yields

$$213 214 \ell(\tilde{Y}, y) - \ell(\hat{Y}, y) \leq -\frac{\alpha^2}{2\beta} \|g\|^2. \quad (13)$$

215 The proof is given in Appendix B.5.

216 *Remark.* With MSE, $g = \hat{Y} - y$, so the improvement is proportional to the squared residual mag-
 217 nitude. Further, with a bounded adapter family, the trained A^out (minimizing batch loss) produces d
 218 that correlates with $-g$ unless capacity is zero.

219 **Theorem 3** (Descent for input adapters). *Let $\tilde{X} = X + \delta v$ with $v := A_\phi^\text{in}(X)$. Assume F is
 220 differentiable at X with Jacobian $J_F(X)$. Define the effective prediction step $s := J_F(X)v$. Then
 221 for δ small,*

$$\ell(F(\tilde{X}), y) \leq \ell(\hat{Y}, y) + \delta \langle g, s \rangle + \frac{\beta}{2} \delta^2 \|d\|^2 + O(\delta^2). \quad (14)$$

224 If $\langle g, s \rangle \leq -\alpha \|g\| \|s\|$, there exists $\bar{\delta} > 0$ such that $\forall \delta \in (0, \bar{\delta})$ the loss strictly decreases. Moreover,
 225 optimizing the quadratic upper bound in δ yields the same margin as Theorem 2 up to $O(1)$ terms.
 226

227 The proof is given in Appendix B.6. Theorems 2 and 3 show that for sufficiently small δ and mild
 228 alignment, both adapter types reduce the loss locally, with explicit improvement margins.

229 3.3 COMPOSITIONAL STABILITY (INPUT + OUTPUT)

231 Let the full edit be $\tilde{X} = X + \delta v$, $\hat{Y}' = F(\tilde{X})$, then $\tilde{Y} = \hat{Y}' + \delta d(\hat{Y}', X)$. Under the same conditions
 232 as Prop. 3.1 and Theorems 2 and 3, we have:

233 **Proposition 3.2** (Composite drift and loss bound).

$$\|\tilde{Y} - \hat{Y}\| \leq \|\hat{Y}' - \hat{Y}\| + \delta \|d(\hat{Y}', X)\| \leq \delta L_F \|v\| + \delta C_d, \quad (15)$$

234 so the model drift is $O(\delta)$. Further, for the loss,

$$\ell(\tilde{Y}, y) \leq \ell(\hat{Y}, y) + \delta \langle g, s + d \rangle + \frac{\beta}{2} \delta^2 \|s + d\|^2 + O(\delta^2), \quad (16)$$

235 The proof is given in Appendix B.7. If the combined step $s + d$ aligns with $-g$ by parameter-sharing
 236 or a learned gate, we inherit the same descent guarantee as Theorem 2.

237 4 IMPLEMENTATION

238 4.1 δ -ADAPTER

239 δ -Adapter targets structured residuals (bias, scale miscalibration, phase lag) while preserving F 's
 240 inductive biases. We encode this through three principles: 1) Boundedness: Enforce small edits via
 241 δ and penalties on $\|A_\theta(\cdot)\|$; 2) Low capacity: Use tiny architectures to avoid overfitting and respect
 242 production budgets. 3) Horizon awareness: Allow horizon-specific corrections without destabilizing
 243 temporal coherence. Concretely, we use a tiny MLP as the backbone and impose:

$$\|A_\theta^\text{in}(X)\|_\infty \leq 1, \quad \|A_\theta^\text{out}(\cdot)\|_\infty \leq 1, \quad (17)$$

244 via tanh squashing and optional clipping, so that δ is a direct bound on the maximum per-entry
 245 change. For multiplicative edits we ensure positivity where required by applying $\exp(\delta A_\theta(\cdot))$ as
 246 an alternative to $1 + \delta A_\theta$. For compositional adapters (input+output), as stated in Prop. 3.2, their
 247 parameters can be optimized in parallel during the training process.

248 4.2 FEATURE SELECTOR

249 A particularly transparent instantiation of our input adapter is to cast it as a learnable mask (selector)
 250 that selects the parts of the input that are most consequential for the frozen forecaster F . Concretely,
 251 for a context window $X \in \mathbb{R}^{L \times d}$, we parametrize an adapter A_θ that outputs a mask $M(X; \theta) \in$
 252 $[0, 1]^{L \times d}$, and apply it multiplicatively,

$$X' = X \odot M(X; \theta). \quad (18)$$

253 The mask is trained end-to-end while keeping F fixed. Intuitively, M plays the role of a soft selector:
 254 values near 1 keep information intact, values near 0 suppress it. To obtain discrete, human-readable
 255 selections without sacrificing differentiability, we employ relaxed Bernoulli parameterizations. Let
 256 $\alpha(X; \theta) \in \mathbb{R}^{L \times d}$ be adapter logits. We form a Gumbel-Sigmoid (Concrete) relaxation

$$M(X; \theta, \tau) = \sigma\left(\frac{\log \alpha(X; \theta) + G}{\tau}\right), \quad (19)$$

where G is i.i.d. Gumbel noise, $\sigma(\cdot)$ is the logistic function, and $\tau > 0$ is a temperature annealed from a high value (smooth masks) to a low value (nearly binary). At inference, we may harden the mask via a threshold $M_{\text{hard}} = \mathbf{1}\{M > 0.5\}$ or keep it soft to avoid distributional brittleness. As a simpler alternative, we use a straight-through estimator: threshold in the forward pass, back-propagate through the corresponding sigmoid in the backward pass. Training the mask as a selector requires explicit structure in the objective. Given predictions $\tilde{Y} = F(X \odot M)$, we minimize

$$\min_{\theta} \underbrace{\mathcal{L}_{\text{pred}}(\tilde{Y}, Y)}_{\text{forecasting error}} + \lambda_1 \underbrace{\|M\|_1}_{\text{sparsity}} + \lambda_{\text{ent}} \underbrace{\sum H(M_{t,j})}_{\text{low entropy}} + \lambda_{\text{tv}} \underbrace{\text{TV}(M)}_{\text{temporal smoothness}} + \lambda_{\text{bud}} \underbrace{(\bar{m} - \kappa)_+}_{\text{budget}}, \quad (20)$$

where $\bar{m} = \frac{1}{Ld} \sum_{t,j} M_{t,j}$ is the average keep-rate and $\kappa \in (0, 1]$ is a user-specified budget, which stabilizes selection under correlations by constraining the feasible keep set, e.g., use at most 10% of inputs. The ℓ_1 and entropy terms encourage sparse, nearly binary masks; the total-variation penalty $\text{TV}(M)$ promotes temporal contiguity, reflecting the fact that relevant patterns often span short intervals rather than isolated instants. See the specific expressions of each part in Appendix C.3.

4.3 DISTRIBUTION CALIBRATOR

Now, we introduce how to use the proposed adapter as a calibrator when the forecaster F is frozen and produces only fixed-point predictions.

4.3.1 QUANTILE CALIBRATOR

If a distributional assumption is undesirable, the adapter can directly output horizon-wise quantiles as bounded offsets from the point forecast:

$$q_{\tau,\theta}(X) = \hat{Y} + \varepsilon a_{\theta}(X, \hat{Y}, \tau) \odot s_{\theta}(X, \hat{Y}), \quad (21)$$

where $a_{\theta} \in [-1, 1]^{H \times m}$ and $s_{\theta} > 0$ is a learned scale. To ensure monotonicity in τ , we parameterize

$$q_{\tau_{j+1},\theta} = q_{\tau_j,\theta} + \text{softplus}(d_{j,\theta}(X, \hat{Y})), \quad \tau_1 < \tau_2 < \dots < \tau_J, \quad (22)$$

where $d_{j,\theta}$ is the adapter's raw increment for the gap between two adjacent quantile levels τ_j and τ_{j+1} . Eq. 22 anchored at a central level (e.g., $\tau_{J/2}$) via the bounded offset around \hat{Y} . Then, for the training objective, we replace the point losses with pinball loss and add reliability regularization:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^J \ell_{\tau_j}(Y^{(i)}, q_{\tau_j,\theta}(X^{(i)})) + \lambda_{\text{cal}} \mathcal{C}_{\text{rel}}(\theta) + \lambda_{\text{mag}} \|a_{\theta}\|_2^2. \quad (23)$$

where ℓ_{τ} is the pinball loss; \mathcal{C}_{rel} can be the same soft-coverage penalty as above, or a PIT-uniformity term computed by interpolating the predicted quantiles into a differentiable CDF and matching the PIT distribution to Uniform(0, 1).

4.3.2 CONFORMAL CALIBRATOR

When strict distribution-free guarantees are needed, we combine a learned scale function with conformal prediction, i.e., we train $w_{\theta}(X, \hat{Y}) > 0$ (small adapter) to predict residual magnitude while keeping the mean at \hat{Y} :

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^N |Y^{(i)} - \hat{Y}^{(i)}| / w_{\theta}(X^{(i)}, \hat{Y}^{(i)}) + \lambda \|w_{\theta}\|_2^2, \quad (24)$$

subject to a mild regularizer to keep w_{θ} near 1 on average. Then, we can use conformal scaling on a held-out calibration set \mathcal{D}_{cal} to compute normalized residuals as

$$r^{(i)} = \frac{\|Y^{(i)} - \hat{Y}^{(i)}\|}{w_{\theta}(X^{(i)}, \hat{Y}^{(i)})}, \quad (X^{(i)}, Y^{(i)}) \in \mathcal{D}_{\text{cal}}. \quad (25)$$

Then, the calibrated marginally valid prediction sets can be obtained by

$$\mathcal{C}_{\alpha}(X) = \{y : \|y - \hat{Y}\| \leq \kappa_{\alpha} w_{\theta}(X, \hat{Y})\}, \quad (26)$$

where κ_{α} is the empirical $(1 - \alpha)$ -quantile of $\{r^{(i)}\}$. This yields finite-sample coverage $1 - \alpha$ under exchangeability. The adapter w_{θ} personalizes interval width while F remains untouched.

324 **5 EXPERIMENTS**

325

326 We validate the δ -Adapter method on a variety of widely used datasets, see Appendix C.1. We test its
 327 gains when applied to pre-trained and state-of-the-art (SOTA) models (Section 5.1), its application
 328 as a feature selector (Section 5.2), and its effectiveness as an interval calibrator (Section 5.3). In this
 329 paper, we set $\delta = 0.1$ (0.01 for ETT datasets) and the learning rate of Adam to 1E-4, and conduct
 330 an ablation study on them at Section 5.4.

331 **5.1 EFFECTIVENESS OF δ -ADAPTER**

332

333 Table 1: The improvement of δ -Adapter on Pre-Trained models.

Model	Sundial-S (Univariate)								TTM-R2 (Multivariate)								
	original		Ada-X			Ada-Y			original		Ada-X			Ada-Y			
Type	MSE	MAE	MSE	MAE	IMP	MSE	MAE	IMP	MSE	MAE	IMP	MSE	MAE	IMP	MSE	MAE	IMP
Dataset																	
ELC	0.427	0.463	0.334	0.410	17%	0.404	0.451	4%	0.180	0.272	0.167	0.262	6%	0.168	0.262	5%	
Traffic	0.237	0.314	0.220	0.301	6%	0.224	0.302	5%	0.517	0.344	0.492	0.329	5%	0.492	0.325	5%	
Exchange	0.249	0.332	0.241	0.332	2%	0.235	0.329	3%	0.094	0.213	0.090	0.206	3%	0.092	0.210	1%	
Weather	0.427	0.463	0.025	0.005	96%	0.039	0.059	89%	0.150	0.196	0.148	0.193	2%	0.143	0.191	4%	
ETTm1	0.121	0.217	0.078	0.190	24%	0.087	0.202	18%	0.338	0.357	0.329	0.357	1%	0.331	0.353	3%	
ETTm2	0.348	0.420	0.201	0.325	32%	0.254	0.371	19%	0.177	0.259	0.174	0.243	4%	0.175	0.240	4%	

342 We first verify the performance gains of δ -Adapter on pre-trained models, including Sundial-S (Univariate)
 343 Liu et al. (2025) and TTM-R2 (Multivariate) Ekambaram et al. (2024). The experimental
 344 results in Table 1 show that δ -Adapter consistently enhances forecasting performance across all
 345 datasets and backbone models, confirming its effectiveness and generality. Both the input adapter
 346 (Ada-X) and the output adapter (Ada-Y) have achieved significant performance gains. These results
 347 highlight that training lightweight adapters while keeping the backbone frozen is a powerful and
 348 efficient way to boost predictive accuracy.

349 Table 2: Comparison of various adapter methods and online methods (averaged across all lengths).

Model	DistPred					iTransformer					Autoformer					Others		
	Offline	SOLID	TAFAS	LoRA	Ada-X+Y	Offline	SOLID	TAFAS	LoRA	Ada-X+Y	Offline	SOLID	TAFAS	Ada-X+Y	OneNet [†]	FSNet [†]		
Dataset																		
ELC	0.182	0.182	0.182	0.180	0.175	0.190	0.190	0.190	0.186	0.180	0.515	0.502	0.510	0.478	0.417	0.537		
ETTh1	0.461	0.460	0.476	0.454	0.451	0.454	0.458	0.477	0.448	0.449	0.593	0.589	0.591	0.577	0.618	0.877		
ETTh2	0.390	0.391	0.402	0.385	0.379	0.388	0.393	0.448	0.384	0.377	0.438	0.435	0.436	0.426	0.581	0.587		
ETTm1	0.412	0.406	0.411	0.407	0.396	0.417	0.414	0.420	0.414	0.403	0.664	0.661	0.638	0.597	0.548	0.851		
ETTm2	0.285	0.285	0.288	0.281	0.274	0.300	0.298	0.304	0.293	0.290	0.339	0.339	0.338	0.321	1.171	1.113		
Exchange	0.350	0.347	0.363	0.346	0.297	0.383	0.376	0.392	0.376	0.316	0.509	0.491	0.495	0.465	0.647	0.878		
Traffic	0.453	0.453	0.455	0.449	0.440	0.475	0.475	0.476	0.468	0.461	0.972	0.959	0.975	0.942	0.567	0.701		
Weather	0.256	0.255	0.256	0.251	0.242	0.259	0.257	0.259	0.255	0.244	0.325	0.316	0.325	0.299	0.390	0.541		

[†] OneNet and FSNet are implemented based on the public library provided in their paper with no label leakage. For more details, please refer to Table 10 in Appendix C.9.

350 Then, we compared the proposed δ -Adapter with other adapter methods and online learning methods
 351 by removing label leakage Liang et al. (2024); see Ava Lau et al. (2025). It is worth noting that when
 352 removing label leakage, some methods have a certain degree of performance degradation. This may
 353 be because the design of these methods relies excessively on future true values. Table 2 shows
 354 that the δ -adapter achieves the lowest error on every dataset across all three backbones. The gains
 355 are sizeable on challenging sets, while remaining consistent on the ETT variants. Moreover, when
 356 contrasted with OneNet and FSNet, δ -Adapter paired with standard backbones yields substantially
 357 lower errors on all datasets, underscoring its plug-and-play effectiveness and robustness.

358 Table 3: Gains of δ -Adapter on SOTA models (averaged across all lengths. See Table 9 for details).

Model	DistPred			iTransformer			FourierGNN			FreTS			Autoformer		
	Original	Ada-X	Ada-Y	Original	Ada-X	Ada-Y	Original	Ada-X	Ada-Y	Original	Ada-X	Ada-Y	Original	Ada-X	Ada-Y
Dataset															
ELC	0.182	0.178	0.169	0.190	0.187	0.181	0.267	0.255	0.241	0.209	0.203	0.194	0.515	0.488	0.450
Exchange	0.350	0.302	0.319	0.383	0.348	0.349	0.380	0.393	0.379	0.416	0.412	0.422	0.509	0.481	0.462
Traffic	0.453	0.448	0.442	0.475	0.470	0.461	0.777	0.749	0.740	0.596	0.590	0.572	0.972	0.959	0.918
Weather	0.256	0.251	0.245	0.259	0.249	0.245	0.255	0.251	0.244	0.255	0.249	0.243	0.325	0.306	0.299
ETTh1	0.461	0.457	0.458	0.454	0.453	0.456	0.561	0.546	0.542	0.482	0.474	0.471	0.593	0.583	0.577
ETTh2	0.390	0.386	0.387	0.388	0.385	0.390	0.545	0.499	0.506	0.537	0.492	0.498	0.438	0.420	0.423
ETTm1	0.412	0.399	0.402	0.417	0.407	0.406	0.456	0.447	0.447	0.405	0.401	0.401	0.664	0.604	0.637
ETTm2	0.285	0.279	0.282	0.300	0.292	0.293	0.445	0.386	0.439	0.335	0.285	0.323	0.339	0.316	0.320

358 Next, we verify whether the δ -Adapter provides gains to the SOTA forecaster. Table 3 shows that
 359 δ -Adapter provides consistent and significant improvements across multiple SOTA models. For

nearly all datasets, Ada-X and Ada-Y lead to lower prediction errors compared to the original models, demonstrating that the proposed adapters generalize well to diverse forecasting architectures. Notably, Ada-X again delivers the largest gains, particularly on challenging datasets such as Exchange, Traffic, and ETT series, confirming that refining the input signals before model inference is the most impactful strategy. Also, δ -Adapter yields clear benefits, highlighting its plug-and-play nature and ability to enhance high-performing models. These results further validate that δ -Adapter is a broadly applicable, efficient, and effective enhancement method for modern forecaster.

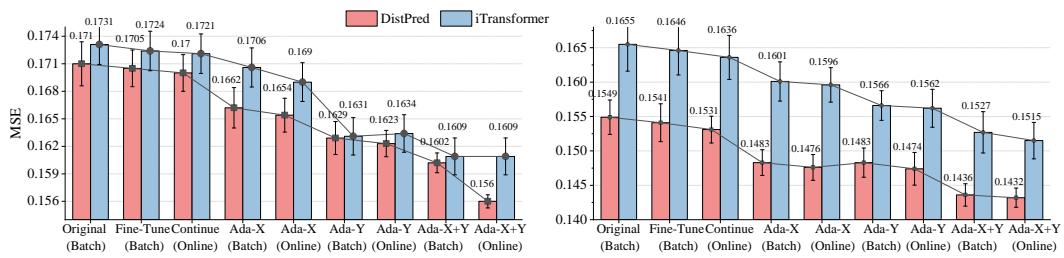


Figure 2: Performances of the forecaster F and δ -Adapter under batch or online training.

Finally, we test whether δ -Adapter is effective under different compositions and training methods. **Implementation and training details of Ada-X+Y are in Appendix C.5.** Figure 2 shows that δ -Adapter consistently reduces error under batch and online training. Each single adapter improves over the frozen forecaster and also outperforms conventional fine-tuning or continue-training. And training the adapters online yields further gains over batch. Importantly, Ada-X+Y delivers the lowest MSE in all settings, indicating robust and statistically reliable improvements.

5.2 EFFECTIVENESS OF THE FEATURE SELECTOR

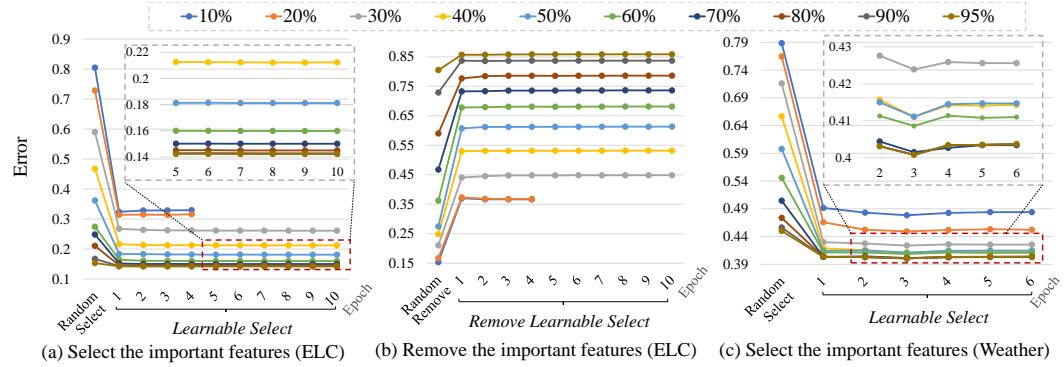


Figure 3: Changes of forecaster’s performance after selecting or removing valid features.

To verify the effectiveness of the mask adapter as a feature selector, we visualized its training process, as shown in Figure 3. It demonstrates that a learnable mask adapter reliably identifies the most informative input features under varying sparsity budgets. In subfigure (a), selected features yields markedly lower errors than random selection across all retention rates (10–95%) and converges within a few epochs. Conversely, when the learned features are removed (b), the forecaster’s error rises substantially, often worse than removing an equal number of randomly chosen features. This shows that these features are uniquely critical to performance rather than incidental. Table 4 shows the mask ratio of the mask adapter when the best performance is achieved (no budget added), and Figure 4 visualizes important features in different proportions (most important features remain unchanged). These confirming that the learned selections consistently outperform random picks, and removing them degrades accuracy the most.

Table 4: Best performance of the mask adapter and its mask ratio.

Dataset	ELC	ETTh1	ETTh2	ETTm1	ETTm2	Traffic	Weather	Exchange
Original	0.163	0.390	0.296	0.345	0.182	0.444	0.173	0.099
Masked	0.159	0.382	0.291	0.334	0.176	0.436	0.171	0.093
Mask Ratio	97%	96%	95%	97%	96%	98%	96%	92%

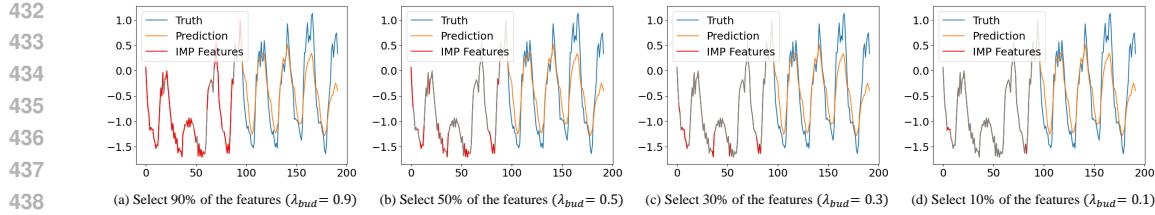


Figure 4: Visualization of different important features learned by the mask adapter.

5.3 PERFORMANCE OF CALIBRATOR

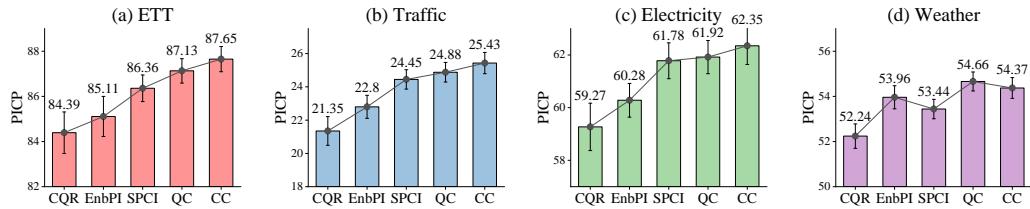


Figure 5: Comparisons among the Quantile (QC), Conformal (CC) calibrators and others.

Now, we verify the effect of δ -Adapter as the Quantile Calibrator (QC) and Conformal Calibrator (CC). As shown in Figure 5, our calibrators consistently deliver the highest PICP, indicating better coverage reliability than strong baselines (CQR Romano et al. (2019), EnbPI Xu & Xie (2021), SPCI Xu & Xie (2023)). Further, in Figure 6, we illustrate that both calibrators produce well-calibrated intervals that expand near peaks and usually enclose the ground truth. QC tends to yield slightly wider, more conservative bands, while CC delivers comparably high coverage with tighter intervals.

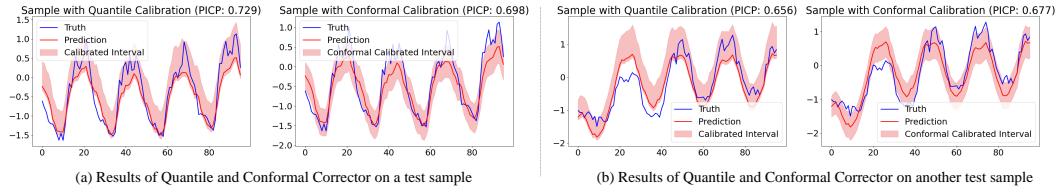
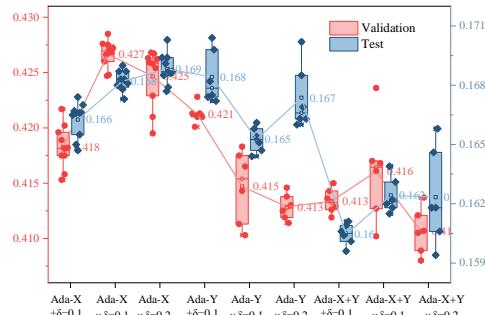


Figure 6: Visualization of the Quantile and Conformal calibrator predictions.

5.4 ABLATION STUDIES

The impact of δ is important. Figure 7 indicates that all adapter variants reduce error versus the frozen model, but the combined adapter (Ada-X+Y) delivers the lowest median errors and the tightest variability. Across placements, a moderate adjustment size is the most reliable, e.g., pushing to $\delta = 0.2$ yields smaller or inconsistent gains, suggesting overly aggressive corrections. It is confirmed that composing input and output adapters with a modest multiplicative trust-region produces the most accurate and stable forecasts.

Then, we used PatchTST Nie et al. (2023) and TimeMixer Wang et al. (2024a) as backbones to compare the performance of additive and multiplicative composite δ -Adapter. As shown in Table 5, after adding the δ -Adapter to PatchTST and TimeMixer, their performance has been significantly improved. the additive and multiplicative adapters reduce the MSE of PatchTST by 5.6% and 5.1%

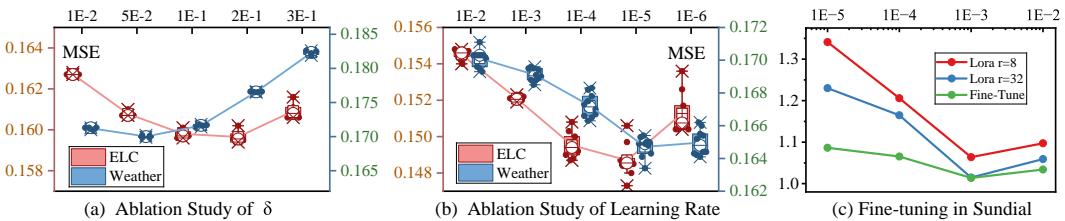
Figure 7: Performance of δ -Adapter's variants.

486 respectively across various datasets. However, for TimeMixer, the MSE reductions from the additive
 487 and multiplicative adapters are 1.6% and 1.8% respectively. This indicates that both have their
 488 respective advantages, and the increase of the additive adapter is relatively more significant.
 489

490 Table 5: Comparison of additive and multiplicative composite δ -Adapter.
 491

	PatchTST		+ Ada-X+Y		+ Ada-X \times Y		TimeMixer		+ Ada-X+Y		+ Ada-X \times Y	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ELC	0.167	0.252	0.159	0.245	0.159	0.246	0.145	0.243	0.143	0.241	0.143	0.241
Weather	0.178	0.219	0.161	0.220	<u>0.165</u>	<u>0.224</u>	0.168	0.216	<u>0.166</u>	<u>0.214</u>	<u>0.164</u>	<u>0.214</u>
Traffic	0.463	0.297	<u>0.451</u>	<u>0.292</u>	0.448	0.290	0.475	0.317	0.465	0.307	0.467	<u>0.310</u>

492 Forecasters are susceptible to hyperparameters. Thus, we investigated the impact of two key factors:
 493 δ and its learning rate. As shown in Figure 8, despite the large variation ranges of δ and the learning
 494 rate, the forecaster (iTransformer) can still maintain relatively stable prediction performance.
 495 However, other models, such as those that attempt to fine-tune pre-trained large models using Lora
 496 Hu et al. (2022), not only exhibit large performance variance but also lead to degradation in per-
 497 formance, which remains a problem worthy of further exploration. These experiments demonstrate
 498 that the proposed δ -Adapter has a stable training process and brings performance gains.
 499

500 Figure 8: Variances of Ada-X with different δ (a) and learning rates (b) and fine-tuned Sundial (c).
 501

502 Finally, we tested the efficiency of δ -Adapter. Table 6 shows that δ -Adapter is the most time-efficient
 503 adaptive method overall, it is consistently faster than other methods across all horizons. This is
 504 because Ada-X+Y itself is lightweight and only uses the most recent single sample to update the
 505 model. Compared to other adapters that use or select a large number of recent samples for updates,
 506 it is obviously faster. The δ -Adapter is designed to be extremely lightweight. Compared to the
 507 backbone model (specifically, for Sundial (128M) and TabPFN (48M), the adapter introduces less
 508 than 2%-6% additional parameters, validating the lightweight claim.
 509

510 Table 6: Time (S) and memory (MB) of adapters (backbone is TabPFN) and online methods.
 511

TabPFN 48M		Ada-X+Y 3M			SOLID 0.5M			TAFAS 6M			OneNet 3M			FSNet 2M		
Time	Memory	Train	Test	Memory	Train	Test	Memory	Train	Test	Memory	Train	Time	Memory	Train	Time	Memory
281	1840	392	395	1983	511	667	2401	603	861	3468	693	471	1512	621	485	1504
307	1848	386	379	2132	481	624	2423	589	895	3790	681	445	1537	618	472	1531
326	1852	385	415	2622	484	593	2446	583	1152	4186	631	452	1559	599	466	1517
351	1856	369	431	3102	505	398	2501	916	1803	6809	530	465	1567	554	458	1526

527

6 CONCLUSION

 528

529 We present δ -Adapter, a lightweight and post-hoc framework that improves frozen forecasters via
 530 bounded input nudges and output residual corrections. we provide theory guaranteeing local descent
 531 and stable composition. To enhance interpretability and robustness, we introduce a feature-selector
 532 adapter that learns a sparse, horizon-aware mask under budget priors, exposing the most consequen-
 533 tial inputs while constraining edits. Beyond point forecasts, we deliver calibrated uncertainty via two
 534 distributional correctors: a Quantile Calibrator that learns quantile offsets trained with pinball loss,
 535 and a Conformal Corrector that estimates heteroscedastic scales for normalized-residual conformal
 536 prediction, yielding finite-sample coverage with personalized intervals. Across diverse backbones
 537 and datasets, δ -Adapter yields consistent accuracy and calibration gains.
 538

540 **ETHICS STATEMENT**
 541

542 Biases in benchmark creation: The authors are aware of the potential for bias in the creation of our
 543 benchmark entries. The selection and definition of dark patterns, as well as the design of benchmark
 544 prompts, may inadvertently reflect the authors' perspectives and biases. This includes assumptions
 545 about user interactions and model behaviors that may not be universally accepted or relevant.

546 Misuse potential: While our intention with this benchmark is to identify and reduce the presence
 547 of dark design patterns in LLMs, we acknowledge the potential for misuse. There is a risk that
 548 malicious actors could use this benchmark to fine-tune models in ways that intentionally enhance
 549 these dark patterns, thereby exacerbating their negative impact.

551 **REPRODUCIBILITY STATEMENT**
 552

553 The code used in this paper can be found here. And we use notebooks to write some simple examples
 554 so that readers can quickly implement the results of the paper. The steps to reproduce the paper are:

- 556 • 1. Download the code.
 557 • 2. Install the necessary environment.
 558 • 3. Run “bash run.sh”.
 559 • 4. Or, run the provided notebook.

560 The code is given in this anonymous link: **Anonymous Repository**.
 561

564 **LLM USAGE STATEMENT**
 565

566 The authors affirm that no LLM was used in our work.
 567

569 **REFERENCES**

570 Md Atik Ahamed and Qiang Cheng. Timemachine: A time series is worth 4 mambas for long-term
 571 forecasting. *arXiv preprint arXiv:2403.09898*, 2024.

573 O. D. Anderson. Time-series. *Journal of the Royal Statistical Society. Series D (The Statistician)*,
 574 25(4):308–310, 1976. ISSN 00390526, 14679884.

575 Lucas Baier, Marcel Hofmann, Niklas Kühl, Marisa Mohr, and Gerhard Satzger. Handling concept
 576 drifts in regression problems—the error intersection approach. *arXiv preprint arXiv:2004.00438*,
 577 2020.

579 Dan Biderman, Jacob P. Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Con-
 580 nor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and
 581 John Patrick Cunningham. Lora learns less and forgets less. *Trans. Mach. Learn. Res.*, 2024,
 582 2024.

583 Manoel Castro-Neto, Young-Seon Jeong, Myong-Kee Jeong, and Lee D Han. Online-svr for short-
 584 term traffic flow prediction under typical and atypical traffic conditions. *Expert Systems with*
 585 *Applications*, 36(3):6164–6173, 2009.

586 Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler
 587 Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecast-
 588 ing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997,
 589 2023.

591 Mouxiang Chen, Lefei Shen, Han Fu, Zhuo Li, Jianling Sun, and Chenghao Liu. Calibration
 592 of time-series forecasting: Detecting and adapting context-driven distribution shift. *KDD*
 593 '24, pp. 341–352, New York, NY, USA, 2024. Association for Computing Machinery. ISBN
 9798400704901.

- 594 Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-
 595 trend decomposition. *J. Off. Stat.*, 6(1):3–73, 1990.
 596
- 597 Jerome T Connor, R Douglas Martin, and Les E Atlas. Recurrent neural networks and robust time
 598 series prediction. *IEEE Transactions on Neural Networks*, 5(2):240–254, 1994.
 599
- 600 Luke Nicholas Darlow, Qiwen Deng, Ahmed Hassan, Martin Asenov, Rajkarn Singh, Artjom
 601 Joosen, Adam Barker, and Amos Storkey. Dam: Towards a foundation model for forecasting.
 602 In *The Twelfth International Conference on Learning Representations*, 2024.
 603
- 604 Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for
 605 time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.
 606
- 607 Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam Nguyen, Wesley M
 608 Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained
 609 models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural
 610 Information Processing Systems*, 37:74147–74181, 2024.
 611
- 612 Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):
 613 1–28, 1985.
 614
- 615 Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint
 616 arXiv:2310.03589*, 2023.
 617
- 618 Shivam Grover and Ali Etemad. Shift-aware test-time adaptation and benchmarking for time-series
 619 forecasting. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML
 620 2025*, 2025.
 621
- 622 Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo,
 623 Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular
 624 foundation model. *Nature*, 01 2025. doi: 10.1038/s41586-024-08328-6.
 625
- 626 Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, An-
 627 drea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp,
 628 2019.
 629
- 630 Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,
 631 Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *The Tenth International
 632 Conference on Learning Representations*, 1(2):3, 2022.
 633
- 634 Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.
 635
- 636 Yuxin Jia, Youfang Lin, Jing Yu, Shuo Wang, Tianhao Liu, and Huaiyu Wan. Pgn: The rnn’s new
 637 successor is effective for long-range time series forecasting. *Advances in Neural Information
 638 Processing Systems*, 37:84139–84168, 2024.
 639
- 640 Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yux-
 641 uan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-lm: Time series forecasting by reprogramming
 642 large language models. In *The Twelfth International Conference on Learning Representations*,
 643 2024.
 644
- 645 HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time se-
 646 ries forecasting via test-time adaptation. In *Proceedings of the Thirty-Ninth AAAI Conference
 647 on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artifi-
 648 cial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence,
 649 AAAI’25/IAAI’25/EAAI’25*. AAAI Press, 2025. ISBN 978-1-57735-897-8.
 650
- 651 Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term
 652 temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference
 653 on research & development in information retrieval (SIGIR)*, pp. 95–104, Ann Arbor, MI, USA,
 654 2018.

- 648 Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A
 649 unified approach to action segmentation. In *European conference on computer vision*, pp. 47–54.
 650 Springer, 2016.
- 651 Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to
 652 document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- 653 Thomas L Lee, William Toner, Rajkarn Singh, Artjom Joosen, and Martin Asenov. Lightweight
 654 online adaption for time series foundation model forecasts. In *Forty-second International Con-
 655 ference on Machine Learning*, 2025.
- 656 Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li.
 657 Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution.
 658 *ACM Transactions on Knowledge Discovery from Data*, 17(1):1–21, 2023.
- 659 Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhui Chen, Yu-Xiang Wang, and Xifeng
 660 Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series
 661 forecasting. In *Advances in 33rd Neural Information Processing Systems (NeurIPS)*, volume 32,
 662 pp. 5243–5253, Vancouver, Canada, 2019.
- 663 Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation.
 664 In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th
 665 Annual Meeting of the Association for Computational Linguistics and the 11th International Joint
 666 Conference on Natural Language Processing*, pp. 4582–4597. Association for Computational
 667 Linguistics, 2021.
- 668 Daojun Liang. Distpred: A distribution-free probabilistic inference method for regression and fore-
 669 casting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data
 670 Mining V. 1*, pp. 753–764, 2025.
- 671 Daojun Liang, Haixia Zhang, Jing Wang, Dongfeng Yuan, and Minggao Zhang. Act now: A novel
 672 online forecasting framework for large-scale streaming data. *arXiv preprint arXiv:2412.00108*,
 673 2024.
- 674 Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R News*, 2(3):
 675 18–22, 2002.
- 676 Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet:
 677 Time series modeling and forecasting with sample convolution and interaction. In *Advances in
 678 Neural Information Processing Systems*, pp. 5816–5828, 2022a.
- 679 Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann.
 680 Unitime: A language-empowered unified model for cross-domain time series forecasting. In
 681 *Proceedings of the ACM Web Conference 2024*, pp. 4095–4106, 2024.
- 682 Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring
 683 the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*,
 684 35:9881–9893, 2022b.
- 685 Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long.
 686 itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint
 687 arXiv:2310.06625*, 2023.
- 688 Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and
 689 Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv
 690 preprint arXiv:2502.00816*, 2025.
- 691 Haoyu Ma, Yushu Chen, Wenlai Zhao, Jinzhe Yang, Yingsheng Ji, Xinghua Xu, Xiaozhu Liu, Hao
 692 Jing, Shengzhuo Liu, and Guangwen Yang. A mamba foundation model for time series forecast-
 693 ing. *arXiv preprint arXiv:2411.02941*, 2024.
- 694 Heitor R. Medeiros, Hossein Sharifi-Noghabi, Gabriel L. Oliveira, and Saghar Irandoust. Accurate
 695 parameter-efficient test-time adaptation for time series forecasting. In *Second Workshop on Test-
 696 Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025.

- 702 Zelin Ni, Hang Yu, Shizhan Liu, Jianguo Li, and Weiyao Lin. Basisformer: Attention-based time
 703 series forecasting with learnable and interpretable basis. *Advances in Neural Information Pro-*
 704 *cessing Systems*, 36:71222–71241, 2023.
- 705 Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64
 706 words: Long-term forecasting with transformers. In *The Eleventh International Conference on*
 707 *Learning Representations*, 2022.
- 708 Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64
 709 words: Long-term forecasting with transformers. In *International Conference on Learning*
 710 *Representations*, 2023.
- 711 Boris N Oreshkin, Dmitri Carpow, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural ba-
 712 sis expansion analysis for interpretable time series forecasting. In *International Conference on*
 713 *Learning Representations*, 2019.
- 714 Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based
 715 Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on*
 716 *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7654–7673, Online, Novem-
 717 ber 2020. Association for Computational Linguistics.
- 718 Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven CH Hoi. Learning fast and slow for online
 719 time series forecasting. In *8th International Conference on Learning Representations (ICLR)*,
 720 2023.
- 721 Domenico Piccolo. A distance measure for classifying arima models. *Journal of time series analysis*,
 722 11(2):153–164, 1990.
- 723 Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Ad-*
 724 *vances in neural information processing systems*, 32, 2019.
- 725 David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic fore-
 726 casting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–
 727 1191, 2020.
- 728 Mohammad Amin Shabani, Amir H Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative
 729 multi-scale refining transformers for time series forecasting. In *The Eleventh International Con-*
 730 *ference on Learning Representations*, 2022.
- 731 Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45,
 732 2018.
- 733 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,
 734 Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in 31st Neural*
 735 *Information Processing Systems (NeurIPS)*, volume 30, pp. 6000–6010, Long Beach, USA, 2017.
- 736 Vladimir Vovk, Jie Shen, Valery Manokhin, and Min-ge Xie. Nonparametric predictive distribu-
 737 tions based on conformal prediction. In *Conformal and probabilistic prediction and applications*,
 738 pp. 82–102. PMLR, 2017.
- 739 Vladimir Vovk, Ilia Nouretdinov, Valery Manokhin, and Alexander Gammerman. Cross-conformal
 740 predictive distributions. In *conformal and probabilistic prediction and applications*, pp. 37–51.
 741 PMLR, 2018.
- 742 Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and
 743 JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The*
 744 *Twelfth International Conference on Learning Representations*, 2024a.
- 745 Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jian-
 746 min Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting
 747 with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498,
 748 2024b.

- 756 Qingsong Wen, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, Tieniu Tan, et al.
 757 Onenet: Enhancing time series forecasting models under concept drift by online ensembling.
 758 *Advances in Neural Information Processing Systems*, 36:69949–69980, 2023.
- 759
 760 Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition trans-
 761 formers with auto-correlation for long-term series forecasting. In *Advances in Neural Information*
 762 *Processing Systems (NeurIPS)*, volume 34, pp. 22419–22430, Virtual Conference, 2021.
- 763 Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet:
 764 Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International*
 765 *Conference on Learning Representations*, 2022.
- 766 Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for
 767 deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint*
 768 *Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence
 769 Organization, 2019.
- 770
 771 Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In *International*
 772 *Conference on Machine Learning*, pp. 11559–11569. PMLR, 2021.
- 773 Chen Xu and Yao Xie. Sequential predictive conformal inference for time series. In *International*
 774 *Conference on Machine Learning*, pp. 38707–38727. PMLR, 2023.
- 775
 776 Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for
 777 non-stationary time series forecasting. *Advances in Neural Information Processing Systems*, 37:
 778 31350–31379, 2024.
- 779 Ying yee Ava Lau, Zhiwen Shao, and Dit-Yan Yeung. Fast and slow streams for online time series
 780 forecasting without information leakage. In *The Thirteenth International Conference on Learning*
 781 *Representations*, 2025.
- 782
 783 Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filternet: Harnessing
 784 frequency filters for time series forecasting. *Advances in Neural Information Processing Systems*,
 785 37:55115–55140, 2024a.
- 786 Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Long-
 787 bing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series
 788 forecasting. *Advances in Neural Information Processing Systems*, 36, 2024b.
- 789
 790 Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series
 791 forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp.
 792 11121–11128, 2023.
- 793 Qianru Zhang, Changlei Yu, Haixin Wang, Yudong Yan, Yuansheng Cao, Siu-Ming Yiu, Tailin
 794 Wu, and Hongzhi Yin. Fldmamba: Integrating fourier and laplace transform decomposition with
 795 mamba for enhanced time series prediction. *arXiv preprint arXiv:2507.12803*, 2025.
- 796 Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang.
 797 Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings*
 798 *of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 11106–11115,
 799 Virtual Conference, 2021.
- 800
 801 Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Fre-
 802 quency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of*
 803 *the 39th International Conference on Machine Learning (ICML)*, volume 162, pp. 27268–27286,
 804 Baltimore, Maryland, 2022.
- 805 Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis
 806 by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.
- 807
 808
 809

810 **A RELATED WORK**
811

812 **A.1 CLASSICAL MODELS FOR TS FORECASTING**
813

814 TS forecasting is a classic research field where numerous methods have been invented to utilize
815 historical series to predict future missing values. Early classical methods Piccolo (1990); Gardner Jr
816 (1985) are widely applied because of their well-defined theoretical guarantee and interpretability.
817 For example, ARIMA Piccolo (1990) initially transforms a non-stationary TS into a stationary one
818 via differencing, and subsequently approximates it using a linear model with several parameters.
819 Exponential smoothing Gardner Jr (1985) predicts outcomes at future horizons by computing a
820 weighted average across historical data. In addition, some regression-based methods, e.g., random
821 forest regression (RFR) Liaw et al. (2002) and support vector regression (SVR) Castro-Neto et al.
822 (2009), etc., are also applied to TS forecasting. These methods are straightforward and have fewer
823 parameters to tune, making them a reliable workhorse for TS forecasting. However, their short-
824 coming is insufficient data fitting ability, especially for high-dimensional series, resulting in limited
825 performance.
826

827 **A.2 DEEP MODELS FOR TS FORECASTING**
828

829 The advancement of deep learning has greatly boosted the progress of TS forecasting. Specifically,
830 convolutional neural networks (CNNs) LeCun et al. (1998) and recurrent neural networks (RNNs)
831 Connor et al. (1994) have been adopted by many works to model nonlinear dependencies of TS, e.g.,
832 LSTNet Lai et al. (2018) improve CNNs by adding recursive skip connections to capture long- and
833 short-term temporal patterns; DeepAR Salinas et al. (2020) predicts the probability distribution by
834 combining autoregressive methods and RNNs. Several works have improved the series aggregation
835 forms of Attention mechanism, such as operations of exponential intervals adopted in LogTrans Li
836 et al. (2019), ProbSparse activations in Informer Zhou et al. (2021), frequency sampling in FED-
837 former Zhou et al. (2022) and iterative refinement in Scaleformer Shabani et al. (2022). Besides,
838 GNNs and Temporal convolutional networks (TCNs) Lea et al. (2016) have been utilized in some
839 methods Wu et al. (2019); Li et al. (2023); Liu et al. (2022a); Wu et al. (2022) for TS forecasting
840 on graph data. The aforementioned methods solely concentrate on the forms of aggregating input
841 series, overlooking the challenges posed by the concept drift problem.
842

843 **A.3 TRANSFORMER-LIKE MODELS**
844

845 Since TS exhibit a variety of patterns, it is meaningful and beneficial to decompose them into several
846 components, each representing an underlying category of patterns that evolving over time Anderson
847 (1976). Several methods, e.g., STL Cleveland et al. (1990), Prophet Taylor & Letham (2018) and
848 N-BEATS Oreshkin et al. (2019), commonly utilize decomposition as a preprocessing phase on
849 historical series. There are also some methods, e.g., Autoformer Wu et al. (2021), FEDformer
850 Zhou et al. (2022), Non-stationary Transformers Liu et al. (2022b) and DistPred Liang (2025),
851 that harness decomposition into the Attention module. The aforementioned methods attempt to
852 apply decomposition to input series to enhance predictability, reduce computational complexity,
853 or ameliorate the adverse effects of non-stationarity. Nevertheless, these prevalent methods are
854 susceptible to significant concept drift when applied to non-stationary TS.
855

856 Furthermore, there are four themes that use deep learning to predict time series: (1) smarter trans-
857 formers Vaswani et al. (2017), such as PatchTST Nie et al. (2022), iTransformer Liu et al. (2023)
858 BasisFormer Ni et al. (2023), and TimeXer Wang et al. (2024b) which restructure attention or add
859 learnable bases to extend context length, cut computation and boost accuracy; (2) competitive non-
860 transformer backbones, including N-HiTS (hierarchical MLP) Challu et al. (2023), DLinear Zeng
861 et al. (2023), PGN Jia et al. (2024) and state-space models like TSMamba Ma et al. (2024), TimeMa-
862 chine Ahamed & Cheng (2024) and FLDMamba Zhang et al. (2025), which deliver linear-time infer-
863 ence and rival or surpass transformers on long horizons; (3) foundation-model initiatives, TimeGPT
864 Garza et al. (2023), OneFitAll Zhou et al. (2023), TimeLLM Jin et al. (2024), UniTime Liu et al.
865 (2024) and DAM Darlow et al. (2024) that pre-train on massive heterogeneous corpora and achieve
866 impressive zero-shot or few-shot performance across domains; and (4) training and interpretability
867 advances, such as frequency-adaptive normalization (FAN) Ye et al. (2024), e.g., FrTS Yi et al.
868

864 (2024b), FilterNet Yi et al. (2024a), and decomposition-aware architectures, e.g., which tackle non-
 865 stationarity, quantify uncertainty and make forecasts more transparent.
 866

867 A.4 ONLINE LEARNING 868

869 Online learning strategies embed concept drift adaptation within the forecasting models themselves.
 870 One example is FSNet c, which leverages complementary learning systems theory to pair a slow-
 871 learning base forecaster with fast-adapting components. Another line of work is OneNet Wen et al.
 872 (2023), an online ensembling approach that dynamically combines two neural models: one special-
 873 izes in capturing temporal dependencies within each series, and the other focuses on cross-series
 874 (covariate) relationships. Each of these deep learning techniques illustrates how integrating drift-
 875 awareness (through dual-model architectures, ensembling, or proactive adjustment) can improve TS
 876 forecasting performance in online.
 877

878 A.5 POST-PROCESSING METHODS IN TIME SERIES

879 Testing-Time adaption (TTA) is very important for Time Series Forecasting. The adapter-based
 880 methods include SOLID Chen et al. (2024), TAFAS Kim et al. (2025) and its follow-ups PETSA
 881 Medeiros et al. (2025) and DynaTTA Grover & Etemad (2025), ELF Lee et al. (2025), etc., and
 882 online approaches, e.g., FSNet Pham et al. (2023) and OneNet Wen et al. (2023), aim to mitigate
 883 test-time concept drift. Specifically, SOLID retrains selected predictor layers using the most recent
 884 similar samples; TAFAS updates linear adapters by online detection of temporal cycles; PETSA and
 885 DynaTTA extends TAFAS with additional losses and dynamic gating to further enhance adaptability.
 886 These methods are either based on linear adapters, parallel fusion, or overall fine-tuning; and, they
 887 do not consider the impact of label leakage Liang et al. (2024); yee Ava Lau et al. (2025). On
 888 the contrary, δ -Adapter can perform non-linear adaptation on both input and output, with good
 889 theoretical guarantees. And it only relies on the most recent sample for fast updates. In addition, it
 890 can be used as a feature selector or a corrector.
 891

892 A.6 POST-PROCESSING METHODS IN NLP

893 Our work is conceptually related to the general parameter-efficient adaptation methods that have
 894 been developed primarily in NLP. Adapter modules for BERT and other Transformers add small
 895 task-specific bottleneck layers between pre-trained weights, keeping the backbone frozen while
 896 achieving near-fine-tuning performance on many downstream tasks Houlsby et al. (2019). This
 897 idea has been extended to multilingual and multi-task settings (e.g., MAD-X), where language and
 898 task adapters are stacked to enable cross-lingual transfer Pfeiffer et al. (2020). A complementary
 899 direction is low-rank adaptation (LoRA), which inserts trainable low-rank matrices into attention
 900 and feed-forward projections to adapt large language models with only a small number of additional
 901 parameters Hu et al. (2022). Another family of methods performs input-side adaptation via prompt
 902 and prefix tuning: instead of changing internal weights, they learn continuous prompts or prefixes at
 903 the embedding level that condition a frozen language model for each task Li & Liang (2021).
 904

905 Compared with the above methods, δ -Adapter adopts the same high-level principle of learning a
 906 small δ -module around a frozen backbone, but it is tailored to TSF and operates strictly at the
 907 input/output interface of a possibly black-box forecaster. Specifically, we introduce horizon-aware
 908 input adapters, feature-masking modules, and output-side uncertainty adapters, and we analyze their
 909 behavior through Lipschitz-style stability and descent guarantees. To our knowledge, such an I/O
 910 level, theoretically characterized adapter framework for multi-horizon forecasting is not present in
 911 the existing NLP adapter or prompt-tuning literature, which primarily modifies internal layers or
 912 token embeddings of language models.

913 A.7 CONFORMAL PREDICTION

914 Conformal prediction presents an alternative framework for distribution prediction, diverging from
 915 traditional parametric approaches. In their study, the authors in (Vovk et al., 2017; 2018) introduced
 916 a random prediction system and proposed a nonparametric prediction method grounded in confor-
 917 mal assumptions. By integrating conformal prediction with quantile regression in (Romano et al.,
 918 2019; Xu & Xie, 2021; 2023), they developed a method for constructing prediction intervals for the

918 response variable. However, the practical application of conformal prediction is not without limitations.
 919 Its effectiveness is often constrained by the assumption of exchangeability of residuals, which
 920 may not hold in all contexts, particularly in the presence of temporal dependencies. This limitation
 921 can lead to less reliable prediction intervals when applied to non-independent and identically dis-
 922 tributed (non-i.i.d.) data, thereby challenging its robustness in real-world scenarios where data often
 923 exhibit complex dependencies.

924 A.8 TERMINOLOGY EXPLANATION

925 **Conditions drift**, which refers to gradual changes in the data-generating process (e.g., seasonal
 926 regime shifts, covariate shifts in demand patterns) that occur after the model has been deployed,
 927 making full retraining costly; **Low-complexity residual structure** means that residual errors often
 928 exhibit simple patterns (e.g., horizon-wise bias, scale miscalibration, calendar offsets) that can be
 929 captured by a small function class (tiny MLPs/low-rank heads) rather than requiring a new high-
 930 capacity backbone, but the base model fails to absorb them.

931 B THEORETICAL PROOF

932 B.1 PROOF OF PROPOSITION 2.1

933 Define risks (squared error):

$$934 \mathcal{R}_{\text{out}}(\delta) = \frac{1}{2} \mathbb{E}[\|Y - (F(X) + \delta g(X))\|^2] = \frac{1}{2} \mathbb{E}[\|R(X) - \delta g(X)\|^2], \quad (27)$$

935 and

$$936 \mathcal{R}_{\text{in}}(\delta) = \frac{1}{2} \mathbb{E}[\|Y - F(X + \delta u(X))\|^2]. \quad (28)$$

937 *Proof.* Let $A := \mathbb{E}\|g(X)\|^2$ and $B := \mathbb{E}\langle R(X), g(X) \rangle$. Then

$$938 \mathcal{R}_{\text{out}}(\delta) = \frac{1}{2} \mathbb{E}\|R\|^2 - \delta B + \frac{1}{2} \delta^2 A. \quad (29)$$

939 Hence \mathcal{R}_{out} is a strictly convex quadratic in δ whenever $A > 0$, with unique minimizer $\delta^* = B/A$
 940 and minimal value

$$941 \mathcal{R}_{\text{out}}(\delta^*) = \frac{1}{2} \mathbb{E}\|R\|^2 - \frac{1}{2} \frac{B^2}{A}. \quad (30)$$

942 In particular, if $B > 0$ and $A > 0$ then for all $0 < \delta < 2B/A$, $\mathcal{R}_{\text{out}}(\delta) < \mathcal{R}_{\text{out}}(0) = \frac{1}{2} \mathbb{E}\|R\|^2$.
 943 Then, expand the square:

$$944 \|R - \delta g\|^2 = \|R\|^2 - 2\delta \langle R, g \rangle + \delta^2 \|g\|^2. \quad (31)$$

945 Taking expectations and multiplying by $\frac{1}{2}$ yields the displayed quadratic form. If $A > 0$, the derivative
 946 $d\mathcal{R}_{\text{out}}/d\delta = -B + \delta A$ vanishes uniquely at $\delta^* = B/A$; strict convexity gives the minimal
 947 value above. If $B > 0$, then near $\delta = 0$ the derivative is negative, so every $\delta \in (0, 2B/A)$ strictly
 948 improves the risk over $\delta = 0$. If $A = 0$ then $g = 0$ a.s. and risk is constant; if $B \leq 0$ there is no
 949 positive δ improving over $\delta = 0$.

950 *Remark.* (i) This is exactly the first (shrunk) step of residual boosting. (ii) The achievable drop at
 951 the optimal δ^* is $\frac{1}{2}(B^2/A)$, which is positive iff $B \neq 0$ and $A > 0$.

952 \square

953 B.2 PROOF OF PROPOSITION 2.2

954 Let $F : \mathbb{R}^d \rightarrow \mathbb{R}^H$ be differentiable, $u : \mathcal{X} \rightarrow \mathbb{R}^d$ a measurable nudging field, and define for $\delta \geq 0$

$$955 \hat{Y}_{\text{in}}(X; \delta) = F(X + \delta u(X)), \quad \mathcal{R}_{\text{in}}(\delta) = \frac{1}{2} \mathbb{E} [\|y - \hat{Y}_{\text{in}}(X; \delta)\|_2^2]. \quad (32)$$

956 Write $r(X) = y - F(X)$, $J_F(X)$ for the Jacobian of F at X , and $A := \mathbb{E}[\langle r(X), J_F(X) u(X) \rangle]$.

957 If $A > 0$, then there exists $\varepsilon > 0$ such that $\mathcal{R}_{\text{in}}(\delta) < \mathcal{R}_{\text{in}}(0)$ for all $\delta \in (0, \varepsilon]$. If, in addition, F is
 958 affine in a neighborhood of the support of X (J_F is constant and the Hessian is zero), then

$$959 \mathcal{R}_{\text{in}}(\delta) = \frac{1}{2} \mathbb{E} [\|r(X) - \delta J_F u(X)\|_2^2], \quad (33)$$

972 is a quadratic function of δ whose unique minimizer is
 973

$$\delta^* = \frac{\mathbb{E}[\langle r(X), J_F u(X) \rangle]}{\mathbb{E}[\|J_F u(X)\|_2^2]}. \quad (34)$$

976 Based on the conditions, we know that: F is C^1 (continuously differentiable) on an open set con-
 977 taining $\{x + \delta u(X) : \delta \in [0, \delta_0]\}$ for some $\delta_0 > 0$. And $\|J_F(X + \delta u(X)) u(X)\|$ is integrable
 978 uniformly for $\delta \in [0, \delta_0]$, and $\|y - F(X + \delta u(X))\|$ is integrable.
 979

980 First, we have the following lemma,

981 **Lemma 4** (Improvement via Jacobian-aligned nudging). *If $\mathbb{E}[\langle r, J_F u \rangle] > 0$, then sufficiently small
 982 $\delta > 0$ reduces risk. As before, the optimal small-step size is $\delta^* = \frac{\mathbb{E}[\langle r, J_F u \rangle]}{\mathbb{E}[\|J_F u\|^2]}$.*
 983

984 *Proof.* For each X , by the fundamental theorem of calculus in Banach spaces,
 985

$$F(X + \delta u(X)) = F(X) + \int_0^\delta J_F(X + t u(X)) u(X) dt. \quad (35)$$

988 Hence,

$$R_\delta = R_0 - \int_0^\delta J_F(X + t u(X)) u(X) dt. \quad (36)$$

991 Let $F(\delta) := \mathcal{R}_{\text{in}}(\delta) = \frac{1}{2} \mathbb{E}\|R_\delta\|^2$. Using $\frac{d}{d\delta}\|v\|^2 = 2\langle v, v' \rangle$,

$$F'(\delta) = \mathbb{E}\langle R_\delta, -J_F(X + \delta u(X)) u(X) \rangle. \quad (37)$$

994 Under the domination assumption, dominated convergence allows $\delta \rightarrow 0$ inside the expectation,
 995 giving

$$F'(0) = -\mathbb{E}\langle R_0, J_F(X) u(X) \rangle. \quad (38)$$

997 If $C > 0$, then $F'(0) = -C < 0$. By continuity of F' near 0 (again from dominated convergence
 998 and continuity of J_F), there exists $\varepsilon > 0$ so that F is strictly decreasing on $(0, \varepsilon)$, hence $F(\delta) <$
 999 $F(0)$ for all $\delta \in (0, \varepsilon)$.

1000 If, in addition, $\|J_F(z)\| \leq L_F$ and $\|u(X)\| \leq U(X)$ with $\mathbb{E}U(X)^2 < \infty$, then for $|\delta| \leq 1$, we have
 1001

$$\|F(X + \delta u) - F(X)\| \leq L_F |\delta| \|u(X)\|, \quad (39)$$

1003 and the same quadratic expansion as in Proposition 2.1 yields

$$\mathcal{R}_{\text{in}}(\delta) \leq \frac{1}{2} \mathbb{E}\|R\|^2 - \delta \mathbb{E}\langle R, J_F u \rangle + \frac{1}{2} \delta^2 L_F^2 \mathbb{E}\|u\|^2, \quad (40)$$

1005 making the “improvement for small δ ” explicit whenever $\mathbb{E}\langle R, J_F u \rangle > 0$. \square

1007 B.2.1 PROOF OF STEP 1: EXACT SMALL-STEP DECREASE

1009 *Proof.* Define, for each (X, y) ,

$$f(\delta; x, y) := \frac{1}{2} \|y - F(X + \delta u(X))\|_2^2 = \frac{1}{2} \|r(\delta; x)\|_2^2, \quad r(\delta; x) := y - F(X + \delta u(X)). \quad (41)$$

1012 By the chain rule,

$$\frac{\partial}{\partial \delta} f(\delta; x, y) = \left\langle r(\delta; x), \frac{\partial}{\partial \delta} r(\delta; x) \right\rangle = -\left\langle r(\delta; x), J_F(X + \delta u(X)) u(X) \right\rangle. \quad (42)$$

1015 By the domination assumptions and Lemma 4, we can pass the derivative through the expectation to
 1016 get

$$\mathcal{R}'_{\text{in}}(\delta) = \mathbb{E} \left[\frac{\partial}{\partial \delta} f(\delta; x, y) \right] = -\mathbb{E} \left[\left\langle r(\delta; x), J_F(X + \delta u(X)) u(X) \right\rangle \right]. \quad (43)$$

1019 Evaluating at $\delta = 0$,

$$\mathcal{R}'_{\text{in}}(0) = -\mathbb{E}[\langle r(X), J_F(X) u(X) \rangle] = -A. \quad (44)$$

1021 If $A > 0$, then $\mathcal{R}'_{\text{in}}(0) < 0$. By continuity of \mathcal{R}'_{in} at 0, there exists $\varepsilon > 0$ such that $\mathcal{R}'_{\text{in}}(\delta) \leq -\frac{A}{2} < 0$
 1022 for all $\delta \in [0, \varepsilon]$. Therefore, for any $\delta \in (0, \varepsilon]$,

$$\mathcal{R}_{\text{in}}(\delta) - \mathcal{R}_{\text{in}}(0) = \int_0^\delta \mathcal{R}'_{\text{in}}(t) dt \leq -\frac{A}{2} \delta < 0, \quad (45)$$

1025 which proves the strict risk decrease for sufficiently small positive δ . \square

1026 B.2.2 PROOF OF STEP 2: CLOSED-FORM δ^* UNDER AN AFFINE F
1027

1028 *Proof.* Assume F is affine: $F(X) = Ax + b$ with a constant matrix $A \in \mathbb{R}^{H \times d}$. Then $J_F \equiv A$ and
1029 $F(X + \delta u(X)) = F(X) + \delta Au(X)$. (46)

1030 Hence,

$$1031 \mathcal{R}_{\text{in}}(\delta) = \frac{1}{2} \mathbb{E} [\|r(X) - \delta Au(X)\|_2^2] = \frac{1}{2} \mathbb{E} [\|r(X)\|^2] - \delta \mathbb{E} [\langle r(X), Au(X) \rangle] + \frac{1}{2} \delta^2 \mathbb{E} [\|Au(X)\|^2]. \\ 1032 \quad (47)$$

1034 This is a strictly convex quadratic in δ provided $\mathbb{E}[\|Au(X)\|^2] > 0$. Differentiating and setting to 0,
1035

$$1036 \mathcal{R}'_{\text{in}}(\delta) = -\mathbb{E} [\langle r(X), Au(X) \rangle] + \delta \mathbb{E} [\|Au(X)\|^2] = 0 \quad (48)$$

1037 yields the unique minimizer

$$1038 \delta^* = \frac{\mathbb{E} [\langle r(X), Au(X) \rangle]}{\mathbb{E} [\|Au(X)\|^2]} = \frac{\mathbb{E} [\langle r(X), J_F u(X) \rangle]}{\mathbb{E} [\|J_F u(X)\|^2]}. \quad (49)$$

1040 This completes the proof for affine F . The same expression arises if, instead of assuming affine F ,
1041 we optimize the first-order surrogate obtained by linearizing F at $\delta = 0$:

$$1042 F(X + \delta u(X)) \approx F(X) + \delta J_F(X) u(X), \quad (50)$$

1043 which leads to the quadratic proxy

$$1044 \tilde{\mathcal{R}}_{\text{in}}(\delta) := \frac{1}{2} \mathbb{E} [\|r(X) - \delta J_F(X) u(X)\|^2], \quad (51)$$

1046 whose unique minimizer is the same δ^* as above. Further, we denote by $H_F(X)[v, w] \in \mathbb{R}^H$ the
1047 second directional derivative of F at X along v, w , then

$$1048 \mathcal{R}_{\text{in}}''(0) = \mathbb{E} [\|J_F(X)u(X)\|^2 - \langle r(X), H_F(X)[u(X), u(X)] \rangle]. \quad (52)$$

1049 If there exists $\eta \in [0, 1]$ such that

$$1050 |\mathbb{E} [\langle r(X), H_F(X)[u(X), u(X)] \rangle]| \leq \eta \mathbb{E} [\|J_F(X)u(X)\|^2], \quad (53)$$

1051 then $\mathcal{R}_{\text{in}}''(0) \in [(1 - \eta)B_0, (1 + \eta)B_0]$ where $B_0 = \mathbb{E} [\|J_F u\|^2]$. In that case, the true local
1052 minimizer δ^\dagger of \mathcal{R}_{in} satisfies the bracket

$$1053 \frac{A}{(1 + \eta)B_0} \leq \delta^\dagger \leq \frac{A}{(1 - \eta)B_0}, \quad (54)$$

1054 quantifying how curvature perturbs the first-order optimizer. When F is affine or the curvature term
1055 averages to zero, $\eta = 0$ and $\delta^\dagger = \delta^*$. \square
1056

1058 B.3 PROOF OF PROPOSITION 3.1
1059

1060 *Proof.* By Lipschitzness of F :

$$1061 \|\tilde{y} - \hat{y}\| = \|F(\tilde{X}) - F(X)\| \leq L_F \|\tilde{X} - X\| = L_F \delta \|A_\phi^{\text{in}}(X)\|. \quad (55)$$

1063 According to $\|A_\phi^{\text{in}}(X)\|_\infty \leq 1$, $\|A_\phi^{\text{out}}(\hat{Y}, X)\|_\infty \leq 1$ and $\delta \in (0, \delta_{\max}]$ with $\delta_{\max} \leq 1$, we have
1064 $\|A_\phi^{\text{in}}(X)\| \leq \sqrt{Ld} \|A_\phi^{\text{in}}(X)\|_\infty \leq \sqrt{Ld}$. Combining yields the claim. \square
1065

1066 B.4 PROOF OF COROLLARY 1
1067

1068 *Proof.* Coordinatewise, $\tilde{x}_i - x_i = x_i(e^{\delta a_i} - 1)$. By the mean value theorem for $t \mapsto e^t$, for each i
1069 there exists $\xi_i \in (0, \delta a_i)$ such that

$$1070 e^{\delta a_i} - 1 = \delta a_i e^{\xi_i} \Rightarrow |\tilde{x}_i - x_i| = |x_i| \delta |a_i| e^{\xi_i} \leq B_X \delta |a_i| e^{|\xi_i|} \leq B_X \delta |a_i| e^{\delta \|a\|_\infty}. \quad (56)$$

1072 According to $\|A_\phi^{\text{in}}(X)\|_\infty \leq 1$, $\|A_\phi^{\text{out}}(\hat{Y}, X)\|_\infty \leq 1$ and $\delta \in (0, \delta_{\max}]$ with $\delta_{\max} \leq 1$, we have
1073 $\|a\|_\infty \leq 1$, hence $e^{\delta \|a\|_\infty} \leq e^\delta$. Summing squares,

$$1075 \|\tilde{x} - x\| = \sqrt{\sum_i |\tilde{x}_i - x_i|^2} \leq \sqrt{\sum_i (B_X \delta |a_i| e^\delta)^2} = \delta e^\delta B_X \|a\|. \quad (57)$$

1077 Then apply Lipschitz step in Proposition 3.1, we have

$$1078 \|\tilde{y} - \hat{y}\| = \|F(\tilde{X}) - F(X)\| \leq L_F \|\tilde{x} - x\| \leq \delta e^\delta L_F B_X \|a\|. \quad (58)$$

1079 For $\delta \leq 1$, $e^\delta \leq e$, so the bound is $O(\delta)$. \square

1080 **B.5 PROOF OF THEOREM 2**
 1081

1082 *Proof.* By β -smoothness with $u = \tilde{y}$, $v = \hat{y}$,

$$1083 \quad \ell(\tilde{y}, y) \leq \ell(\hat{y}, y) + \nabla \ell(\hat{y}, y)^\top (\tilde{y} - \hat{y}) + \frac{\beta}{2} \|\tilde{y} - \hat{y}\|^2 = \ell(\hat{y}, y) + \delta \langle g, d \rangle + \frac{\beta}{2} \delta^2 \|d\|^2. \quad (59)$$

1084 By alignment condition, $\langle g, d \rangle \leq -\alpha \|g\| \|d\|$. Substitute:

$$1086 \quad \ell(\tilde{y}, y) - \ell(\hat{y}, y) \leq -\delta \alpha \|g\| \|d\| + \frac{\beta}{2} \delta^2 \|d\|^2. \quad (60)$$

1087 The RHS is a convex quadratic in δ with unique minimizer $\delta^* = \frac{\alpha \|g\|}{\beta \|d\|}$. Plugging δ^* gives $-\frac{\alpha^2}{2\beta} \|g\|^2$.
 1088 Strict descent holds whenever the derivative at 0 is negative and the second-order term does not
 1089 dominate, equivalently $\delta \in (0, \frac{2\alpha \|g\|}{\beta \|d\|})$. \square

1091 **B.6 PROOF OF THEOREM 3**
 1092

1093 *Proof.* By first-order Taylor expansion of F at x ,

$$1094 \quad F(x + \delta v) = F(x) + \delta Jv + r_F(\delta), \quad \text{with } \|r_F(\delta)\| = O(\delta). \quad (61)$$

1095 Set $\delta := \delta s + r_F(\delta)$, so $\tilde{y} = \hat{y} + \delta$. Apply β -smoothness of ℓ :

$$1097 \quad \ell(\hat{y} + \delta, y) \leq \ell(\hat{y}, y) + \langle g, \delta \rangle + \frac{\beta}{2} \|\delta\|^2. \quad (62)$$

1098 Then, compute the terms:

$$1099 \quad \langle g, \delta \rangle = \delta \langle g, s \rangle + \langle g, r_F(\delta) \rangle \quad \text{and } \|\delta\|^2 = \delta^2 \|s\|^2 + 2\delta \langle s, r_F(\delta) \rangle + \|r_F(\delta)\|^2. \quad (63)$$

1100 Since $\|r_F(\delta)\| = O(\delta)$, we have $\langle g, r_F(\delta) \rangle = O(\delta)$ and $\|\delta\|^2 = \delta^2 \|s\|^2 + O(\delta^2)$. Therefore,

$$1101 \quad \ell(F(x + \delta v), y) \leq \ell(\hat{y}, y) + \delta \langle g, s \rangle + \frac{\beta}{2} \delta^2 \|s\|^2 + O(\delta^2). \quad (64)$$

1103 If $\langle g, s \rangle \leq -\alpha \|g\| \|s\|$, then for sufficiently small δ the negative linear term dominates the $O(\delta^2)$
 1104 remainder, yielding strict descent. Optimizing the quadratic upper bound in δ gives the minimizer
 1105 $\delta^* = \frac{\alpha \|g\|}{\beta \|s\|}$ and value $-\frac{\alpha^2}{2\beta} \|g\|^2$ up to $O(1)$, establishing the last claim. \square

1107 **B.7 PROOF OF THEOREM 3.2**
 1108

1109 *Proof.* We formalize the two claims: (i) $O(\delta)$ bound on prediction drift, and (ii) loss upper bound
 1110 under composition. Let the composed edit be: $\tilde{x} = x + \delta v$, $\hat{y}' := F(\tilde{X})$, and $\tilde{y} := \hat{y}' + \delta d(\hat{y}', X)$.
 1111 As before, $\hat{y} = F(X)$.

1112 (i) For $O(\delta)$ bound on prediction drift, using the triangle inequality, we have:

$$1113 \quad \|\tilde{y} - \hat{y}\| \leq \|\hat{y}' - \hat{y}\| + \delta \|d(\hat{y}', X)\|. \quad (65)$$

1115 Further, according to $\|A_\phi^{\text{in}}(X)\|_\infty \leq 1$, $\|A_\phi^{\text{out}}(\hat{Y}, X)\|_\infty \leq 1$ and $\delta \in (0, \delta_{\max}]$ with $\delta_{\max} \leq 1$, we
 1116 have $\|\hat{y}' - \hat{y}\| = \|F(\tilde{X}) - F(X)\| \leq L_F \|\tilde{x} - x\| = \delta L_F \|v\|$. The bound follows.

1117 (ii) For loss upper bound under composition, by definition we have

$$1118 \quad \tilde{y} = \hat{y}' + \delta d' = \hat{y} + \delta s + r_F(\delta) + \delta d'. \quad (66)$$

1119 Set $\Delta := \delta(s + d') + r_F(\delta)$. By β -smoothness, we have

$$1121 \quad \ell(\hat{y} + \Delta, y) \leq \ell(\hat{y}, y) + \langle g, \Delta \rangle + \frac{\beta}{2} \|\Delta\|^2, \quad (67)$$

1122 which can be decomposed into:

$$1123 \quad \langle g, \Delta \rangle = \delta \langle g, s + d' \rangle + \langle g, r_F(\delta) \rangle, \quad (68)$$

$$1124 \quad \text{where } \|\Delta\|^2 = \delta^2 \|s + d'\|^2 + 2\delta \langle s + d', r_F(\delta) \rangle + \|r_F(\delta)\|^2. \quad (69)$$

1126 Since $\|r_F(\delta)\| = O(\delta)$, we have $\langle g, r_F(\delta) \rangle = O(\delta)$ and $\|\Delta\|^2 = \delta^2 \|s + d'\|^2 + O(\delta^2)$. Thus

$$1127 \quad \ell(\tilde{y}, y) \leq \ell(\hat{y}, y) + \delta \langle g, s + d' \rangle + \frac{\beta}{2} \delta^2 \|s + d'\|^2 + O(\delta^2). \quad (70)$$

1128 Here, if $\langle g, s + d' \rangle \leq -\alpha \|g\| \|s + d'\|$, the linear term is strictly negative whenever $s + d' \neq 0$. For
 1129 sufficiently small δ , the negative linear term dominates the $O(\delta^2)$ remainder, giving strict descent.

1130 *Remark.* If a learned gate $\gamma \in [0, 1]^q$ combines input- and output-induced steps as $s_\gamma = \gamma \odot s + (1 -$
 1131 $\gamma) \odot d'$, then alignment for s_γ follows from mild conditions (e.g., selecting γ to minimize $\langle g, s_\gamma \rangle$
 1132 subject to $\gamma \in [0, 1]^q$ ensures $\langle g, s_\gamma \rangle \leq \min\{\langle g, s \rangle, \langle g, d' \rangle\}$). \square

1134 **C EXPERIMENTAL SETUP AND RESULTS**

1135 **C.1 DATASET**

1136 **C.2 COMMONLY USED TS DATASETS**

1137
 1138 The information of the experiment datasets used in this paper are summarized as follows: (1) Electricity Transformer Temperature (ETT) dataset Zhou et al. (2021), which contains the data collected
 1139 from two electricity transformers in two separated counties in China, including the load and the oil
 1140 temperature recorded every 15 minutes (ETTm) or 1 hour (ETTh) between July 2016 and July 2018.
 1141 (2) Electricity (ECL) dataset ¹ collects the hourly electricity consumption of 321 clients (each column)
 1142 from 2012 to 2014. (3) Exchange Lai et al. (2018) records the current exchange of 8 different
 1143 countries from 1990 to 2016. (4) Traffic dataset ² records the occupation rate of freeway system
 1144 across State of California measured by 861 sensors. (5) Weather dataset ³ records every 10 minutes
 1145 for 21 meteorological indicators in Germany throughout 2020. The detailed statistics information of
 1146 the datasets is shown in Table 7.

1147
 1148
 1149
 1150 **Table 7: Details of the seven TS datasets.**

Dataset	length	features	frequency
ETTh1	17,420	7	1h
ETTh2	17,420	7	1h
ETTm1	69,680	7	15m
ETTm2	69,680	7	15m
Electricity	26,304	321	1h
Exchange	7,588	8	1d
Traffic	17,544	862	1h
Weather	52,696	21	10m

1160
 1161 **C.3 TRAINING OBJECTIVE**

1162 We train θ on \mathcal{D} while backpropagating through F but not updating it. Let $\tilde{Y}_\theta(X)$ denote the
 1163 adapted prediction. For point forecasts we minimize a horizon-aware loss: Here are explicit formulas
 1164 for each loss term when the input-adaptor is a learnable mask $M(X; \phi) \in [0, 1]^{L \times d}$ applied as
 1165 $X' = X \odot M$. Let $\mathcal{D} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^N$, $\hat{Y}^{(i)} = F(X^{(i)} \odot M(X^{(i)}; \phi))$, and H, m be horizon
 1166 and target dims. Expectations \mathbb{E} below are over the empirical data distribution (mini-batches in
 1167 practice).

1168
 1169 MSE (point forecasts):

$$\mathcal{L}_{\text{pred}}^{\text{MSE}} = \mathbb{E}_{(X, Y) \sim \mathcal{D}} \left[\frac{1}{Hm} \sum_{h=1}^H \sum_{k=1}^m w_h (\hat{Y}_{h,k} - Y_{h,k})^2 \right]. \quad (71)$$

1170
 1171 MAE (point forecasts):

$$\mathcal{L}_{\text{pred}}^{\text{MAE}} = \mathbb{E} \left[\frac{1}{Hm} \sum_{h=1}^H \sum_{k=1}^m w_h |\hat{Y}_{h,k} - Y_{h,k}| \right]. \quad (72)$$

1172
 1173 Pinball (quantile $\tau \in \mathcal{T}$). If \hat{Y}^τ predicts the τ -quantile,

$$\mathcal{L}_{\text{pred}}^{\text{QB}} = \mathbb{E} \left[\frac{1}{|\mathcal{T}| Hm} \sum_{\tau \in \mathcal{T}} \sum_{h,k} \rho_\tau(Y_{h,k} - \hat{Y}_{h,k}^\tau) \right], \quad \rho_\tau(u) = u(\tau - \mathbf{1}\{u < 0\}). \quad (73)$$

1¹<https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014>

2²<http://pems.dot.ca.gov>

3³<https://www.bgc-jena.mpg.de/wetter>

1188 Sparsity (L1) on the mask:
 1189

$$\mathcal{L}_{\ell_1} = \mathbb{E}_{X \sim \mathcal{D}} \left[\frac{1}{Ld} \sum_{t=1}^L \sum_{j=1}^d M_{t,j}(X; \phi) \right]. \quad (74)$$

1193 Entropy (pushes mask toward 0 or 1):
 1194

$$\mathcal{L}_{\text{ent}} = \mathbb{E}_X \left[-\frac{1}{Ld} \sum_{t,j} \left(M_{t,j} \log(M_{t,j} + \delta) + (1 - M_{t,j}) \log(1 - M_{t,j} + \delta) \right) \right]. \quad (75)$$

1198 Temporal smoothness:
 1199

$$\mathcal{L}_{\text{TV}} = \mathbb{E}_X \left[\frac{1}{(L-1)d} \sum_{t=2}^L \sum_{j=1}^d |M_{t,j}(X; \phi) - M_{t-1,j}(X; \phi)| \right]. \quad (76)$$

1204 Budget (fraction of active entries not to exceed κ):
 1205

$$\bar{m}(X; \phi) = \frac{1}{Ld} \sum_{t,j} M_{t,j}(X; \phi). \quad (77)$$

1208 A hinge penalty enforces $\bar{m} \leq \kappa$:
 1209

$$\mathcal{L}_{\text{bud}} = \mathbb{E}_X \left[(\bar{m}(X; \phi) - \kappa)_+ \right], \quad (u)_+ \equiv \max\{u, 0\}. \quad (78)$$

1211 Group sparsity:
 1212

$$\mathcal{L}_{\text{group}} = \mathbb{E}_X \left[\frac{1}{d} \sum_{j=1}^d \sqrt{\sum_{t=1}^L M_{t,j}(X; \phi)^2 + \delta} \right]. \quad (79)$$

1218 C.4 ONLINE LEARNING SETUP

1220 During online testing, we set the batch size to 1 to ensure that data arrives in order. Meanwhile,
 1221 we used a streaming buffer, where only one updated data point is cached at each moment/iteration
 1222 (avoid label leakage raised by Liang et al. (2024); see Ava Lau et al. (2025), while returning a
 1223 complete sample from a previous moment. E.g., at time t , the input used for online update returned
 1224 from the buffer is $X_{t-H-L:t-H}$, the label is $\hat{Y}_{t-H:t}$, where H is the prediction length and L is the
 1225 input length.

1226 C.5 TRAINING DETAILS OF ADA-X+Y

1228 Ada-X+Y is composed of Ada-X and Ada-Y, and Ada-X and Ada-Y are **trained jointly** in an end-
 1229 to-end manner, not sequentially. We minimize a single combined loss \mathcal{L} over the union of parameters
 1230 A_θ^{in} (Ada-X) and A_θ^{out} (Ada-Y). The forward pass is:

$$\hat{Y} = F(X + \delta A_\theta^{in}(X)) \quad (80)$$

$$\tilde{Y} = \hat{Y} + \delta A_\theta^{out}(\hat{Y}) \quad (81)$$

1235 During the backward pass, gradients flow from the loss through Ada-Y (Eq. 2), then through the
 1236 backbone F , and finally to Ada-X (Eq. 1). This ensures that Ada-X learns input perturbations that
 1237 specifically help the backbone produce features that Ada-Y can best correct.

1238 Experimental Setup: In our experiments, we instantiate two separate Adam optimizers (both learning
 1239 rate are 1E-4) for modular flexibility. However, they are stepped simultaneously after a single
 1240 backward pass, making the process equivalent to optimizing a joint objective. As derived in Propo-
 1241 sition 3.2, this joint update rule maintains the $O(\delta)$ drift bounds and descent guarantees, ensuring
 1242 the two adapters do not destabilize each other.

Table 8: Performances of the forecaster F and δ -Adapter under batch or online training.

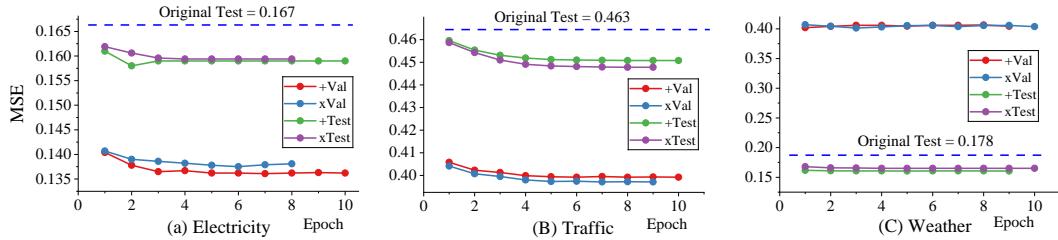
Dataset	Model	Original (Batch)	Fine-Tuning (Batch)	Continue (Online)	Ada-X (Batch)	Ada-X (Online)	Ada-Y (Batch)	Ada-Y (Online)	Ada-X+Y (Batch)	Ada-X+Y (Online)
Weather	DistPred	0.1710	0.1715	0.1700	0.1662	0.1654	0.1629	0.1623	0.1602	0.1560
	iTransformer	0.1731	0.1724	0.1721	0.1706	0.169	0.1631	0.1634	0.1609	0.1609
Traffic	DistPred	0.4229	0.4229	0.4229	0.4182	0.4179	0.4117	0.4119	0.4028	0.4029
	iTransformer	0.4437	0.4411	0.4414	0.4361	0.4360	0.4294	0.4294	0.4202	0.4202
ELC	DistPred	0.1546	0.1546	0.1545	0.1483	0.1476	0.1483	0.1474	0.1436	0.1432
	iTransformer	0.1655	0.1646	0.1636	0.1601	0.1596	0.1566	0.1562	0.1527	0.1515

C.6 USING δ -ADAPTERS TO IMPROVE MULTIVARIATE TIME SERIES

Tables 8, 9 and Figure 10 show that δ -Adapter provides consistent improvements across multiple forecasting models. For nearly all datasets, Ada-X and Ada-Y lead to lower prediction errors compared to the original models, demonstrating that the proposed adapters generalize well to diverse forecasting architectures. Notably, Ada-X again delivers the largest gains, particularly on challenging datasets such as Exchange, Traffic, and ETT series, confirming that refining the input signals before model inference is the most impactful strategy. These results further validate that δ -Adapter is a broadly applicable, efficient, and effective enhancement method for modern time series forecasting.

C.7 δ -ADAPTER’S VALIDATION AND TESTING PERFORMANCE CHANGES WITH EPOCHS

We also present the performance changes of additive and multiplicative adapters on different datasets over epochs (the blank is due to early stopping). In Figure 9, we visualize the changes in validation and test losses of δ -Adapter across different datasets. Across Electricity, Traffic, and Weather, adding Ada-X+Y drives the test MSE consistently below the original frozen model from the very first epoch and then decreases further before plateauing after 5 epochs. Validation and test curves track closely (no divergence), indicating stable training without overfitting. The gains are monotonic or near-monotonic on Electricity and Traffic, while Weather shows an immediate, steady improvement that remains well under the original baseline. Overall, Ada-X+Y delivers fast convergence and robust generalization across datasets. These experiments show that the loss curve of the δ -Adapter gradually decreases with epochs and has stable and consistent boundaries. Meanwhile, the composite adapter (X+Y) can achieve better performance (Stability Analysis of Section 3), which also proves the robustness of the δ -Adapter and the correctness of its theoretical foundation.

Figure 9: Validation and testing performance changes with epochs when adding δ -Adapter.

C.8 QUANTILE CALIBRATOR (QC) AND CONFORMAL CALIBRATOR (CC)

Figure 11 illustrates that both calibrators produce well-calibrated intervals. QC attains higher coverage than CC, while on another sample CC is better. QC tends to yield slightly wider, more conservative bands. CC delivers comparably high coverage with tighter intervals. Overall, the two methods are complementary and reliably improve uncertainty quantification over the raw predictor.

Now, let’s discuss how to choose between QC and CC. Both modules turn a frozen point forecaster into a calibrated probabilistic predictor, but they are aimed at slightly different desiderata: QC directly learns horizon-wise conditional quantiles as bounded offsets around the point forecast, which

1296

1297

1298

1299 Table 9: Multivariate time series forecasting results on the benchmark datasets.

1300

Dataset	DistPred			iTTransformer			FourierGNN			FreTS			Autoformer			
	Length	Original	Ada-X	Ada-Y	Original	Ada-X	Ada-Y	Original	Ada-X	Ada-Y	Original	Ada-X	Ada-Y	Original	Ada-X	Ada-Y
ELC	96	0.155	0.149	0.148	0.163	0.160	0.157	0.250	0.235	0.224	0.189	0.183	0.176	0.228	0.211	0.221
	192	0.169	0.166	0.162	0.175	0.173	0.167	0.255	0.245	0.230	0.193	0.189	0.180	0.437	0.383	0.384
	336	0.185	0.181	0.176	0.193	0.189	0.182	0.267	0.256	0.240	0.207	0.203	0.192	0.612	0.590	0.527
	720	0.221	0.217	0.190	0.231	0.226	0.218	0.298	0.284	0.268	0.246	0.239	0.229	0.782	0.767	0.670
Avg		0.182	0.178	0.169	0.190	0.187	0.181	0.267	0.255	0.241	0.209	0.203	0.194	0.515	0.488	0.450
ETTh1	96	0.389	0.385	0.384	0.390	0.385	0.386	0.506	0.502	0.503	0.397	0.398	0.396	0.449	0.437	0.444
	192	0.451	0.448	0.446	0.444	0.444	0.440	0.540	0.540	0.540	0.458	0.456	0.453	0.571	0.566	0.558
	336	0.498	0.493	0.497	0.479	0.478	0.483	0.583	0.585	0.584	0.507	0.508	0.511	0.656	0.644	0.638
	720	0.505	0.502	0.503	0.504	0.502	0.514	0.615	0.634	0.632	0.568	0.574	0.563	0.695	0.686	0.669
Avg		0.461	0.457	0.458	0.454	0.453	0.456	0.561	0.566	0.565	0.482	0.484	0.481	0.593	0.583	0.577
ETTh2	96	0.303	0.300	0.301	0.296	0.293	0.293	0.396	0.383	0.388	0.342	0.314	0.330	0.375	0.358	0.375
	192	0.378	0.372	0.373	0.383	0.380	0.380	0.507	0.468	0.477	0.468	0.427	0.437	0.438	0.432	0.444
	336	0.447	0.438	0.443	0.429	0.427	0.434	0.558	0.500	0.500	0.548	0.501	0.506	0.464	0.460	0.459
	720	0.431	0.433	0.431	0.445	0.440	0.453	0.718	0.646	0.660	0.791	0.725	0.717	0.473	0.470	0.494
Avg		0.390	0.386	0.387	0.388	0.385	0.390	0.545	0.499	0.506	0.537	0.492	0.498	0.438	0.420	0.423
ETTm1	96	0.339	0.324	0.330	0.345	0.334	0.334	0.405	0.399	0.397	0.340	0.334	0.337	0.586	0.464	0.569
	192	0.384	0.374	0.378	0.382	0.369	0.374	0.435	0.427	0.430	0.380	0.378	0.380	0.627	0.572	0.602
	336	0.416	0.409	0.404	0.431	0.423	0.419	0.464	0.457	0.455	0.417	0.414	0.415	0.691	0.650	0.656
	720	0.510	0.487	0.495	0.511	0.501	0.496	0.519	0.506	0.507	0.483	0.478	0.474	0.754	0.729	0.720
Avg		0.412	0.399	0.402	0.417	0.407	0.406	0.456	0.447	0.447	0.405	0.401	0.401	0.664	0.604	0.637
ETTm2	96	0.179	0.175	0.179	0.182	0.179	0.183	0.220	0.204	0.213	0.191	0.179	0.187	0.271	0.236	0.288
	192	0.245	0.242	0.245	0.255	0.251	0.253	0.329	0.289	0.322	0.275	0.241	0.270	0.290	0.286	0.289
	336	0.309	0.304	0.302	0.327	0.317	0.323	0.380	0.359	0.380	0.342	0.309	0.333	0.359	0.350	0.350
	720	0.406	0.394	0.404	0.435	0.423	0.414	0.852	0.694	0.842	0.531	0.413	0.501	0.435	0.431	0.432
Avg		0.285	0.279	0.282	0.300	0.292	0.293	0.445	0.386	0.439	0.335	0.285	0.323	0.339	0.316	0.320
Exchange	96	0.084	0.088	0.084	0.099	0.102	0.099	0.106	0.118	0.105	0.098	0.105	0.195	0.170	0.179	
	192	0.190	0.186	0.182	0.180	0.173	0.168	0.208	0.216	0.203	0.186	0.181	0.187	0.260	0.243	0.246
	336	0.319	0.281	0.294	0.352	0.304	0.329	0.365	0.396	0.368	0.383	0.380	0.386	0.437	0.414	0.402
	720	0.809	0.653	0.714	0.901	0.814	0.800	0.841	0.843	0.841	0.989	0.989	0.1011	1.144	1.095	1.021
Avg		0.350	0.302	0.319	0.383	0.348	0.349	0.380	0.393	0.379	0.416	0.412	0.422	0.509	0.481	0.462
Traffic	96	0.423	0.416	0.412	0.444	0.436	0.429	0.779	0.753	0.731	0.563	0.555	0.538	0.659	0.657	0.650
	192	0.441	0.435	0.429	0.460	0.455	0.446	0.756	0.721	0.710	0.568	0.562	0.545	0.829	0.814	0.804
	336	0.458	0.453	0.447	0.479	0.475	0.465	0.765	0.739	0.739	0.595	0.589	0.572	1.094	1.072	1.025
	720	0.490	0.487	0.480	0.517	0.513	0.502	0.806	0.781	0.780	0.659	0.653	0.634	1.307	1.292	1.195
Avg		0.453	0.448	0.442	0.475	0.470	0.461	0.777	0.749	0.740	0.596	0.590	0.572	0.972	0.959	0.918
Weather	96	0.171	0.166	0.162	0.173	0.165	0.163	0.184	0.183	0.172	0.185	0.177	0.172	0.262	0.240	0.243
	192	0.224	0.220	0.213	0.223	0.213	0.210	0.226	0.222	0.214	0.224	0.218	0.212	:3003	0.282	0.275
	336	0.278	0.270	0.264	0.284	0.271	0.267	0.273	0.268	0.260	0.272	0.267	0.260	0.323	0.315	0.312
	720	0.353	0.347	0.340	0.357	0.349	0.339	0.338	0.330	0.329	0.341	0.335	0.328	0.389	0.385	0.367
Avg		0.256	0.251	0.245	0.259	0.249	0.245	0.255	0.251	0.244	0.255	0.249	0.243	0.325	0.306	0.299

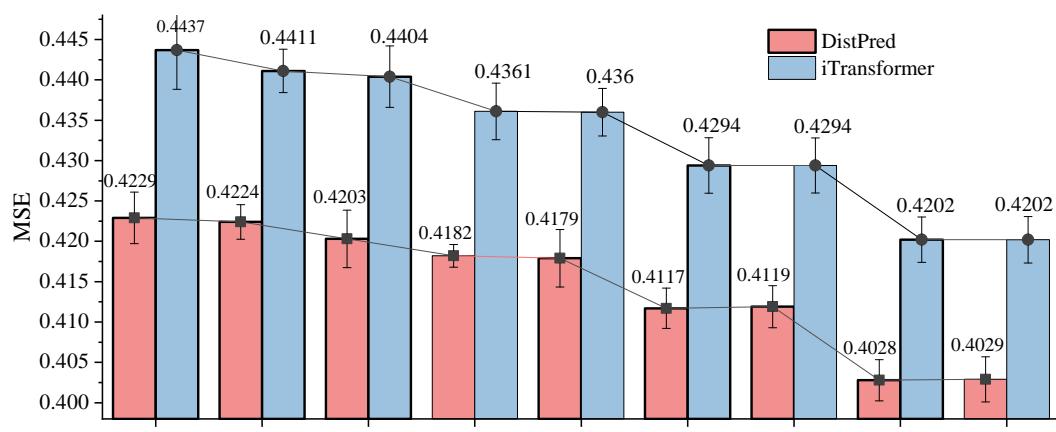


Figure 10: Performances of δ-Adapter under batch or online training

1346

1347

1348

1349

produces a smooth quantile function over multiple levels without assumptions about the underlying distribution. CC learns only a heteroscedastic scale function and combines it with normalized-residual conformal prediction on a held-out calibration set, yielding symmetric but input-dependent intervals with finite-sample marginal coverage under exchangeability.

Empirically, both variants achieve strong coverage, but QC tends to produce marginally wider and more conservative bands, while CC attains similar coverage with somewhat tighter intervals (see Figs. 5 and 6). For a new real-world dataset, our recommendation is therefore: If strict coverage guarantees are the main requirement, CC is preferable, because the conformal step provides finite-sample marginal coverage at the target level. If one needs a rich predictive distribution or multiple coverage levels from a single model, QC is more convenient, as it directly returns a full quantile curve while remaining non-parametric w.r.t. the underlying distribution.

C.9 COMPARISON BETWEEN ADAPTERS AND ONLINE LEARNING METHODS

The adapter-based methods we reviewed include SOLID, TAFAS, etc., and online approaches, e.g., FSNet and OneNet. These methods aim to mitigate test-time concept drift via selective layer retraining (SOLID), online adapter updates (TAFAS), auxiliary loss (PETSA), layer-wise adjustment and memory (FSNet), and dynamic model selection (OneNet). However, according to works by Liang et al. (2024); yee Ava Lau et al. (2025), the above methods have used future labels to some extent, causing label leakage in long-term forecasting, where future ground truth is adopted in advance for adaptation. To achieve a fair comparison, we removed label leakage (it may cause performance degradation of some methods) to test their performance. As shown in Table 10, it can be found that our method achieves the lowest error on every dataset across all backbones.

Table 10: Comparison of various adapters and online methods.

Model		DistPred				iTransformer				Autoformer				Others	
Dataset	Length	Offline	SOLID	TAFAS	Ada-X+Y	Offline	SOLID	TAFAS	Ada-X+Y	Offline	SOLID	TAFAS	Ada-X+Y	OneNet [†]	FSNet [†]
ELC	96	0.155	0.154	0.156	0.146	0.163	0.165	0.165	0.156	0.228	0.211	0.230	0.196	0.247	0.310
	192	0.169	0.170	0.170	0.164	0.175	0.177	0.176	0.167	0.437	0.433	0.442	0.379	0.300	0.442
	336	0.185	0.182	0.183	0.178	0.193	0.189	0.189	0.176	0.612	0.583	0.588	0.580	0.325	0.483
	720	0.221	0.220	0.220	0.213	0.231	0.230	0.231	0.222	0.782	0.780	0.781	0.757	0.798	0.913
Avg		0.182	0.182	0.182	0.175	0.190	0.190	0.190	0.180	0.515	0.502	0.510	0.478	0.417	0.537
ETTh1	96	0.389	0.392	0.393	0.381	0.390	0.391	0.394	0.382	0.449	0.444	0.442	0.431	0.524	0.730
	192	0.451	0.450	0.452	0.445	0.444	0.450	0.437	0.440	0.571	0.566	0.565	0.561	0.571	0.820
	336	0.498	0.495	0.534	0.480	0.479	0.481	0.512	0.474	0.656	0.652	0.653	0.635	0.614	0.899
	720	0.505	0.502	0.524	0.497	0.504	0.509	0.563	0.499	0.695	0.694	0.705	0.681	0.762	1.060
Avg		0.461	0.460	0.476	0.451	0.454	0.458	0.477	0.449	0.593	0.589	0.591	0.577	0.618	0.877
ETTh2	96	0.303	0.320	0.319	0.294	0.296	0.311	0.311	0.286	0.375	0.371	0.381	0.353	0.515	0.515
	192	0.378	0.371	0.413	0.364	0.383	0.385	0.412	0.375	0.438	0.439	0.437	0.430	0.568	0.572
	336	0.447	0.445	0.447	0.432	0.429	0.429	0.429	0.429	0.464	0.453	0.452	0.456	0.602	0.615
	720	0.431	0.428	0.430	0.427	0.445	0.446	0.570	0.425	0.473	0.479	0.475	0.465	0.637	0.646
Avg		0.390	0.391	0.402	0.379	0.388	0.393	0.448	0.377	0.438	0.435	0.436	0.426	0.581	0.587
ETTh1	96	0.339	0.340	0.339	0.318	0.345	0.341	0.345	0.331	0.586	0.588	0.512	0.461	0.435	0.655
	192	0.384	0.389	0.387	0.373	0.382	0.381	0.384	0.362	0.627	0.628	0.642	0.564	0.496	0.825
	336	0.416	0.412	0.414	0.408	0.431	0.423	0.437	0.420	0.691	0.689	0.673	0.642	0.585	0.867
	720	0.510	0.484	0.504	0.484	0.511	0.510	0.512	0.500	0.754	0.740	0.725	0.721	0.676	1.055
Avg		0.412	0.406	0.411	0.396	0.417	0.414	0.420	0.403	0.664	0.661	0.638	0.597	0.548	0.851
ETTh2	96	0.179	0.179	0.180	0.171	0.182	0.182	0.183	0.176	0.271	0.267	0.273	0.233	0.434	0.334
	192	0.245	0.247	0.249	0.237	0.255	0.254	0.259	0.249	0.290	0.298	0.295	0.281	0.602	0.873
	336	0.309	0.311	0.311	0.298	0.327	0.325	0.332	0.315	0.359	0.357	0.353	0.348	0.829	1.156
	720	0.406	0.402	0.411	0.391	0.435	0.432	0.440	0.420	0.435	0.432	0.431	0.423	2.819	2.090
Avg		0.285	0.285	0.288	0.274	0.300	0.298	0.304	0.290	0.339	0.339	0.338	0.321	1.171	1.113
ETTm2	96	0.084	0.081	0.080	0.085	0.099	0.097	0.098	0.082	0.195	0.166	0.175	0.165	0.338	0.709
	192	0.190	0.189	0.189	0.182	0.180	0.177	0.176	0.167	0.260	0.234	0.237	0.236	0.591	0.771
	336	0.319	0.313	0.374	0.279	0.352	0.359	0.420	0.300	0.437	0.434	0.432	0.407	0.617	0.848
	720	0.809	0.806	0.808	0.642	0.901	0.870	0.873	0.714	1.144	1.130	1.134	1.050	1.041	1.183
Avg		0.350	0.347	0.363	0.297	0.383	0.376	0.392	0.316	0.509	0.491	0.495	0.465	0.647	0.878
Exchange	96	0.423	0.424	0.424	0.410	0.444	0.445	0.443	0.426	0.659	0.634	0.664	0.653	0.546	0.677
	192	0.441	0.447	0.447	0.431	0.460	0.463	0.467	0.448	0.829	0.827	0.844	0.804	0.549	0.690
	336	0.458	0.451	0.457	0.443	0.479	0.475	0.472	0.456	1.094	1.080	1.096	1.030	0.571	0.705
	720	0.490	0.490	0.493	0.475	0.517	0.515	0.523	0.513	1.307	1.296	1.297	1.282	0.603	0.732
Avg		0.453	0.453	0.455	0.440	0.475	0.475	0.476	0.461	0.972	0.959	0.975	0.942	0.567	0.701
Traffic	96	0.171	0.168	0.170	0.160	0.173	0.170	0.172	0.162	0.262	0.260	0.299	0.235	0.251	0.322
	192	0.224	0.224	0.226	0.211	0.223	0.221	0.222	0.210	:0.3003	0.298	0.295	0.276	0.295	0.465
	336	0.278	0.275	0.276	0.254	0.284	0.281	0.282	0.261	0.323	0.321	0.321	0.309	0.316	0.514
	720	0.353	0.353	0.353	0.342	0.357	0.357	0.358	0.345	0.389	0.386	0.384	0.375	0.697	0.862
Avg		0.256	0.255	0.256	0.242	0.259	0.257	0.259	0.244	0.325	0.316	0.325	0.299	0.390	0.541

[†] OneNet and FSNet are implemented based on the public library provided in this paper, with their backbone models derived from their respective literatures. This implementation removes concept drift, and as a result, the online learning performance has deteriorated.

C.10 ABLATION STUDIES OF δ -ADAPTER'S DEPTH, WIDTH AND VALUE

Table 11 shows that ablation studies of δ -Adapter's depth and width. It can be found that the depth has little impact on performance, while the greater the width, the slight improvement in performance.

Table 11: Ablation studies of δ -Adapter’s depth and width.

Depth	2				3				4			
Width	64	128	256	512	64	128	256	512	64	128	256	512
ELC	0.159	0.157	0.155	0.1533	0.158	0.157	0.154	0.152	0.159	0.157	0.154	0.152
Weather	0.162	0.16	0.159	0.158	0.162	0.161	0.159	0.158	0.162	0.161	0.16	0.158
Traffic	0.439	0.437	0.433	0.43	0.44	0.436	0.433	0.43	0.44	0.436	0.433	0.43

C.11 THE CHOICE OF HYPERPARAMETER δ

δ is related to the properties of the dataset (e.g., noise level, degree of concept drift). In our work, we divided the datasets into two categories: one with severe concept drift ($\delta = 0.1$, e.g., Traffic, Weather, etc.) and the other with non-severe concept drift ($\delta = 0.01$, e.g., ETT, etc.). We did not perform hyperparameter searches based on models or datasets; instead, for datasets with severe concept drift, setting $\delta = 0.1$ is sufficient. In addition, we conducted ablation experiments on δ . As shown in Table 12, a better value of $\delta = 0.1$ might yield better results. In our paper, we only reported the two settings ($\delta=0.1$ or 0.01).

Table 12: Ablation studies of δ -Adapter’s value.

+0.1X		$\times 0.1X$		$\times 0.2X$		$+0.1Y$		$\times 0.1Y$		$\times 0.2Y$		$+0.1(X\&Y)$		$\times 0.1(X\&Y)$		$\times 0.2(X\&Y)$	
Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test
0.418	0.166	0.427	0.168	0.425	0.169	0.421	0.168	0.415	0.165	0.413	0.167	0.413	0.160	0.416	0.162	0.411	0.162

C.12 PERFORMANCE OF δ -ADAPTER ON BLACK-BOX MODELS.

Table 13 shows the performance of the δ -Adapter on the black-box models. Specifically, we used TabPFN Hollmann et al. (2025) and TimesFM Das et al. (2024) as frozen black-box models to conduct zero-shot testing on various datasets. For comparison, we corrected the output results of these black-box models by adding Ada-Y. As shown in the table below, it can be found that after adding Ada-Y, the prediction error of the model is significantly reduced, which further proves the effectiveness of the proposed method.

C.13 VISUALIZATION OF FEATURE SELECTOR AND CORRECTOR

Figure 11 shows that both Quantile Calibrator (QC) and Conformal Calibrator (CC) produce adaptive, heteroscedastic prediction intervals that track signal volatility—widening near peaks/troughs and typically enclosing the ground truth across diverse samples. QC tends to be more conservative (wider bands) and sometimes attains higher per-sample coverage (e.g., PICP≈0.729), while CC achieves tighter intervals with comparable coverage (e.g., PICP≈0.677 on multiple samples). The consistent behavior across the two test sets indicates that the calibrators are complementary and robust, yielding reliable uncertainty quantification beyond the raw predictor.

Figure 12 present the visualization results of the feature selector. We selected 90%, 50%, 30%, and 10% of the input data respectively to test the pre-trained iTransformer model, in order to observe their impact on the output results. It can be seen from the figure that most of the important features selected by the δ -Adapter determine the performance of the model, while other features are relatively less important.

Table 13: Performance of δ -Adapter on black-box models.

	Traffic	Weather	ELC	Exchange	ETTh1	ETTh2	ETTm1	ETTm2
TabPFN	0.367	0.875	0.115	0.129	0.129	0.180	0.037	0.114
Ada-Y	0.342	0.552	0.089	0.096	0.095	0.171	0.034	0.103
TimesFM	0.211	0.168	0.084	0.239	0.029	0.135	0.028	0.267
Ada-Y	0.196	0.157	0.081	0.215	0.024	0.104	0.025	0.223

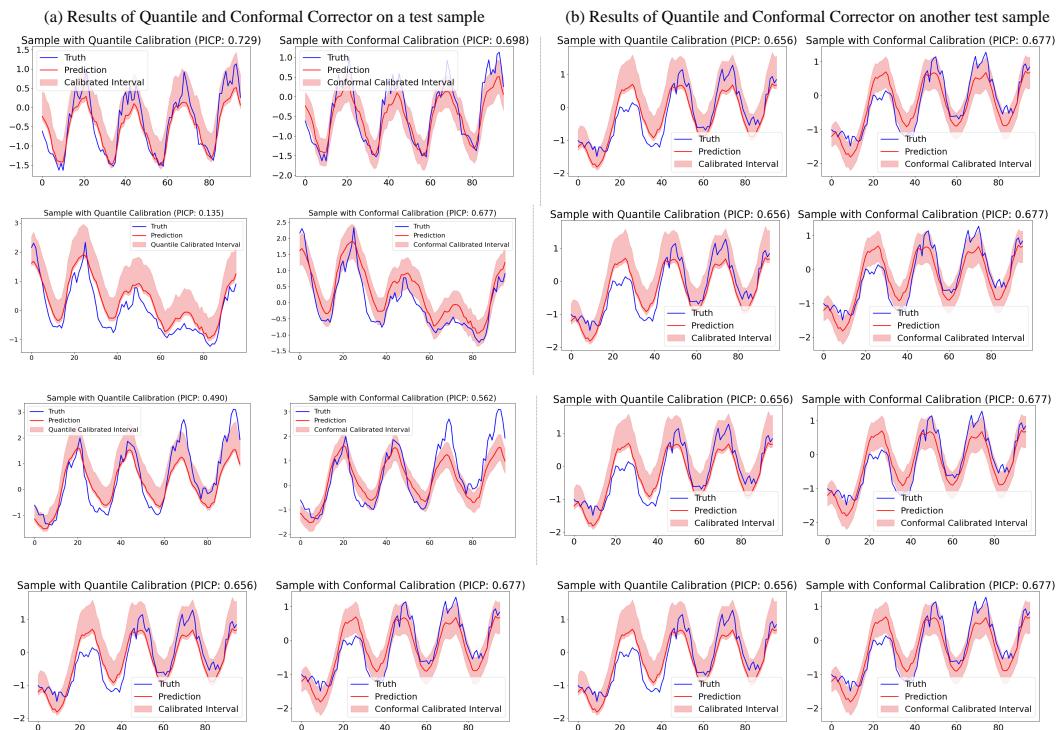


Figure 11: Visualization of the Quantile and Conformal calibrator predictions.

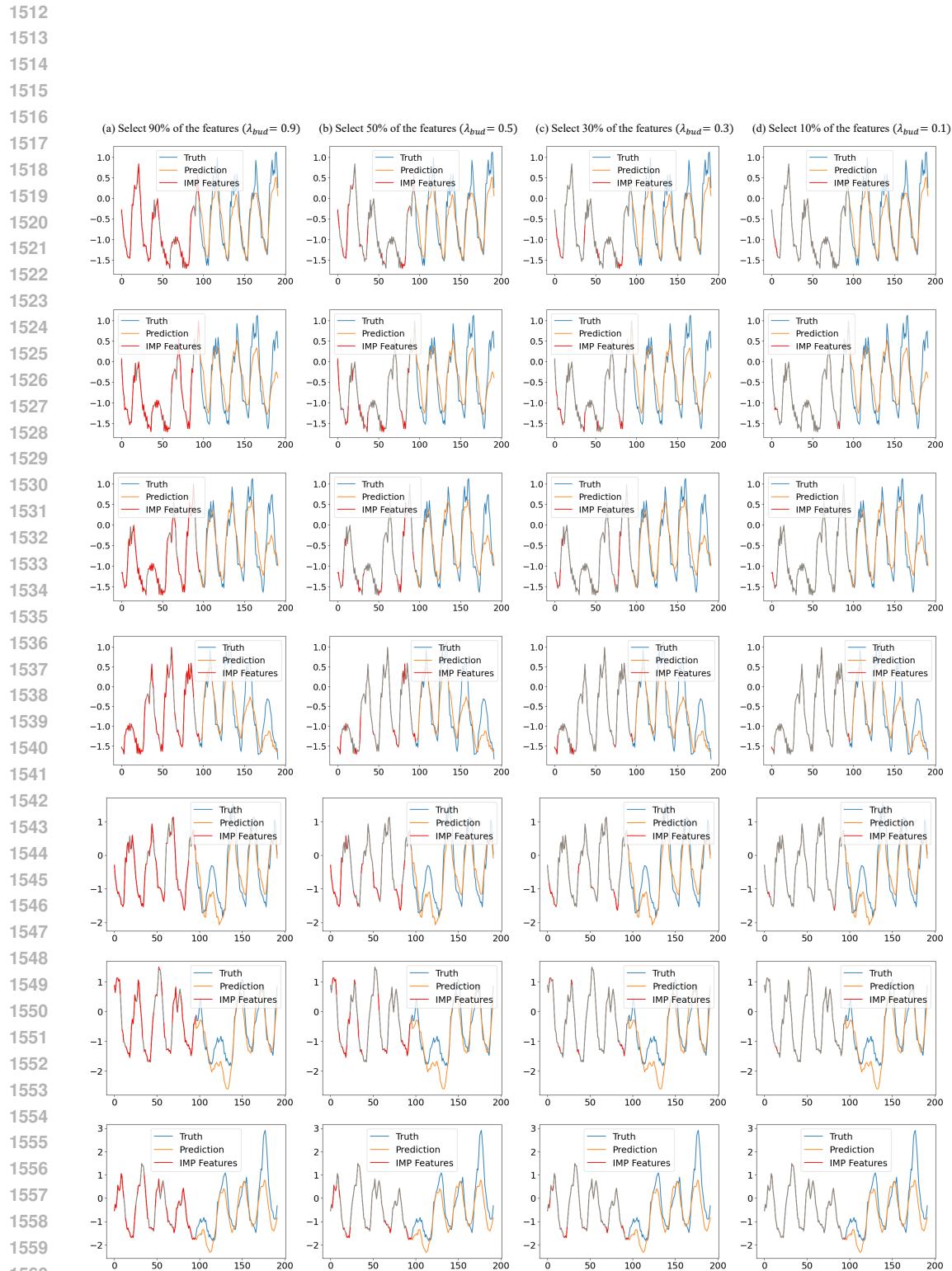


Figure 12: Visualization of different important features learned by the mask adapter.