TTC4TS: SCALING TEST-TIME COMPUTE FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time compute has become an important paradigm in computer vision and language modeling, where additional computation at inference time enables models to explore multiple candidate solutions and select or refine the most reliable one. However, time series forecasting models typically produce predictions through a single forward pass, leaving them vulnerable to distribution shifts and non-stationarity that frequently arise in real-world temporal data. We propose **T**ime **T**est **C**ompute for **T**ime **S**eries (TTC4TS), an architecture-agnostic framework that improves the predictive performance of frozen forecasting models at inference time. Given a pretrained backbone, the generator produces multiple candidate forecast trajectories using augmentation, retrieval, adapter. The synthesizer then selects or combines these candidates using aggregation, classification, or candidate-scoring and ranking methods to produce the final forecast. By increasing the number of candidates, TTC4TS scales test-time compute without retraining the forecasting backbone. Beyond improving point-forecast accuracy, TTC4TS can exploit candidate forecasts for uncertainty quantification, which supports improved interval forecasts. We also theoretically analyze when additional test-time computation improves forecasting Experiments across diverse datasets and backbones demonstrate consistent gains in accuracy and calibration. Code is provided in the *supplementary materials*.

1 INTRODUCTION

Time Series Forecasting (TSF) is a fundamental problem in machine learning, supporting decision-making in energy, transportation, finance, healthcare, retail, scientific discovery, etc. Recent years have witnessed remarkable progress in neural forecasting models, including recurrent (Connor et al., 1994; Salinas et al., 2020) or convolutional neural networks (Lea et al., 2016; Wu et al., 2022; Luo & Wang, 2024), Transformers (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Nie et al., 2023; Liu et al., 2023; Wang et al., 2024), linear architectures (Zeng et al., 2023; Yue et al., 2026), spectral models (Yi et al., 2024b;a), and emerging time series foundation models (Garza et al., 2023; Jin et al., 2024; Liu et al., 2024; Das et al., 2024; Ansari et al., 2024). Despite these advances, most forecasting systems still follow the same inference paradigm: after training, a model receives a historical context window X and produces a future trajectory $\hat{Y} = F(X)$ through a single forward propagation step. This single-shot prediction paradigm is simple and efficient, but often fragile because the model tends to be overly confident in its single-shot deterministic outputs (Lakshminarayanan et al., 2017).

In contrast, recent progress in vision and Large Language Models (LLM) suggests a different direction: test-time compute (Ma et al., 2025; Snell et al., 2025). Rather than relying on a single output, a model can allocate additional computation during inference to generate, refine, compare, or select among multiple candidate solutions. This paradigm has shown that better decisions can often be obtained not only by scaling training, model size, or data, but also by using computation more effectively at test time. However, this powerful tool remains largely unexplored in TSF. Existing forecasting models, including large time-series models (Das et al., 2024; Ansari et al., 2024) and LLM-based reasoning models (Le et al., 2026), still rely on deterministic single-shot prediction or iterative coarse-to-fine refinement and lack a framework for generating and synthesizing heterogeneous candidates while keeping the backbone frozen.

This paper asks a simple question: *Can we introduce test-time compute into TSF without retraining or modifying the deployed forecasting backbone?*

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

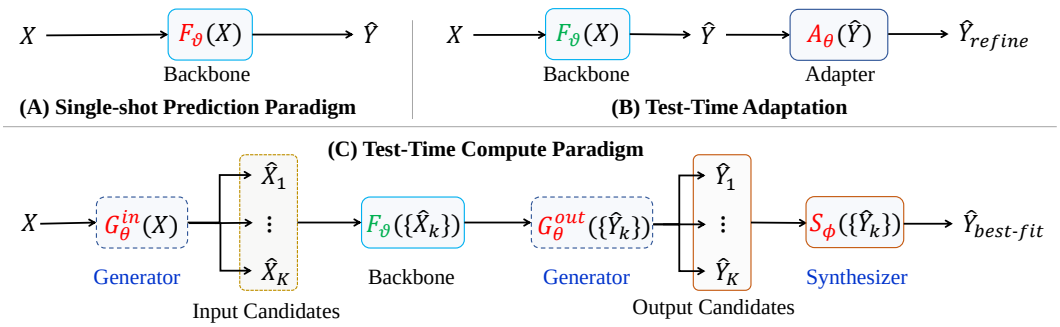

Figure 1: Comparison of time series inference paradigms (red: trainable; green: frozen). (A) Standard learning relies on a single deterministic forward pass. (B) Test-Time Adaptation refines the single output, often requiring parameter updates. (C) Our proposed Test-Time Compute framework scales compute at inference by generating multiple diverse candidates, followed by a synthesis step.

We answer this question affirmatively by proposing **T**ime **T**est **C**ompute for **T**ime **S**eries (TTC4TS), a test-time compute paradigm tailored to TSF. The main idea is to keep the pretrained forecaster F unchanged, generate a large number of candidate trajectories via the **Generator** G_θ, and then select or aggregate these candidates through the **Synthesizer** S_ϕ to produce the best-fit prediction. Figure 1 illustrates three different inference paradigms for TSF. The traditional single-shot prediction paradigm setting shown in Figure 1(A), the model is trained or updated using observed data, but prediction itself still depends on a single deterministic forward pass. Test-time adaptation, shown in Figure 1(B), attempts to refine the prediction at inference time, often by adapting model parameters or correcting the single output under some auxiliary objective. Although such methods improve flexibility, they still typically operate around one forecast trajectory and may require delicate update rules to avoid instability or leakage.

In contrast, the test-time compute paradigm shown in Figure 1(C) changes the nature of inference: instead of committing immediately to one trajectory, the model generates multiple candidate forecasts and then performs a selection or aggregation step before producing the final prediction. This final step is the key "thinking" process in our framework. TTC4TS generates multiple candidate forecasts by applying small, bounded modifications (e.g., multiple adapters) or augmentation (e.g., tunable transformation operators) to either the input or the output of the frozen forecaster F, or by retrieving similar historical contexts to produce diverse candidate trajectories $\{\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_K\}$. The synthesizer then selects or combines these candidates to produce the final forecast. The synthesizer can be implemented in several forms, including simple averaging, weighted aggregation, horizon-wise selection, or a lightweight classifier/scorer that chooses the most reliable trajectory. In this way, TTC4TS converts TSF from a single-shot prediction process into a candidate-generation and synthesis process, enabling additional inference-time reasoning while preserving the original forecasts.

Beyond improving point forecasting accuracy, TTC4TS also provides an additional benefits: The multiple candidate trajectories naturally support Uncertainty Quantification (UQ). The diversity among candidates can be used to construct prediction intervals and estimate coverage that provide good and personalized uncertainty estimates.

Our contributions are summarized as follows:

- An architecture-agnostic framework is introduced, which improves frozen time-series forecasters through candidate generation and aggregation, instantiating 3 generator families and 8 synthesis strategies with explicit pre-test fitting and causal information access.
- We characterize when expanding candidate pools improves forecasting by separating gains in attainable prediction quality from aggregation error, explaining why additional candidates require reliable synthesis rather than guaranteeing lower squared-error risk.
- We extend TTC4TS through separate conformal calibration of candidate-derived prediction intervals, improving aggregate interval quality while demonstrating that candidate diversity alone does not provide calibrated uncertainty.
- Extensive experiments show that TTC4TS delivers substantial performance gains, achieving an average improvement of 6.4% and reducing errors in 420/432 evaluated configurations.

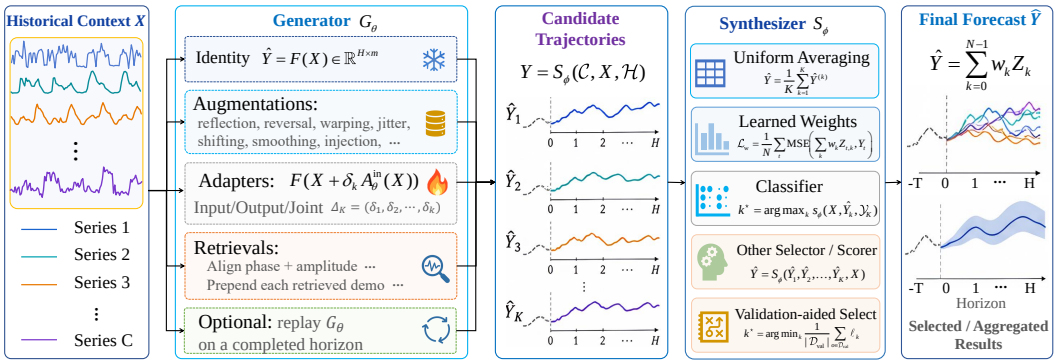

Figure 2: Overview of the TTC4TS framework for test-time compute in time series forecasting.

2 Methodology

The overall architecture of TTC4TS is shown in Figure 2. We first present the problem setup (Section 2.1), and then introduce the generator (Section 2.2) and the synthesizer (Section 2.3).

2.1 Problem Setup

Let $\mathcal{D} = \{(X^{(t)}, Y^{(t)})\}_{t=1}^{T}$ denote a training set of context–target pairs, where each context window $X^{(t)} \in \mathbb{R}^{L \times C}$ covers a history of length L over C covariates, and each future target $Y^{(t)} \in \mathbb{R}^{H \times m}$ spans a forecast horizon H across m target dimensions (we omit the superscript (t) hereafter for brevity). A pre-trained forecaster F maps a context window X to predictions $\hat{Y} = F(X) \in \mathbb{R}^{H \times m}$. The proposed framework, TTC4TS, comprises two components: a generator G_θ, which produces a set of K candidate forecasts, and a synthesizer S_ϕ, which combines them. The framework replaces single-shot inference with a bank of candidate trajectories followed by synthesis:

$$\mathcal{C} = G_\theta(F, X; \mathcal{B}) = [Z_0, \ldots, Z_{N-1}], \quad \text{then,} \quad \hat{Y} \quad = S_\phi(\mathcal{C}, X, \mathcal{H}). \tag{1}$$

Here, $\mathcal{B}$ is an eligible historical reference bank, and $\mathcal{H} = X^{\{0:t\}}$ denotes already observed history. The generator explores alternative forecasts through residual adapters, transformations, and retrieved demonstrations. The synthesizer assigns reliability weights using fixed coefficients, an offline-trained scorer, or observed retrospective errors. The current target Y is never an input to either component. Auxiliary learning is permitted, but no backbone or auxiliary parameter is updated on a test query.

2.2 Generator: Candidate Trajectories Generation

Augmentation-based Candidates: For an input transformation T_m and its associated forecast correction U_m, we form $Z_m^{\text{aug}} = U_m(F(T_m(X)))$. The implemented bank contains A candidates, including magnitude reflection, time reversal, magnitude warping, time warping, exponentially weighted moving-average smoothing, jitter, horizontal shifting, positive and negative vertical shifts, positive and negative slope changes, and spike injection. Transformation scales depend only on the observed context. Specifically, with $\mu_c = L^{-1} \sum_t X_{t,c}$, define $s_c = \max(\sqrt{\delta_\mu}, \delta)$, where $\delta_\mu = L^{-1} \sum_t (X_{t,c} - \mu_c)^2$.

In particular, reversing the input history does not reverse the output forecast. Operators with U_m equal to the identity are candidate-generating perturbations, not assumed invertible forecasting symmetries. All operator definitions, constants, and the jitter construction are provided in Appendix B.1.

Multi-Bounded Adapters: For $Z \in \mathbb{R}^{T \times C}$, an adapter predicts a residual using a lightweight neural network. Rather than fitting a separate adapter for every candidate, we use two shared adapters, r_{θ_x} and r_{θ_y}, and vary their adjustment strength. For $K \geq 2$, $\delta_j = \frac{j}{K-1} \delta_{\max}, j \in [1, K]$. The input, output, and joint paths are

$$Z_j^{\text{in}} = F(X + \delta_j r_{\theta_x}(X)), \quad Z_j^{\text{out}} \quad = Z_0 + \delta_j r_{\theta_y}(Z_0), \quad Z_j^{\text{joint}} = Z_j^{\text{in}} + \delta_j r_{\theta_y}(Z_j^{\text{in}}). \tag{2}$$

Each direct additive correction is bounded elementwise by δ_j in standardized units. This does not impose a corresponding bound on the backbone's response to an input perturbation. The input and joint paths share θ_x, and the output and joint paths share θ_y. Further, the adapters are fitted on $\mathcal{D}_{\text{adapt}}$ by minimizing the mean forecast error across all paths:

$$\mathcal{L}_{\text{adapt}} = \frac{1}{|\mathcal{D}_{\text{adapt}}|} \sum_{(X,Y)\in\mathcal{D}_{\text{adapt}}} \frac{1}{3K} \sum_{j=0}^{K-1} \sum_{v\in\{\text{in,out,joint}\}} \frac{\|Z_j^v - Y\|_F^2}{HC}. \tag{3}$$

Please refer to Appendix B.2 for more implementation details.

Retrieval-based Candidates: Retrieval supplies a separate candidate for each of M historical neighbors. A training-bank entry contains both a reference past and its already observed continuation: $R_n = \{X^{a_n} : X^{a_n+S+H}\}$, where a_n is the starting index and S is the length of the reference past. Further, using $Q = X[L-S:L]$ and $P_n = R_n[0:S]$, retrieval compares each channel after local normalization:

$$\mathcal{Z}(z)_t = \frac{z_t - \text{mean}(z)}{\max(\text{std}(z), 0.01)}, \qquad d_{n,c} = \frac{1}{S}\sum_{t=0}^{S-1} \left[\mathcal{Z}(Q_{\cdot,c})_t - \mathcal{Z}(P_{n,\cdot,c})_t\right]^2. \tag{4}$$

For each channel c, we retain the M smallest-distance entries and assemble the jth demonstration as $R_{t,c}^{(j)} = R_{n_{j,c},t,c}$. Let $R^{\text{phase},j}$ be the shifted pair. Its first S positions determine $\mu_{r,j,c}$ and $\sigma_{r,j,c}$; the query determines $\mu_{q,c}$ and $\sigma_{q,c}$. The amplitude alignment is given by

$$\alpha_{j,c} = \frac{\sigma_{q,c}\sigma_{r,j,c}}{\sigma_{r,j,c}^2 + (\sigma_{q,c}/K)^2}, \quad \beta_{j,c} = \mu_{q,c} - \alpha_{j,c}\mu_{r,j,c}, \quad \widetilde{R}_{t,c}^{(j)} = \alpha_{j,c}R_{t,c}^{\text{phase},j} + \beta_{j,c}. \tag{5}$$

The continuation inside each demonstration is historical data, not the current query's future. For further details, please refer to Appendix B.3, which specifies the exact phase-alignment calculation.

2.3 Synthesizer: Candidate Selection or Aggregation

Candidate-Bank Construction: Consistent with Eq. 1, the generator concatenates the baseline, nonzero-strength adapter candidates, transformed-input candidates, and retrieved-demonstration candidates in a fixed order:

$$\mathcal{C} = \left[Z_0, \{Z_m^{\text{aug}}\}_{m=1}^{A}, \{Z_j^{\text{in}}\}_{j=1}^{K-1}, \quad \{Z_j^{\text{out}}\}_{j=1}^{K-1}, \{Z_j^{\text{joint}}\}_{j=1}^{K-1}, \{Z_j^{\text{ret}}\}_{j=1}^{M}\right]. \tag{6}$$

Because every zero-strength adapter path recovers $F(X)$, this shared candidate is included only once. Thus, $N = 1 + 3(K-1) + A + M$. Please refer to Appendix E for the specific algorithm.

Synthesis Strategies: The synthesis strategies S_ϕ encompass a diverse range of approaches. In this work, we implement 8 distinct Synthesis Strategies, each designed to identify the synthesized forecast from a large pool of candidates. Therefore, after generating N candidate trajectories $\mathcal{Y}_K = \{\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_K\}$, we map them to the final forecast using a function $\hat{Y} = S_\phi(\mathcal{Y}_K, X)$. Here, S_ϕ follows a general rule, i.e., the synthesizer selects or combines generated candidates through a unified convex aggregation rule

$$\widehat{Y} = \sum_{k=0}^{N-1} w_k Z_k, \qquad w_k \geq 0, \qquad \sum_{k=0}^{N-1} w_k = 1. \tag{7}$$

Each candidate receives one weight for its entire forecast trajectory. One-hot weights implement hard selection, while distributed weights implement aggregation.

We consider uniform averaging, learned input-dependent synthesis (ensemble learning, learned weights, classifier, risk regressor, pairwise ranker), Retrospective Weighting and Validation-aided Selection. Please refer to Appendix C for implementation details.

3 When Does Additional Computation Help?

Forecast Geometry and Useful Diversity: Condition on available information $\mathcal{H}_t$, including fitted parameters and generator draws. Set $d = HC$, $\langle a, b\rangle_d = a^\top b/d$, $\mu = \mathbb{E}[Y_t \mid \mathcal{H}_t]$, and

$v = \mathbb{E}[\|Y_t - \mu\|_d^2 \mid \mathcal{H}_t]$. For observable simplex weights and $z_w = \sum_k w_k Z_k$,

$$R(w) = v + \|z_w - \mu\|_d^2 = w^\top Q w = \sum_k w_k R_k - \tfrac{1}{2} \sum_{i,j} w_i w_j \|Z_i - Z_j\|_d^2. \tag{8}$$

Here, $Q_{ij} = \mathbb{E}[\langle Z_i - Y_t, Z_j - Y_t \rangle_d \mid \mathcal{H}_t]$ includes bias and $R_k = Q_{kk}$. Diversity reduces risk relative to weighted individual risks, but need not improve accuracy. z^* projects μ onto the candidate hull. Adding z strictly improves this oracle exactly when $\langle z^* - \mu, z - z^* \rangle_d < 0$ (Appendix D.2).

Moreover, adapter-generated candidates are not independent. Appendix D.3 derives approximation bounds under smoothness assumptions. Refitting adapters changes the paths and invalidates fixed-generator nesting.

Aggregation Error and the Value of Computation: For static learned weights, let $\bar{Q}$ be the deployment error second-moment and $\widehat{Q}$ its fitting estimate. If their largest entrywise difference is at most η and empirical optimization error is at most γ, then

$$\bar{R}(\widehat{w}) - \min_{w \in \Delta_K} \bar{R}(w) \leq 2\eta + \gamma, \tag{9}$$

where Δ_K is the simplex. A nested pool improves deployed risk exactly when its oracle gain exceeds the increase in aggregation excess (Appendix D.4). For exponential weights $w_k \propto \exp(-s_k/T)$, $T > 0$, the separate assumption $\max_k |s_k - R_k| \leq \varepsilon$ gives

$$R(w) \leq \min_k R_k + 2\varepsilon + T \log K - D(w). \tag{10}$$

Appendices D.5 and D.6 prove the prior-dependent extension, exact identity-shrinkage correction, and historical-score concentration bound with a temporal-drift term. The recovered accuracy gain must be assessed against the measured computational cost. (Appendix D.7).

4 Candidate-informed Uncertainty Quantification

From Candidate Diversity to Prediction Intervals: In this section, we study whether candidate information helps a separate uncertainty head, followed by residual calibration. This extension follows locally adaptive conformal prediction and conformalized quantile regression (Romano et al., 2019). Let $\mu_{t,h,c}$ be the selected point forecast and let $s_{t,h,c} > 0$ be an uncertainty scale. The family-balanced displacement is

$$d_{t,h,c}^2 = \frac{1}{3} \sum_{g \in \{\text{adapter,augmentation,retrieval}\}} \frac{1}{|\mathcal{K}_g|} \sum_{k \in \mathcal{K}_g} (Z_{t,k,h,c} - \mu_{t,h,c})^2. \tag{11}$$

where $\mathcal{K}_g$ is the set of candidate indices belonging to generator family g. The identity is excluded from these three families but retained in empirical candidate quantiles, and equal-family mass prevents the many adapter grid points from dominating this dispersion statistic.

Candidate-scale CP: After the trained forecaster is fixed, a separate calibration block gives

$$R_{i,h,c} = \frac{|Y_{i,h,c} - \mu_{i,h,c}|}{s_{i,h,c}}, \qquad q_{h,c}(\alpha) = R_{(\lceil (n+1)(1-\alpha) \rceil),h,c}. \tag{12}$$

Here, n is the number of calibration origins and parentheses denote order statistics across origins, separately for each lead and channel. An order above n returns infinity rather than an unjustified finite maximum. The interval is

$$I_{t,h,c}(\alpha) = \left[\mu_{t,h,c} - q_{h,c}(\alpha) s_{t,h,c},\ \mu_{t,h,c} + q_{h,c}(\alpha) s_{t,h,c}\right]. \tag{13}$$

Under exchangeability of calibration and future scores, this construction has the usual marginal split-conformal interpretation. Temporal dependence and shifts can violate exchangeability. Our experiments report empirical coverage, without claiming finite-sample coverage under arbitrary time-series shifts or coverage conditional on every history.

5 Experiments

We validate TTC4TS on 6 datasets, 6 backbones, 4 horizons and 3 seeds (432 cells, see Appendix F.1 for more details). Results on three additional datasets are reported in Appendix G. We test its gains when applied to pre-trained and SOTA models (Section 5.1), the effectiveness of generator (Section 5.2), the effectiveness of synthesizer (Section 5.3), and UQ performance in Section 5.4.

5.1 Main Results

Table 1: TTC4TS and matched adaptation/retrieval variants on 6 datasets, 6 backbones, 4 horizons and 3 seeds (432 cells). Relative errors are geometric means of ratios to Frozen in each cell. Detailed results are provided in Appendix G.1. Seed SD summarizes 3 seed-level aggregates. Wins counts lower MSE than Frozen, and TTC wins counts cells where TTC4TS beats the row.

Method	Rel. MSE	Rel. MAE	Seed SD	Wins	TTC wins
Frozen backbone	1.000	1.000	0.0000	0/432	420/432
SOLID (Chen et al., 2024)	1.019	1.055	0.0002	2/432	428/432
TAFAS (Kim et al., 2025)	1.002	1.042	0.0014	166/432	402/432
DynaTTA (Grover & Etemad, 2025)	1.015	1.067	0.0068	112/432	424/432
AdaPTS (Benechehab et al., 2025)	0.993	1.009	0.0034	216/432	410/432
δ-Adapter input (Liang et al., 2026b)	0.981	1.013	0.0097	251/432	372/432
δ-Adapter output (Liang et al., 2026b)	0.994	1.017	0.0067	212/432	410/432
δ-Adapter composed (Liang et al., 2026b)	0.992	1.049	0.0163	189/432	386/432
TS-RAG (Ning et al., 2026)	0.985	1.007	0.0033	281/432	414/432
Align-RAG (Asadi et al., 2026)	0.981	1.018	0.0000	246/432	347/432
TTC4TS + Validation-select	0.940	0.993	0.0065	334/432	364/432
TTC4TS + Uniform	0.965	0.997	0.0012	406/432	377/432
TTC4TS + Risk regressor	0.961	0.996	0.0029	407/432	412/432
TTC4TS + Learned weights	**0.936**	**0.988**	0.0023	**420/432**	0/432

We conducted a baseline comparison and component ablation study to evaluate whether TTC4TS aggregation improves forecasting accuracy. Table 1 (see Appendix G.1 for details) shows that TTC4TS with fixed learned weights achieves the best aggregate performance across 432 evaluation cells, with geometric-mean relative errors, reductions of 6.4% compared with the frozen backbone. It improves MSE in 420/432 cells (97.2%), the highest win count among all methods. Compared with the strongest adaptation/retrieval baselines, it achieves lower MSE than each baseline in 347-428 cells. All four TTC4TS variants improve both aggregate metrics. These results demonstrate broad empirical gains across the evaluated settings and highlight the additional benefit of learned weighting.

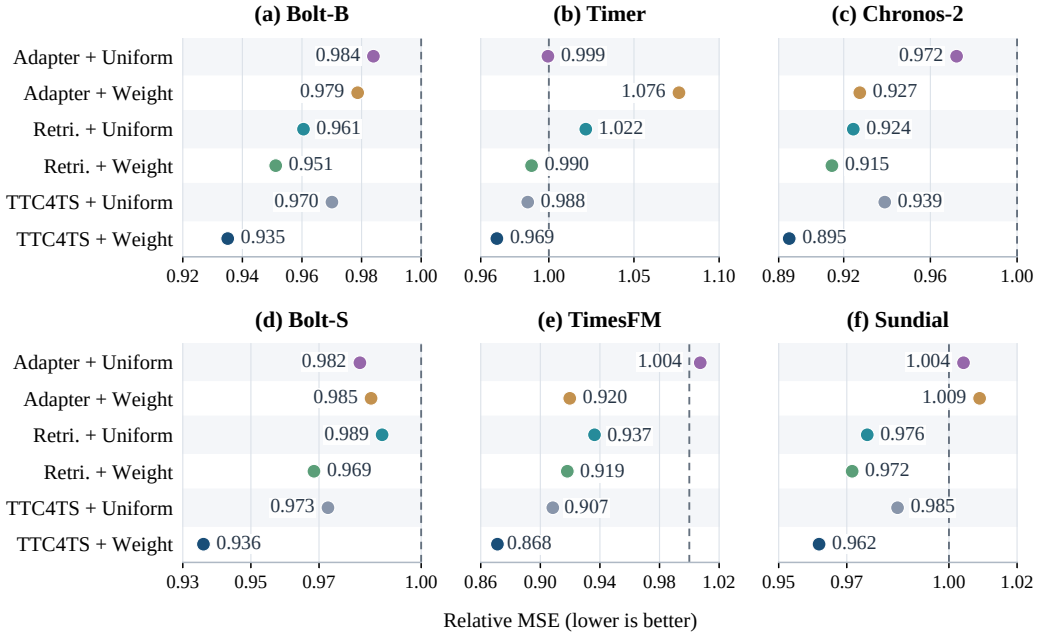

Figure 3: Ablation studies of TTC4TS. Each subplot aggregates all 6 datasets, 4 horizons, and 3 seeds for one frozen checkpoint, using geometric MSE ratios to Frozen. The dashed line is the frozen reference, and horizontal axes are truncated and their ranges vary to keep values legible.

Figure 3 shows that TTC4TS with learned weights achieves the lowest relative error across all 6 models, yielding reductions of 3.1-13.2% over the frozen backbones. It consistently outperforms standalone adapters, retrieval-based variants, and uniform aggregation. Within TTC4TS, weighting provides the largest improvement on Chronos-2. These results support the effectiveness of combining TTC4TS with weighting across the evaluated models.

5.2 Effectiveness of the Generator

We conduct a generator ablation study to evaluate the individual contributions and complementarity of Adapters (A), Augmentations (U), and Retrieval (R). By comparing these strategies individually and in combination under fixed Uniform aggregation, we examine how candidate-pool composition affects forecasting accuracy relative to the frozen backbone. The results across 6 datasets and 6 backbones are shown in Figure 4, and the detailed results are in Appendix Table 11. All 3 generation strategies reduce MSE relative to the frozen backbone. Specifically, all pairwise combinations outperform their individual components, with U+R achieving the lowest relative MSE of 0.953 among these combinations. Combining all 3 strategies yields the best result, reducing relative MSE to 0.942, a 5.8% improvement over the frozen backbone. These results suggest that the 3 generation strategies provide complementary candidates that improve forecasting accuracy when combined. In addition, every generator is evaluated separately in Appendix Table 21, including poor stress candidates.

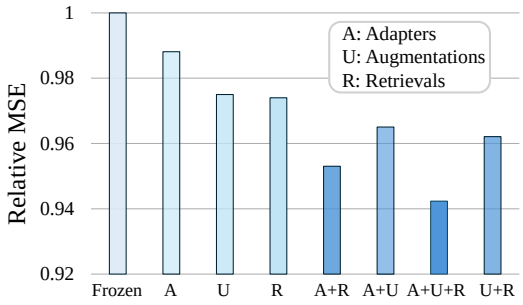

Figure 4: Generator-family comparison across 6 datasets. Results are averaged and normalized.

5.3 Effectiveness of the Synthesizer

Table 2: Geometric mean MSE relative to the frozen backbone on the six datasets over four horizons and three seeds. The experimental results are averaged and normalized for a better comparison.

Backbone	Frozen	Uniform	Weights	Class.	Risk	Rank	Ensem.	Retro.	Single	V.Select	Oracle
Bolt-S	1.000	0.973	0.936	1.286	0.977	0.992	**0.927**	0.980	0.988	0.957	0.736
Bolt-B	1.000	0.970	0.935	1.149	0.974	0.979	**0.913**	0.977	0.979	0.954	0.730
TimesFM	1.000	0.907	**0.868**	1.163	0.904	0.913	**0.868**	0.945	0.930	0.876	0.659
Timer	1.000	0.988	0.969	1.192	0.987	1.019	0.975	0.986	1.012	0.980	0.804
Sundial	1.000	0.985	**0.962**	1.290	0.985	1.027	**0.962**	0.985	1.001	0.974	0.794
Chronos-2	1.000	0.939	0.895	1.300	0.939	0.972	**0.876**	0.948	0.937	0.907	0.721

We conducted a comparative evaluation of aggregation strategies across 6 frozen forecasting backbones and 6 datasets to assess how candidate selection and combination affect forecasting accuracy and whether these strategies consistently improve upon the frozen backbone. The results are presented in Table 2. Relative errors are computed against the identity forecast in each task before taking a geometric mean, giving equal weight to datasets, horizons, and seeds. The domain-clustered sensitivity is reported in Appendix Table 8. It is noted that the test-label Oracle provides a non-deployable reference for the remaining scope to improve candidate selection.

Table 2 shows that ensemble learning (Ensem.) and learned weights (Weights) achieve the strongest non-oracle performance, with results aggregated over six datasets, four forecasting horizons, and three seeds. Ensem. reduces MSE by 2.5-13.2% relative to Frozen across the six backbones and achieves the best or joint-best non-oracle result on five, with the largest gains on TimesFM (0.868) and Chronos-2 (0.876). Weights performs best on Timer (0.969) and matches Ensem. on TimesFM and Sundial. Uniform, Risk, Retro., and V.Select also improve over Frozen on every backbone, whereas Class. consistently increases MSE, and Rank and Single occasionally degrade performance. Despite these improvements, a clear gap remains to Oracle. This indicates a very promising direction: in the future, we can achieve better results with improved synthesizers.

Table 3 shows that Ensemble learning has the lowest aggregate MSE among deployable rules (0.919), followed by learned weights. Validation-based Selection does not outperform either fixed rule in aggregate. Selecting the best validation rule is therefore useful relative to Frozen but is not a reliable way to beat the strongest combination baseline. The trajectory Classifier increases relative MSE to 1.228. Pairwise ranking lowers MSE to 0.983 while increasing MAE to 1.026, so its effect depends on the metric. The Oracle uses current test labels and is unavailable at deployment. Its relative MSE of 0.739 diagnoses candidate opportunity rather than achievable selection performance.

Table 3: Aggregation over the same 80-candidate pool across 6 datasets. Seed SD is the standard deviation of seed-level geometric mean relative MSE. Wins count model–dataset–horizon–seed cells. Oracle (targets) cannot be deployed.

Aggregation	Rel. MSE	Rel. MAE	Seed SD	Wins
Frozen backbone	1.000	1.000	0.0000	0/432
Uniform	0.965	0.997	0.0012	406/432
Learned weights	0.936	0.988	0.0023	**420/432**
Classifier	1.228	1.156	0.0186	73/432
Risk regressor	0.961	0.996	0.0029	407/432
Pairwise ranker	0.983	1.026	0.0086	213/432
Ensemble learning	**0.919**	**0.985**	0.0015	384/432
Retrospective	0.970	0.996	0.0011	375/432
Validation-best candidate	0.974	1.005	0.0006	240/432
Validation-Selection	0.940	0.993	0.0065	334/432
Oracle (unavailable)	0.739	0.905	0.0003	432/432

In addition, ensemble learning achieves the lowest errors, whereas Learned weights improves over the backbone in more configurations. This distinction reflects the magnitude and frequency of improvements: win counts ignore their size, while relative MSE accounts for both gains and degradations.

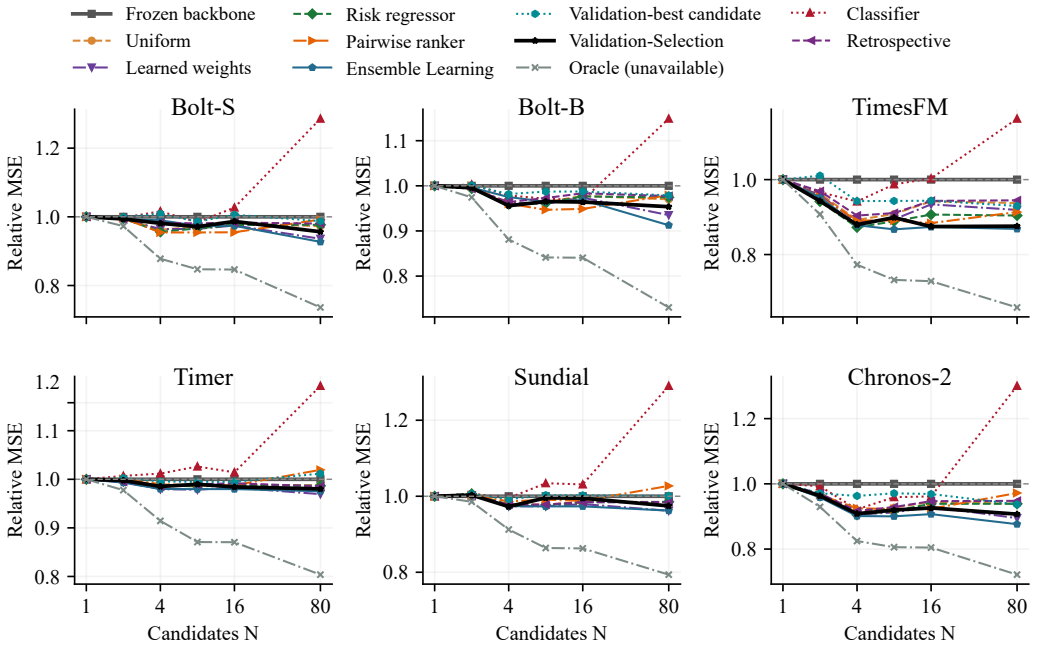

Figure 5: Candidate-budget curves on 6 datasets for all eleven rules in the legend. Values are geometric mean relative MSE over four horizons and three seeds. Ordering is fixed before evaluation, and Oracle uses unavailable test labels. Candidate count differs from wall-clock cost.

Further, we conducted a candidate-budget scaling experiment to investigate whether increasing the number of candidate forecasts improves prediction accuracy and how this relationship depends on the aggregation strategy. Specifically, we varied the candidate budget from 1 to 80 across 6 frozen forecasting backbones and compared different aggregation methods. The results shown in Figure 5 shows that the benefits of increasing the candidate budget depend strongly on the aggregation strategy. At $N = 80$, Learned weights and Ensemble learning improve over the frozen backbone across all 6 models, whereas the Classifier consistently deteriorates. In addition, the Oracle error continues to decrease as the pool expands, revealing useful candidates that practical synthesizers do not fully exploit. These results highlight that effective aggregation is essential for converting additional candidates into forecasting gains.

Table 4: Comparison of methods for quantifying uncertainty. C80/C90/C95 are empirical pointwise coverage percentages, and W90 is mean 90% interval width in fitting-standardized units. R-WIS is the geometric mean WIS ratio to Frozen split CP over the three interval levels; lower is better. Coverage and width are arithmetic means over cells.

Method	C80	C90	C95	W90	R-WIS
Frozen Gaussian (Hyndman & Athanasopoulos, 2018)	84.9	90.5	93.5	2.930	1.155
Frozen split CP (Lei et al., 2018)	81.6	89.9	94.9	2.168	1.000
Raw candidate quantiles (Koenker & Hallock, 2001)	25.6	31.4	38.8	0.314	1.596
History-scale CP (Romano et al., 2019)	81.0	89.7	94.9	2.254	1.022
CQR (non-shrinking) (Romano et al., 2019)	83.8	93.1	97.0	2.453	1.029
Candidate-spread CP (Xu & Xie, 2023)	80.9	**90.0**	95.2	2.539	1.043
TTC4TS split CP	81.3	89.8	94.9	2.126	0.985
TTC4TS-quantile CP	82.2	90.7	95.7	2.116	**0.971**
TTC4TS-scale CP	80.8	89.8	**95.0**	2.276	1.018
TTC4TS bounded ACI	**80.3**	90.2	95.4	2.143	0.974

Note. Coverage ranks minimize deviation from nominal, and WIS ranks minimize error within each block.

5.4 Candidate-informed UQ

We investigate whether TTC4TS candidates can support informative prediction intervals in addition to accurate point forecasts. The uncertainty experiment covers all 6 backbones, all 6 datasets, 3 seeds, totally 108 complete cells. Comparators include Gaussian intervals, split Conformal Prediction (CP), a non-shrinking CQR implementation (Romano et al., 2019), and bounded adaptive conformal inference (Gibbs & Candes, 2021). Protocol details appear in Appendix I.

Table 4 summarizes the uncertainty quantification performance across 108 experimental settings. It shows that TTC4TS-quantile CP achieves the lowest R-WIS (0.971), corresponding to a 2.9% improvement over Frozen split CP, while reducing W90 from 2.168 to 2.116 (2.4%). Its empirical coverage is slightly conservative at 82.2%, 90.7%, and 95.7%. TTC4TS bounded ACI achieves a similarly low R-WIS with coverage closer to the nominal targets (80.3%, 90.2%, and 95.4%). TTC4TS-scale CP attains 95.0% coverage at the 95% level, but produces wider intervals (W90 = 2.276) and a higher R-WIS (1.018) than Frozen split CP. In contrast, raw candidate quantiles produce very narrow intervals but exhibit severe undercoverage, with C90 reaching only 31.4%. Overall, the quantile and bounded ACI variants offer favorable balances between coverage calibration and interval sharpness, although the improvements do not extend to every TTC4TS variant.

It is worth noting that calibration is essential: uncalibrated candidate quantiles cover only 31.4% of targets at nominal 90%. Moreover, the learned TTC4TS-scale CP head has relative WIS 1.018, showing that a more complex uncertainty estimator does not necessarily improve interval quality. Overall, the results demonstrate the utility of conformally calibrated candidate quantiles, while distinguishing empirical coverage from guarantees under temporal dependence.

6 Conclusion

In this paper, we proposed TTC4TS, a generate-and-synthesize framework for improving forecasts from a fixed backbone. Through augmentation-, adapter-, and retrieval-based candidates, it makes the inference procedure an additional design choice without requiring backbone retraining or test-time parameter updates. Across the 432 main evaluation cells, learned-weight synthesis achieves a 6.4% reduction in geometric-mean relative MSE and improves on the frozen backbone in 420 cells. The central finding is conditional rather than monotonic: additional computation helps when candidates provide useful alternatives and synthesis recovers their benefit. Our analysis and candidate-budget experiments distinguish this benefit from redundant sampling and unreliable selection. Separately calibrated candidate quantiles further improve aggregate interval quality, while raw candidate disagreement does not constitute calibrated uncertainty. Future work should focus on cost-aware candidate generation, more reliable synthesis, and calibration under dependence. More broadly, TTC4TS shifts attention from scaling the forecaster alone to determining which additional computations are worth performing before issuing a forecast.

AI USE STATEMENT

In this work, we used generative-AI tools to polish our paper (to check whether our writing is correct). We did not employ generative-AI tools to participate in the writing process, and we have not used generative AI tools for other tasks. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study does not involve research on human participants or animals. The authors declare no conflicts of interest and full compliance with research integrity and publication ethics.

REPRODUCIBILITY STATEMENT

The code can be obtained from the **supplementary materials** uploaded along with the paper. We have reproduced the experiments in this paper using a **notebook**, so that readers can quickly replicate the results presented in the paper. Therefore, we recommend that you open and run our notebook to view or execute the results. The steps to reproduce the paper are:

- 1. Download the code.
- 2. Install the necessary environment.
- 3. Run "python scripts/reproduce_results.py".
- 4. Or, run the provided notebook in "./notebooks/*".

REFERENCES

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

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

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Mohammad Asadi, Soheil Hor, Bardiya Akhbari, Jack W O'Sullivan, Tahoura Nedaee, Layne C Price, Raviteja Anantha, Euan Ashley, and Ehsan Adeli. Align-rag: Alignment is all you need for tsfm in-context learning. *arXiv preprint arXiv:2608.05571*, 2026.

Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.

Abdelhakim Benechehab, Vasilii Feofanov, Giuseppe Paolo, Albert Thomas, Maurizio Filippone, and Balázs Kégl. Adapts: adapting univariate foundation models to probabilistic multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25, 2025.

Johannes Bracher, Evan L Ray, Tilmann Gneiting, and Nicholas G Reich. Evaluating epidemic forecasts in an interval format. *PLoS computational biology*, 17(2):e1008618, 2021.

Luis Candanedo. Appliances energy prediction. UCI Machine Learning Repository, 2017. URL `https://doi.org/10.24432/C5VC8G`. Dataset, CC BY 4.0.

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

Nicolo Cesa-Bianchi, Gábor Lugosi, et al. *Prediction, learning, and games*, volume 1. Cambridge university press Cambridge, 2006.

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

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

Song Chen. Beijing pm2.5. UCI Machine Learning Repository, 2015. URL `https://doi.org/10.24432/C5JS49`. Dataset, CC BY 4.0.

Jerome T Connor, R Douglas Martin, and Les E Atlas. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5(2):240–254, 1994.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR, 21–27 Jul 2024.

Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28, 1985.

Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Shivam Grover and Ali Etemad. Shift-aware test-time adaptation and benchmarking for time-series forecasting. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025.

Xiao Hong, Guijuan Zhang, Dianjie Lu, Hong Liu, Lei Zhu, and Mingliang Xu. Personalized crowd emotional contagion coupling the virtual and physical cyberspace. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(3):1638–1652, 2022.

Xiao Hong, Yuexing Han, and Bing Wang. Integrated extensive detection, contact tracing and dynamical social distancing interventions to prevent future epidemic waves. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(10):6281–6292, 2023.

Xiao Hong, Guijuan Zhang, Zena Tian, Bing Wang, and Dianjie Lu. Integrating adaptive rewiring and compulsory regulating: Co-evolutionary epidemic dynamics driven by multi-source information. *IEEE Transactions on Network Science and Engineering*, 13:6622–6639, 2026.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *The Tenth International Conference on Learning Representations*, 1(2):3, 2022.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8.

Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4): 143–156, 2001.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval (SIGIR)*, pp. 95–104, Ann Arbor, MI, USA, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Xuan-May Le, Minh-Tuan Tran, Ling Luo, Uwe Aickelin, Dinh Phung, and Trung Le. Efficient test-time scaling for llm-based time series forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 2437–2448, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3818201. URL https://doi.org/10.1145/3770855.3818201.

Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European conference on computer vision*, pp. 47–54. Springer, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Thomas L Lee, William Toner, Rajkarn Singh, Artjom Joosen, and Martin Asenov. Lightweight online adaption for time series foundation model forecasts. In *Forty-second International Conference on Machine Learning*, 2025.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in 33rd Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 5243–5253, Vancouver, Canada, 2019.

Daojun Liang. Distpred: A distribution-free probabilistic inference method for regression and forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 753–764, 2025.

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

Daojun Liang, Haixia Zhang, Dongfeng Yuan, Bingzheng Zhang, and Minggao Zhang. Minusformer: Improving time series forecasting by progressively learning residuals. *arXiv preprint arXiv:2402.02332*, 2024b.

Daojun Liang, Haixia Zhang, Dongfeng Yuan, and Minggao Zhang. Periodformer: An efficient long-term time series forecasting method based on periodic attention. *Knowledge-Based Systems*, 304:112556, 2024c. ISSN 0950-7051.

Daojun Liang, Jing Chen, Xiao Wang, Yinglong Wang, and Shuo Li. Deepboots: Dual-stream residual boosting for drift-resilient time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23391–23400, 2026a.

Daojun Liang, Qi Li, Yinglong Wang, Jing Chen, Hu Zhang, Xiaoxiao Cui, Qizheng Wang, and Shuo Li. The forecast after the forecast: A post-processing shift in time series. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=syfWdclGE1.

Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R News*, 2(3): 18–22, 2002.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, pp. 4095–4106, 2024.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35: 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *International conference on learning representations*, volume 2024, pp. 31728–31770, 2024.

Haoyu Ma, Yushu Chen, Wenlai Zhao, Jinzhe Yang, Yingsheng Ji, Xinghua Xu, Xiaozhu Liu, Hao Jing, Shengzhuo Liu, and Guangwen Yang. A mamba foundation model for time series forecasting. *arXiv preprint arXiv:2411.02941*, 2024.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2523–2534, 2025.

Heitor R. Medeiros, Hossein Sharifi-Noghabi, Gabriel L. Oliveira, and Saghar Irandoust. Accurate parameter-efficient test-time adaptation for time series forecasting. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. Ts-rag: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. *Advances in Neural Information Processing Systems*, 38:163170–163199, 2026.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7654–7673, Online, November 2020. Association for Computational Linguistics.

Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven CH Hoi. Learning fast and slow for online time series forecasting. In *8th International Conference on Learning Representations (ICLR)*, 2023.

Domenico Piccolo. A distance measure for classifying arima models. *Journal of time series analysis*, 11(2):153–164, 1990.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in 31st Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 6000–6010, Long Beach, USA, 2017.

Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min-ge Xie. Nonparametric predictive distributions based on conformal prediction. In *Conformal and probabilistic prediction and applications*, pp. 82–102. PMLR, 2017.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024.

Qingsong Wen, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, Tieniu Tan, et al. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. *Advances in Neural Information Processing Systems*, 36:69949–69980, 2023.

Robert L Winkler and Spyros Makridakis. The combination of forecasts. *Journal of the Royal Statistical Society: Series A (General)*, 146(2):150–157, 1983.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 22419–22430, Virtual Conference, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2022.

Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, pp. 11559–11569. PMLR, 2021.

Chen Xu and Yao Xie. Sequential predictive conformal inference for time series. In *International Conference on Machine Learning*, pp. 38707–38727. PMLR, 2023.

Ying yee Ava Lau, Zhiwen Shao, and Dit-Yan Yeung. Fast and slow streams for online time series forecasting without information leakage. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filternet: Harnessing frequency filters for time series forecasting. *Advances in Neural Information Processing Systems*, 37:55115–55140, 2024a.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36, 2024b.

Wenzhen Yue, Yong Liu, Hao Wang, Haoxuan Li, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. Olinear: A linear model for time series forecasting in orthogonally transformed domain. *Advances in Neural Information Processing Systems*, 38:60573–60632, 2026.

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

Qianru Zhang, Chenglei Yu, Haixin Wang, Yudong Yan, Yuansheng Cao, Siu-Ming Yiu, Tailin Wu, and Hongzhi Yin. Fldmamba: Integrating fourier and laplace transform decomposition with mamba for enhanced time series prediction. *arXiv preprint arXiv:2507.12803*, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 11106–11115, Virtual Conference, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162, pp. 27268–27286, Baltimore, Maryland, 2022.

A Related Work

A.1 Deep Learning for Time Series Forecasting

Time series (TS) forecasting is a fundamental problem characterized by a rich trajectory of methodological advancements. While classical statistical methods (e.g., ARIMA (Piccolo, 1990) and Exponential Smoothing (Gardner Jr, 1985)) and traditional machine learning algorithms (e.g., Random Forests (Liaw et al., 2002) and SVR (Castro-Neto et al., 2009)) offer strong theoretical guarantees and interpretability, they often struggle to capture the complex, high-dimensional dependencies inherent in real-world data. Consequently, deep learning paradigms have dominated recent literature. Early deep models leveraged CNNs (LeCun et al., 1998; Lai et al., 2018) and RNNs (Connor et al., 1994; Salinas et al., 2020) to capture non-linear spatiotemporal dynamics. Subsequently, Transformers (Vaswani et al., 2017) revolutionized the field, with variants introducing mechanisms like sparse attention (Zhou et al., 2021; Li et al., 2019), frequency-domain analysis (Zhou et al., 2022; Yi et al., 2024b;a), and period-based formulations (Wu et al., 2022; Liang et al., 2024c).

Recently, the literature has diversified into several advanced paradigms: (1) *Smarter Transformers* that optimize context length and computational efficiency, such as PatchTST (Nie et al., 2022), iTransformer (Liu et al., 2023), and TimeXer (Wang et al., 2024); (2) *Linear and State-Space Models* that rival Transformers at a fraction of the computational cost, including DLinear (Zeng et al., 2023), N-HiTS (Challu et al., 2023), and Mamba-based architectures (Ma et al., 2024; Ahamed & Cheng, 2024; Zhang et al., 2025); and (3) *Foundation Models* (e.g., TimeGPT (Garza et al., 2023), TimeLLM (Jin et al., 2024), UniTime (Liu et al., 2024)), which leverage massive cross-domain corpora to enable zero-shot capabilities. Furthermore, some models explicitly integrate physical laws (Hong et al., 2023; 2026; 2022) or utilize decomposition architectures (Wu et al., 2021; Liu et al., 2022; Liang et al., 2024b; Liang, 2025) to disentangle complex patterns. Despite these architectural innovations, most of these static models implicitly assume stationarity, leaving them highly vulnerable to concept drift when deployed in dynamic, real-world environments.

A.2 Adapting to Concept Drift: Online Learning and Test-Time Adaptation

To circumvent the performance degradation caused by non-stationarity (Anderson, 1976; Liang et al., 2024a), researchers have increasingly explored continuous adaptation strategies.

Online learning approaches integrate drift-awareness directly into the forecasting pipeline. For instance, FSNet (Pham et al., 2023) balances a slow-learning backbone with fast-adapting components inspired by complementary learning systems, whereas OneNet (Wen et al., 2023) dynamically ensembles models specialized in temporal and cross-series dependencies, significantly improving online forecasting robustness (Liang et al., 2026a).

Parallel to online learning, Test-Time Adaptation (TTA) aims to adjust pre-trained models during inference. Prominent adapter-based methods include SOLID (Chen et al., 2024), which retrains specific layers using retrieved similar samples, and TAFAS (Kim et al., 2025), which updates linear adapters based on detected temporal cycles. Extensions like PETSA (Medeiros et al., 2025), DynaTTA (Grover & Etemad, 2025), and ELF (Lee et al., 2025) further refine this via dynamic gating and auxiliary losses. However, these methods typically rely on linear adapters, parallel fusion, or computationally expensive fine-tuning of internal weights. More critically, they predominantly overlook the insidious problem of label leakage (Liang et al., 2024a; yee Ava Lau et al., 2025). In contrast, our proposed TTC4TS fundamentally departs from these paradigms by executing non-linear adaptations exclusively at the input and output boundaries. It achieves rapid updates utilizing only the most recent samples, serves dual functions as a feature selector and corrector, and provides robust theoretical guarantees against label leakage.

A.3 Parameter-Efficient Fine-Tuning and Uncertainty Calibration

Our approach draws conceptual inspiration from Parameter-Efficient Fine-Tuning (PEFT) methods originally pioneered in Natural Language Processing (NLP). Techniques such as bottleneck adapters (Houlsby et al., 2019; Pfeiffer et al., 2020), Low-Rank Adaptation (LoRA) (Hu et al., 2022). While δ-Adapter shares the high-level philosophy of freezing the backbone, it is uniquely tailored to the continuous, multi-horizon nature of TS forecasting. Unlike NLP adapters that require white-box

access to modify internal attention or feed-forward layers, δ-Adapter operates strictly at the input or output interface, allowing it to wrap around any black-box forecaster.

Furthermore, handling output uncertainty is critical in TS adaptation. Conformal prediction (Vovk et al., 2017; Romano et al., 2019) has emerged as a popular non-parametric framework to construct valid prediction intervals (Xu & Xie, 2021; 2023; Liang et al., 2026b). However, conformal prediction heavily relies on the assumption of exchangeable residuals, which is routinely violated in TS data due to strong temporal autocorrelations and non-i.i.d. dynamics. In contrast, TTC4TS keeps the forecasting backbone and all fitted auxiliary parameters fixed during testing. It generates alternative forecasts through transformations, bounded residual adapters, and historical retrieval, then selects or aggregates these candidates using information available at the forecast origin.

B Generator Operator Definitions

This appendix specifies the transformations used in the candidate generator. We omit the forecast-origin subscript and use zero-based indices $t = 0, \dots, L-1$ and $h = 0, \dots, H-1$. All quantities operate on the series standardized using the training prefix. Let

$$\mu_c = \frac{1}{L}\sum_{t=0}^{L-1} X_{t,c}, \qquad s_c = \max\left(\sqrt{\frac{1}{L}\sum_{t=0}^{L-1}(X_{t,c}-\mu_c)^2}, \delta\right). \tag{14}$$

For transformation m, write $B^{(m)} = F(T_m(X))$ for the uncorrected backbone forecast and $Z^{(m)}$ for the resulting candidate.

B.1 Augmentation-based Candidates

Magnitude Flip: Reflect the history around its channel mean and apply the same reflection to the forecast:

$$T_{\mathrm{mf}}(X)_{t,c} = 2\mu_c - X_{t,c}, \qquad Z^{(\mathrm{mf})}_{h,c} = 2\mu_c - B^{(\mathrm{mf})}_{h,c}. \tag{15}$$

Time Flip: Reverse the observed history without reversing the forecast:

$$T_{\mathrm{tf}}(X)_{t,c} = X_{L-1-t,c}, \qquad Z^{(\mathrm{tf})} = B^{(\mathrm{tf})}. \tag{16}$$

This is a perturbation-based candidate, not an assertion of time-reversal equivariance.

Magnitude Warp: Define the positive modulation

$$g(u) = \exp\left(\epsilon \sin\frac{2\pi u}{L}\right), \tag{17}$$

where, ϵ is a small constant factor. Then, we use

$$T_{\mathrm{mw}}(X)_{t,c} = \mu_c + (X_{t,c}-\mu_c)g(t), \qquad Z^{(\mathrm{mw})}_{h,c} = \mu_c + \frac{B^{(\mathrm{mw})}_{h,c}-\mu_c}{g(L+h)}. \tag{18}$$

Time Warp: Distort the sampling locations by

$$u_t = \frac{t}{L-1}, \qquad \tau_t = (L-1)\left[u_t + 0.05\sin(2\pi u_t)\right]. \tag{19}$$

Let $a_t = \lfloor \tau_t \rfloor$, $b_t = \min(a_t+1, L-1)$, and $\lambda_t = \tau_t - a_t$. Linear interpolation yields

$$T_{\mathrm{tw}}(X)_{t,c} = (1-\lambda_t)X_{a_t,c} + \lambda_t X_{b_t,c}, \qquad Z^{(\mathrm{tw})} = B^{(\mathrm{tw})}. \tag{20}$$

EWMA Smoothing: Construct a smoothed history recursively:

$$\widetilde{X}_{0,c} = X_{0,c}, \qquad \widetilde{X}_{t,c} = 0.3X_{t,c} + 0.7\widetilde{X}_{t-1,c} \quad (t \geq 1), \qquad Z^{(\mathrm{ewma})} = F(\widetilde{X}). \tag{21}$$

Jitter: The jittered history and candidate are

$$\widetilde{X}_{t,c} = X_{t,c} + \alpha s_c \epsilon_{t,c}, \qquad \epsilon_{t,c} \sim \mathcal{N}(0,1), \tag{22}$$

where s_c is the history's standard deviation and α controls the noise strength. The implementation constructs reproducible pseudo-random noise rather than drawing fresh independent noise on each call. For seed s, define

$$\widetilde{X}^{(j)} = X + \alpha_j s \odot \epsilon^{(j)}, \qquad \widehat{Y}^{(j)} = F(\widetilde{X}^{(j)}). \tag{23}$$

Specifically, in our paper implementation, the noise is deterministic for a fixed query and seed, and the construction should not be described as supplying independent posterior samples.

Horizontal Shift: With $d = \max(1, \lfloor L/K \rfloor)$, shift the history using left-endpoint replication:

$$T_{\mathrm{hs}}(X)_{t,c} = X_{\max(0,t-d),c}, \qquad Z^{(\mathrm{hs})} = B^{(\mathrm{hs})}. \tag{24}$$

Positive and Negative Vertical Shifts: Each sign produces a separate candidate. For $b_c^{\pm} = \pm 0.5 s_c$,

$$T_{\mathrm{vs}\pm}(X)_{t,c} = X_{t,c} + b_c^{\pm}, \qquad Z_{h,c}^{(\mathrm{vs}\pm)} = B_{h,c}^{(\mathrm{vs}\pm)} - b_c^{\pm}. \tag{25}$$

Positive and Negative Slopes: Each sign again gives a separate candidate. For $a_c^{\pm} = \pm 0.5 s_c / L$,

$$T_{\mathrm{sl}\pm}(X)_{t,c} = X_{t,c} + a_c^{\pm}(t - L + 1), \qquad Z_{h,c}^{(\mathrm{sl}\pm)} = B_{h,c}^{(\mathrm{sl}\pm)} - a_c^{\pm}(h+1). \tag{26}$$

The added historical slope is zero at the last observation, and the forecast correction uses the corresponding future offset.

Spike Injection: Let $t_\star = L - \max(2, \lfloor \frac{L}{2K} \rfloor)$. Inject a channel-scaled spike:

$$T_{\mathrm{sp}}(X)_{t,c} = X_{t,c} + 2 s_c \mathbf{1}[t = t_\star], \qquad Z^{(\mathrm{sp})} = B^{(\mathrm{sp})}. \tag{27}$$

The ten transformation types above yield twelve candidates because vertical shifts and slopes each have two signs. Only magnitude flip, magnitude warp, vertical shifts, and slope changes apply non-identity forecast corrections.

B.2 Multi-Bounded Adapters

Here, we omit the batch dimension and adopt zero-based indexing for the time dimension. All operations are performed on data standardized using statistics computed from the training prefix. The adapters generate candidate representations by scaling a learned residual. Specifically, given an input tensor $Z \in \mathbb{R}^{T \times C}$, the Adapter computes

$$\begin{aligned} u &= \mathrm{vec}(Z), \\ h_1 &= \mathcal{N}(\mathrm{ReLU}(W_1 u + b_1)), \\ h_2 &= \mathcal{N}(\mathrm{ReLU}(W_2 h_1 + b_2)), \\ r_\theta(Z) &= \mathrm{reshape}_{T \times C}\left[\tanh(W_3 h_2 + b_3)\right]. \end{aligned} \tag{28}$$

Here, $\mathcal{N}$ is the normalization used by the implementation. Because of $\tanh$,

$$|r_\theta(Z)_{t,c}| \le 1. \tag{29}$$

For $K \ge 2$, the adjustment strengths are

$$\delta_j = \frac{j}{K-1}\delta_{\max}, \qquad j = 0, \ldots, K-1. \tag{30}$$

Then, the input, output, and joint candidates are respectively

$$\begin{aligned} \widehat{Y}_j^{\mathrm{in}} &= F(X + \delta_j r_{\theta_x}(X)) \\ \widehat{Y}_j^{\mathrm{out}} &= F(X) + \delta_j r_{\theta_y}(F(X)) \\ \widehat{Y}_j^{\mathrm{joint}} &= \widehat{Y}_j^{\mathrm{in}} + \delta_j r_{\theta_y}\left(\widehat{Y}_j^{\mathrm{in}}\right). \end{aligned} \tag{31}$$

The same input Adapter θ_x is shared by the input and joint paths; the same output Adapter θ_y is shared by the output and joint paths. Their training objective is

$$\min_{\theta_x,\theta_y} \frac{1}{|\mathcal{D}_{\text{adapt}}|} \sum_{(X,Y)\in\mathcal{D}_{\text{adapt}}} \frac{1}{3K} \sum_{j=0}^{K-1} \sum_{v\in\{\text{in,out,joint}\}} \frac{\|\widehat{Y}_j^v - Y\|_F^2}{HC}, \tag{32}$$

with all parameters of F fixed. Backpropagation through F remains enabled when training the input Adapter. Further, the code also supports the optional multiplicative correction

$$A_{\theta,\delta}^{\text{mul}}(Z) = Z \odot (1 + \delta r_\theta(Z)). \tag{33}$$

Since $|r_\theta(Z)_{t,c}| \le 1$, the multiplicative factor lies in $[1-\delta, 1+\delta]$. This bounds the relative factor rather than the absolute size of the correction independently of Z.

B.3 Historical Retrieval and Alignment

References are complete historical pairs $R_n = V[16n : 16n + S + H]$, with $S = 64$, and must lie entirely inside the training bank. At origin o, only entries satisfying $16n + S + H \le o$ are eligible. The query is $Q = X[L - S : L]$. For each channel, retrieve the M smallest local-z-score distances between Q and the reference past, using the standard-deviation floor 0.01. Candidate j assembles channel c from its jth nearest reference, yielding $R_{t,c}^{(j)} = R_{n_{j,c},t,c}$. This is channel-wise retrieval, not retrieval of one common multivariate window for all channels.

Phase Alignment: Center the query and each reference past:

$$q_{t,c}^0 = Q_{t,c} - \overline{Q}_c, \qquad p_{j,t,c}^0 = R_{t,c}^{(j)} - \frac{1}{S}\sum_{u=0}^{S-1} R_{u,c}^{(j)}. \tag{34}$$

For lag ℓ, define the valid overlap

$$I_\ell = \{t : 0 \le t < S,\ 0 \le t - \ell < S\}. \tag{35}$$

The selected lag is

$$\ell_{j,c}^\star = \arg\max_{\ell\in\{-8,\dots,8\}} \frac{\sum_{t\in I_\ell} q_{t,c}^0 p_{j,t-\ell,c}^0}{\max\Big(\sqrt{\sum_{t\in I_\ell}(q_{t,c}^0)^2 \sum_{t\in I_\ell}(p_{j,t-\ell,c}^0)^2}, 10^{-6}\Big)}. \tag{36}$$

Apply that lag to the complete historical pair, replicating endpoints:

$$R_{t,c}^{\text{phase},j} = R_{\text{clip}(t-\ell_{j,c}^\star,0,S+H-1),c}^{(j)}. \tag{37}$$

Amplitude Alignment: Compute $\mu_{r,j,c}$ and $\sigma_{r,j,c}$ from the first S positions of $R^{\text{phase},j}$, and $\mu_{q,c}, \sigma_{q,c}$ from Q. Then

$$\alpha_{j,c} = \frac{\sigma_{q,c}\sigma_{r,j,c}}{\sigma_{r,j,c}^2 + (\sigma_{q,c}/5)^2 + 10^{-6}}, \qquad \beta_{j,c} = \mu_{q,c} - \alpha_{j,c}\mu_{r,j,c}, \tag{38}$$

$$\widetilde{R}_{t,c}^{(j)} = \alpha_{j,c} R_{t,c}^{\text{phase},j} + \beta_{j,c}. \tag{39}$$

The denominator is the implemented stabilized alignment rule; it is not replaced by ordinary variance-ratio rescaling. One forecast is generated from each concatenation $[\widetilde{R}^{(j)}, X]$. Historical neighbors are not averaged before forecasting.

Causal and Interface Requirements: The full retrieval configuration requires $L \ge 64$, at least M eligible entries per channel, and a backbone that accepts $L + 64 + H$ input steps. At retrospective origin $o - H$, causal replay requires the eligibility bound $16n + 64 + H \le o - H$. This requirement concerns the historical replay call, not just current-origin retrieval.

C SYNTHESIZER FOR TRAJECTORY-LEVEL SYNTHESIS

C.1 STATIC AGGREGATION

Uniform Averaging: It sets $w_k = 1/N$ and requires no synthesizer fitting. Learned weights introduce one logit a_k per candidate and solve

$$w_k = \mathrm{softmax}(a)_k, \qquad \min_a \frac{1}{|\mathcal{D}_{o,\mathrm{syn}}|} \sum_{o \in \mathcal{D}_{o,\mathrm{syn}}} \ell\left(\sum_k w_k Z_{o,k}, Y_o\right). \tag{40}$$

Here and below, $o \in \mathcal{D}_{o,\mathrm{syn}}$ indexes forecast windows in a fitting block. These weights are fixed across test inputs and are fitted separately for each model–dataset–horizon–seed configuration.

C.2 LEARNING-BASED SYNTHESIZER

Ensemble learning: The Ensemble learning instead performs 40 selections with replacement on $\mathcal{D}_{\mathrm{syn}}$. If $E_{n,o}$ averages the first n selections, the next candidate is

$$j_{n+1} = \arg\min_j \frac{1}{|\mathcal{D}_{o,\mathrm{syn}}|} \sum_{o \in \mathcal{D}_{o,\mathrm{syn}}} \ell\left(\frac{nE_{n,o} + Z_{o,j}}{n+1}, Y_o\right), \tag{41}$$

with the initial selection minimizing single-candidate fitting loss. A candidate selected c_k times receives $w_k = c_k/40$. These coefficients also remain fixed at test time.

Learned Weights: Learn one logit a_k per candidate and convert the logits into convex weights:

$$w_k = \mathrm{softmax}(a)_k. \tag{42}$$

Optimize the combined forecast directly on the aggregator-training block:

$$\mathcal{L}_{\mathrm{weights}} = \frac{1}{N} \sum_t \mathrm{MSE}\left(\sum_k w_k Z_{t,k}, Y_t\right). \tag{43}$$

Classifier: The classifier attempts to predict the best candidate. For each fitting window, let $k^\star = \arg\min_k \ell_k$. The classifier minimizes

$$\mathcal{L}_{\mathrm{class}} = -\frac{1}{|\mathcal{D}_{\mathrm{syn}}|} \sum_{o \in \mathcal{D}_{\mathrm{syn}}} \log \mathrm{softmax}(s)_{k^\star}. \tag{44}$$

At inference it returns $Z_{\arg\max_k s_k}$. Thus, classification implements multiclass hard selection rather than probability-weighted averaging.

Risk Regressor: The risk regressor aggregates candidate-level features to predict relative error, with the regression target defined as a stabilized transformation of each candidate's error relative to that of the backbone:

$$q_k = \log\left[1 + \min\left(100, \frac{\ell_k}{\max(\ell_0, 10^{-5})}\right)\right]. \tag{45}$$

The regression objective applies SmoothL1 loss to the score and target arrays over fitting windows and candidates:

$$\mathcal{L}_{\mathrm{risk}} = \mathrm{SmoothL1}\big(\{s_k\}_{o \in \mathcal{D}_{\mathrm{syn}},k}, \{q_k\}_{o \in \mathcal{D}_{\mathrm{syn}},k}\big). \tag{46}$$

At inference,

$$w_k = \mathrm{softmax}(-s)_k. \tag{47}$$

Lower predicted transformed relative error therefore receives higher weight. The risk regressor softly combines trajectories; it neither predicts raw MSE nor selects only the minimum-score candidate. Adapter candidate-count experiments use this synthesis rule.

Pairwise Ranker: The pairwise ranker learns to assign higher scores to lower-error candidates through pairwise comparisons of training errors, then selects the highest-scoring forecast trajectory at inference. For unequal-error pairs, define $d_{ij} = \mathrm{sign}(\ell_i - \ell_j)$ and optimize

$$\mathcal{L}_{\mathrm{rank}} = \mathrm{mean}_{i,j:\, d_{ij} \neq 0} \,\mathrm{softplus}[d_{ij}(s_i - s_j)]. \tag{48}$$

This objective assigns higher scores to lower-error candidates. Equal-error pairs are excluded. Inference selects $Z_{\arg\max_k s_k}$, as in classification, but supervision uses pairwise preferences rather than only the winning index.

C.3 Retrospective Weighting

At origin o, the immediately preceding horizon is already observed. We replay each candidate operator with context $X[o-H-L:o-H]$ and historical origin $o-H$, obtaining $Z_{o-H,k}$. Its retrospective error is

$$r_k = \ell(Z_{o-H,k}, V[o-H:o]). \tag{49}$$

To ensure causal replay, retrieval eligibility in this call must be checked against $o-H$, not o. Retrospective synthesis then uses

$$w_k = (1-\lambda)\,\mathrm{softmax}_k\left(-\frac{r_k}{\tau \max(r_0, 10^{-5})}\right) + \lambda \mathbf{1}[k=0]. \tag{50}$$

Validation selects $\tau \in \{0.1, 0.3, 1\}$ and $\lambda \in \{0, 0.5, 1\}$. The temperature controls concentration on recently successful operators, while λ shrinks the forecast toward the frozen backbone. This policy changes weights without fitting a synthesizer or updating parameters at test time. It requires at least $L+H$ historical observations and an additional candidate bank unless the relevant forecasts are cached.

C.4 Validation-aided Selection

Adapter supervision uses $\mathcal{D}_{\mathrm{adapt}}$, while learned synthesis rules use the aggregator-training block $\mathcal{D}_{\mathrm{syn}}$. Validation targets select hyperparameters and deployment rules; current test-future targets are unavailable throughout generation and synthesis. The learned-synthesizer optimization configuration uses 160 AdamW updates, learning rate 0.01, and weight decay 0.03. Offline auxiliary fitting is distinct from backbone retraining and from online parameter adaptation.

Validation-best Candidate: In addition to the synthesis rules above, validation-best selects a single candidate operator:

$$k^\star = \arg\min_k \frac{1}{|\mathcal{D}_{\mathrm{val}}|} \sum_{o \in \mathcal{D}_{\mathrm{val}}} \ell_k, \qquad \widehat{Y} = Z_{k^\star}. \tag{51}$$

The operator is fixed throughout testing, but its trajectory is regenerated for each input. This is not per-query oracle selection.

Validation-aided Selection: The validation-based selection chooses an entire deployment rule from

$$\begin{aligned}\mathcal{M} = \{&\text{Frozen}, \text{Uniform}, \text{Learned weights}, \text{Classifier}, \text{Risk},\\ &\text{Ranker}, \text{Greedy}, \text{Retrospective}, \text{Validation-best}\},\end{aligned} \tag{52}$$

using

$$m^\star = \arg\min_{m \in \mathcal{M}} \frac{1}{|\mathcal{D}_{\mathrm{val}}|} \sum_{o \in \mathcal{D}_{\mathrm{val}}} \ell(\widehat{Y}^{(m)}, Y). \tag{53}$$

The selected rule is held fixed across the test split, although its candidate weights may remain input-dependent. This outer selection is a deployment policy, not an additional learned aggregator or a per-query selector between synthesis methods. It may choose the unchanged backbone.

D When Additional Computation is Beneficial

This appendix develops Section 3 in three stages: the geometry of the candidate set, the error incurred when learning or scoring its combination, and the cost of obtaining that set. All claims concern squared error in the fitting-standardized coordinates. They do not assert corresponding guarantees for MAE, geometric averages of relative errors, or probabilistic coverage. The arguments specialize classical convex projection, forecast combination, and exponential weighting (Winkler & Makridakis, 1983; Cesa-Bianchi et al., 2006); their purpose is to make the conditions relevant to TTC4TS explicit.

D.1 Information, randomness, and the risk being compared

Flatten an $H \times C$ forecast to $\mathbb{R}^d$, $d = HC$, with inner product $\langle a, b\rangle_d = a^\top b/d$ and norm $\|a\|_d$. Let $\mathcal{H}_t$ contain all observed information used at the forecast origin: history, fitted parameters,

normalization, the causal reference bank, and the realized algorithmic random draws. Such draws use no future information; conditionally on the observed history they are independent of the future target. Each candidate Z_k and deployed weight is $\mathcal{H}_t$-measurable. Assume finite conditional second moments and a finite nonempty candidate set. Conditional equalities below hold almost surely; suppress t to simplify notation. Define

$$\mu = \mathbb{E}[Y \mid \mathcal{H}_t], \qquad v = \mathbb{E}[\|Y - \mu\|_d^2 \mid \mathcal{H}_t], \qquad R(z) = \mathbb{E}[\|z - Y\|_d^2 \mid \mathcal{H}_t]. \tag{54}$$

The simplex is $\Delta_K = \{w \in \mathbb{R}^K : w_k \geq 0, \sum_k w_k = 1\}$. We write $R(w) = R(z_w)$ for $z_w = \sum_k w_k Z_k$, and $R_k = R(Z_k)$. The conditional mean μ and conditional risks are analytical quantities, not inputs to the implementation. No claim below permits reading the current future target.

Conditional risk describes one observed history with its remaining future uncertainty. In contrast, static Learned weights minimize a single risk averaged over a deployment distribution of histories and futures. We denote that risk by $\bar{R}$ and analyze it separately in Appendix D.4. An oracle allowed to choose different weights for every history is generally stronger than the best static weights. Neither oracle is the test-label selector reported in the experiments.

D.2 Exact risk geometry and the value of a new candidate

Proposition 1 (Risk and Diversity Identities): Define $Q_{ij} = \mathbb{E}[\langle Z_i - Y, Z_j - Y\rangle_d \mid \mathcal{H}_t]$. For every $w \in \Delta_K$,

$$Q_{ij} = v + \langle Z_i - \mu, Z_j - \mu\rangle_d, \tag{55}$$

$$R(w) = w^\top Q w = v + \|z_w - \mu\|_d^2 = \sum_k w_k R_k - D(w), \tag{56}$$

$$D(w) = \sum_k w_k \|Z_k - z_w\|_d^2 = \tfrac{1}{2}\sum_{i,j} w_i w_j \|Z_i - Z_j\|_d^2. \tag{57}$$

In particular, Q is positive semidefinite and $D(w) \geq 0$.

Proof: Write $Z_i - Y = (Z_i - \mu) - (Y - \mu)$. The cross terms have conditional mean zero, proving Equation 55. Since $\sum_k w_k = 1$, the mixture error is $\sum_k w_k (Z_k - Y)$, whose expected squared norm is $w^\top Q w$. Expanding $\|z_w - Y\|_d^2$ around μ gives the second equality in Equation 56. Also,

$$\tfrac{1}{2}\sum_{i,j} w_i w_j \|Z_i - Z_j\|_d^2 = \sum_k w_k \|Z_k\|_d^2 - \|z_w\|_d^2 = \sum_k w_k \|Z_k - z_w\|_d^2.$$

Subtracting this identity from $\sum_k w_k R_k = v + \sum_k w_k \|Z_k - \mu\|_d^2$ proves the remaining equality. Finally, for any vector a, $a^\top Q a = \mathbb{E}[\|\sum_k a_k (Z_k - Y)\|_d^2 \mid \mathcal{H}_t] \geq 0$.

Interpretation: $D(w)$ is observable, but the weighted individual risk in Equation 56 is not. Increasing both by adding a bad forecast need not improve their difference. After conditioning on the history and generator draws, candidates are fixed; their errors share the same random future. They are not independent draws of that future. Across random histories, an error second moment can be decomposed into bias products and centered cross moments. Calling the whole matrix an error covariance would omit the bias term.

For example, under the restrictive unconditional assumptions that every error has common bias b, centered second moment σ^2, and pairwise centered cross moment $\rho\sigma^2$, uniform averaging has risk

$$\|b\|_d^2 + \frac{K\sigma^2 + K(K-1)\rho\sigma^2}{K^2} = \|b\|_d^2 + \sigma^2\left(\rho + \frac{1-\rho}{K}\right). \tag{58}$$

For a family with fixed $0 \leq \rho \leq 1$, only the last term decreases with K. Bias and correlated error remain. This derivation explains a possible saturation mechanism; the heterogeneous TTC4TS pool is not assumed to satisfy these exchangeable-moment conditions.

Proposition 2 (Necessary and Sufficient Improvement of The Convex Oracle): Let $\mathcal{C} = \mathrm{conv}\{Z_0, \ldots, Z_{K-1}\}$ and let z^* be the projection of μ onto $\mathcal{C}$. Then

$$R_{\mathcal{C}}^* = \min_{z \in \mathcal{C}} R(z) = v + \mathrm{dist}_d(\mu, \mathcal{C})^2. \tag{59}$$

For an additional candidate z, put $u = z - z^*$, $c = \langle z^* - \mu, u\rangle_d$, and $q = \|u\|_d^2$. The enlarged hull has strictly smaller optimal risk if and only if $c < 0$. Its risk reduction is at least

$$g(c,q) = \begin{cases} 0, & q = 0 \text{ or } c \geq 0, \\ c^2/q, & q > 0,\ -q \leq c < 0, \\ -2c - q, & q > 0,\ c < -q. \end{cases} \tag{60}$$

More generally, nested candidate hulls have non-increasing optimal risk.

Proof: The finite hull is compact and convex. The squared distance has a minimizer, and strict convexity in the forecast z makes it unique (its weight representation need not be unique). Equation 56 proves Equation 59. The projection satisfies

$$\langle z^* - \mu, x - z^*\rangle_d \geq 0 \quad \text{for every } x \in \mathcal{C}. \tag{61}$$

Indeed, a negative inner product would make a sufficiently short feasible step toward x decrease the squared distance. Conversely, expanding the squared distance shows that this inequality implies optimality.

Every point of the enlarged hull has the form $(1-a)x + az$ with $x \in \mathcal{C}$ and $a \in [0,1]$. If $c \geq 0$, its displacement from z^* has nonnegative inner product with $z^* - \mu$ by Equation 61; hence z^* remains optimal. If $c < 0$, the feasible segment $z^* + au$ gives

$$R(z^* + au) - R(z^*) = 2ac + a^2 q. \tag{62}$$

This is negative for sufficiently small positive a. For $q > 0$, its minimizer on $[0,1]$ is $a^* = \min\{1, \max\{0, -c/q\}\}$. Substituting the three cases proves Equation 60. If $q = 0$, then $u = 0$ and $c = 0$. The full enlarged hull may improve more than this single segment, which is why g is a lower bound on its gain. Finally, minimizing over a superset cannot increase the minimum.

D.2.1 Why Uniform Can Get Worse

Let $\bar{z}$ be the old mean of K candidates and add z. Set $u = z - \bar{z}$, $c = \langle \bar{z} - \mu, u\rangle_d$, and $q = \|u\|_d^2$; these quantities now use $\bar{z}$, not z^*. The new mean is $\bar{z} + u/(K+1)$. Equation 62 gives strict improvement exactly when

$$2(K+1)c + q < 0. \tag{63}$$

A direction can be helpful for a small weight yet harmful at Uniform's forced weight. For a scalar deterministic target $Y = 0$, adding $z = M \neq 0$ to the perfect frozen forecast $Z_0 = 0$ raises Uniform's risk to $M^2/4$. Conversely, candidates 1 and -1 each have risk one but their equal mixture has risk zero. Neither trajectory distance nor the best individual risk fully characterizes the best ensemble.

D.3 Adapter sample count: resolution, curvature, and refitting

Proposition 3 (A Fixed Output Adapter has a One-Dimensional Hull): Fix Z_0, a residual r, and an interval $[0, m]$, where $m = \delta_{\max} > 0$. For any finite grid $\mathcal{D} \subset [0, m]$ containing both endpoints,

$$\operatorname{conv}\{Z_0 + \delta r : \delta \in \mathcal{D}\} = \{Z_0 + \delta r : 0 \leq \delta \leq m\}. \tag{64}$$

Thus its optimal convex risk is independent of grid size. A uniform linspace grid with $N \geq 2$ points has mean forecast $Z_0 + (m/2)r$ for every N. If h is the maximum adjacent grid gap, the best single grid candidate obeys

$$0 \leq \min_{\delta \in \mathcal{D}} R(Z_0 + \delta r) - \min_{0 \leq \delta \leq m} R(Z_0 + \delta r) \leq \tfrac{1}{4}\|r\|_d^2 h^2. \tag{65}$$

Proof: Every convex combination has effective adjustment $\sum_j w_j \delta_j \in [0, m]$. Conversely, each $\delta \in [0, m]$ is a convex combination of 0 and m, proving Equation 64. The arithmetic mean of linspace is $m/2$. For hard selection, set $b = \langle Z_0 - \mu, r\rangle_d$ and $q = \|r\|_d^2$. Risk is $R(Z_0) + 2b\delta + q\delta^2$. If $q = 0$, the claim is immediate. Otherwise the continuous minimizer is $\delta^* = \min\{m, \max\{0, -b/q\}\}$. When it is an endpoint, the grid attains it. When it is interior, the derivative vanishes and

$$R(Z_0 + \delta r) - R(Z_0 + \delta^* r) = q(\delta - \delta^*)^2.$$

A nearest grid point lies within $h/2$, yielding Equation 65.

D.3.1 Scope for The Implemented Adapters

Here $r = R_\psi(F(X))$ is fixed after fitting and $m = 0.3$. The statement concerns the output-only grid with one identity, not the mean of the entire mixed pool. In that pool, family proportions and the identity's mass also matter. Two linspace grids with different sizes need not be nested. For example, with $Z_0 = 0$, $r = 1$, and deterministic $Y = m/2$, a three-point grid attains zero selection error while a four-point grid does not. A smaller upper bound on discretization error is not a monotonicity theorem for non-nested grids. Refitting adapters, as in the sample-count experiment, additionally changes r and the input/composed paths.

Proposition 4 (Discretizing a Nonlinear Adapter Path): Let $z : [0, m] \to \mathbb{R}^d$ be a twice continuously differentiable forecast path with $\sup_\delta \|z''(\delta)\|_d \le B$. Let $\mathcal{C}_{\text{path}}$ be the convex hull of the full path, $\mathcal{C}_{\text{grid}}$ the hull of an endpoint-inclusive grid with maximum gap h, and $a = \text{dist}_d(\mu, \mathcal{C}_{\text{path}})$. With $\epsilon_h = Bh^2/8$,

$$0 \le R^*_{\mathcal{C}_{\text{grid}}} - R^*_{\mathcal{C}_{\text{path}}} \le 2a\epsilon_h + \epsilon_h^2. \tag{66}$$

If only L_z-Lipschitz continuity is known, the same statement holds with $\epsilon_h = L_z h/2$.

Proof: For neighboring grid points a_0, b_0 with gap $\ell \le h$ and $\delta = (1-\theta)a_0 + \theta b_0$, linearly interpolate $\widetilde{z}(\delta) = (1-\theta)z(a_0) + \theta z(b_0)$. Taylor expansion about δ cancels the first-order terms in this weighted sum. The integral remainders have norms bounded by $B|a_0 - \delta|^2/2$ and $B|b_0 - \delta|^2/2$. Therefore

$$\|\widetilde{z}(\delta) - z(\delta)\|_d \le \tfrac{B}{2}\left[(1-\theta)\theta^2 + \theta(1-\theta)^2\right]\ell^2 = \tfrac{B}{2}\theta(1-\theta)\ell^2 \le Bh^2/8.$$

Under Lipschitz continuity the corresponding bound is $(1-\theta)L_z\theta\ell + \theta L_z(1-\theta)\ell \le L_z h/2$. The continuous path is compact, and its finite-dimensional convex hull is compact: each point admits a combination of at most $d+1$ path points. Replace those points in a representation of the projection onto $\mathcal{C}_{\text{path}}$ by their interpolants. Their combination lies in $\mathcal{C}_{\text{grid}}$ and is within ϵ_h of that projection. Its distance from μ is at most $a + \epsilon_h$. Equation 59 then gives the upper bound $(a+\epsilon_h)^2 - a^2$; hull inclusion gives the lower bound.

Input and composed adapters in Equation 65 can have curvature because adjusted contexts pass through F. Their neural wrappers are not assumed globally twice differentiable with a known B; even a useful Lipschitz constant is not established here. Proposition 4 is a conditional approximation result, not a certified rate for these checkpoints. It also bounds approach to a continuous-path oracle, not recovery of that oracle by a fitted scorer.

D.4 Static Learned weights and aggregation estimation error

Condition on the fitted generator. Let P be the deployment distribution of a history–future pair, with finite second moments. For one weight vector shared across origins, define

$$\bar{Q}_{ij} = \mathbb{E}_P[\langle Z_i(X) - Y, Z_j(X) - Y\rangle_d], \qquad \bar{R}(w) = w^\top \bar{Q} w. \tag{67}$$

The fitting matrix $\widehat{Q}$ averages error products on aggregator-training windows. It can differ from $\bar{Q}$ through finite data, temporal dependence, or distribution shift. Our implementation optimizes mixture MSE through softmax logits; it does not explicitly need to store $\widehat{Q}$.

Proposition 5 (Estimation and Optimization Bound): Suppose $\max_{i,j} |\widehat{Q}_{ij} - \bar{Q}_{ij}| \le \eta$ and the returned $\widehat{w} \in \Delta_K$ satisfies

$$\widehat{w}^\top \widehat{Q} \widehat{w} \le \min_{w \in \Delta_K} w^\top \widehat{Q} w + \gamma, \qquad \gamma \ge 0. \tag{68}$$

Then Equation 9 holds. These are deterministic conditions on the realized fitting problem; no independence of candidate errors is required.

Proof: For every simplex vector w,

$$|w^\top(\widehat{Q} - \bar{Q})w| \le \sum_{i,j} w_i w_j \eta = \eta.$$

Let w^* minimize $w^\top \bar{Q} w$ on the compact simplex. Successively applying this inequality, Equation 68, and the same inequality again gives

$$\bar{R}(\widehat{w}) \le \widehat{w}^\top \widehat{Q} \widehat{w} + \eta \le (w^*)^\top \widehat{Q} w^* + \gamma + \eta \le \bar{R}(w^*) + 2\eta + \gamma.$$

This proves the claim. The optimization condition compares against the closed simplex, including zero weights. Finite softmax logits give positive weights, but can approximate its boundary; any remaining approximation or optimization gap belongs in γ.

D.4.1 Why a Larger Pool Need Not Improve Learned Aggregation

For nested sets $S \subset S'$ of unchanged candidate rules, define $\bar{R}^*_S = \min_{w \in \Delta_{|S|}} \bar{R}_S(w)$ and the actual excess $E_S = \bar{R}_S(\widehat{w}_S) - \bar{R}^*_S \geq 0$. Then

$$\bar{R}_S(\widehat{w}_S) - \bar{R}_{S'}(\widehat{w}_{S'}) = \underbrace{\bar{R}^*_S - \bar{R}^*_{S'}}_{\text{candidate opportunity}} - \underbrace{(E_{S'} - E_S)}_{\text{change in aggregation excess}} . \tag{69}$$

The first term is nonnegative because the new weights can put zero mass on added rules. The second can have either sign. Under Proposition 5 for S', an oracle gain exceeding $2\eta_{S'} + \gamma_{S'}$ is sufficient for improvement over the old deployed rule, since $E_S \geq 0$; this is not necessary. Candidate count may increase estimation difficulty and optimization error, while a redundant hull gives no additional opportunity.

This comparison holds for the same deployment distribution and unchanged candidate rules. For static weights it concerns a convex hull of forecast *functions*, not separate pointwise optima. Pointwise hull equality in Proposition 3 also implies equality for static mixtures of the fixed affine output path, because the effective δ is constant across histories. Refitting generators at each budget invalidates simple nesting. AdamW with a fixed number of updates, including the 160-update budget study, supplies no certificate for γ or η. Thus the bound diagnoses requirements; it does not declare every measured configuration statistically certified.

D.5 Risk scoring, temperature, and identity shrinkage

Proposition 6 (Conditional Exponential-Weight Bound): Return to the conditional setting in Appendix D.1. Let observable scores s_k satisfy $\max_k |s_k - R_k| \leq \varepsilon$. Let the temperature $T > 0$ and prior $\pi \in \Delta_K$ be observable, with every $\pi_k > 0$. Set

$$w_k = \frac{\pi_k \exp(-s_k/T)}{\sum_j \pi_j \exp(-s_j/T)}. \tag{70}$$

For any comparison weights $q \in \Delta_K$, with $0 \log 0 = 0$,

$$\begin{aligned} R(w) \leq R(q) + D(q) - D(w) + 2\varepsilon \\ + T\big[\mathrm{KL}(q\|\pi) - \mathrm{KL}(w\|\pi) - \mathrm{KL}(q\|w)\big]. \end{aligned} \tag{71}$$

Consequently,

$$R(w) \leq \min_k \{R_k + T \log(1/\pi_k)\} + 2\varepsilon - D(w). \tag{72}$$

Uniform π gives Equation 10, strengthening the version without $D(w)$.

Proof. Let $A = \sum_j \pi_j e^{-s_j/T}$. Then $\log(w_k/\pi_k) = -s_k/T - \log A$. Substituting this into the definition of KL divergence yields, for every q,

$$q^\top s + T\,\mathrm{KL}(q\|\pi) = T\,\mathrm{KL}(q\|w) - T \log A.$$

For $q = w$ the right side is $-T \log A$. Subtracting the two identities gives

$$w^\top s = q^\top s + T\big[\mathrm{KL}(q\|\pi) - \mathrm{KL}(w\|\pi) - \mathrm{KL}(q\|w)\big].$$

Writing $\mathbf{R} = (R_0, \ldots, R_{K-1})^\top$, score accuracy implies $w^\top \mathbf{R} \leq w^\top s + \varepsilon$ and $q^\top s \leq q^\top \mathbf{R} + \varepsilon$. Applying Equation 56 to both mixtures proves Equation 71. For the vertex selecting candidate k, $D(q) = 0$ and $\mathrm{KL}(q\|\pi) = \log(1/\pi_k)$. Dropping the other nonpositive KL terms and minimizing over k proves Equation 72. All steps are conditional and allow fully dependent candidates.

D.5.1 What The Bound Does and Does Not Establish

Writing $\Delta = R_0 - \min_k R_k \geq 0$ when identity is retained, Equation 10 guarantees improvement over Frozen whenever

$$\Delta + D(w) > 2\varepsilon + T \log K. \tag{73}$$

This is a sufficient analytical condition, not an observable acceptance test: Δ and a valid ε are unknown. Lower temperature reduces the entropy allowance, but does not repair inaccurate score ordering. In the hard-selection limit, the usual 2ε comparison with the best individual risk remains; hard selection can still miss a better convex combination. Taking q as an optimal mixture in Equation 71 leaves a $D(q)$ term. Scores of individual losses alone therefore do not ensure competitive optimization of mixture loss, which depends on off-diagonal error products.

Even exact scores do not imply monotonicity at fixed temperature. Let $Y = 0$, with identity 0 and n copies of forecast 1. At $T = 1$ and uniform prior, all bad candidates have exact risk one, yet their total weight is $ne^{-1}/(1 + ne^{-1})$. The ensemble risk is its square and strictly increases with n. The hull never changes. A prior assigning fixed total mass to a group of identical forecasts, divided among its copies, would remove this duplication effect; the implemented uniform-prior rule does not do so.

D.5.2 Identity Shrinkage Has an Exact Diversity Correction

For any observable $\lambda \in [0, 1]$ and $z_\lambda = (1 - \lambda)z_w + \lambda Z_0$, Proposition 1 with two candidates gives

$$R(z_\lambda) = (1 - \lambda)R(w) + \lambda R_0 - \lambda(1 - \lambda)\|z_w - Z_0\|_d^2. \tag{74}$$

Replacing $R(w)$ by any valid upper bound gives a corresponding shrinkage bound, including its negative diversity term. It does not guarantee that a validation-chosen $\lambda < 1$ improves on Frozen after a shift. Indeed, for any observed history where a rule outputs $z \neq Z_0$, a compatible future law with $Y = Z_0$ deterministically makes Frozen perfect and that rule strictly worse. With no constraint on changes in the future conditional law, universal dominance is impossible. Identity membership only makes a safe forecast *available*; it does not make an imperfect selector safe.

The Risk regressor uses a clipped, normalized log-loss target; Classifier and Pairwise ranker use classification objectives. Their outputs are not calibrated absolute MSE estimates by construction. Proposition 6 must not be applied to those scores without an additional calibration argument. For Retrospective, $s_k = r_{t,k}$ and $T = \tau \max(r_{t,0}, \epsilon)$ are observable; the algebra applies if score accuracy holds, but the use of an observed horizon alone does not establish that assumption.

D.6 Historical scores under dependence and temporal drift

The following sufficient conditions explain what would be needed to turn historical losses into a score-accuracy bound. They are not asserted for the reported data. Fix K candidate rules before an audit sequence. Let $\mathcal{F}_0 \subset \cdots \subset \mathcal{F}_n \subseteq \mathcal{H}_t$ be a filtration in which the prediction for block i is available in $\mathcal{F}_{i-1}$ and its fully resolved loss L_{ik} is measurable in $\mathcal{F}_i$. Completed, non-overlapping forecast blocks with intervening history included in the filtration are one admissible construction. Overlapping target blocks cannot simply be treated as independent observations.

Assume $0 \leq L_{ik} \leq M$ almost surely for a known constant M, and use deterministic nonnegative weights a_i summing to one. Define

$$\begin{aligned} p_{ik} &= \mathbb{E}[L_{ik} \mid \mathcal{F}_{i-1}], & s_k &= \sum_i a_i L_{ik}, \\ \beta_t &= \max_k \left| \sum_i a_i p_{ik} - R_k \right|, & n_{\text{eff}} &= \left(\sum_i a_i^2 \right)^{-1}. \end{aligned} \tag{75}$$

Here, R_k is current conditional risk, whereas p_{ik} is the earlier predictable risk. Their difference includes changing regimes and histories, even for a stationary process with state-dependent predictability.

Proposition 7 (Concentration Plus Drift): For any $0 < \alpha < 1$, with probability at least $1 - \alpha$ over the observed sequence,

$$\max_k |s_k - R_k| \leq \beta_t + M\sqrt{\frac{\log(2K/\alpha)}{2n_{\text{eff}}}}. \tag{76}$$

No independence across candidates or across blocks is assumed. The probability statement is over histories, not a post-observation guarantee conditional on each realized history.

Proof: For each k, $L_{ik} - p_{ik}$ is a martingale difference. Given $\mathcal{F}_{i-1}$ it lies in an interval of length M. To verify the needed exponential bound, let U be any conditionally centered variable in an interval of length M. Its conditional log moment generating function $\psi(\xi) = \log \mathbb{E}[e^{\xi U} \mid \mathcal{F}_{i-1}]$ satisfies $\psi(0) = \psi'(0) = 0$. Its second derivative is the variance under exponential tilting. Tilting preserves the interval, so that variance is at most $M^2/4$: squared distance to the interval midpoint is at most $M^2/4$, and variance is no larger than the expected squared distance to any fixed point. Integrating the derivative bound twice gives $\psi(\xi) \le \xi^2 M^2/8$ for either sign of ξ. Applied to the scaled increment, this conditional Hoeffding argument (Azuma, 1967; Cesa-Bianchi et al., 2006) yields

$$\mathbb{E}[\exp\{\xi a_i(L_{ik} - p_{ik})\} \mid \mathcal{F}_{i-1}] \le \exp(\xi^2 a_i^2 M^2/8).$$

Iterating conditional expectation across i bounds the moment generating function of $S_k = \sum_i a_i(L_{ik} - p_{ik})$ by $\exp(\xi^2 M^2 \sum_i a_i^2/8)$. Markov's inequality and minimization over $\xi > 0$, attained at $4u/(M^2 \sum_i a_i^2)$, give

$$\Pr(S_k \ge u) \le \exp\left(-\frac{2u^2}{M^2 \sum_i a_i^2}\right).$$

Applying the same argument to $-S_k$ and a union bound over K rules bounds $\Pr(\max_k |S_k| \ge u)$ by $2K \exp(-2n_{\text{eff}} u^2/M^2)$. Choose $u = M\sqrt{\log(2K/\alpha)/(2n_{\text{eff}})}$, then use $|s_k - R_k| \le |S_k| + |\sum_i a_i p_{ik} - R_k|$. This proves Equation 76; $M = 0$ is immediate without division.

Implications and Limits: For equal weights $n_{\text{eff}} = n$. This is a concentration parameter for martingale innovations around their changing predictable means, not a claim that dependent blocks behave as n independent samples of current risk. The drift term can remain large as n increases. If an external assumption bounds $\beta_t \le b$, substituting $\varepsilon = b + M\sqrt{\log(2K/\alpha)/(2n_{\text{eff}})}$ into Proposition 6 yields a high-probability current-risk comparison. A high-probability drift bound requires its own failure probability, added by a union bound. Without drift control, a longer scoring history offers no such certification.

The implemented retrospective rule uses one completed horizon, so this construction has $n = 1$; the HC entries of that horizon are not automatically independent audit samples. Standardized squared error is also not known to be bounded by a universal M. Clipping audit losses would change the risk being estimated; a claim about unclipped MSE would additionally require a tail bound. Data-dependent weighting, choosing rules after seeing audit labels, and overlapping forecasts require separate arguments. Thus Proposition 7 explains why score reliability can fail under shift without attributing an unverified statistical guarantee to the experiments.

D.7 Compute budgets, oracle definitions, and experimental scope

D.7.1 A Decision Criterion that Includes Cost

On a fixed deployment distribution, let $C(S)$ denote the expected measured cost of generating and scoring set S, and let $\nu \ge 0$ convert that cost to the units of risk. Define $J(S) = \bar{R}_S(\widehat{w}_S) + \nu C(S)$. For nested unchanged rules, Equation 69 gives exactly

$$J(S) - J(S') = (\bar{R}^*_S - \bar{R}^*_{S'}) - (E_{S'} - E_S) - \nu\{C(S') - C(S)\}. \tag{77}$$

Increasing computation is beneficial for this objective if and only if the right side is positive. This is a decomposition, not an algorithm that knows future risk. Output-only candidates share $F(X)$; input and composed paths require adjusted-context forecasts. Retrieval changes both the number of calls and context length. Historical scoring may need another complete pool. Consequently, K is not a substitute for $C(S)$, and duplicated outputs can raise scoring cost without enlarging the forecast hull. Fitting a separate aggregator or adapter for each K also incurs offline costs beyond the inference budget.

D.7.2 The Reported Oracle is a Different Comparator

The unavailable test-label selector chooses k after observing Y. Its conditional expected loss satisfies

$$\mathbb{E}[\min_k \|Z_k - Y\|_d^2 \mid \mathcal{H}_t] \le \min_k \mathbb{E}[\|Z_k - Y\|_d^2 \mid \mathcal{H}_t] = \min_k R_k, \tag{78}$$

because the minimum realized loss is at most each realized loss before taking expectations. It is not ordered in general relative to the conditional convex optimum. With $Y = 0$ and candidates $-1, 1$, hindsight hard selection loses one whereas their mean loses zero. With $Y \in \{-1, 1\}$ equiprobable and the same candidates, hindsight selection loses zero whereas the best observable convex forecast is zero with risk one. An observed Oracle improvement therefore measures hindsight candidate opportunity; it neither proves a deployable gain nor estimates the convex optimum in Proposition 2.

D.7.3 Reading the Experimental Evidence

Uniform has an exact risk identity but no general monotonicity guarantee. Learned weights and Greedy ensemble depend on mixture estimation and optimization; Greedy's restricted search can add approximation error. Classifier and Pairwise ranker aim at selection and need generalization and ranking accuracy, rather than automatically satisfying Proposition 6. Risk regression and Retrospective require additional score calibration to invoke that proposition. Validation-best candidate and Validation-Selection further depend on validation-to-test transfer. None gains a guarantee merely by including the frozen forecast.

The measured budget and adapter sweeps can test whether a particular fitted configuration improves, but cannot establish the conditional means, drift bounds, or uniform estimation guarantees assumed above. Seed variation, correlated datasets, and finite test origins further separate observed aggregate metrics from these population statements. The appropriate conclusion is conditional: additional candidates help when they extend useful forecast directions, their benefit survives aggregation error, and that recovered benefit justifies actual computation. More samples by themselves satisfy none of these three requirements.

E ALGORITHM

Algorithm 1 specifies the complete workflow. Its generator includes every family, and its weight procedure covers every deployable rule. Algorithms 2 expand these calls.

Algorithm 1 TTC4TS: offline fitting, validation selection, and test-time inference

Require: Pretrained F; pre-test series; context L, horizon H; observed test history
Ensure: Fitted adapters and aggregators, selected rule m^*, and forecasts $\widehat{Y}_t$

Offline fitting: freeze F; form purged chronological blocks D_A, D_S, D_V
Fit normalization on D_A; standardize data; build reference bank $\mathcal{R}$ from D_A
Initialize input/output residuals R_ϕ, R_ψ; set $\Delta = \mathrm{linspace}(0, 0.3, 20)$
for each adapter update on a minibatch $(X, Y) \subset D_A$ **do**
    Construct $Z_j^{\mathrm{in}}, Z_j^{\mathrm{out}}, Z_j^{\mathrm{joint}}$ by Equation 2, for all $\delta_j \in \Delta$
    Update only (ϕ, ψ) using $\frac{1}{3|\Delta|} \sum_j \sum_{v \in \{\mathrm{in,out,joint}\}} \mathrm{MSE}(Z_j^v, Y)$
end for
Freeze adapters; fix the ordered pool: identity, adapters, augmentations, Retrieval-demo
for each origin u in $D_S \cup D_V$ **do**
    $Z_u \leftarrow \textsc{Generate}(X_u, u)$; $\widetilde{Z}_u \leftarrow \textsc{Generate}(X_{u-H}, u-H)$ ▷ Alg. 2
    $r_{u,k} \leftarrow \mathrm{MSE}(\widetilde{Z}_{u,k}, x_{u-H:u})$; $f_u \leftarrow \mathrm{Features}(X_u, Z_u, r_u)$
end for
On D_S, fit learned weights, Classifier, Risk, ranker and Greedy (Appendix C.2)
Validation: choose Retrospective (τ, λ) and best fixed candidate k^* on D_V
Select m^* by mean validation MSE over the nine deployment options; exclude Oracle
Freeze all fitted parameters, feature statistics and validation choices
for each test origin t with only observations through $t-1$ available **do**
    $Z_t \leftarrow \textsc{Generate}(X_t, t)$; $r_t, f_t \leftarrow \emptyset$ ▷ Standardized history
    if $m^* \in$ {Risk, ranker, Retrospective} **then**
        $\widetilde{Z}_t \leftarrow \textsc{Generate}(X_{t-H}, t-H)$
        $r_{t,k} \leftarrow \mathrm{MSE}(\widetilde{Z}_{t,k}, x_{t-H:t})$; $f_t \leftarrow \mathrm{Features}(X_t, Z_t, r_t)$
    end if
    $w_t \leftarrow \textsc{Weights}(m^*, Z_t, f_t, r_t)$ ▷ no future labels
    Output $\widehat{Y}_t = \sum_k w_{t,k} Z_{t,k}$ for this origin; no parameter update
end for

E.1 COMPLETE PROCEDURES AND CODE MAPPING

Algorithm 1 covers the complete training and deployment protocol; the two procedures below expand its generator and aggregator. D_A, D_S, and D_V denote the adapter-fitting, aggregator-fitting, and validation blocks, respectively. $X_u = x_{u-L:u}$ and $x_{u-H:u}$ use half-open time intervals. All tensors below use the fixed normalization fitted on D_A. Returned forecasts are in those coordinates; the inverse scaler is applied when original-unit predictions are requested. $M = |\Delta| = 20$ is the number of samples per adapter view, whereas $K = 80$ is the total candidate count and $Q = 10$ is the retrieval count.

The output-coordinate map h_a undoes magnitude reflection, vertical translation, magnitude warping, or the future continuation of an added slope. It is the identity for perturbations without a justified forecast-space inverse, including time reversal. The $\delta = 0$ identity is included only once. The ordered pool has $1 + 3(M-1) + 12 + Q = 80$ candidates; retrieval contributes only independently generated demonstration forecasts. No direct retrieved future or cumulative retrieval average is a candidate. The separate input/joint calls in Algorithm 2 preserve the measured implementation's 61 wrapper calls per current-origin pool, even though reusing them could reduce cost.

Transformation details. Magnitude warping applies a positive sinusoidal scale to the mean-centered history and divides the forecast's centered values by its continuation through the horizon. Added slopes are removed using the corresponding future slope. Time reversal, smoothing, time warping, horizontal shifts, jitter and spikes perturb the conditioning history; they do not define invertible transformations of future dynamics. In particular, the predicted future is not reversed for time-flipped inputs. Stateless jitter depends only on the observed window and fixed seed. Historical

Algorithm 2 GENERATE: all TTC4TS candidate families

Require: Frozen F, R_ϕ, R_ψ; fixed Δ, 12 augmentations $\mathcal{A}$, bank $\mathcal{R}$; $Q = 10$

function GENERATE(X, u)
  $B \leftarrow F(X)$; $\mathcal{I}, \mathcal{O}, \mathcal{J}, \mathcal{U}, \mathcal{D} \leftarrow$ empty ordered lists
  for each $\delta \in \Delta \setminus \{0\}$ in increasing order **do**
    Append $F(X + \delta R_\phi(X))$ to $\mathcal{I}$ ▷ Input adapter
    Append $B + \delta R_\psi(B)$ to $\mathcal{O}$ ▷ Output adapter
    $P \leftarrow F(X + \delta R_\phi(X))$ ▷ Separate call in the measured code
    Append $P + \delta R_\psi(P)$ to $\mathcal{J}$ ▷ Composed adapter
  end for
  for each augmentation $a \in \mathcal{A}$ in fixed order **do**
    $(X^a, h_a) \leftarrow \mathrm{Transform}(a, X, H, \mathrm{seed})$
    Append $h_a(F(X^a))$ to $\mathcal{U}$
  end for
  Mask bank pairs whose observed continuation ends after u
  For each channel, retrieve Q nearest standardized 64-step pasts to $X_{L-64:L}$
  Assemble Q multichannel pairs (P_q, V_q) from these channel-wise neighbors
  for $q = 1, \ldots, Q$ **do**
    $(\bar{P}_q, \bar{V}_q) \leftarrow \mathrm{Align}(P_q, V_q; X_{L-64:L})$ ▷ Phase and amplitude
    $C_q \leftarrow [\bar{P}_q; \bar{V}_q; X]$ ▷ Aligned past, observed continuation, query
    Check $\mathrm{length}(C_q) \le F$'s context capacity; otherwise raise an error
    Append $F(C_q)$ to $\mathcal{D}$ ▷ One forecast per demonstration
  end for
  $Z \leftarrow \mathrm{stack}([B] + \mathcal{I} + \mathcal{O} + \mathcal{J} + \mathcal{U} + \mathcal{D})$
  Check finite values and common shape $H \times C$; **return** Z
end function

forecasts use the same transformations and timestamp masks as current forecasts; on a horizon-spaced stream, they may be cached once their targets become observed.

E.2 FITTING OBJECTIVES AND VALIDATION CHOICES

For each origin u in D_S, form candidate errors $\ell_{u,k} = \mathrm{MSE}(Z_{u,k}, Y_u)$ and historical errors $r_{u,k} = \mathrm{MSE}(Z_{u-H,k}, x_{u-H:u})$. The latter use only an already observed horizon. Fit the following components on D_S only; $\mathbb{E}_S$ denotes the empirical mean over its fitting origins:

$$b^* = \arg\min_b \ \mathbb{E}_S \,\mathrm{MSE}\left(\sum_k \mathrm{softmax}(b)_k Z_{u,k}, Y_u\right), \tag{79}$$

$$\theta_C^* = \arg\min_{\theta_C} \ \mathbb{E}_S \,\mathrm{CE}\left(c_{\theta_C}(Z_u), \arg\min_k \ell_{u,k}\right), \tag{80}$$

$$\theta_R^* = \arg\min_{\theta_R} \ \mathbb{E}_{S,k} \,\mathrm{smoothL}_1(s_{\theta_R}(f_{u,k}), \rho_{u,k}), \tag{81}$$

$$\rho_{u,k} = \log\left(1 + \min\left(100, \frac{\ell_{u,k}}{\max(\ell_{u,0}, 10^{-5})}\right)\right),$$

$$\theta_P^* = \arg\min_{\theta_P} \ \mathbb{E}_{S,i,j:\ell_{u,i} \neq \ell_{u,j}} \,\mathrm{softplus}[(s_{\theta_P}(f_{u,i}) - s_{\theta_P}(f_{u,j}))\,\mathrm{sign}(\ell_{u,i} - \ell_{u,j})]. \tag{82}$$

These are numerical optimization objectives, not claims that a global optimum is attained. c is the supplied trajectory Classifier, applied independently to flattened HC-dimensional candidates; s is the feature MLP plus candidate bias. Feature means/scales are fitted on D_S. All four objectives use 160 full-batch AdamW updates at learning rate 0.01 and weight decay 0.03 in the main comparison. Adapter fitting in Algorithm 1 instead uses 32 Adam minibatch updates, differentiating through F with its parameters frozen. The zero-Delta loss is constant and contributes no parameter gradient.

Ensemble learning starts with $\bar{Z}_u^{(0)} = 0$ and zero counts. At step $n = 1, \ldots, 40$, select

$$k_n = \arg\min_k \mathbb{E}_S \,\mathrm{MSE}\left(\frac{(n-1)\bar{Z}_u^{(n-1)} + Z_{u,k}}{n}, Y_u\right),$$
$$\bar{Z}_u^{(n)} = \frac{(n-1)\bar{Z}_u^{(n-1)} + Z_{u,k_n}}{n}. \tag{83}$$

Its fixed weight g_k is the fraction of the 40 selections assigned to candidate k. Selection is with replacement; ties use the first candidate in the fixed order. Uniform needs no fitting. Validation-best uses $k^* = \arg\min_k \mathbb{E}_V \ell_{u,k}$. Retrospective selects (τ, λ) by mixture MSE on D_V over $\{0.1, 0.3, 1\} \times \{0, 0.5, 1\}$. Validation-selected TTC4TS selects one rule from the nine options, in that order for exact ties. Oracle is an evaluation-only diagnostic, $Z_{t,\arg\min_k \mathrm{MSE}(Z_{t,k}, Y_t)}$, and is never available to that choice or deployment.

F Experimental Setup and Results

F.1 Implementation Details

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 from two electricity transformers in two separated counties in China, including the load and the oil temperature recorded every 15 minutes (ETTm) or 1 hour (ETTh) between July 2016 and July 2018. (2) Electricity (ECL) dataset [1] collects the hourly electricity consumption of 321 clients (each column) from 2012 to 2014. (3) Exchange Lai et al. (2018) records the current exchange of 8 different countries from 1990 to 2016. (4) Traffic dataset [2] records the occupation rate of freeway system across State of California measured by 861 sensors. (5) Weather dataset [3] records every 10 minutes for 21 meteorological indicators in Germany throughout 2020. Appliances, Beijing PM2.5 dataset provides by Candanedo (2017) and Chen (2015). The detailed statistics information of the datasets is shown in Table 5.

ETT uses the standard chronological 12/4/4-month boundaries. The other datasets use 70/10/20 chronological splits. The first 60% of the nominal training period fits adapters, normalization, and the retrieval bank; the rest fits aggregators after a one-horizon purge. The middle development block selects hyperparameters and the deployment rule. Test origins are spaced by H, for $H \in \{96, 192, 336, 720\}$, and all channels are retained. We report MSE and MAE in fitting-standardized units. These choices differ from stride-one LTSF evaluation and from fitting a scaler on the complete nominal training block; only comparisons within the same protocol are directly interpretable.

Table 5: Summary statistics of the datasets, including sequence length, number of variables, and sampling frequency.

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

The context length is 512, with at most 96 fitting windows per component. Adapters receive 32 Adam updates at learning rate 10^{-4}, averaging the three adapter views and all 20 Delta values; batch size is one for horizon 720 or checkpoints named base, otherwise two. Adapter and Classifier hidden widths are 512; maximum Delta is 0.3 and retrieval uses top-$K = 10$. Aggregators receive 160 updates at learning rate 0.01 and weight decay 0.03; budget-specific fits use 40 updates. Seeds 0–2 vary initialization and adapter minibatches. The 80-candidate pool contains identity, 57 adapter candidates, 12 augmentations, and 10 retrieval demonstrations. Saved predictions hold targets and alternatives fixed across rules, and parameter hashes verify the frozen backbone. PyTorch experiments use eight 32GB V100 GPUs, one task per device. Appendix Table 10 reports synchronized cold-query latency including retrospective forecasts.

[1] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[2] http://pems.dot.ca.gov

[3] https://www.bgc-jena.mpg.de/wetter

F.2 Reproduction details

With half-open index intervals, adapter origins satisfy $t \geq L$ and $t + H \leq e_{\text{adapt}}$. Aggregator origins satisfy $t \geq e_{\text{adapt}} + H$ and $t + H \leq e_{\text{train}}$; validation uses $t \geq e_{\text{train}}$ and $t + H \leq e_{\text{val}}$; testing uses $t \geq e_{\text{val}}$ and $t + H \leq e_{\text{test}}$. Retrospective targets are $[t - H, t)$, so none used to train an aggregator lies before the adapter boundary. Audits verify these inequalities; earlier test observations may legitimately enter a later test origin's history.

The supplied PyTorch implementation stores per-origin targets, all candidate forecasts, learned adapter and aggregator states, fitted feature statistics, exact split boundaries, dataset hashes, frozen-backbone hashes, and software versions. Every result table is reconstructed from these files. Model parameters are checked before and after adapter fitting. Test labels enter only the metric computation and the explicitly marked oracle diagnostic. Training retrieval excludes reference pairs ending after the relevant origin. Unit tests cover these information boundaries, degenerate normalization gradients, forecast-space inverses, and the shared-residual adapter forward and joint-gradient path.

All six checkpoints use controlled point decoders. Bolt emits median predictions in 64-step blocks, TimesFM in 128-step blocks, and Timer uses its point head recursively. Chronos-2 predicts channel-independent median quantiles directly. Sundial shares one transformer representation across eight fixed Gaussian noise trajectories and applies 50 flow-decoder steps to each trajectory before averaging. A recorded backbone call means one wrapper prediction, not one internal transformer or flow-network pass; latency includes all internal computation.

G Detailed Experimental Results

G.1 Comparison with SOTA Methods

This comparison fixes the TTC4TS synthesizer to **learned weights**: one global simplex vector per model–dataset–horizon–seed cell, fitted on the aggregator-training block.

The lower block of Table 1 keeps the existing shared TTC4TS adapters: one input, output, or composed prediction at $\delta = 0.3$, a validation-best adapter candidate including identity, and learned weights over identity plus each generator family. These are component ablations rather than independently trained prior methods. Full Uniform and aligned-demo averaging with identity are retained to distinguish changing the generator pool from changing its weights.

G.2 Measured comparison

Component controls further limit the claim that the complete generator is necessary. Learned weights over identity and augmentations alone reach relative MSE 0.935, slightly below the full pool's 0.936; adapter-only learned weights reach 0.985. Full TTC4TS has better relative MAE than the augmentation-only control (0.988 versus 0.990), but it cannot claim uniformly better error or efficiency. All three generator families provide an interface for experimentation; this evaluation does not establish that retaining every family is optimal.

The 432-cell audit recomputes 7,344 test metric records and their validation counterparts from saved forecasts, verifies frozen model and source hashes, and checks every validation-selected placement and checkpoint. One preliminary TimesFM identity check required reproducing the original cache batch to remove batch-layout rounding differences; its prediction then matched exactly. The correction changed only verification, and both source versions and the initial failure record are retained. All final cell statuses are successful. Seed standard deviations summarize repeated fitting, not independent datasets; deterministic retrieval variants reuse verified seed-zero predictions and accordingly have zero seed dispersion.

Table 6: Geometric relative MSE by model and dataset on 6 datasets, aggregating horizons 96/192/336/720 and seeds 0–2. Each ratio uses the matching Frozen result.

Dataset	AdaPTS	δ-Adp+In	δ-Apt+Out	TS-RAG	Align-RAG	TTC4TS+U	TTC4TS+RR	TTC4TS+LW
Bolt-S								
ETTh1	0.985	0.998	0.986	0.996	0.996	0.996	0.995	**0.906**
ETTh2	0.996	1.011	1.010	0.997	**0.983**	1.000	0.995	0.988
ETTm1	0.970	0.990	0.965	0.976	1.022	0.996	0.958	**0.888**
ETTm2	1.003	1.012	1.024	1.004	1.016	0.973	**0.972**	0.980
Appliances	1.025	0.991	1.023	0.990	1.045	1.071	0.992	**0.969**
BeijingWeather	0.920	0.974	0.880	0.931	**0.861**	0.902	0.903	0.892
Bolt-B								
ETTh1	0.967	1.000	0.969	0.992	0.996	0.980	0.981	**0.889**
ETTh2	1.004	1.013	1.015	1.004	**0.931**	0.950	0.973	0.970
ETTm1	0.970	0.999	0.970	0.970	1.001	0.981	0.975	**0.893**
ETTm2	0.982	1.015	1.000	0.982	1.039	**0.960**	0.970	0.963
Appliances	1.003	0.999	1.028	0.988	1.039	1.018	**0.982**	0.987
BeijingWeather	0.977	0.981	0.983	0.968	0.878	0.880	**0.835**	0.913
TimesFM								
ETTh1	0.996	1.012	0.999	1.006	0.975	0.989	0.985	**0.938**
ETTh2	1.011	1.012	1.039	1.021	1.058	**0.927**	0.947	0.974
ETTm1	0.995	1.009	1.005	1.013	1.000	0.962	0.930	**0.897**
ETTm2	1.002	1.016	1.010	0.997	1.085	0.970	0.969	**0.931**
Appliances	1.011	0.999	1.011	1.007	0.970	0.925	0.921	**0.918**
BeijingWeather	0.743	0.981	0.881	0.981	0.557	**0.551**	**0.551**	0.861
Timer								
ETTh1	1.001	1.018	1.017	1.011	0.985	**0.971**	0.982	0.972
ETTh2	1.014	1.017	1.031	1.016	1.060	**0.974**	0.976	0.976
ETTm1	0.995	0.998	1.000	0.999	1.246	1.269	1.018	**0.978**
ETTm2	1.005	1.015	1.026	1.001	1.165	**0.956**	0.964	0.964
Appliances	0.956	1.009	0.966	0.961	1.065	0.994	0.983	**0.954**
BeijingWeather	0.998	1.009	1.011	1.009	**0.964**	0.997	1.017	0.972
Sundial								
ETTh1	1.005	1.012	1.016	1.008	0.984	0.994	0.990	**0.978**
ETTh2	0.996	1.011	1.006	1.000	1.007	**0.968**	0.970	0.972
ETTm1	1.015	1.007	1.025	1.007	0.993	1.000	0.978	**0.963**
ETTm2	0.999	1.013	1.013	1.005	1.125	0.970	0.969	**0.963**
Appliances	1.029	1.000	1.034	1.025	0.998	0.941	**0.937**	0.940
BeijingWeather	0.986	0.992	0.986	0.981	0.956	0.985	0.986	**0.955**
Chronos-2								
ETTh1	0.990	1.001	0.994	0.994	0.943	0.962	0.965	**0.883**
ETTh2	0.992	1.011	1.009	0.994	0.993	**0.936**	0.951	0.950
ETTm1	1.023	0.990	1.026	1.013	0.913	0.917	0.917	**0.885**
ETTm2	1.003	1.013	1.028	1.002	1.004	0.972	0.967	**0.960**
Appliances	0.958	0.984	0.961	0.984	0.937	0.927	0.913	**0.898**
BeijingWeather	0.847	0.979	0.798	0.854	0.791	0.839	**0.788**	0.803

Table 7: Geometric relative MAE by model and dataset on 6 datasets, aggregating horizons 96/192/336/720 and seeds 0–2. Each ratio uses the matching Frozen result.

Dataset	AdaPTS	δ-Adp+In	δ-Apt+Out	TS-RAG	Align-RAG	TTC4TS+U	TTC4TS+RR	TTC4TS+LW
Bolt-S								
ETTh1	1.004	1.026	1.030	1.023	1.002	0.998	0.997	**0.982**
ETTh2	1.007	1.027	1.034	1.009	1.013	1.003	**0.998**	1.000
ETTm1	1.005	1.036	1.036	1.022	1.056	1.048	0.990	**0.978**
ETTm2	1.016	1.037	1.052	1.020	1.021	0.990	**0.984**	0.996
Appliances	1.035	1.014	1.039	1.015	1.018	1.029	**0.997**	1.009
BeijingWeather	1.040	1.128	1.126	1.092	**1.007**	1.009	1.010	**1.007**
Bolt-B								
ETTh1	0.987	1.026	1.014	1.020	0.998	0.986	0.989	**0.976**
ETTh2	1.012	1.030	1.035	1.014	0.970	**0.963**	0.983	0.983
ETTm1	0.996	1.040	1.034	1.005	1.020	1.004	0.990	**0.980**
ETTm2	1.002	1.039	1.040	1.006	1.031	**0.977**	0.985	0.990
Appliances	1.024	1.018	1.039	1.017	1.010	1.004	**0.991**	1.021
BeijingWeather	1.058	1.126	1.167	1.084	1.011	1.009	**0.998**	1.010
TimesFM								
ETTh1	1.010	1.036	1.034	1.032	1.000	1.005	1.002	**0.988**
ETTh2	1.006	1.029	1.039	1.017	1.043	**0.962**	0.966	0.984
ETTm1	1.000	1.038	1.035	1.018	1.019	1.003	0.977	**0.964**
ETTm2	1.000	1.038	1.033	0.999	1.061	0.985	0.984	**0.968**
Appliances	1.022	1.018	1.035	1.022	0.990	0.962	**0.961**	0.969
BeijingWeather	1.027	1.133	1.191	1.133	0.907	0.900	**0.899**	1.026
Timer								
ETTh1	1.002	1.033	1.033	1.017	1.012	**0.986**	0.990	0.993
ETTh2	1.007	1.027	1.032	1.012	1.068	0.985	0.985	**0.984**
ETTm1	0.991	1.020	1.011	1.008	1.117	1.118	1.012	**0.990**
ETTm2	1.003	1.024	1.028	1.003	1.143	**0.987**	0.991	0.988
Appliances	0.989	1.012	1.000	0.994	1.024	0.986	**0.985**	0.993
BeijingWeather	1.000	1.081	1.073	1.039	0.995	0.994	0.994	**0.982**
Sundial								
ETTh1	1.009	1.022	1.032	1.018	1.014	1.009	1.000	**0.997**
ETTh2	1.002	1.024	1.025	1.012	1.053	**0.982**	0.984	0.987
ETTm1	1.016	1.029	1.044	1.022	1.030	1.010	0.993	**0.989**
ETTm2	1.007	1.030	1.034	1.013	1.123	0.982	0.982	**0.979**
Appliances	1.028	1.010	1.035	1.027	0.993	**0.964**	0.967	0.974
BeijingWeather	1.026	1.069	1.077	1.055	0.990	0.985	0.984	**0.966**
Chronos-2								
ETTh1	1.007	1.030	1.036	1.021	0.986	0.987	0.987	**0.975**
ETTh2	1.010	1.033	1.041	1.017	1.009	**0.977**	0.980	0.985
ETTm1	1.015	1.039	1.054	1.026	1.003	0.993	**0.977**	0.980
ETTm2	1.018	1.045	1.061	1.018	1.021	0.989	**0.986**	0.992
Appliances	1.029	1.015	1.036	1.015	0.981	0.977	**0.969**	0.979
BeijingWeather	1.053	1.145	1.132	1.121	**0.947**	0.971	0.956	1.003

G.3 Experiments on More Datasets

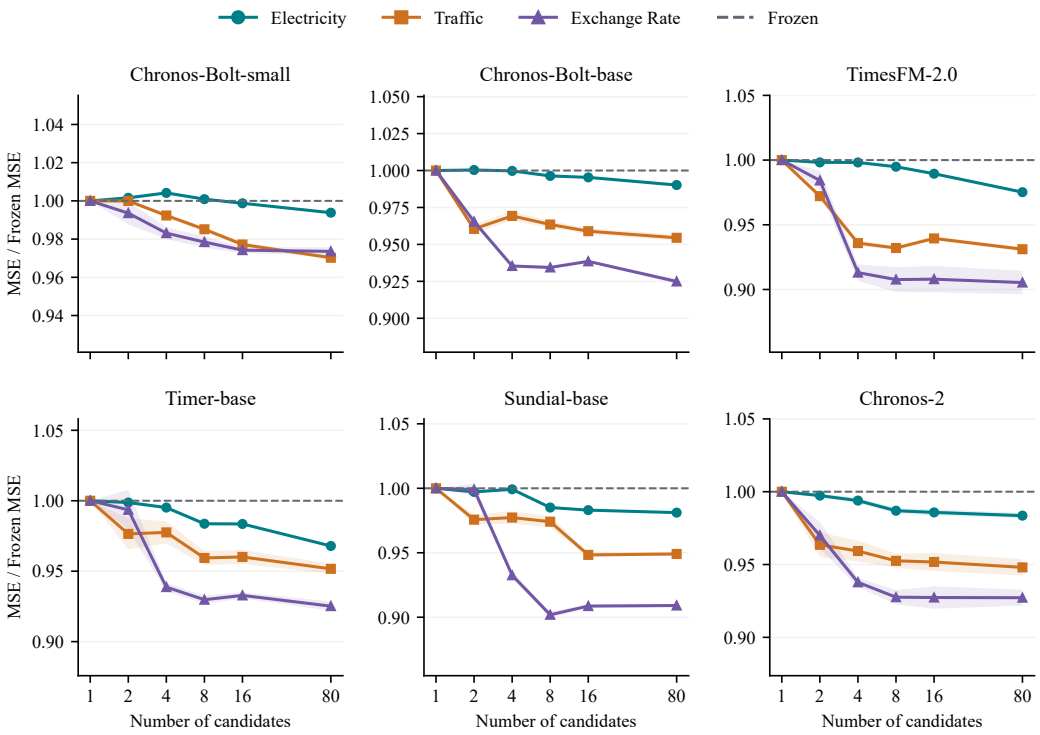

Figure 6: Candidate-budget curves on other 3 common datasets.

Figure 6 illustrates the impact of the number of candidates on forecasting performance across six pretrained forecasting backbones and three datasets. As the number of candidates increases, most models consistently achieve lower MSE ratios, demonstrating the effectiveness of candidate selection in improving forecasting accuracy. Overall, the results indicate that leveraging more candidate forecasts can effectively enhance the performance of pretrained time-series models.

G.4 Domain sensitivity and deployment cost

Giving equal weight to the ETT family, Appliances, and BeijingWeather yields TTC4TS relative MSE 0.919, compared with 0.940 when each dataset receives equal weight. The domain table shows that improvement is largest on BeijingWeather; the overall Classifier failure also hides a gain on that domain. The three domain groups are too few for a broad generalization claim.

Table 8: Domain sensitivity. Within-domain relative MSE averages checkpoints, horizons and seeds geometrically. Lower is better. VS denotes Validation-Selection. VS denotes Validation-Selection.

Domain	Classifier	Risk	Val.-best	VS
Appliances	1.419	0.971	1.025	**0.957**
BeijingWeather	0.863	0.890	0.868	**0.842**
ETT family	1.294	0.976	0.989	**0.962**

Timing uses batch-one float32 inference, one warmup and eight timed repeats with CUDA synchronization; median and 90th-percentile latency and peak allocated GPU memory are saved. Loading, input transfer, and service overhead are excluded. The measured implementation makes 61 wrapper calls for a full current-origin pool and 122 when retrospective features are needed, versus one for

Frozen. Input and joint views recompute the same input-adapted backbone forecasts separately, and sharing them is a possible implementation optimization but is not credited in these timings.

G.5 Measured computation

Table 9: Median synchronized inference latency (milliseconds) on ETTh1 at horizon 96, seed 0, first test origin, batch one, one warmup and three repeats on a V100. Point decoders and context length 512 are fixed; loading and transfers are excluded. Learned weights do not require retrospective forecasts. Timings are measured in the concurrent experiment environment and are not isolated-service throughput benchmarks.

Backbone	Frozen	Input	Output	Joint	Raw demo	Align demo	TTC4TS-LW
Bolt-S	42.9	**40.7**	43.0	41.5	415.2	461.6	2594.7
Bolt-B	**73.5**	75.8	74.7	75.4	759.6	839.4	4618.1
TimesFM	50.7	52.8	51.1	**50.3**	512.6	564.8	3233.6
Timer	**9.1**	9.7	9.8	10.5	91.0	110.2	633.7
Sundial	118.5	**104.8**	105.1	105.2	1465.9	1164.5	7069.2
Chronos-2	35.6	**32.9**	33.1	33.6	345.4	387.1	2235.3

Note. Best: bold; second-best: double underline. Ties use displayed precision.

Table 10: Median cold-query inference latency in milliseconds on ETTh1 at horizon 96 and seed 0, measured on one V100 with all channels and a batch of one origin. The 80-candidate pool includes top-10 retrieval demonstrations. Risk includes retrospective forecasts; Classifier scores trajectories directly. Eight timed repeats follow a warmup; data and models reside on the GPU. VS denotes Validation-Selection.

Backbone	Frozen	Uniform	Classifier	Risk	VS
Bolt-S	**39.9**	2516.6	2401.2	4919.6	2502.3
Bolt-B	**68.5**	4298.8	4375.0	8342.6	4060.2
TimesFM	**44.8**	3038.4	2717.2	5733.2	2920.4
Timer	**8.6**	598.8	591.3	1186.2	1217.8
Sundial	**98.4**	7819.9	6047.5	12397.8	6056.9
Chronos-2	**31.5**	1989.3	2094.1	3976.2	1958.7

The median synchronized inference latency and the median cold-query inference latency are shown in Tables 9 and 10. It can seen that single-forecast adapters require one backbone prediction, plus their small residual network(s). Ten separate unaligned demonstrations require ten predictions. The current aligned generator also evaluates identity, for eleven wrapper calls. Full TTC4TS evaluates 61 wrapper calls: shared input and composed adapter paths still recompute their input-adjusted forecasts separately. Learned weights require no retrospective candidate pool. These implementation costs must accompany the accuracy comparison, and more candidates are not evidence of matched-latency superiority.

Offline fitting is also measured separately. Across 432 cells per placement, median standalone fitting times are 38.5 seconds for input adapters, 11.7 seconds for output adapters, and 38.0 seconds for composed adapters. Output-only fitting caches frozen training forecasts and avoids backbone backpropagation. These medians summarize heterogeneous cells in the concurrent experiment environment; they are not a matched-training-cost comparison with TTC4TS.

H Detailed Experimental Results of Generator

Tables 11–17 disaggregate Figures 4 and 5. Entries average per-cell MSE ratios to Frozen geometrically over $H \in \{96, 192, 336, 720\}$ and seeds 0–2. Re-aggregation reproduces every plotted value; these ratios are not absolute errors.

Table 11: Detailed experiment results of Figure 4 by model and dataset: ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather. Relative MSE is averaged geometrically over four horizons 96, 192,336, and 720 and three seeds. A denotes adapters, U augmentations, and R top-10 Retrieval-demo. Every family pool includes the identity once and uses Uniform. Frozen is 1.000, and lower is better.

Dataset	Frozen	A	U	R	A+R	A+U	A+U+R	U+R
Chronos-Bolt-small								
ETTh1	1.000	0.997	0.970	0.972	0.995	**0.916**	0.934	0.995
ETTh2	1.000	1.002	0.994	0.993	0.999	**0.972**	0.978	0.999
ETTm1	1.000	0.991	0.953	0.943	0.970	**0.911**	0.922	0.982
ETTm2	1.000	1.001	0.992	0.980	0.986	0.971	**0.951**	0.969
Appliances	1.000	1.010	1.000	**0.994**	1.001	0.998	1.002	1.053
BeijingWeather	1.000	0.979	0.966	0.954	0.962	0.930	0.910	**0.907**
Chronos-Bolt-base								
ETTh1	1.000	0.993	0.963	0.964	0.989	**0.899**	0.921	0.980
ETTh2	1.000	1.001	0.991	0.982	0.989	0.957	**0.946**	0.952
ETTm1	1.000	0.990	0.954	0.950	0.978	0.926	**0.923**	0.976
ETTm2	1.000	0.994	0.984	0.975	0.982	0.963	**0.945**	0.958
Appliances	1.000	1.008	1.001	0.993	0.997	1.009	**0.991**	1.008
BeijingWeather	1.000	0.988	0.972	0.957	0.967	0.916	0.894	**0.888**
TimesFM-2.0								
ETTh1	1.000	0.995	0.974	0.966	0.982	0.947	**0.935**	0.982
ETTh2	1.000	0.996	0.987	0.973	0.978	0.962	0.934	**0.928**
ETTm1	1.000	0.975	0.952	0.934	0.947	0.967	**0.928**	0.947
ETTm2	1.000	0.960	0.950	0.940	0.946	0.946	**0.930**	0.961
Appliances	1.000	0.956	0.953	0.939	0.939	0.991	0.933	**0.921**
BeijingWeather	1.000	0.944	0.924	0.875	0.883	0.871	0.718	**0.577**
Timer-base								
ETTh1	1.000	1.002	0.995	0.988	0.993	1.017	0.981	**0.971**
ETTh2	1.000	1.004	0.998	0.990	0.993	0.983	**0.968**	0.972
ETTm1	1.000	1.000	**0.999**	1.005	1.001	1.078	1.135	1.220
ETTm2	1.000	1.002	0.996	0.985	0.988	0.978	0.957	**0.955**
Appliances	1.000	0.977	0.976	0.970	**0.969**	0.991	0.979	0.989
BeijingWeather	1.000	0.999	0.991	0.989	0.995	**0.970**	0.976	0.995
Sundial-base								
ETTh1	1.000	1.002	0.990	0.985	0.992	1.001	**0.982**	0.990
ETTh2	1.000	0.997	0.994	0.987	0.988	0.989	0.971	**0.968**
ETTm1	1.000	1.006	0.988	**0.981**	0.993	1.028	0.990	0.992
ETTm2	1.000	0.996	0.994	0.985	0.985	1.006	0.974	**0.968**
Appliances	1.000	1.022	1.016	0.994	0.994	1.013	0.954	**0.937**
BeijingWeather	1.000	0.988	0.983	0.978	0.982	0.972	**0.971**	0.983
Chronos-2								
ETTh1	1.000	0.994	0.964	0.961	0.985	**0.897**	0.910	0.963
ETTh2	1.000	0.991	0.980	0.970	0.977	0.950	**0.934**	0.937
ETTm1	1.000	1.003	0.966	0.945	0.970	0.926	**0.886**	0.913
ETTm2	1.000	0.999	0.984	0.975	0.984	0.952	**0.944**	0.968
Appliances	1.000	0.964	0.951	0.935	0.941	0.958	**0.917**	0.920
BeijingWeather	1.000	0.860	0.863	0.854	0.849	0.910	0.868	**0.846**

Note. Best: bold; second-best: double underline. Ties use displayed precision.

H.1 Candidate-budget values for every aggregation method

The following six tables retain all methods and budgets in Figure 5. Columns denote Frozen (F), Uniform (U), Learned weights (W), Classifier (C), Risk regressor (Risk), Pairwise ranker (Rank), Ensemble learning (E), Retrospective (Rt), Validation-best candidate (VB), Validation-selected TTC4TS (TTC), and unavailable Oracle (O). K is the total candidate budget, not the per-adapter sample count. All tables use the measured budget-specific fits and tie rules in Appendix C.2; in

particular, $K = 80$ is taken from the budget experiment rather than substituted from the main aggregation table. Bold uses unrounded values and excludes Oracle.

Table 12: Chronos-Bolt-small: dataset-specific candidate-budget relative MSE underlying Figure 5. Every entry averages four horizons and three seeds geometrically; candidate budget is K. Column definitions are given in Appendix H. O is the unavailable test-label Oracle.

Dataset	K	F	U	W	C	Risk	Rank	E	Rt	VB	VS	O
ETTh1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.997	**0.996**	0.999	0.997	1.002	**0.996**	0.999	1.001	1.000	0.987
	4	1.000	**0.963**	0.965	1.010	0.966	1.001	0.964	0.980	1.040	0.966	0.918
	8	1.000	0.984	0.972	1.026	0.985	0.995	**0.961**	0.998	1.040	0.993	0.899
	16	1.000	0.988	0.978	1.081	0.987	0.997	**0.961**	0.999	1.040	0.990	0.899
	80	1.000	0.972	0.906	1.098	0.982	0.985	**0.879**	1.005	0.941	0.927	0.703
ETTh2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	1.003	1.004	1.005	1.003	1.003	1.008	1.001	1.002	1.003	0.984
	4	1.000	**0.982**	0.983	1.000	**0.982**	0.983	0.991	**0.982**	0.992	0.991	0.925
	8	1.000	**0.981**	0.983	0.997	**0.981**	0.998	0.991	0.982	0.999	1.006	0.866
	16	1.000	0.990	0.991	0.997	0.990	0.991	0.991	**0.989**	0.998	1.000	0.866
	80	1.000	0.993	0.988	1.058	0.995	1.039	1.006	**0.978**	1.020	1.020	0.751
ETTm1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	**0.987**	0.989	0.997	**0.987**	0.996	0.990	0.993	1.000	0.994	0.964
	4	1.000	**0.949**	0.954	0.978	0.951	0.994	0.962	0.970	1.000	0.966	0.835
	8	1.000	**0.952**	0.953	1.025	**0.952**	0.992	0.953	0.968	0.995	0.963	0.802
	16	1.000	0.969	0.961	1.065	0.969	0.998	**0.953**	0.975	0.995	0.972	0.802
	80	1.000	0.943	0.888	1.884	0.962	1.065	**0.885**	0.966	0.967	0.908	0.655
ETTm2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	1.000	1.003	1.004	1.000	1.000	1.007	**0.998**	1.002	**0.998**	0.983
	4	1.000	**0.966**	0.971	0.996	0.967	0.995	0.986	0.970	1.005	0.974	0.907
	8	1.000	**0.971**	0.974	0.994	0.972	0.991	0.983	0.973	1.003	0.980	0.861
	16	1.000	**0.983**	**0.983**	1.006	0.984	0.986	0.984	0.984	1.002	0.986	0.860
	80	1.000	**0.980**	**0.980**	1.457	0.983	1.106	0.987	0.986	1.009	1.021	0.775
Appliances	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	1.021	1.004	1.017	1.019	1.015	**0.998**	0.999	1.000	1.002	0.983
	4	1.000	1.010	1.002	1.084	1.008	0.999	**0.997**	**0.997**	1.018	**0.997**	0.959
	8	1.000	1.012	1.006	1.088	1.011	1.005	**0.997**	1.000	1.027	1.025	0.939
	16	1.000	1.001	0.998	1.082	1.001	1.009	**0.997**	0.999	1.023	1.005	0.935
	80	1.000	0.994	0.969	1.466	0.994	1.023	**0.955**	0.993	1.007	0.963	0.853
BeijingWeather	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.968	0.964	0.963	0.964	**0.961**	**0.961**	0.968	0.989	0.967	0.936
	4	1.000	0.907	0.913	1.032	0.870	**0.777**	1.042	0.995	0.999	0.993	0.742
	8	1.000	0.930	0.904	0.802	0.900	**0.771**	0.958	0.966	0.864	0.869	0.733
	16	1.000	0.953	0.939	0.938	0.929	**0.773**	0.958	0.954	0.978	0.966	0.733
	80	1.000	0.954	0.892	0.966	0.947	**0.773**	0.861	0.954	0.984	0.907	0.696

Table 13: Chronos-Bolt-base: dataset-specific candidate-budget relative MSE underlying Figure 5. Datasets are ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather. Every entry averages four horizons and three seeds geometrically; candidate budget is K. Column definitions are given in Appendix H. O is the unavailable test-label Oracle.

Dataset	K	F	U	W	C	Risk	Rank	E	Rt	VB	VS	O
ETTh1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.990	0.991	0.995	**0.989**	0.997	0.992	0.997	1.001	0.999	0.982
	4	1.000	**0.959**	0.962	1.009	0.961	0.982	0.965	0.985	1.035	0.993	0.899
	8	1.000	0.973	0.966	1.037	0.974	1.014	**0.960**	0.994	1.030	1.005	0.880
	16	1.000	0.982	0.972	1.028	0.982	1.043	**0.960**	0.995	1.029	1.008	0.880
	80	1.000	0.964	0.889	1.030	0.973	1.006	**0.858**	0.988	0.949	0.956	0.683
ETTh2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	1.002	1.003	1.006	1.002	1.003	1.006	1.001	1.004	1.004	0.988
	4	1.000	0.969	0.971	0.987	0.969	0.980	0.995	**0.968**	1.001	0.987	0.908
	8	1.000	**0.969**	0.975	0.987	0.970	0.988	0.993	**0.969**	1.003	1.003	0.871
	16	1.000	0.983	0.987	0.990	0.984	**0.976**	0.993	0.983	1.003	0.999	0.870
	80	1.000	0.982	0.970	1.088	0.983	1.087	0.972	**0.965**	0.992	0.992	0.735
ETTm1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	**0.985**	0.988	0.995	**0.985**	0.991	0.989	0.988	0.998	0.986	0.964
	4	1.000	0.956	0.959	1.007	**0.955**	0.974	0.966	0.961	0.998	0.957	0.839
	8	1.000	0.963	0.960	1.025	0.960	0.971	**0.957**	0.965	0.977	0.958	0.815
	16	1.000	0.976	0.969	1.032	0.974	0.973	0.957	0.977	0.980	**0.955**	0.815
	80	1.000	0.950	0.893	1.375	0.966	1.015	**0.884**	0.972	0.925	0.888	0.682
ETTm2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	**0.989**	0.991	1.006	**0.989**	0.997	0.993	0.991	0.999	0.992	0.964
	4	1.000	**0.956**	0.957	0.993	0.958	0.981	0.959	**0.956**	0.987	0.964	0.901
	8	1.000	0.963	0.963	0.968	0.965	0.970	**0.959**	0.964	0.987	0.966	0.858
	16	1.000	0.978	0.976	0.980	0.979	0.971	**0.959**	0.976	0.986	0.966	0.857
	80	1.000	0.975	0.963	1.371	0.978	1.019	**0.955**	0.984	1.006	0.962	0.757
Appliances	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	1.010	0.999	1.016	1.010	1.017	**0.996**	0.997	1.000	1.001	0.981
	4	1.000	0.997	0.993	1.040	0.997	1.015	0.992	**0.991**	1.000	1.001	0.946
	8	1.000	0.999	0.995	1.071	0.999	1.012	**0.990**	0.993	1.067	1.029	0.930
	16	1.000	0.998	0.994	1.049	0.998	1.008	**0.990**	0.996	1.067	1.019	0.927
	80	1.000	0.993	0.987	1.357	0.993	1.038	**0.982**	0.990	1.060	0.984	0.868
BeijingWeather	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.990	0.993	1.000	0.988	0.992	1.002	**0.987**	1.006	0.991	0.968
	4	1.000	0.922	0.933	0.848	0.918	**0.841**	0.975	0.924	0.878	0.843	0.802
	8	1.000	0.916	0.905	0.774	0.904	**0.754**	0.932	0.953	0.871	0.842	0.712
	16	1.000	0.950	0.946	0.765	0.944	**0.753**	0.952	0.975	0.871	0.849	0.712
	80	1.000	0.957	0.913	0.802	0.954	**0.751**	0.837	0.966	0.946	0.943	0.675

Table 14: TimesFM-2.0: dataset-specific candidate-budget relative MSE underlying Figure 5. Datasets are ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather. Every entry averages four horizons and three seeds geometrically; candidate budget is K. Column definitions are given in Appendix H. O is the unavailable test-label Oracle.

Dataset	K	F	U	W	C	Risk	Rank	E	Rt	VB	VS	O
ETTh1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.994	0.994	1.004	**0.993**	0.998	0.994	0.997	1.005	0.994	0.973
	4	1.000	**0.934**	0.938	1.003	0.935	0.968	0.943	0.967	1.026	1.013	0.821
	8	1.000	0.956	**0.947**	0.982	0.957	0.977	0.949	0.966	1.023	1.003	0.780
	16	1.000	0.971	0.956	1.014	0.971	1.003	**0.950**	0.979	1.023	0.996	0.779
	80	1.000	0.966	**0.938**	1.030	0.971	0.987	**0.938**	0.990	1.043	0.972	0.713
ETTh2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	**1.000**	1.007	1.009	**1.000**	**1.000**	1.023	**1.000**	1.007	**1.000**	0.977
	4	1.000	0.940	0.948	0.953	**0.938**	0.961	0.995	0.940	0.962	0.944	0.850
	8	1.000	0.945	0.947	0.944	0.945	0.945	0.978	0.943	0.938	**0.926**	0.801
	16	1.000	0.967	0.967	**0.936**	0.966	0.962	0.989	0.949	0.938	0.937	0.799
	80	1.000	0.973	0.974	1.106	0.971	1.117	1.010	**0.955**	0.996	0.996	0.722
ETTm1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	**0.968**	0.969	0.997	**0.968**	1.003	0.969	0.976	1.010	0.978	0.911
	4	1.000	0.926	0.926	1.047	**0.921**	0.973	0.932	0.943	1.011	0.948	0.748
	8	1.000	0.929	0.928	1.088	0.928	0.977	**0.925**	0.935	1.008	0.944	0.716
	16	1.000	0.946	0.933	1.107	0.946	0.983	**0.926**	0.952	1.009	0.958	0.711
	80	1.000	0.934	0.897	1.932	0.943	1.022	**0.878**	0.963	0.992	0.906	0.602
ETTm2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	**0.964**	0.974	1.015	**0.964**	0.988	1.003	0.966	1.008	0.971	0.938
	4	1.000	**0.924**	0.928	1.003	**0.924**	0.981	0.940	0.930	1.015	0.940	0.851
	8	1.000	0.932	**0.931**	1.020	0.932	0.997	0.939	0.937	1.006	0.938	0.800
	16	1.000	0.940	**0.933**	1.009	0.940	0.978	0.935	0.946	0.996	0.939	0.791
	80	1.000	0.940	**0.931**	1.329	0.941	1.055	0.949	0.948	1.026	0.953	0.738
Appliances	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.963	**0.958**	1.010	0.963	1.013	**0.958**	0.963	1.034	0.963	0.957
	4	1.000	**0.924**	**0.924**	1.011	**0.924**	1.001	**0.924**	0.931	1.024	0.929	0.888
	8	1.000	0.936	0.926	1.047	0.937	0.991	**0.918**	0.935	1.068	0.928	0.867
	16	1.000	0.939	0.929	1.048	0.939	0.983	**0.917**	0.940	1.083	0.967	0.862
	80	1.000	0.939	0.918	1.477	0.939	0.991	**0.912**	0.941	1.087	0.964	0.813
BeijingWeather	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.886	0.854	0.787	0.780	**0.751**	0.789	0.914	1.000	0.774	0.719
	4	1.000	0.729	0.691	0.687	0.640	**0.563**	0.605	0.735	0.677	0.585	0.540
	8	1.000	0.766	0.711	0.860	0.674	**0.561**	0.575	0.764	0.677	0.690	0.497
	16	1.000	0.891	0.888	0.917	0.709	**0.523**	0.596	0.896	0.677	0.552	0.496
	80	1.000	0.875	0.861	0.572	0.694	**0.493**	0.595	0.875	0.561	0.559	0.439

Table 15: Timer-base: dataset-specific candidate-budget relative MSE underlying Figure 5. Datasets are ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather. Every entry averages four horizons and three seeds geometrically; candidate budget is K. Column definitions are given in Appendix H. O is the unavailable test-label Oracle.

Dataset	K	F	U	W	C	Risk	Rank	E	Rt	VB	VS	O
ETTh1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	1.004	1.001	1.003	1.004	**1.000**	**1.000**	**1.000**	1.004	1.003	0.999
	4	1.000	1.000	0.989	1.014	0.999	1.005	**0.985**	0.995	1.006	0.993	0.942
	8	1.000	1.001	**0.994**	1.062	1.002	1.003	0.995	1.001	1.003	1.003	0.920
	16	1.000	0.997	0.997	1.068	0.997	1.005	**0.995**	1.000	1.003	1.004	0.920
	80	1.000	0.988	**0.972**	1.121	0.989	1.034	0.975	0.992	1.019	1.012	0.850
ETTh2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	1.008	1.004	1.004	1.008	1.005	**1.000**	1.002	**1.000**	1.002	0.994
	4	1.000	0.986	0.983	**0.979**	0.985	0.983	0.992	0.988	0.983	0.983	0.932
	8	1.000	0.978	**0.975**	0.982	0.977	1.002	0.986	0.996	0.987	0.992	0.846
	16	1.000	0.987	0.982	**0.947**	0.986	0.995	0.986	0.992	0.987	0.968	0.846
	80	1.000	0.990	0.976	1.036	0.991	1.126	1.012	**0.971**	1.016	1.009	0.760
ETTm1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	1.000	**0.997**	1.002	0.999	0.998	0.998	0.998	1.000	0.999	0.974
	4	1.000	1.047	1.013	1.086	1.039	1.004	**0.995**	0.997	1.000	1.005	0.871
	8	1.000	1.026	1.005	1.104	1.021	1.005	**0.994**	1.000	1.018	1.023	0.835
	16	1.000	1.002	0.997	1.080	1.002	1.009	**0.995**	1.001	1.019	1.019	0.835
	80	1.000	1.005	0.978	1.597	1.000	1.026	**0.959**	0.994	1.055	0.987	0.745
ETTm2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	1.007	1.002	1.010	1.007	**0.997**	0.999	0.998	1.002	0.998	0.985
	4	1.000	0.967	0.961	0.979	0.966	0.964	0.972	0.969	0.988	**0.960**	0.878
	8	1.000	0.974	0.969	1.019	0.973	**0.959**	0.981	0.977	0.963	0.961	0.827
	16	1.000	0.984	0.977	1.028	0.983	**0.958**	0.981	0.986	0.965	0.961	0.826
	80	1.000	0.985	**0.964**	1.218	0.986	0.988	0.980	0.985	0.972	0.970	0.766
Appliances	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	**0.960**	0.965	1.021	0.961	1.002	0.967	0.973	1.009	0.982	0.936
	4	1.000	0.960	**0.953**	1.027	0.960	0.997	0.956	0.969	1.012	0.984	0.920
	8	1.000	0.963	0.953	1.007	0.963	0.989	**0.947**	0.979	1.015	0.982	0.883
	16	1.000	0.976	0.963	0.995	0.974	0.992	**0.947**	0.980	1.011	0.979	0.883
	80	1.000	0.970	**0.954**	1.273	0.969	1.002	0.957	0.978	1.006	0.958	0.845
BeijingWeather	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	1.002	0.997	1.002	1.001	1.001	**0.996**	0.998	1.000	1.002	0.979
	4	1.000	0.981	**0.980**	0.989	**0.980**	0.986	**0.980**	0.994	0.984	0.987	0.946
	8	1.000	0.979	0.976	0.989	0.977	**0.967**	0.976	0.986	0.984	0.978	0.919
	16	1.000	0.987	0.982	0.974	0.986	**0.968**	0.976	0.992	0.984	0.974	0.918
	80	1.000	0.989	0.972	1.000	0.985	**0.949**	0.969	0.995	1.004	**0.949**	0.864

Table 16: Sundial-base: dataset-specific candidate-budget relative MSE underlying Figure 5. Datasets are ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather. Every entry averages four horizons and three seeds geometrically; candidate budget is K. Column definitions are given in Appendix H. O is the unavailable test-label Oracle.

Dataset	K	F	U	W	C	Risk	Rank	E	Rt	VB	VS	O
ETTh1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	1.005	1.002	1.009	1.005	1.003	**1.000**	1.004	1.010	1.009	0.994
	4	1.000	0.990	**0.984**	1.021	0.989	1.007	0.987	0.992	0.997	0.990	0.935
	8	1.000	0.996	0.999	1.121	0.999	1.008	1.000	0.993	**0.990**	0.995	0.897
	16	1.000	0.992	0.994	1.119	0.992	1.032	1.000	0.993	**0.990**	1.010	0.897
	80	1.000	0.985	0.978	1.251	0.987	1.051	**0.970**	0.995	0.987	0.977	0.828
ETTh2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.998	1.001	1.005	0.998	1.002	1.010	**0.997**	1.000	0.998	0.982
	4	1.000	0.985	0.981	**0.965**	0.983	0.973	0.988	0.992	0.994	0.985	0.904
	8	1.000	0.974	0.973	**0.972**	**0.972**	1.006	0.974	0.988	0.983	1.000	0.822
	16	1.000	0.985	0.980	**0.942**	0.984	0.978	0.974	0.994	0.983	0.987	0.822
	80	1.000	0.987	**0.972**	1.054	0.986	1.032	0.983	0.982	0.990	0.991	0.734
ETTm1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	1.010	1.004	1.010	1.009	1.001	**1.000**	1.002	**1.000**	1.002	0.992
	4	1.000	0.996	0.991	1.005	0.994	1.000	0.985	0.983	1.000	**0.977**	0.882
	8	1.000	0.993	0.988	1.108	0.990	1.000	**0.984**	0.992	1.005	0.988	0.841
	16	1.000	0.994	0.989	1.130	0.993	1.001	**0.984**	0.993	1.003	**0.984**	0.841
	80	1.000	0.981	0.963	1.827	0.987	1.016	**0.954**	0.989	1.005	0.962	0.717
ETTm2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	1.000	1.005	1.006	1.000	1.004	1.018	**0.998**	1.000	1.000	0.983
	4	1.000	0.984	0.978	1.030	0.983	1.005	0.984	0.987	0.989	**0.976**	0.919
	8	1.000	0.976	**0.969**	1.042	0.974	0.995	0.986	0.977	0.993	0.993	0.842
	16	1.000	0.982	**0.975**	1.038	0.981	0.984	0.986	0.983	0.993	0.998	0.840
	80	1.000	0.985	**0.963**	1.265	0.984	1.081	0.974	0.992	0.998	0.992	0.794
Appliances	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	1.034	1.014	1.026	1.035	1.021	1.007	1.020	1.028	1.021	0.988
	4	1.000	0.966	0.962	0.997	0.966	0.979	0.962	0.962	1.020	**0.958**	0.922
	8	1.000	0.972	0.961	1.007	0.972	0.987	**0.955**	0.964	1.087	1.054	0.888
	16	1.000	0.988	0.975	1.003	0.987	0.987	**0.955**	0.975	1.087	1.044	0.884
	80	1.000	0.994	0.940	1.445	0.993	1.035	**0.928**	0.978	1.036	0.961	0.834
BeijingWeather	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.991	0.988	**0.987**	0.990	0.990	0.989	0.990	1.000	0.990	0.974
	4	1.000	0.953	0.946	0.956	0.950	0.956	**0.940**	0.954	0.954	0.951	0.915
	8	1.000	0.957	0.952	0.967	0.953	0.950	**0.942**	0.956	0.963	0.949	0.896
	16	1.000	0.973	0.963	0.969	0.968	0.949	**0.942**	0.969	0.963	0.949	0.895
	80	1.000	0.978	0.955	1.048	0.973	**0.952**	0.965	0.972	0.993	0.965	0.868

Table 17: Chronos-2: dataset-specific candidate-budget relative MSE underlying Figure 5. Datasets are ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather. Every entry averages four horizons and three seeds geometrically; candidate budget is K. Column definitions are given in Appendix H. O is the unavailable test-label Oracle.

Dataset	K	F	U	W	C	Risk	Rank	E	Rt	VB	VS	O
ETTh1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.996	**0.995**	1.000	0.996	**0.995**	**0.995**	0.997	1.007	1.001	0.982
	4	1.000	0.935	0.932	0.959	0.934	0.965	**0.930**	0.950	1.024	0.953	0.853
	8	1.000	0.960	0.947	0.986	0.956	0.975	**0.933**	0.968	1.017	0.964	0.846
	16	1.000	0.973	0.957	0.986	0.971	0.977	**0.933**	0.986	1.017	0.980	0.845
	80	1.000	0.961	0.883	1.155	0.968	0.979	**0.839**	0.978	0.975	0.869	0.709
ETTh2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.990	0.989	**0.987**	0.990	1.005	0.990	0.989	0.992	0.992	0.965
	4	1.000	0.928	0.922	0.918	0.927	0.956	**0.913**	0.931	0.923	0.931	0.835
	8	1.000	0.938	0.929	0.911	0.940	0.952	**0.907**	0.944	0.956	0.947	0.817
	16	1.000	0.963	0.951	0.917	0.963	0.951	**0.909**	0.966	0.956	0.942	0.817
	80	1.000	0.970	0.950	1.051	0.971	1.130	**0.915**	0.974	0.965	0.953	0.725
ETTm1	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	**1.000**	1.006	1.002	1.017	1.005	1.014	1.002	1.005	1.004	1.011	0.968
	4	1.000	**0.932**	0.937	0.962	0.933	0.964	0.934	0.933	0.984	0.933	0.814
	8	1.000	0.947	0.938	1.017	0.945	0.977	**0.935**	0.941	0.994	**0.935**	0.787
	16	1.000	0.968	0.951	1.037	0.968	0.972	**0.935**	0.964	0.992	**0.935**	0.786
	80	1.000	0.945	0.885	1.945	0.961	0.988	**0.878**	0.958	0.907	0.899	0.640
ETTm2	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.999	0.998	1.013	0.999	1.007	1.002	**0.997**	1.002	**0.997**	0.972
	4	1.000	0.951	0.952	0.991	**0.950**	0.955	0.962	**0.950**	0.993	**0.950**	0.875
	8	1.000	0.960	**0.957**	1.006	0.959	0.961	0.961	0.959	0.994	0.961	0.839
	16	1.000	0.975	0.967	1.015	0.974	0.962	**0.961**	0.972	0.992	0.966	0.838
	80	1.000	0.975	**0.960**	1.526	0.977	1.029	0.965	0.986	1.006	0.999	0.785
Appliances	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.954	0.956	0.956	0.956	0.980	0.957	0.976	1.000	**0.948**	0.916
	4	1.000	0.911	0.906	0.933	0.909	0.969	**0.905**	0.920	1.010	0.914	0.848
	8	1.000	0.918	0.908	0.950	0.918	0.936	**0.899**	0.923	1.054	0.941	0.833
	16	1.000	0.945	0.927	0.947	0.942	0.950	**0.899**	0.948	1.054	0.933	0.830
	80	1.000	0.935	0.898	1.511	0.937	0.972	**0.887**	0.946	0.959	0.920	0.780
BeijingWeather	1	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
	2	1.000	0.868	0.841	0.984	0.849	**0.812**	0.819	0.866	0.834	0.843	0.789
	4	1.000	0.821	0.793	0.783	0.788	**0.762**	0.776	0.823	0.855	0.781	0.730
	8	1.000	0.845	0.805	0.892	0.796	**0.756**	0.779	0.844	0.828	0.783	0.720
	16	1.000	0.851	0.834	0.875	0.824	**0.752**	0.814	0.852	0.820	0.810	0.719
	80	1.000	0.854	0.803	0.888	0.829	**0.770**	0.784	0.853	0.822	0.815	0.700

I CANDIDATE-INFORMED UQ

A small network receives five common observable features: log-one-plus context standard deviation, log-one-plus RMS first difference of context, log-one-plus square root of the identity's retrospective MSE, relative forecast lead, and log-one-plus absolute displacement of the forecast from the last observation. Retrospective MSE averages the already observed horizon and channels. Seven additional features describe current candidates: log-one-plus standard deviation within each of the 3 generator families, standard deviation of the 3 family means, family-balanced RMS displacement from μ, and the signed log-one-plus displacements of the empirical 10th and 90th candidate percentiles from μ.

I.1 Comparators and information access

All methods use the same targets and calibration origins. The selected TTC point predictor is shared across TTC interval methods, except that raw candidate quantiles and CQR use their own central quantile for scoring.

- **Frozen Gaussian** centers intervals at Frozen and uses Gaussian quantiles times per-lead/channel residual RMS estimated on the selection block.
- **Frozen/TTC split CP** use absolute calibration residuals around their respective point forecasts, giving a constant half-width for each lead and channel.
- **Raw candidate quantiles** use empirical lower/upper candidate percentiles without calibration. **Candidate-quantile CP** expands those endpoints using the corrected quantile of $\max(\ell_i - Y_i, Y_i - u_i, 0)$ on calibration.
- **History-scale CP** matches the scale head and training objective but uses only the five common features. **Candidate-spread CP** uses $s = \sqrt{d^2 + b^2}$, where b is the greater of 0.01 and 5% of selection-block residual RMS, separately for each lead and channel. **Candidate-scale CP** uses the twelve-feature learned scale.
- **CQR (non-shrinking)** fits a $12 \to 32 \to 7$ residual-quantile MLP with pinball loss at levels 0.025, 0.05, 0.10, 0.50, 0.90, 0.95, and 0.975. Outputs are sorted before the loss to prevent crossing. Optimization matches the scale heads. Calibration uses $\max(\ell_i - Y_i, Y_i - u_i, 0)$, a conservative non-shrinking variant of CQR (Romano et al., 2019), with the same candidate features available to the proposed scale head.
- **Frozen/TTC rolling CP** maintain the most recent n complete residual horizons, initially the calibration block, and replace the oldest after each observed test horizon. **Candidate-scale rolling CP** does the same with normalized residuals and a fixed learned scale head.
- **TTC bounded ACI** combines rolling residuals with $\alpha_{t+1,h,c} = \Pi_{[1/(n+1),\,0.5]}[\alpha_{t,h,c} + 0.01(\alpha - \mathbf{1}\{Y_{t,h,c} \notin I_{t,h,c}\})]$, initialized at the desired α. This explicit finite-width, projected adaptation of ACI (Gibbs & Candes, 2021) does not inherit the unrestricted ACI guarantee. Adaptive updates occur only after the current horizon has become observed.

Fixed-calibration methods never use test labels for interval updates. Online comparators use previously completed test horizons as historical feedback, never the current or future horizon. Generator, point-aggregator, and uncertainty-head parameters remain frozen in both settings. Sequential residual modeling such as SPCI (Xu & Xie, 2023) is related work, not a reproduced comparator here; neither native probabilistic foundation-model interfaces nor bootstrap-refitted backbones are evaluated.

I.2 Metrics and measured comparison

We evaluate pointwise coverage at nominal 80%, 90%, and 95%, interval width, and the interval score

$$\mathrm{IS}_\alpha(\ell, u; y) = (u - \ell) + \frac{2}{\alpha}(\ell - y)_+ + \frac{2}{\alpha}(y - u)_+. \tag{84}$$

Weighted interval score (WIS) combines these three intervals with the supplied central prediction m:

$$\mathrm{WIS} = \frac{0.5|y - m| + \sum_{\alpha \in \{0.2, 0.1, 0.05\}} (\alpha/2)\mathrm{IS}_\alpha}{3.5}. \tag{85}$$

This follows interval-format forecast evaluation (Bracher et al., 2021). The central prediction is the fixed point forecast for scale/residual methods and the 0.5 quantile for candidate-quantile/CQR methods. Table 18 reports model–dataset variation rather than treating aggregate coverage as uniform validity.

Simultaneous per-channel trajectory bands further illustrate the coverage–width tradeoff. Candidate max-score CP covers 90.1% of complete trajectories at mean width 5.875, compared with 88.5% and 4.916 for TTC max-residual CP. Higher trajectory coverage therefore comes with wider bands. The extension supports calibrated interval construction from TTC4TS candidates, while showing that candidate diversity alone and a more complex scale head do not ensure better uncertainty estimates.

Table 18: Fixed-calibration uncertainty by model and dataset at horizon 96 on ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather. Each pair gives 90% pointwise coverage (C, percent) and geometric relative WIS (R) over seeds 0–2. WIS uses 80/90/95% intervals and is normalized to Frozen split CP in each cell. CQR uses the declared non-shrinking correction.

	Frozen split		TTC split		Candidate-scale		CQR	
Dataset	C	R	C	R	C	R	C	R
Chronos-Bolt-small								
ETTh1	94.6	1.000	94.9	**0.967**	**93.4**	1.037	95.5	1.020
ETTh2	86.2	**1.000**	86.6	1.005	**87.7**	1.072	92.4	1.068
ETTm1	92.8	1.000	93.4	**0.958**	**92.1**	1.023	94.8	1.110
ETTm2	87.8	1.000	87.9	**0.994**	87.9	1.026	**90.8**	1.030
Appliances	87.7	1.000	86.1	**0.998**	87.8	1.006	**91.3**	1.020
BeijingWeather	88.0	1.000	**88.7**	0.988	**88.7**	**0.948**	94.0	0.976
Chronos-Bolt-base								
ETTh1	95.0	1.000	94.9	**0.961**	**93.5**	1.016	95.9	1.004
ETTh2	87.3	1.000	87.6	**0.992**	**87.8**	1.088	**92.2**	1.065
ETTm1	92.9	1.000	93.2	**0.979**	**92.4**	1.036	94.4	1.129
ETTm2	87.9	1.000	88.0	**0.992**	87.6	1.029	**90.5**	1.038
Appliances	87.3	**1.000**	86.1	1.001	87.1	1.016	**91.4**	1.036
BeijingWeather	87.8	1.000	88.8	0.998	**89.5**	**0.956**	93.5	0.957
TimesFM-2.0								
ETTh1	95.9	1.000	95.1	**0.976**	**94.7**	1.009	95.9	1.017
ETTh2	84.8	**1.000**	84.3	1.020	86.1	1.032	**90.8**	1.090
ETTm1	94.0	1.000	94.3	**0.939**	**92.8**	0.957	94.4	1.011
ETTm2	**88.2**	1.000	**88.2**	**0.967**	87.5	1.006	**91.8**	1.009
Appliances	86.7	1.000	87.0	**0.962**	86.9	0.985	**91.2**	0.994
BeijingWeather	89.5	**1.000**	89.5	1.035	**90.4**	1.056	93.8	1.036
Timer-base								
ETTh1	95.3	1.000	95.5	**0.976**	**93.4**	0.987	95.1	1.015
ETTh2	86.8	1.000	86.9	**0.990**	87.5	1.089	**91.3**	1.086
ETTm1	93.9	1.000	94.0	**0.990**	**92.3**	1.027	94.6	1.095
ETTm2	87.6	1.000	87.3	**0.983**	86.3	1.053	**90.7**	1.020
Appliances	86.9	1.000	87.0	**0.994**	87.9	0.996	**91.7**	1.010
BeijingWeather	88.1	1.000	88.4	1.026	**89.1**	1.020	94.1	**0.974**
Sundial-base								
ETTh1	95.2	1.000	95.0	**0.982**	**94.9**	0.984	95.4	1.024
ETTh2	87.0	**1.000**	87.0	1.007	**87.6**	1.114	92.7	1.089
ETTm1	94.5	1.000	94.7	**0.996**	**93.4**	1.015	94.7	1.079
ETTm2	**89.0**	1.000	88.3	**0.994**	88.0	1.047	92.2	1.032
Appliances	**89.0**	1.000	87.4	**0.973**	88.1	0.983	91.8	0.986
BeijingWeather	88.5	1.000	88.6	0.915	**88.8**	0.904	94.0	**0.902**
Chronos-2								
ETTh1	95.2	1.000	**94.9**	**0.943**	95.2	0.984	96.1	0.986
ETTh2	86.4	**1.000**	85.5	1.037	85.8	1.112	**91.9**	1.058
ETTm1	93.8	1.000	93.9	**0.973**	**92.5**	1.031	94.7	1.090
ETTm2	87.8	1.000	88.4	**0.997**	88.1	1.040	**91.0**	1.043
Appliances	87.6	1.000	87.7	**0.972**	**88.7**	0.975	92.9	1.001
BeijingWeather	88.1	1.000	88.9	1.001	**91.0**	1.010	93.6	**0.993**

Note. Coverage ranks minimize distance to 90 percent; WIS ranks minimize error.

A marginal 90% interval at each step does not imply 90% coverage of an entire trajectory. For a simultaneous band within each channel, we calibrate $R_{i,c}^{\max} = \max_{h \le 96} |Y_{i,h,c} - \mu_{i,h,c}| / s_{i,h,c}$, then use its corrected 90th percentile for all leads. Table 19 compares constant-scale and candidate-scale versions. These bands target each channel's trajectory separately, not the joint event across every channel, and retain the temporal-dependence limitation of the marginal experiment.

Table 19: Simultaneous 90% bands for each channel's full 96-step trajectory, over ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather and seeds 0–2. Calibration uses the maximum normalized error across the horizon, separately for each channel. C is whole-trajectory coverage (percent), and W is mean width in fitting-standardized units. These bands do not target simultaneous coverage across all channels.

	TTC max-residual CP		Candidate max-score CP	
Model	C	W	C	W
Chronos-Bolt-small	88.7	4.856	**90.2**	5.848
Chronos-Bolt-base	88.3	4.855	**90.1**	5.879
TimesFM-2.0	89.5	5.003	**90.0**	5.844
Timer-base	88.3	5.013	**90.0**	6.025
Sundial-base	88.1	4.912	**90.3**	5.821
Chronos-2	88.1	4.856	**90.2**	5.834

Note. Best: bold; second-best: double underline. Ties use displayed precision. Ranks minimize distance to nominal 90 percent coverage; width is descriptive.

J ILLUSTRATIVE SUCCESSFUL FORECASTS

Figures 7 and 8 visualize saved forecasts from six fixed model–dataset pairs, one for each checkpoint and dataset, at $H = 96$ and seed 0. Each panel shows 96 observed steps and the next 96 true and predicted values. The curves come from the revised campaign's saved Frozen and validation-selected TTC4TS predictions, and no fitting or forecast regeneration was performed for these plots.

These are explicitly post-hoc success illustrations. Within each fixed pair, eligible origin–channel combinations have positive TTC4TS MSE reduction and TTC4TS MSE below half the variance of the true future. We select the observed gain nearest the 75th percentile of eligible gains, resolving ties by origin and channel order. This selection uses test targets only to choose an illustration; it does not alter the deployed predictor, validation choices, or aggregate results. The examples are not a random sample or an estimate of typical performance. Panels use fitting-standardized coordinates and individual vertical scales. Table 20 identifies the selected origins, channels and deployment rules; the accompanying CSV and manifest include timestamps, exact errors and source hashes.

Table 20: Provenance for the six post-hoc forecast illustrations on ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather. All use horizon 96 and seed 0. Origin t is a zero-based dataset index; channel MSEs use adapter-prefix standardization. The displayed rule was chosen on validation before these examples were selected.

Panel / model	Dataset	Channel	Rule	t	Gain (%)
(a) Bolt-S	ETTh1	MUFL	Ensemble learning	11808	22.9
(b) Bolt-B	ETTh2	MUFL	Val.-best	14112	8.2
(c) TimesFM	ETTm1	HUFL	Val.-best	53664	43.3
(d) Timer	ETTm2	OT	Ensemble learning	46560	35.6
(e) Sundial	Appliances	T4	Ensemble learning	19148	61.6
(f) Chronos-2	BeijingWeather	TEMP	Retro.	35539	15.3

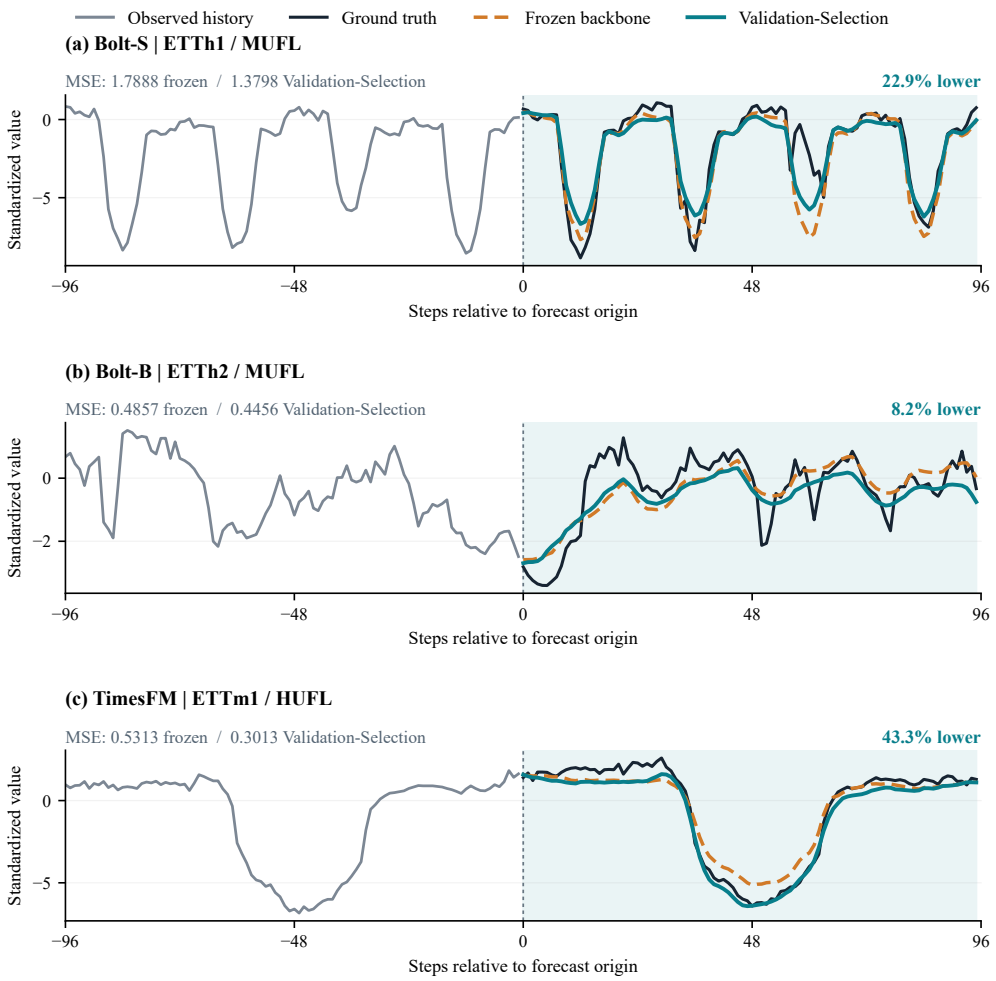

Figure 7: Saved successful forecasts for (a) Chronos-Bolt-small on ETTh1 (MUFL), (b) Chronos-Bolt-base on ETTh2 (MUFL), and (c) TimesFM-2.0 on ETTm1 (HUFL), at horizon 96 and seed 0. Gray is observed history, black is ground truth, orange dashed is Frozen, and teal is validation-selected TTC4TS. Shading marks the forecast interval. MSE and percentage reductions refer only to the plotted channel and horizon. These post-hoc illustrations follow Appendix J's selection rule.

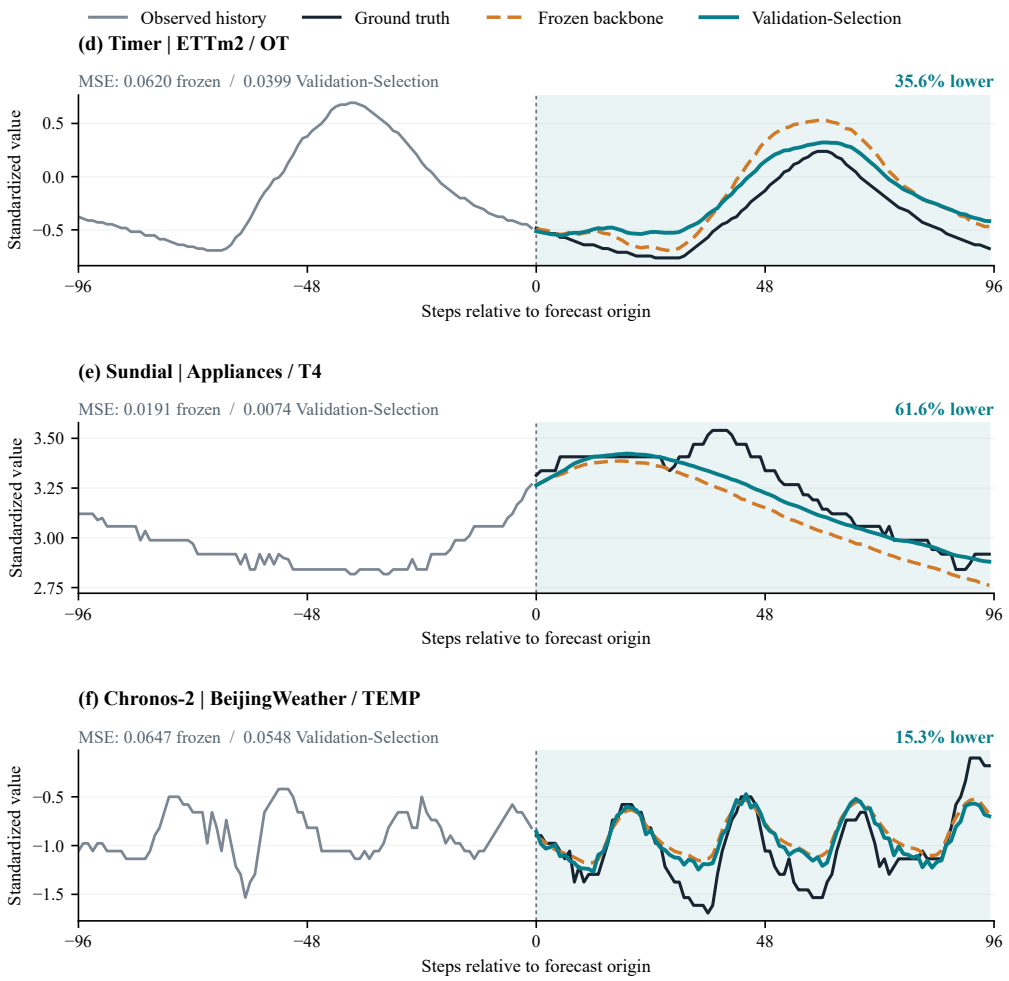

Figure 8: Saved successful forecasts for (d) Timer-base on ETTm2 (OT), (e) Sundial-base on Appliances (T4), and (f) Chronos-2 on BeijingWeather (TEMP), at horizon 96 and seed 0. Each panel compares the same observed history and future targets for Frozen and validation-selected TTC4TS. Colors, shading and error annotations follow Figure 7. These are post-hoc success illustrations, not representative estimates of average gains.

J.1 INTERVALS ON PRESPECIFIED FORECAST WINDOWS

Figures 9 and 10 show the first test origin for six prespecified model–dataset pairs, at seed 0 and horizon 96. The channel is OT for ETT, Appliances for the energy dataset, and TEMP for BeijingWeather. No origin or channel is selected by test coverage, interval width, or prediction error. Each row uses identical vertical limits for Frozen split CP and Candidate-scale CP, with 48 observed steps followed by the 96-step forecast. Annotations report coverage within that single window; they are descriptive, not evidence of nominal coverage for every query. All curves are extracted from saved experiment endpoints.

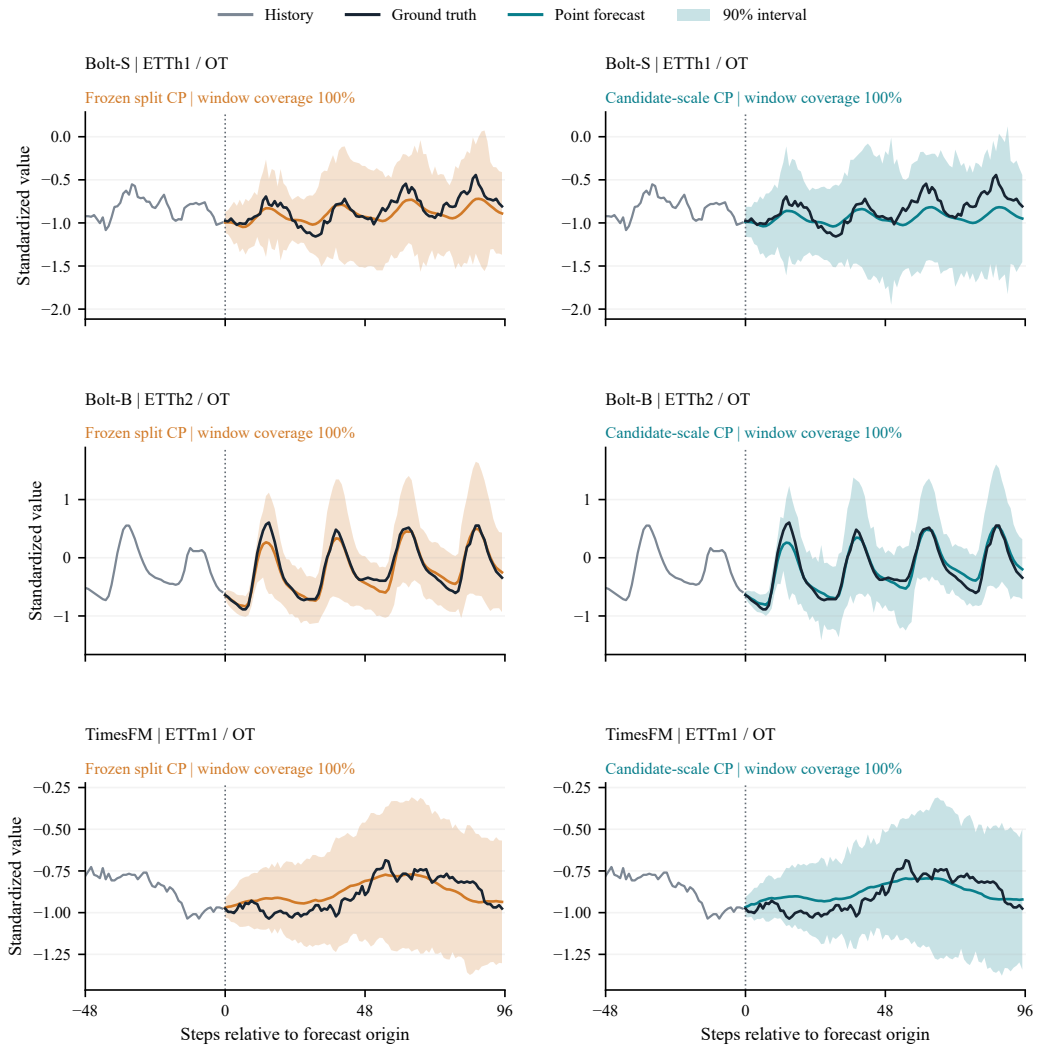

Figure 9: Nominal 90% intervals for Chronos-Bolt-small on ETTh1 (OT), Chronos-Bolt-base on ETTh2 (OT), and TimesFM-2.0 on ETTm1 (OT), at the first test origin, horizon 96, seed 0. Left: Frozen split CP. Right: Candidate-scale CP around the refitted/selected TTC forecast. Shading denotes prediction intervals; black is ground truth, gray is observed history, and colored curves are point predictions. These windows are fixed in advance of the uncertainty results.

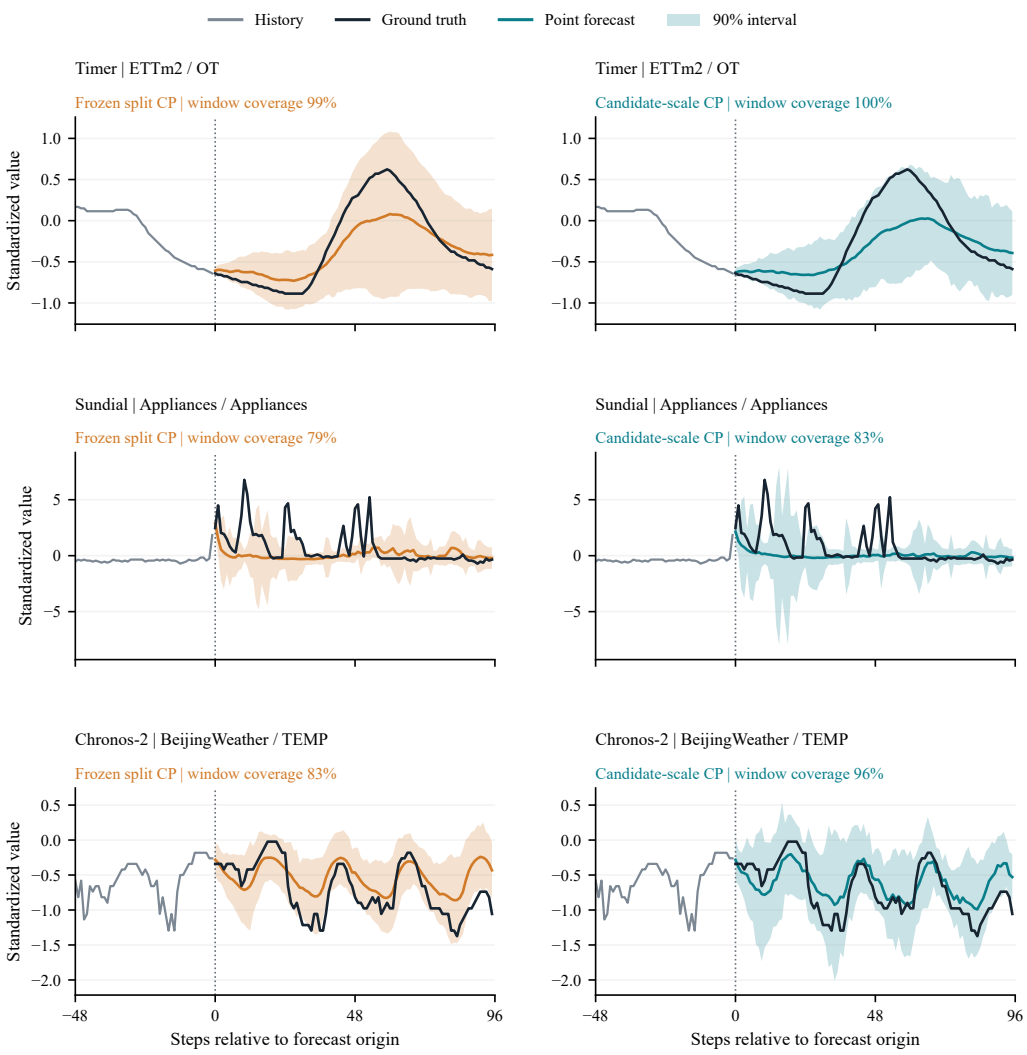

Figure 10: Nominal 90% intervals for Timer-base on ETTm2 (OT), Sundial-base on Appliances (Appliances), and Chronos-2 on BeijingWeather (TEMP), at the first test origin, horizon 96, seed 0. Left: Frozen split CP. Right: Candidate-scale CP. Colors, history length, shading, and matched vertical limits follow Figure 9. Coverage failures are retained rather than filtering examples by success.

K CANDIDATE AND GENERATOR COMPARISONS

Table 21: Every individual generator operator on ETTh1, ETTh2, ETTm1, ETTm2, Appliances, and BeijingWeather, evaluated on identical test origins. No candidate is omitted for poor performance.

Candidate	Rel. MSE	Rel. MAE	Candidate	Rel. MSE	Rel. MAE
ewma	1.029	1.044	joint=0.284211	1.013	1.062
horizontal	1.380	1.236	joint=0.3	1.015	1.067
identity	1.000	**1.000**	magnitude flip	0.995	1.005
input=0.0157895	1.001	**1.000**	magnitude warp	0.981	1.003
input=0.0315789	1.000	1.001	output=0.0157895	1.000	1.001
input=0.0473684	1.001	1.002	output=0.0315789	1.000	1.002
input=0.0631579	1.000	1.003	output=0.0473684	1.000	1.004
input=0.0789474	0.999	1.004	output=0.0631579	1.000	1.005
input=0.0947368	0.998	1.005	output=0.0789474	1.000	1.008
input=0.110526	0.998	1.006	output=0.0947368	1.001	1.010
input=0.126316	0.997	1.007	output=0.110526	1.002	1.012
input=0.142105	0.997	1.008	output=0.126316	1.002	1.015
input=0.157895	0.997	1.009	output=0.142105	1.003	1.018
input=0.173684	0.997	1.009	output=0.157895	1.004	1.021
input=0.189474	0.997	1.010	output=0.173684	1.005	1.024
input=0.205263	0.997	1.011	output=0.189474	1.007	1.027
input=0.221053	0.997	1.012	output=0.205263	1.008	1.031
input=0.236842	0.996	1.013	output=0.221053	1.009	1.034
input=0.252632	0.995	1.014	output=0.236842	1.011	1.038
input=0.268421	0.995	1.015	output=0.252632	1.013	1.042
input=0.284211	0.995	1.016	output=0.268421	1.015	1.046
input=0.3	0.994	1.017	output=0.284211	1.017	1.051
jitter	0.999	1.001	output=0.3	1.019	1.055
joint=0.0157895	1.000	1.001	retrieval=demo1	0.977	1.004
joint=0.0315789	1.000	1.003	retrieval=demo10	1.013	1.018
joint=0.0473684	1.001	1.005	retrieval=demo2	**0.971**	1.006
joint=0.0631579	1.001	1.008	retrieval=demo3	0.983	1.010
joint=0.0789474	0.999	1.011	retrieval=demo4	1.001	1.012
joint=0.0947368	0.999	1.014	retrieval=demo5	0.991	1.011
joint=0.110526	0.999	1.017	retrieval=demo6	0.999	1.014
joint=0.126316	1.000	1.020	retrieval=demo7	0.999	1.014
joint=0.142105	1.001	1.024	retrieval=demo8	1.000	1.012
joint=0.157895	1.002	1.027	retrieval=demo9	0.999	1.014
joint=0.173684	1.003	1.031	slope=minus	1.083	1.071
joint=0.189474	1.004	1.035	slope=plus	1.108	1.083
joint=0.205263	1.006	1.040	spike	0.999	1.001
joint=0.221053	1.007	1.044	time=flip	2.108	1.588
joint=0.236842	1.008	1.048	time=warp	1.269	1.160
joint=0.252632	1.010	1.053	vertical=minus	1.000	**1.000**
joint=0.268421	1.011	1.057	vertical=plus	1.000	**1.000**