STEP: LONG-TERM TIME SERIES FORECASTING THROUGH THE LENS OF THE FORECAST START

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term time-series forecasting typically focuses on modeling the distant future, yet prediction becomes increasingly uncertain as the forecast horizon grows. We revisit this problem from the forecast start and argue that it may play a larger role in long-horizon accuracy than its position suggests: improving the forecast start can benefit subsequent forecasting. We propose STEP, which improves forecasting by providing a more accurate initial forecast and further strengthening the forecast start, while also serving as an enhancement for other forecasting backbones. First, STEP encodes adjacent temporal evolution through historical differences, then enriches these representations with learned feature shifts, cross-variable interaction, and subspace interaction to improve the initial forecast. Second, STEP treats the initial forecast as a natural continuation of history and jointly models the two, reinforced by a transition-aware training objective that explicitly emphasizes the forecast start. Experiments on multiple real-world datasets show that STEP achieves competitive forecasting performance and improves diverse forecasting backbones. With training-based integration, STEP improves seven different backbones in 96.4% of the evaluated settings. At inference time, STEP can provide an improved forecast start to a frozen backbone, reducing subsequent forecasting error. Code is available at `https://anonymous.4open.science/r/STEP-DDD5`.

1 INTRODUCTION

Long-term time-series forecasting plays an important role in many fields, such as energy, transportation, and meteorology (Zhou et al., 2021; Wu et al., 2021; Liu et al., 2024; Gao et al., 2025). Yet the greater uncertainty associated with the distant future makes accurate forecasting over long horizons inherently challenging. Consequently, existing research has naturally focused on capturing complex dependencies within and across time series to better model the distant future (Nie et al., 2023; Zeng et al., 2023; Liu et al., 2024; Wang et al., 2024; 2025; Liu et al., 2026).

However, recent studies indicate that directly improving long-term forecasting accuracy still poses significant challenges (Liu et al., 2025; Kim et al., 2025). Forecast uncertainty generally increases with the horizon (Hyndman & Athanasopoulos, 2021), and long-term forecasting can face severe distribution shifts that are difficult to anticipate (Kim et al., 2025), complicating direct forecasting. Such complexity makes it more important to first address the aspects of forecasting that are more directly controllable.

Here we revisit this problem from a different perspective: the forecast start may play a larger role in long-horizon accuracy than its position suggests. As Fig. 1(a) shows, once the true forecast start becomes observed, providing it to the same frozen forecaster can substantially improve predictions over the subsequent horizon. Compared with the original forecast, the final cumulative error is reduced by **44%–98%** in these examples across eight datasets. This observation points to another way to improve long-term forecasting accuracy: improving how forecasting begins, where history transitions into the predicted future.

Following this observation, we propose **STEP**, which revisits long-term forecasting by modeling adjacent evolution and strengthening the forecast start. As illustrated in Fig. 1(b), STEP aims to reduce first-step errors in two ways: improving the overall initial forecast by capturing adjacent temporal

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.

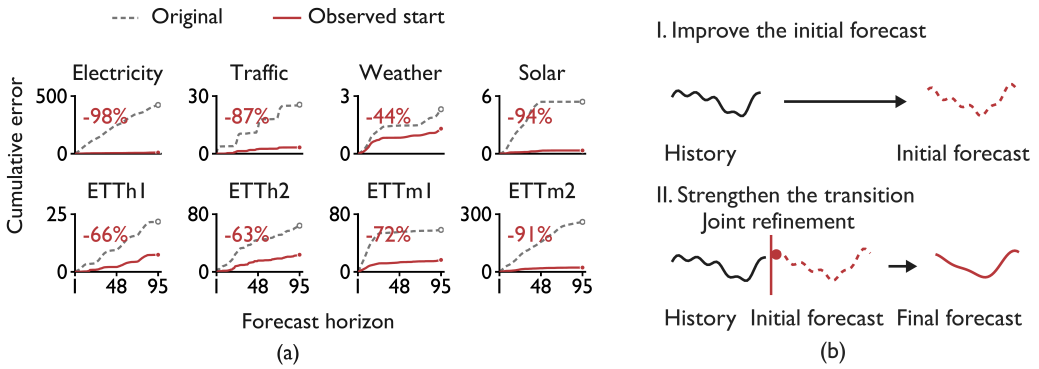


Figure 1: Impact of the forecast start and overview of STEP. (a) Analysis of the forecast start across eight datasets. Original denotes the model's original forecast. Observed start provides the corresponding ground-truth first future value to the same frozen model and re-forecasts the subsequent horizon. The reduction in cumulative error relative to Original quantifies the improvement after the forecast start becomes observed. Cumulative error excludes the observed point. All panels use frozen STEP variants with $L = H = 96$. (b) STEP first improves the initial forecast by capturing adjacent temporal evolution, then strengthens the history-to-future transition through joint refinement.

evolution and strengthening the transition from history to future. The former can improve forecast accuracy, including at the start, while the latter places additional emphasis on the forecast start.

Specifically, STEP first encodes adjacent temporal evolution through historical differences, then enriches these representations through learned feature shifts, cross-variable interaction, and subspace interaction. These operations improve the initial forecast. Beyond a direct mapping from history to future, STEP treats the initial forecast as a continuation of the observed history and jointly models the two. This brings both historical dynamics and the proposed future trajectory to the transition. A transition-aware objective then emphasizes forecast-start accuracy.

Experiments on multiple real-world datasets demonstrate competitive forecasting accuracy. STEP improves seven backbones spanning different forecasting architectures, including strong recent models, in **96.4%** of evaluated settings. STEP can also improve existing models through inference-time enhancement, reducing subsequent forecasting error by supplying an improved forecast start while keeping the backbone frozen. Further analyses connect these gains to forecast-start improvements and examine the two main aspects of STEP: improving the initial forecast and strengthening the forecast start. Our contributions are:

- We revisit long-term time-series forecasting from the perspective of the forecast start and show that long-range forecasting performance is closely related to how prediction begins.
- We propose STEP, which encodes adjacent temporal evolution through historical differences, enriches these representations through learned feature shifts, cross-variable interaction, and subspace interaction to improve the initial forecast, and jointly refines history and the initial forecast under a transition-aware objective that emphasizes the forecast start.
- Extensive experiments show improved long-term forecasting performance across multiple real-world datasets. STEP further improves diverse forecasting architectures, including strong recent models, in **96.4%** of evaluated settings. STEP also enhances existing models at inference time by supplying an improved forecast start to reduce subsequent forecasting error while keeping the backbone frozen.

2 Related Work

Long-term time-series forecasting. Existing research captures temporal structure for long-term forecasting through efficient attention, series decomposition, frequency-domain representations, and trend modeling, including Informer (Zhou et al., 2021), Autoformer, FEDformer, ETSformer, FiLM (Wu et al., 2021; Zhou et al., 2022b; Woo et al., 2022; Zhou et al., 2022a), and DLinear (Zeng

et al., 2023). Another line exploits multiscale, multiresolution, or multiperiod structures, as in Pyraformer, SCINet, TimesNet, MICN (Liu et al., 2022b;a; Wu et al., 2023; Wang et al., 2023), Scaleformer, PDF (Shabani et al., 2023; Dai et al., 2024), TimeMixer (Wang et al., 2024), Pathformer, and TimeMixer++ (Chen et al., 2024; Wang et al., 2025). In addition, PatchTST (Nie et al., 2023), Crossformer, TSMixer, MTS-Mixers (Zhang & Yan, 2023; Chen et al., 2023; Li et al., 2025), iTransformer (Liu et al., 2024), and Client (Gao et al., 2025) reorganize sequence representations or variable interactions to represent historical information for forecasting. Recent models further develop cross-variable modeling: TQNet (Lin et al., 2025) uses temporal queries, VPNet (Liu et al., 2026) combines variate–patch representations with interactions among neighboring variables, and SEED (Xiong et al., 2026) uses spectral entropy to balance temporal and inter-variable dependencies. STEP complements these directions by studying long-term forecasting from the forecast start.

Temporal variation and forecast refinement. Some studies model temporal variation or correct existing forecasts. The differential attention fusion model (Li et al., 2022) uses differences between adjacent time points to capture temporal changes; TDAlign (Xiong et al., 2025) aligns changes between adjacent time steps in predictions and ground truth to strengthen dependencies within the forecast target. GBT (Shen et al., 2023) uses first-stage predictions to provide a better starting point for subsequent forecasting, while δ-Adapter (Liang et al., 2026) improves existing forecasts through input adjustments and output residual corrections with the forecasting model frozen. STEP combines shifted variable interaction for the initial forecast with joint history–future refinement and a transition-aware objective. It can further enhance existing forecasters through training-based integration or inference-time forecast-start enhancement.

3 METHOD

Given historical observations $\mathbf{X} \in \mathbb{R}^{L\times C}$ of C variables, we predict the next H observations $\mathbf{Y} \in \mathbb{R}^{H\times C}$. STEP encodes adjacent evolution to generate an initial forecast trajectory and refines it jointly with history. A transition-aware objective supervises forecast values and adjacent changes, emphasizing the forecast start.

3.1 GENERATING AN INITIAL FORECAST

We first encode historical series and their adjacent differences using instance-normalized history $\bar{\mathbf{X}} = \mathcal{N}(\mathbf{X})$, with statistics computed over the observed history for each variable:

$$\mathbf{Z}^{(0)} = \sum_{k=0}^{K} E_k\big((\Delta^k \bar{\mathbf{X}})^\top\big) \in \mathbb{R}^{C\times d}, \tag{1}$$

where $\Delta^0\bar{\mathbf{X}} = \bar{\mathbf{X}}$, Δ^k for $k \geq 1$ denotes the kth-order adjacent difference with sequence length preserved, and each E_k projects a length-L sequence into d features.

For block input $\mathbf{Z}^{(\ell)}$, we partition each variable representation $\mathbf{z}_c = \mathbf{Z}^{(\ell)}_{c,:}$ into P contiguous segments, $\mathbf{z}_c = [\mathbf{z}_{c,1}, \ldots, \mathbf{z}_{c,P}]$, where $\mathbf{z}_{c,p} \in \mathbb{R}^m$ and $m = d/P$. Each segment is shifted by a learned offset $s_{c,p}$ using integer circular shifts S_a:

$$\widetilde{\mathbf{z}}_{c,p} = (1-\eta_{c,p})S_{a_{c,p}}(\mathbf{z}_{c,p}) + \eta_{c,p}S_{a_{c,p}+1}(\mathbf{z}_{c,p}), \quad a_{c,p} = \lfloor s_{c,p} \rfloor, \quad \eta_{c,p} = s_{c,p} - a_{c,p}.$$

We use two groups of A heads under the same shift-conditioned construction, with zero shifts for one group and learned shifts for the other. Queries follow these shifts, while keys and values remain unshifted, giving $M = 2A$ heads with head width $d_h = d/A$. We omit the block index ℓ below for clarity. We write $\mathbf{q}_{c,a}, \mathbf{k}_{c,a}, \mathbf{v}_{c,a} \in \mathbb{R}^{d_h}$ for the query, key, and value of variable c in head a.

For each head a, variable interaction compares variables c and c' and aggregates their values:

$$s^{\mathrm{var}}_{a,c,c'} = \mathrm{softmax}_{c'}\left(\frac{\mathbf{q}^\top_{c,a}\mathbf{k}_{c',a}}{\sqrt{d_h}}\right), \qquad \mathbf{u}_{c,a} = \sum_{c'=1}^{C} s^{\mathrm{var}}_{a,c,c'}\mathbf{v}_{c',a} \in \mathbb{R}^{d_h}. \tag{2}$$

For each variable c, subspace interaction compares heads a and b and, in the joint composition, aggregates their variable-interaction outputs:

$$s^{\mathrm{sub}}_{c,a,b} = \mathrm{softmax}_{b}\left(\frac{\mathbf{q}^\top_{c,a}\mathbf{k}_{c,b}}{\sqrt{d_h}}\right), \qquad \mathbf{o}_{c,a} = \sum_{b=1}^{M} s^{\mathrm{sub}}_{c,a,b}\mathbf{u}_{c,b} \in \mathbb{R}^{d_h}. \tag{3}$$

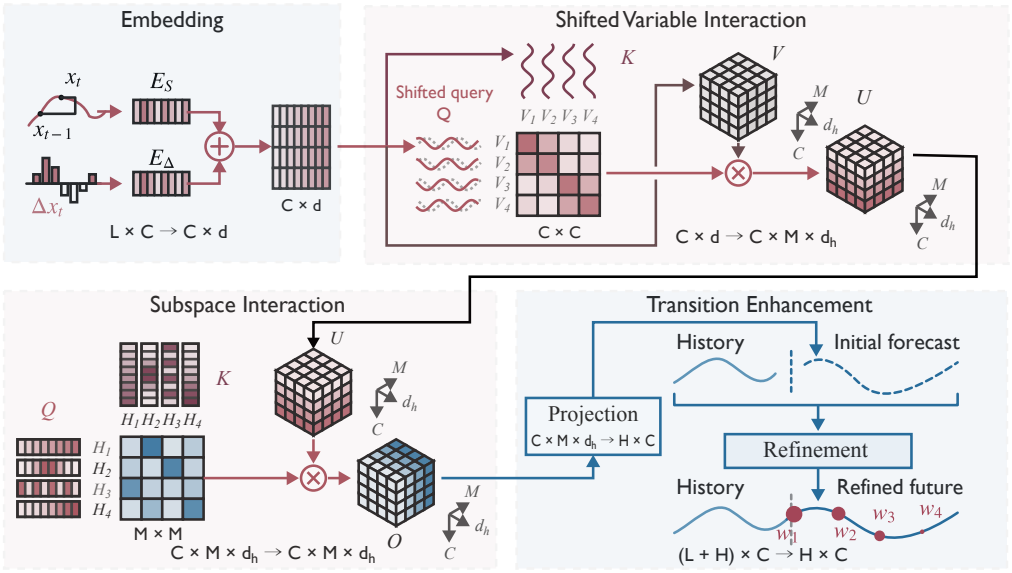


Figure 2: Overview of STEP. Historical series and their adjacent differences are encoded together. Shifted features generate queries, while unshifted features generate keys and values. Variable interaction aggregates information across variables, while subspace interaction combines different projection subspaces. Their output updates the historical representation, which is projected to the initial forecast. Joint refinement uses this forecast and observed history to produce the final prediction. Loss weights w_h decrease with the horizon to emphasize forecast-start errors. Labels V_i and H_a index variables and heads; feature curves, attention weights, and slice shading are schematic.

Softmax normalizes over c' for variable interaction and over b for subspace interaction. The vectors $\mathbf{u}_{c,a}$ and $\mathbf{o}_{c,a}$ are stacked along the variable and head axes to form $\mathbf{U}, \mathbf{O} \in \mathbb{R}^{C\times M\times d_h}$, respectively; thus $\mathbf{U}_{c,a,:} = \mathbf{u}_{c,a}$ and $\mathbf{O}_{c,a,:} = \mathbf{o}_{c,a}$.

The interaction output $\mathbf{O}$ updates the historical representation that produced its queries, keys, and values. Block ℓ computes $\mathbf{O}^{(\ell)}$ from $\mathbf{Z}^{(\ell)}$ using Eqs. (2)–(3) and updates the representation as

$$\begin{aligned} \mathbf{R}_j &= \mathrm{LN}^{\mathrm{attn}}_{\ell,j}\left(\mathbf{Z}_{j-1} + D(\mathbf{O}^{(\ell)}_j W_{\ell,j} + b_{\ell,j})\right), \\ \mathbf{Z}_j &= \mathrm{LN}^{\mathrm{ff}}_{\ell,j}(\mathbf{R}_j + \mathrm{FFN}_{\ell,j}(\mathbf{R}_j)), \qquad j = 1, 2. \end{aligned} \tag{4}$$

Here $\mathbf{Z}_0 = \mathbf{Z}^{(\ell)}$ and $\mathbf{Z}_2 = \mathbf{Z}^{(\ell+1)}$. The original and shifted head groups $\mathbf{O}^{(\ell)}_1, \mathbf{O}^{(\ell)}_2$ are flattened and separately projected to width d. The two groups update the state successively. D, LN, and FFN denote dropout, layer normalization, and a two-layer feed-forward map with activation and dropout. Starting from the embedding $\mathbf{Z}^{(0)}$ in Eq. (1), B blocks produce $\mathbf{Z}^{(B)}$, which is projected to the initial forecast:

$$\widetilde{\mathbf{Y}} = \mathcal{N}^{-1}\left(\left[\mathcal{P}\left(\mathbf{Z}^{(B)}\right)\right]^{\top}\right) \in \mathbb{R}^{H\times C}, \tag{5}$$

We refer to $\widetilde{\mathbf{Y}}$ as the initial forecast below. Here, $\mathcal{P}$ maps each variable's d features to H forecast values. Transposing the $C \times H$ output and applying $\mathcal{N}^{-1}$ restores the forecast layout and input scale.

3.2 Strengthening the History-to-Future Transition

STEP jointly refines the initial forecast and observed history through a temporal linear map:

$$\widehat{\mathbf{Y}}_{:,c} = W_{\mathrm{ref}}\begin{bmatrix}\mathbf{X}_{:,c} \\ \widetilde{\mathbf{Y}}_{:,c}\end{bmatrix}, \qquad W_{\mathrm{ref}} \in \mathbb{R}^{H\times(L+H)}. \tag{6}$$

The same bias-free W_{ref} is shared across variables and applied independently to each variable, mixing temporal positions but not variables. The combined sequence places the last observation and

the forecast start next to each other. Each forecast position can therefore draw on both the observed trajectory and the proposed continuation, giving the final forecast direct access to historical levels and changes.

To strengthen the history-to-future transition, our transition-aware objective penalizes errors in both forecast values and adjacent changes, with greater weight near the forecast start.

Let $e_{h,c} = \widehat{Y}_{h,c} - Y_{h,c}$ denote the value error and $g_{h,c} = e_{h+1,c} - e_{h,c}$ the error in the predicted change between adjacent positions. The objective combines value and adjacent-change losses:

$$\mathcal{L} = \frac{1}{HC} \sum_{h=1}^{H} \sum_{c=1}^{C} w_h \ell_v(e_{h,c}) + \frac{\mu}{(H-1)C} \sum_{h=1}^{H-1} \sum_{c=1}^{C} w_h \ell_d(g_{h,c}), \tag{7}$$

where $\ell_v(e) = e^2 + |e|$ and $\ell_d(g) = g^2 + |g|$ combine squared and absolute error, and μ balances the adjacent-change term. The weights $w_h = \exp[-\lambda(h-1)]$, with $\lambda > 0$, emphasize the forecast start. Both stages are jointly trained using the final forecast.

4 Experiments

Experimental setup. We evaluate STEP on Electricity, Traffic, Weather, Solar, and the four ETT datasets (Zhou et al., 2021; Liu et al., 2024), covering 7 to 862 variables. We use Adam with validation-based early stopping and report mean squared error (MSE) and mean absolute error (MAE). Baselines span linear forecasting, attention, and multiscale mixing: DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), TimeMixer (Wang et al., 2024), Client (Gao et al., 2025), TQNet (Lin et al., 2025), and GTR (Cao et al., 2026). Further experimental settings are provided in Appendix A.3.

4.1 Analyzing the Role of the Forecast Start

We further examine how the forecast trajectory changes after the true forecast start becomes observed. Figure 3 shows three representative patterns: (a) level-dominated, where the updated forecast mainly shifts in level; (b) mixed level–shape, where both the forecast level and temporal evolution change; and (c) shape-dominated, where the main differences appear in later rises and falls. Appendix C provides a complementary controlled analysis of forecast-start correction across correction magnitudes and the full test sets of four datasets.

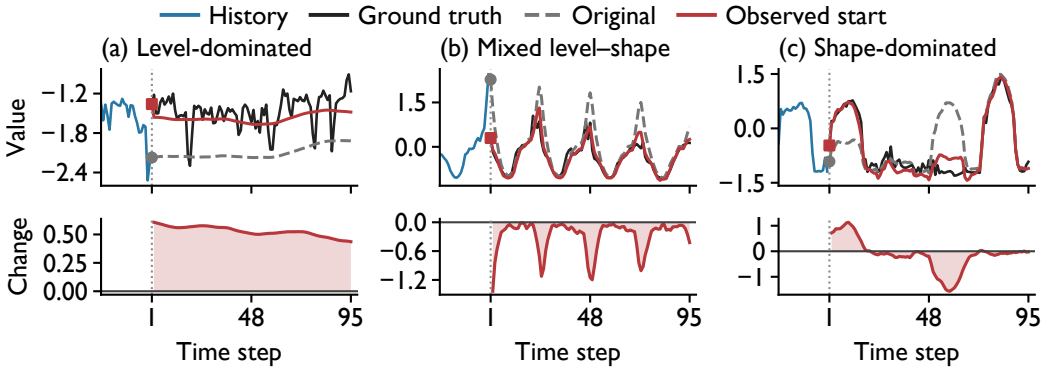


Figure 3: Representative changes in subsequent forecasts after the true forecast start becomes observed: (a) level-dominated, (b) mixed level–shape, and (c) shape-dominated. Top: the original forecast and the forecast obtained after providing the observed start to the same frozen model. Bottom: the difference between Observed start and Original over the aligned subsequent timestamps.

4.2 Long-Term Forecasting

We compare STEP with representative long-term forecasting baselines in Table 1. STEP achieves the lowest MSE at all reported horizons on Electricity, Traffic, and Weather. On Solar, STEP is tied

best at horizon 96, ranks second at 192, and third at 336 and 720. Overall, STEP shows strong and consistent performance across the evaluated datasets.

Table 1: Performance comparison with representative models. MSE is reported at input length 96 and prediction lengths 96, 192, 336, and 720. **Bold** and underline mark the best and second-best distinct values at three-decimal precision, including ties.

	Electricity				Traffic				Weather				Solar			
Method	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
DLinear (2023)	.195	.196	.209	.245	.649	.598	.605	.645	.198	.237	.283	.345	.285	.320	.353	.356
PatchTST (2023)	.167	.188	.204	.246	.447	.466	.482	.514	.178	.225	.278	.354	.225	.267	.290	.289
iTransformer (2024)	.148	.162	.178	.225	.398	.417	.433	.467	.177	.221	.278	.358	.200	.233	.248	.249
TimeMixer (2024)	.153	.166	.185	.225	.487	.473	.498	.506	.161	.208	.265	.339	.218	.222	.231	.243
Client (2025)	.142	.161	.173	.209	.434	.451	.472	.499	.165	.214	.271	.360	.197	.230	.248	.246
TQNet (2025)	.134	.154	.169	.201	.413	.432	.450	.486	.157	.206	.262	.344	**.176**	**.199**	.211	**.209**
GTR (2026)	.134	.153	.170	.208	.428	.440	.457	.494	.156	.204	.262	.342	.194	.202	**.203**	.217
STEP (Ours)	**.127**	**.146**	**.157**	**.183**	**.379**	**.409**	**.428**	**.461**	**.148**	**.196**	**.241**	**.314**	**.176**	.201	.224	.231

4.3 Enhancing Forecasting Backbones

We further examine two ways of enhancing existing forecasting models. Training-based integration transfers STEP to different backbones to improve their forecasting accuracy. Inference-time enhancement considers a different setting: whether a localized correction at the forecast start can improve the broader forecast while leaving the original backbone unchanged. Such a localized intervention improves the forecasting system without modifying the backbone architecture or learned parameters. Figure 4 illustrates both approaches.

(1) Training-based integration. To examine whether STEP can improve existing forecasting models, we integrate it into seven backbones while retaining their native forecasting components or replacing components with functionally similar STEP modules. STEP lowers MSE in 108 of 112 settings in Table 2, with one tie and three regressions: TQNet on Electricity at horizon 720, and GTR on Electricity at horizon 720 and Weather at horizon 192. The other five backbones improve in every setting. These results show that STEP can be applied to improve forecasting accuracy across diverse forecasting architectures. We further compare STEP with δ-Adapter, a recent approach for enhancing existing forecasting models, in Appendix E.3.

Table 2: Training-based STEP integration with $L = 96$ and $H \in \{96, 192, 336, 720\}$. Bold marks the lower MSE in each backbone pair.

	Electricity				Traffic				Weather				Solar			
Backbone	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
DLinear (2023)	.195	.196	.209	.245	.649	.598	.605	.645	.198	.237	.283	.345	.285	.320	.353	.356
+STEP	**.155**	**.168**	**.182**	**.224**	**.544**	**.536**	**.557**	**.597**	**.177**	**.218**	**.264**	**.330**	**.178**	**.194**	**.206**	**.210**
PatchTST (2023)	.167	.188	.204	.246	.447	.466	.482	.514	.178	.225	.278	.354	.225	.267	.290	.289
+STEP	**.134**	**.151**	**.165**	**.222**	**.388**	**.419**	**.432**	**.470**	**.170**	**.218**	**.269**	**.351**	**.189**	**.233**	**.241**	**.250**
iTransformer (2024)	.148	.162	.178	.225	.398	.417	.433	.467	.177	.221	.278	.358	.200	.233	.248	.249
+STEP	**.132**	**.147**	**.158**	**.189**	**.381**	**.408**	**.427**	**.460**	**.158**	**.199**	**.250**	**.319**	**.191**	**.221**	**.233**	**.240**
TimeMixer (2024)	.153	.166	.185	.225	.487	.473	.498	.506	.161	.208	.265	.339	.218	.222	.231	.243
+STEP	**.144**	**.153**	**.168**	**.220**	**.453**	**.458**	**.475**	**.493**	**.160**	**.207**	**.260**	**.337**	**.163**	**.215**	**.219**	**.230**
Client (2025)	.142	.161	.173	.209	.434	.451	.472	.499	.165	.214	.271	.360	.197	.230	.248	.246
+STEP	**.135**	**.152**	**.167**	**.199**	**.411**	**.435**	**.451**	**.481**	**.157**	**.205**	**.262**	**.340**	**.193**	**.228**	**.239**	**.240**
TQNet (2025)	.134	.154	.169	**.201**	.413	.432	.450	.486	.157	**.206**	.262	.344	.176	.199	.211	.209
+STEP	**.131**	**.151**	**.168**	.211	**.398**	**.422**	**.441**	**.476**	**.155**	**.206**	**.261**	**.340**	**.172**	**.193**	**.199**	**.206**
GTR (2026)	.134	.153	.170	**.208**	.428	.440	.457	.494	.156	**.204**	.262	.342	.194	.202	.203	.217
+STEP	**.131**	**.151**	**.166**	.209	**.420**	**.432**	**.451**	**.479**	**.154**	.206	**.261**	**.340**	**.170**	**.188**	**.196**	**.215**

(2) Inference-time enhancement. We next examine whether correcting the forecast start can improve subsequent predictions while keeping the backbone frozen. The backbone remains unchanged.

The supplied first prediction is taken from the backbone itself, STEP, or the ground truth. We therefore focus on Q4, the quartile with the highest forecast-start influence. Within Q4, STEP reduces subsequent MSE relative to the control in all 28 backbone–dataset combinations in Table 3. STEP also reduces subsequent MSE relative to Control in all 28 settings on the full test set, with larger gains in Q4 (Appendix J.1). For example, on Solar, STEP recovers 87% and 89% of the improvement obtained with the ground-truth start for TimeMixer and TQNet, respectively; on Traffic, the corresponding proportions are 78% for DLinear and 71% for TimeMixer. These results show that, within the matched rerun, improving the supplied forecast start can reduce subsequent forecasting error while leaving the backbone unchanged, suggesting that small and controlled interventions at the forecast start can provide a useful direction for enhancing existing forecasting systems.

Table 3: Inference-time forecast-start enhancement ($L = H = 96$), evaluated by MSE over steps 2 to 96. Control, STEP, and GT start supply the frozen backbone with its own prediction, STEP's prediction, or the ground-truth first future value, respectively. Bold and underline mark the best and second-best distinct values; darker blue denotes lower MSE within each triple. Results use Q4, the quartile with the highest forecast-start influence; full-test results appear in Appendix J.1.

	Electricity			Traffic			Weather			Solar		
Backbone	Control	STEP	GT start	Control	STEP	GT start	Control	STEP	GT start	Control	STEP	GT start
DLinear	.2127	<u>.2081</u>	**.2065**	.6623	<u>.6442</u>	**.6392**	.2706	<u>.2661</u>	**.2583**	.4102	<u>.3961</u>	**.3885**
PatchTST	.1974	<u>.1939</u>	**.1913**	.4689	<u>.4613</u>	**.4548**	.2296	<u>.2202</u>	**.2101**	.2669	<u>.2618</u>	**.2426**
iTransformer	.1832	<u>.1808</u>	**.1774**	.4368	<u>.4349</u>	**.4244**	.2336	<u>.2247</u>	**.2163**	.2881	<u>.2680</u>	**.2509**
TimeMixer	.1808	<u>.1779</u>	**.1756**	.4981	<u>.4862</u>	**.4813**	.1977	<u>.1949</u>	**.1864**	.2758	<u>.2367</u>	**.2306**
Client	.1747	<u>.1729</u>	**.1699**	.4686	<u>.4635</u>	**.4554**	.2153	<u>.2093</u>	**.1996**	.2719	<u>.2687</u>	**.2526**
TQNet	.1569	<u>.1561</u>	**.1528**	.4398	<u>.4370</u>	**.4278**	.2080	<u>.2046</u>	**.1946**	.2684	<u>.2490</u>	**.2467**
GTR	.1576	<u>.1563</u>	**.1534**	.4569	<u>.4518</u>	**.4451**	.2074	<u>.2040</u>	**.1943**	.2616	<u>.2590</u>	**.2530**

As an additional sanity check, the inference-time enhancement also achieves lower subsequent MSE than the untouched one-shot forecast in 25 of 28 Q4 backbone–dataset settings. Complete Q4 and full-test comparisons are provided in Appendix J.2.

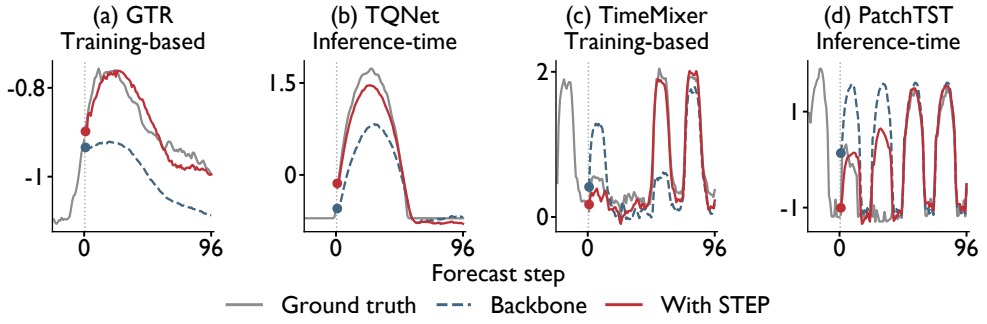


Figure 4: STEP enhancement ($L = H = 96$). (a,c) Training-based integration. (b,d) Inference-time enhancement, where matched reruns differ only in the supplied forecast-start value while the backbone remains frozen.

4.4 Ablation Study

(1) Capacity-matched ablation. To separate STEP's architectural gains from the effect of additional parameters, we compare it with an FFN control matched to within 0.1% in parameter count. Details are provided in Appendix F.1. We measure the MSE improvement of STEP over this control at steps 2 to 96. To examine the role of the forecast start, we group forecasting instances into quartiles according to forecast-start influence, from Q1 with the lowest influence to Q4 with the highest.

Figure 5(a) shows that STEP improves over the matched control on all four datasets, with all 95% confidence intervals above zero. We call $G_{Q4} - G_{Q1}$ the start advantage. Figure 5(b) shows a positive start advantage across all four datasets, with Q4 gains reaching 1.3 to 2.0 times those in Q1 and all confidence intervals above zero. STEP thus provides greater benefits when forecast-start influence is higher. Figure 5(c) shows representative forecast examples.

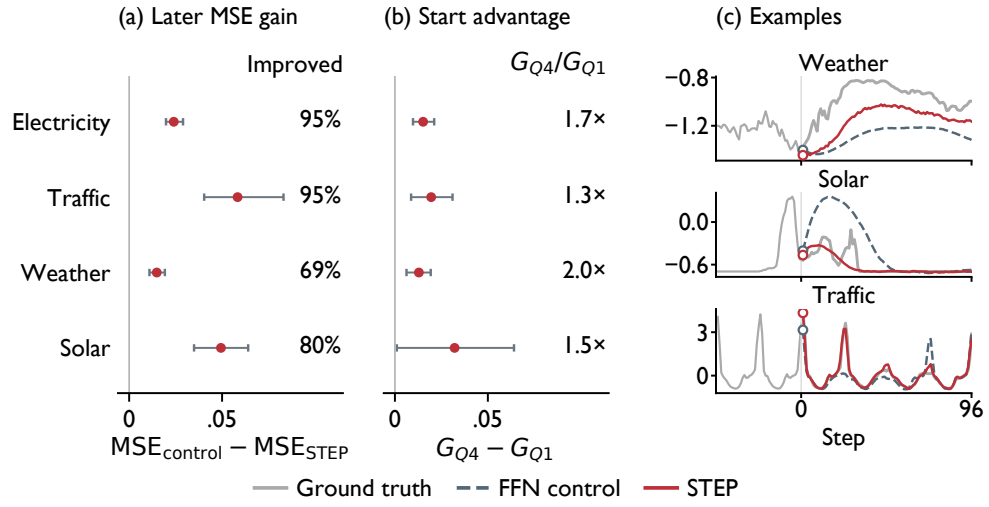


Figure 5: Capacity-matched ablation. (a) MSE improvement of STEP over the FFN control at steps 2 to 96; labels show the percentage of forecasts improved. (b) Difference in STEP gain between Q4 and Q1; labels show the gain ratio G_{Q4}/G_{Q1}. Q1/Q4 are the lowest/highest forecast-start influence quartiles. Error bars in (a) and (b) show 95% confidence intervals. (c) Representative forecasts on Weather, Solar, and Traffic.

(2) Component ablation. We further examine the two main parts of STEP: improving the initial forecast and strengthening the history-to-future transition. For the initial forecast, we separately ablate the shift, shifted variable interaction, and subspace interaction. For the transition design, we separately ablate joint refinement and the STEP objective, and further isolate the contributions of the change term and temporal weighting within the objective (Table 4).

Table 4: Ablation results ($L = H = 96$). Bold/underline: best/second-best distinct values at reported precision. 1st-place count: number of first-place entries across eight metrics, including ties.

Aspect	Variant	Electricity		Traffic		Weather		Solar		1st-place count
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
Overall	STEP	**.127**	**.220**	**.379**	**.248**	**.148**	**.194**	**.176**	**.199**	**8**
	w/o all modules	.149	.236	.440	.275	.165	.206	.224	.250	0
Initial forecast	w/o shift	.130	.223	.386	.251	.156	.201	.177	**.199**	1
	w/o shifted variable interaction	.146	.233	.425	.259	.161	.210	.203	.243	0
	w/o subspace interaction	.130	.223	.384	.251	.150	.198	.186	.227	0
Transition	w/o refinement	.129	.221	.392	.254	.153	.195	.191	.221	0
	w/o STEP objective	.132	.230	.385	.267	.157	.213	.198	.271	0
	STEP objective w/o change term	.130	.223	.382	.250	.152	.202	.178	.202	0
	STEP objective w/o weighting	.129	.223	.381	**.248**	.151	.197	.185	.203	1

STEP achieves the best MSE on all four datasets, including ties at the reported precision (Table 4). Separately removing components of the initial-forecast or transition design degrades performance, while the objective ablations further support the contributions of both adjacent-change supervision and temporal weighting.

4.5 Model Mechanism Analysis

To examine the roles of STEP's two design parts, Figure 6 visualizes learned shifts and refinement weights and compares the forecasting effects of the initial forecast and transition designs. In panel (a), learned shifts spread from zero to nearby offsets with distributions that vary across layers, while refinement weights concentrate around the junction between history and forecast and draw information from both sides. Panel (b) compares STEP with separately trained, parameter-matched FFN controls on Electricity with $H = 720$. The initial forecast design reduces MSE by 2.5% at the first step and by 11.0% over later positions. The overall transition design reduces MSE by 16.9% and

6.2%, respectively. This pattern is consistent with complementary roles: the initial forecast design shows broader gains over later positions, while the transition design shows a stronger effect at the forecast start. Appendix G.2 defines the controls and confidence intervals.

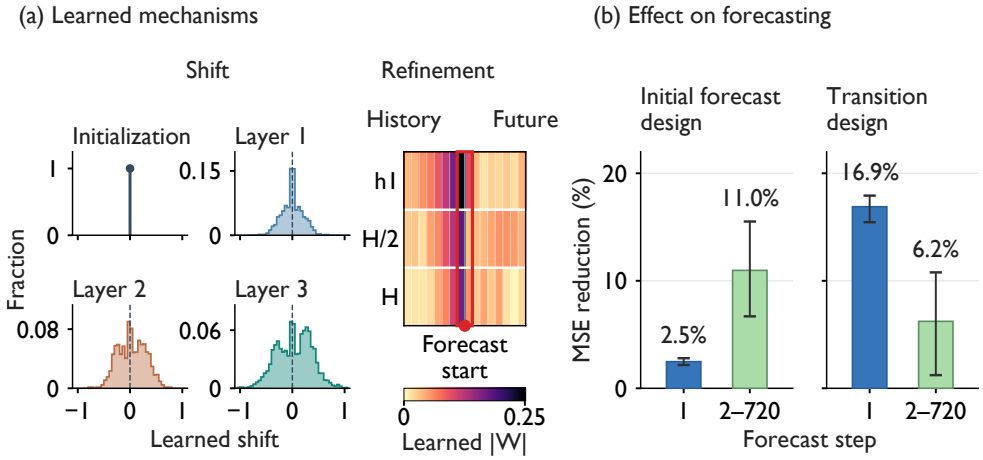


Figure 6: Learned mechanisms and forecasting effects on Electricity ($L = 96$). (a) Shift distributions and absolute refinement weights ($H = 96$); red markers locate the history-to-forecast junction. (b) Relative MSE reductions at step 1 and steps 2–720 ($H = 720$) against parameter-matched FFN controls for the initial forecast interactions or joint refinement; the latter also uses MSE loss. Error bars show 95% paired block bootstrap confidence intervals.

4.6 Historical Context Extension

With $H = 96$, we vary L from 96 to 720 on Weather and Traffic. Traffic MSE decreases from .379 to .341, while Weather performs best at $L = 336$. STEP is best or tied best at all Weather settings and at $L = 96, 720$ on Traffic; VPNet leads at $L = 192, 336$.

Table 5: Performance across history lengths $L \in \{96, 192, 336, 720\}$ with forecast horizon $H = 96$. Bold and underline indicate the best and second-best distinct values.

	Weather (21 variables)								Traffic (862 variables)							
	96		192		336		720		96		192		336		720	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DLinear	.198	.257	.185	.246	.174	.234	.169	.228	.649	.397	.485	.336	.410	.282	.389	.279
PatchTST	.178	.219	.162	.208	.157	.207	.154	.210	.447	.284	.394	.272	.367	.251	.369	.255
iTransformer	.177	.218	.165	.210	.162	.210	.180	.232	.398	.271	.365	.259	.367	.278	.349	.256
TimeMixer	.161	.208	.157	.205	.175	.224	.153	.208	.487	.305	.404	.278	.371	.256	.381	.282
Client	.165	.210	.160	.209	.165	.216	.152	.206	.434	.292	.381	.272	.365	.264	.373	.272
TQNet	.157	.201	.154	.201	.155	.205	.155	.209	.413	.260	.378	.255	.365	.253	.382	.269
VPNet	.157	**.192**	.145	**.182**	.143	**.179**	**.144**	**.181**	.384	.258	**.348**	.245	**.340**	.244	.343	.248
STEP (Ours)	**.148**	.194	**.143**	.189	**.140**	.186	**.144**	.197	**.379**	**.248**	.351	**.236**	.342	**.234**	**.341**	**.235**

5 Conclusion

We revisit long-term time series forecasting from the forecast start and propose STEP to improve how forecasting begins. STEP models adjacent temporal evolution to improve the initial forecast and strengthens the history-to-future transition. Across eight datasets, STEP achieves competitive performance, while training-based integration improves diverse backbones in 96.4% of evaluated settings. With the backbone frozen, improving the supplied forecast start can also reduce subsequent error, showing that localized interventions can improve the broader forecast while preserving the original backbone architecture and its learned parameters throughout inference.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, preparation of scientific figures, and writing utility and automation scripts. All AI-assisted outputs were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Fanpu Cao, Lu Dai, Jindong Han, and Hui Xiong. Enhancing multivariate time series forecasting with global temporal retrieval. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.10847.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *International Conference on Learning Representations*, 2024.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *International Conference on Learning Representations*, 2024.

Jiaxin Gao, Wenbo Hu, Dongxiao Zhang, and Yuntian Chen. Client: Cross-variable linear integrated enhanced transformer for multivariate long-term time series forecasting. *AI Open*, 6: 93–107, 2025. doi: 10.1016/j.aiopen.2025.06.001. URL https://www.sciencedirect.com/science/article/pii/S2666651025000099.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3rd edition, 2021. URL https://otexts.com/fpp3/. Accessed September 15, 2026.

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

Benhan Li, Shengdong Du, and Tianrui Li. A differential attention fusion model based on transformer for time series forecasting. *arXiv preprint arXiv:2202.11402*, 2022. URL https://arxiv.org/abs/2202.11402.

Zhe Li, Xuanxuan Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. MTS-Mixers: Multivariate time series forecasting via factorized temporal and channel mixing. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2025. doi: 10.1109/IJCNN64981.2025.11229402.

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 (ICLR)*, 2026. URL https://openreview.net/forum?id=syfWdclGE1.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37797–37814, 2025. URL https://proceedings.mlr.press/v267/lin25e.html.

Kun Liu, Renjun Jia, Ruifeng Yang, Xirui Zeng, Yuqi Liang, and Cen Chen. Are global dependencies necessary? scalable time series forecasting via local cross-variate modeling. In *International Conference on Learning Representations*, pp. 102794–102817, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a832b65da26c4867a8581cb13d46ad6d-Paper-Conference.pdf.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. SCINet: Time series modeling and forecasting with sample convolution and interaction. In *Advances in Neural Information Processing Systems*, volume 35, 2022a.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-Tao Xia. TimeBridge: Non-stationarity matters for long-term time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39815–39840. PMLR, 2025. URL https://proceedings.mlr.press/v267/liu25cb.html.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022b.

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

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. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Mohammad Amin Shabani, Amir H. Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In *International Conference on Learning Representations*, 2023.

Li Shen, Yuning Wei, and Yangzhu Wang. GBT: Two-stage transformer framework for non-stationary time series forecasting. *Neural Networks*, 165:953–970, 2023. doi: 10.1016/j.neunet.2023.06.044.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *International Conference on Learning Representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, 2025.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. ETSformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022. URL https://arxiv.org/abs/2202.01381.

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*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Feng Xiong, Zongxia Xie, Yanru Sun, Haoyu Wang, and Jianhong Lin. SEED: Spectral entropy-guided evaluation of spatial-temporal dependencies for multivariate time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(32):27153–27161, 2026. doi: 10.1609/aaai.v40i32.39930.

Qi Xiong, Kai Tang, Minbo Ma, Ji Zhang, Jie Xu, and Tianrui Li. Modeling temporal dependencies within the target for long-term time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 37(12):7300–7314, 2025. doi: 10.1109/TKDE.2025.3609415.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. FiLM: Frequency improved legendre memory model for long-term time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 12677–12690, 2022a.

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*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286, 2022b. URL https://proceedings.mlr.press/v162/zhou22g.html.

A Experimental Settings

A.1 Datasets and Preprocessing

Dataset overview. We use eight multivariate forecasting benchmarks widely adopted in previous studies (Zhou et al., 2021; Liu et al., 2024). The four ETT datasets record transformer operating conditions; ETTh1 and ETTh2 are sampled hourly, while ETTm1 and ETTm2 are sampled every 15 minutes. Electricity contains electricity consumption from 321 clients, Traffic records road occupancy from 862 sensors in the San Francisco Bay Area, Weather contains 21 meteorological variables from a weather station in Germany, and Solar records power generation from 137 photovoltaic plants. Table 6 summarizes the dataset characteristics.

Table 6: Statistics of the eight forecasting benchmarks. Observation count gives the number of time points in each complete series, and Variables gives the number of jointly forecast variables.

Dataset	Observation count	Variables	Sampling interval	Domain
ETTh1	17,420	7	1 hour	Transformer operation
ETTh2	17,420	7	1 hour	Transformer operation
ETTm1	69,680	7	15 minutes	Transformer operation
ETTm2	69,680	7	15 minutes	Transformer operation
Electricity	26,304	321	1 hour	Electricity consumption
Traffic	17,544	862	1 hour	Road occupancy
Weather	52,696	21	10 minutes	Meteorology
Solar	52,560	137	10 minutes	Solar power generation

Data splits and preprocessing. All datasets are split chronologically. For ETT, we follow the standard 12/4/4 month training, validation, and test split, with each month represented by 30 days in the data loader. Electricity, Traffic, Weather, and Solar use 70%/10%/20% training, validation, and test splits. Each variable is standardized using the training set mean and standard deviation, which are then applied to the validation and test sets.

A.2 Training and Evaluation Settings

Training and checkpoint selection. We train STEP separately for each dataset and forecast horizon using the objective in Eq. (7). We use Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$, zero weight decay, and single precision. Training configurations are specified for each setting in Section A.3. Early stopping and checkpoint selection are based on validation loss, and the checkpoint with the lowest validation loss is used for evaluation.

Software and hardware. STEP is implemented in Python with PyTorch. Experiments are conducted on an NVIDIA GeForce RTX 4090 with 24 GB of GPU memory using Python 3.10.12, PyTorch 2.8.0, and CUDA 12.8.

A.3 Model and Training Configurations

Tables 7 to 9 report the model and training configurations used for the eight datasets and four forecast horizons with $L = 96$. Model configurations are selected using validation performance. We consider $B \in \{1, 2, 3, 4\}$ encoder blocks, feature width $d \in \{32, 64, 128, 256, 512\}$, feed-forward width $d_{\text{ff}} \in \{32, 64, 128, 256, 512, 1024, 2048\}$, and highest historical difference order $K \in \{0, 1, 2, 3\}$. For each block, the segment width m is selected from $\{4, 8, 16, 32, 64, 128\}$, subject to d/m being an integer, giving $P = d/m$ segments per token. We fix $A = 8$ heads in each original or shifted group for all settings, giving $M = 2A$ heads in total with head width $d_h = d/A$. The segment widths in Table 7 are listed in block order, and $K = 0$ uses historical values only.

Table 7: STEP model configurations for the eight datasets with $L = 96$. Segment widths are listed in block order.

Dataset	B	d	d_{ff}	A	Segment widths m	K
Electricity	3	512	2048	8	64/128/64	1
Traffic	4	512	512	8	32/32/32/32	3
Weather	3	32	32	8	8/8/8	1
Solar	2	512	2048	8	32/32	3
ETTh1	2	128	64	8	4/4	1
ETTh2	2	128	128	8	8/8	0
ETTm1	2	32	32	8	4/4	1
ETTm2	2	32	64	8	4/4	1

Table 8: STEP training configurations for $L = 96$ and forecast horizons $H \in \{96, 192, 336, 720\}$. Epochs denotes the maximum training budget, and learning rates are reported in units of 10^{-4}.

Dataset	Optimizer	L	Batch	Epochs	LR (10^{-4})			
					$H = 96$	192	336	720
Electricity	Adam	96	16	10	5	5	5	5
Traffic	Adam	96	16	10	5	5	5	5
Weather	Adam	96	32	10	8	8	8	8
Solar	Adam	96	16	10	1	1	1	2
ETTh1	Adam	96	16	30	20	18	18	15
ETTh2	Adam	96	32	30	1	1	3	2
ETTm1	Adam	96	16	30	20	20	20	15
ETTm2	Adam	96	16	30	15	15	14	23

The coefficient μ controls the contribution of the adjacent change loss, while λ controls the temporal decay in Eq. (7). We select $\mu \in \{0.5, 1, 2\}$ and $\lambda \in \{0.01, 0.02\}$ using validation performance. The selected pair is shared across all four forecast horizons for each dataset. Table 9 reports the values used in our experiments.

Table 9: Dataset-specific loss coefficients used in Eq. (7), shared across all four forecast horizons.

Dataset	μ	λ
Electricity	0.5	0.01
Traffic	0.5	0.01
Weather	2	0.01
Solar	1	0.01
ETTh1	2	0.02
ETTh2	0.5	0.01
ETTm1	2	0.02
ETTm2	2	0.02

B STEP IMPLEMENTATION

B.1 IMPLEMENTATION DETAILS

This section provides implementation details for the operations introduced in Section 3 and Figure 2. Shapes refer to a single input window, with the batch dimension omitted.

B.1.1 Difference Embedding

For the instance normalized history $\bar{\mathbf{X}}$, we implement the length-preserving differences in Eq. (1) recursively. Let $\mathbf{D}^{(0)} = \bar{\mathbf{X}}$, and for $k = 1, \ldots, K$,

$$D_{t,c}^{(k)} = \begin{cases} D_{1,c}^{(k-1)}, & t = 1, \\ D_{t,c}^{(k-1)} - D_{t-1,c}^{(k-1)}, & 2 \le t \le L. \end{cases} \tag{8}$$

The first entry is retained to preserve the sequence length. Each $E_k : \mathbb{R}^L \to \mathbb{R}^d$ is a learned linear projection shared across variables and distinct across difference orders. The embedding in Eq. (1) is implemented as

$$\mathbf{Z}^{(0)} = \sum_{k=0}^{K} E_k\Big((\mathbf{D}^{(k)})^\top\Big).$$

Setting $K = 0$ uses historical values only.

B.1.2 Shift and Projection

For a block input $\mathbf{Z} \in \mathbb{R}^{C \times d}$, each variable representation is divided into P contiguous segments of width $m = d/P$. For segment $\mathbf{z}_{c,p} \in \mathbb{R}^m$, a learned shift $s_{c,p}$ is applied using linear interpolation:

$$\widetilde{z}_{c,p,r} = (1 - \eta_{c,p}) z_{c,p,(r+a_{c,p}) \bmod m} + \eta_{c,p} z_{c,p,(r+a_{c,p}+1) \bmod m}, \tag{9}$$

where $r \in \{0, \ldots, m-1\}$, $a_{c,p} = \lfloor s_{c,p} \rfloor$, and $\eta_{c,p} = s_{c,p} - a_{c,p}$. Concatenating the shifted segments gives $\widetilde{\mathbf{Z}} \in \mathbb{R}^{C \times d}$.

We use two groups of A heads with width $d_h = d/A$, both following the same shift-conditioned query construction. Let

$$\mathbf{Z}^o = \mathrm{Shift}(\mathbf{Z}; 0) = \mathbf{Z}, \qquad \mathbf{Z}^s = \mathrm{Shift}(\mathbf{Z}; s) = \widetilde{\mathbf{Z}},$$

where the original group uses zero offsets and the shifted group uses learned offsets. Each group generates queries from its corresponding shifted representation, while keys and values are generated from the unshifted representation $\mathbf{Z}$:

$$\begin{aligned} (\mathbf{Q}^o, \mathbf{K}^o, \mathbf{V}^o) &= (\mathbf{Z}^o W_Q^o, \mathbf{Z} W_K^o, \mathbf{Z} W_V^o), \\ (\mathbf{Q}^s, \mathbf{K}^s, \mathbf{V}^s) &= (\mathbf{Z}^s W_Q^s, \mathbf{Z} W_K^s, \mathbf{Z} W_V^s). \end{aligned} \tag{10}$$

Each $C \times d$ projection is reshaped to $C \times A \times d_h$. The groups use separate parameters and are concatenated along the head dimension ($M = 2A$), yielding $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{C \times 2A \times d_h}$. Thus, the original group is the zero-shift instance of the same construction used by the learned-shift group, as abstracted by the Shift–Projection path in Figure 2.

B.1.3 Variable and Subspace Interactions

The variable interaction in Eq. (2) produces a score tensor $\mathbf{S}^{\mathrm{var}} \in \mathbb{R}^{M \times C \times C}$ and output $\mathbf{U} \in \mathbb{R}^{C \times M \times d_h}$. The subspace interaction in Eq. (3) then operates across heads within each variable, producing $\mathbf{S}^{\mathrm{sub}} \in \mathbb{R}^{C \times M \times M}$ and $\mathbf{O} \in \mathbb{R}^{C \times M \times d_h}$.

The output $\mathbf{O}$ is split into the zero-shift and learned-shift head groups. Each group has shape $C \times A \times d_h$ and is flattened to $C \times d$, followed by a separate output projection before the successive residual updates in Eq. (4). Each feed-forward layer follows $C \times d \to C \times d_{\mathrm{ff}} \to C \times d$.

B.1.4 Forecast Refinement

After B encoder blocks, Eq. (5) produces the initial forecast $\widetilde{\mathbf{Y}} \in \mathbb{R}^{H \times C}$. Joint refinement follows Eq. (6), concatenating the observed history and initial forecast along the temporal dimension to form $[\mathbf{X}; \widetilde{\mathbf{Y}}] \in \mathbb{R}^{(L+H) \times C}$. The shared temporal map $W_{\mathrm{ref}} \in \mathbb{R}^{H \times (L+H)}$ is then applied independently to each variable.

B.1.5 Training Objective

We implement the objective in Eq. (7) using

$$e_{h,c} = \widehat{Y}_{h,c} - Y_{h,c}, \qquad g_{h,c} = e_{h+1,c} - e_{h,c}.$$

The temporal weights are $w_h = \exp[-\lambda(h-1)]$, and the coefficient μ controls the contribution of the adjacent change loss. The values of μ and λ are given in Table 9. The loss is averaged over the training batch.

B.2 Forward and Training Procedure

Algorithm 1 summarizes the forward computation and training procedure.

Algorithm 1 Forward computation and training of STEP

Require: Training set $\mathcal{D}_{\mathrm{tr}}$, validation set $\mathcal{D}_{\mathrm{val}}$, parameters θ

function FORWARD($\mathbf{X}; \theta$)
  $\bar{\mathbf{X}} \leftarrow \mathcal{N}(\mathbf{X})$
  $\mathbf{Z}^{(0)} \leftarrow \sum_{k=0}^{K} E_k((\Delta^k \bar{\mathbf{X}})^\top)$ ▷ Eq. (1)
  for $\ell = 0, \ldots, B-1$ **do**
    $\widetilde{\mathbf{Z}}^{(\ell)} \leftarrow \mathrm{Shift}_\ell(\mathbf{Z}^{(\ell)})$ ▷ Eq. (9)
    $(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \leftarrow \mathrm{Project}_\ell(\mathbf{Z}^{(\ell)}, \widetilde{\mathbf{Z}}^{(\ell)})$ ▷ Eq. (10)
    $\mathbf{U} \leftarrow \mathrm{VariableInteraction}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ ▷ Eq. (2)
    $\mathbf{O} \leftarrow \mathrm{SubspaceInteraction}(\mathbf{Q}, \mathbf{K}, \mathbf{U})$ ▷ Eq. (3)
    $\mathbf{Z}^{(\ell+1)} \leftarrow \mathrm{Update}_\ell(\mathbf{Z}^{(\ell)}, \mathbf{O})$ ▷ Eq. (4)
  end for
  $\widetilde{\mathbf{Y}} \leftarrow \mathcal{N}^{-1}([\mathcal{P}(\mathbf{Z}^{(B)})]^\top)$ ▷ Eq. (5)
  $\widehat{\mathbf{Y}} \leftarrow \mathrm{Refine}(\mathbf{X}, \widetilde{\mathbf{Y}})$ ▷ Eq. (6)
  return $\widehat{\mathbf{Y}}$
end function
$\theta^* \leftarrow \theta$
repeat
  for training batch $(\mathbf{X}, \mathbf{Y}) \in \mathcal{D}_{\mathrm{tr}}$ **do**
    $\widehat{\mathbf{Y}} \leftarrow \mathrm{FORWARD}(\mathbf{X}; \theta)$
    $\theta \leftarrow \mathrm{Adam}(\theta, \nabla_\theta \mathcal{L}(\widehat{\mathbf{Y}}, \mathbf{Y}))$ ▷ Eq. (7)
  end for
  if validation loss on $\mathcal{D}_{\mathrm{val}}$ improves **then** $\theta^* \leftarrow \theta$
until early stopping or the maximum epoch is reached
return θ^*

C Forecast Start Correction Analysis

To examine whether subsequent forecasting is specifically sensitive to the accuracy of the supplied forecast start, we conduct a controlled analysis that removes unrelated variation across conditions. Unless otherwise stated, the analyses in this section use frozen STEP variants with $L = H = 96$.

Relation to Figure 1. Figure 1 uses the untouched one-shot forecast as the visual reference, whereas the controlled analysis here uses a matched rerun for both conditions. This removes variation caused by the shifted input window itself, so Control and Corrected start differ only in the supplied forecast-start value. The percentage reductions are therefore not expected to match those in Figure 1.

Controlled correction protocol. Direct multi-horizon forecasters generate the entire future horizon in a single forward pass. Unlike recursive forecasting, the first prediction is not fed back into the model to generate the remaining predictions. Consequently, replacing the first output after forecasting changes only that output and leaves the rest of the forecast unchanged. We therefore use a matched rerun to isolate how the supplied forecast-start value changes subsequent predictions while keeping the model and all other forecasting conditions fixed.

For a history $X = [x_1, \ldots, x_L]$, we construct

$$X^+(a) = [x_2, \ldots, x_L, a],$$

where a is the supplied forecast-start value, and denote the corresponding subsequent forecast by

$$F_f(a; X) = f(X^+(a))_{1:H-1}.$$

The same input construction, timestamps, model parameters, and forecasting procedure are used for all conditions; only a changes. The control uses $a_{\text{ctrl}} = f(X)_1$, whereas the corrected condition uses $a_{\text{gt}} = Y_1$. We quantify the resulting response through changes in the subsequent forecast and its prediction error. The supplied forecast-start point itself is excluded from evaluation, so the reported changes reflect only the subsequent forecast. For each selected variable in the cumulative-error plots, cumulative error at original forecast position h is $C(h) = \sum_{t=2}^{h}(\hat{y}_t - y_t)^2$, where $\hat{y}_t$ and y_t are the aligned prediction and ground truth at position t.

Figure 7 shows this controlled comparison across the eight datasets. Under the matched rerun, correcting the supplied forecast-start value can substantially reduce subsequent cumulative error in these examples.

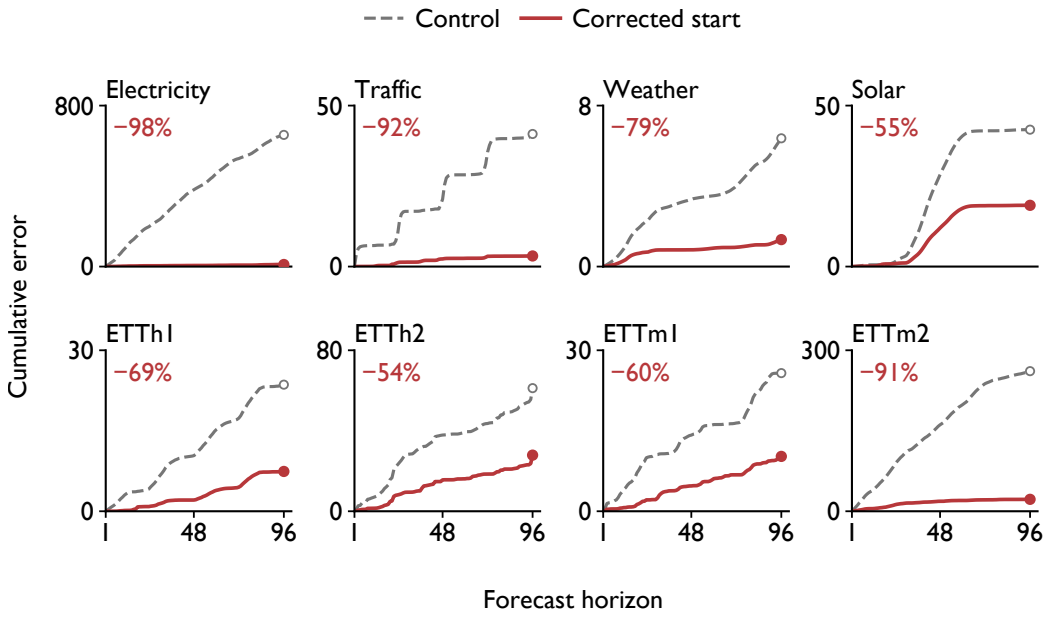


Figure 7: Controlled forecast-start correction across eight datasets. Control supplies the model's own first prediction at the forecast start, whereas Corrected start supplies the corresponding ground-truth value under the same matched rerun. Cumulative error excludes the supplied forecast-start point; percentages denote the reduction relative to Control. The horizontal axis follows the original forecast timestamps; the supplied start corresponds to position 1 and is excluded from cumulative error.

Correction magnitude. We further examine whether the response to forecast-start correction varies systematically with its magnitude and direction. For each test instance, let a denote the model's original first prediction and $\delta = Y_1 - a$. We supply

$$a(q) = a + q\delta,$$

where $q = 0$ retains the original prediction, $q = 1$ gives the ground-truth start, and $q < 0$ moves the supplied value in the opposite direction. The frozen model and all other forecasting conditions remain unchanged for every q. We evaluate the response over the full test sets of Electricity, Traffic, Weather, and Solar, using MSE over forecast positions 2–96.

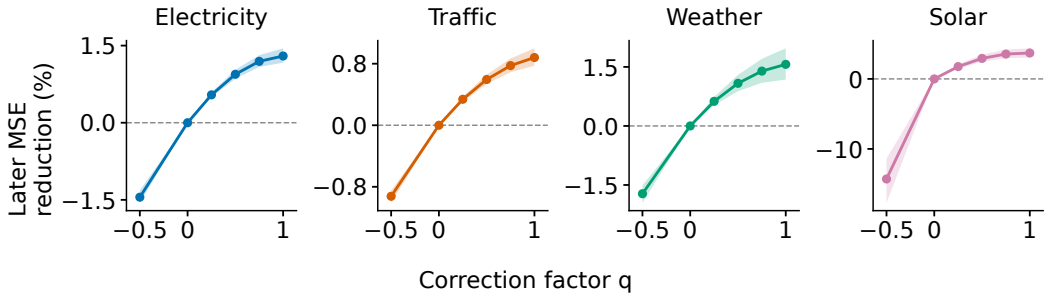


Figure 8: Forecast-start intervention response over the full test sets of four datasets. For each test instance, the supplied forecast start is varied as $a(q) = a + q(Y_1 - a)$, where $q = 0$ retains the model prediction and $q = 1$ gives the ground-truth start. Later MSE is evaluated over forecast positions 2–96 using the same frozen model and forecasting protocol for all q. Curves show $100[\mathrm{MSE}(0) - \mathrm{MSE}(q)]/\mathrm{MSE}(0)$, where MSE averages over all test windows, variables, and positions 2–96. Shading gives 95% percentile intervals from 10,000 paired circular-block bootstrap samples of test windows (block length 168), using the same sampled indices for all q.

As shown in Fig. 8, across all four datasets, later forecasting error decreases consistently as the supplied start moves from the model prediction toward the ground truth, while perturbing the start in the opposite direction ($q = -0.5$) increases error. At $q = 1$, later MSE is reduced by 1.30% on Electricity, 0.88% on Traffic, 1.57% on Weather, and 3.70% on Solar, with all 95% confidence intervals above zero. The corresponding fractions of improved test instances are 84.8%, 77.0%, 61.8%, and 60.4%, respectively. These results show a consistent graded response to the magnitude and direction of forecast-start correction across the full test distributions.

D Forecasting Results

D.1 Complete Forecasting Results

Table 10 reports the complete MSE and MAE results on all eight datasets with $L = 96$ and forecast horizons $H \in \{96, 192, 336, 720\}$. Bold and underline indicate the best and second-best distinct values at the reported precision.

Table 10: Complete forecasting results on eight datasets with $L = 96$. MSE and MAE are reported for forecast horizons $H \in \{96, 192, 336, 720\}$. Bold and underline indicate the best and second-best distinct values at three decimal precision, including ties. iTrans denotes iTransformer.

Dataset	H	Metric	STEP	GTR	TQNet	Client	iTrans	PatchTST	TimeMixer	DLinear
Electricity	96	MSE	**0.127**	0.134	0.134	0.142	0.148	0.167	0.153	0.195
		MAE	**0.220**	0.229	0.230	0.238	0.239	0.252	0.244	0.277
	192	MSE	**0.146**	0.153	0.154	0.161	0.162	0.188	0.166	0.196
		MAE	**0.239**	0.246	0.247	0.254	0.253	0.274	0.256	0.285
	336	MSE	**0.157**	0.170	0.169	0.173	0.178	0.204	0.185	0.209
		MAE	**0.254**	0.264	0.264	0.267	0.269	0.293	0.277	0.301
	720	MSE	**0.183**	0.208	0.201	0.209	0.225	0.246	0.225	0.245
		MAE	**0.284**	0.298	0.294	0.299	0.317	0.324	0.310	0.333
Traffic	96	MSE	**0.379**	0.428	0.413	0.434	0.398	0.447	0.487	0.649
		MAE	**0.248**	0.264	0.260	0.292	0.271	0.284	0.305	0.397
	192	MSE	**0.409**	0.440	0.432	0.451	0.417	0.466	0.473	0.598
		MAE	**0.260**	0.276	0.271	0.298	0.276	0.296	0.296	0.370
	336	MSE	**0.428**	0.457	0.450	0.472	0.433	0.482	0.498	0.605
		MAE	**0.269**	0.288	0.277	0.305	0.283	0.304	0.296	0.373
	720	MSE	**0.461**	0.494	0.486	0.499	0.467	0.514	0.506	0.645
		MAE	**0.289**	0.301	0.295	0.321	0.302	0.322	0.313	0.394

Table 10 (continued)

Dataset	H	Metric	STEP	GTR	TQNet	Client	iTrans	PatchTST	TimeMixer	DLinear
Weather	96	MSE	**0.148**	0.156	0.157	0.165	0.177	0.178	0.161	0.198
		MAE	**0.194**	0.201	0.201	0.210	0.218	0.219	0.208	0.257
	192	MSE	**0.196**	0.204	0.206	0.214	0.221	0.225	0.208	0.237
		MAE	0.246	**0.245**	**0.245**	0.253	0.254	0.259	0.250	0.296
	336	MSE	**0.241**	0.262	0.262	0.271	0.278	0.278	0.265	0.283
		MAE	**0.283**	0.288	0.287	0.294	0.296	0.297	0.293	0.335
	720	MSE	**0.314**	0.342	0.344	0.360	0.358	0.354	0.339	0.345
		MAE	0.345	0.342	0.342	0.346	0.347	0.348	**0.341**	0.381
Solar	96	MSE	**0.176**	0.194	**0.176**	0.197	0.200	0.225	0.218	0.285
		MAE	**0.199**	0.244	0.235	0.246	0.234	0.271	0.308	0.372
	192	MSE	0.201	0.202	**0.199**	0.230	0.233	0.267	0.222	0.320
		MAE	**0.234**	0.261	0.257	0.265	0.261	0.310	0.283	0.398
	336	MSE	0.224	**0.203**	0.211	0.248	0.248	0.290	0.231	0.353
		MAE	**0.252**	0.253	0.263	0.278	0.273	0.315	0.292	0.415
	720	MSE	0.231	0.217	**0.209**	0.246	0.249	0.289	0.243	0.356
		MAE	**0.248**	0.263	0.270	0.279	0.275	0.317	0.294	0.413
ETTh1	96	MSE	**0.358**	0.367	0.371	0.392	0.386	0.414	0.375	0.386
		MAE	**0.381**	0.391	0.393	0.409	0.405	0.419	0.400	0.400
	192	MSE	**0.414**	0.420	0.428	0.445	0.441	0.460	0.429	0.437
		MAE	**0.415**	0.421	0.426	0.436	0.436	0.445	0.421	0.432
	336	MSE	**0.456**	0.476	0.476	0.482	0.487	0.501	0.484	0.481
		MAE	**0.440**	0.447	0.446	0.456	0.458	0.466	0.458	0.459
	720	MSE	**0.479**	0.493	0.487	0.489	0.503	0.500	0.498	0.519
		MAE	0.484	0.476	**0.470**	0.480	0.491	0.488	0.482	0.516
ETTh2	96	MSE	**0.287**	0.290	0.295	0.305	0.297	0.302	0.289	0.333
		MAE	**0.337**	0.342	0.343	0.353	0.349	0.348	0.341	0.387
	192	MSE	0.368	**0.362**	0.367	0.382	0.380	0.388	0.372	0.477
		MAE	**0.388**	0.389	0.393	0.401	0.400	0.400	0.392	0.476
	336	MSE	0.408	0.414	0.417	0.434	0.428	0.426	**0.386**	0.594
		MAE	0.422	0.429	0.427	0.445	0.432	0.433	**0.414**	0.541
	720	MSE	0.417	0.423	0.433	0.424	0.427	0.431	**0.412**	0.831
		MAE	0.436	0.442	0.446	0.444	0.445	0.446	**0.434**	0.657
ETTm1	96	MSE	0.307	**0.305**	0.311	0.336	0.334	0.329	0.320	0.345
		MAE	**0.344**	0.349	0.353	0.369	0.368	0.367	0.357	0.372
	192	MSE	0.351	**0.349**	0.356	0.374	0.377	0.367	0.361	0.380
		MAE	**0.367**	0.375	0.378	0.387	0.391	0.385	0.381	0.389
	336	MSE	0.384	**0.380**	0.390	0.408	0.426	0.399	0.390	0.413
		MAE	**0.392**	0.398	0.401	0.407	0.420	0.410	0.404	0.413
	720	MSE	0.450	**0.435**	0.452	0.477	0.491	0.454	0.454	0.474
		MAE	**0.432**	0.436	0.440	0.442	0.459	0.439	0.441	0.453
ETTm2	96	MSE	**0.167**	0.168	0.173	0.184	0.180	0.175	0.175	0.193
		MAE	**0.248**	0.249	0.256	0.267	0.264	0.259	0.258	0.292
	192	MSE	0.234	**0.232**	0.238	0.252	0.250	0.241	0.237	0.284
		MAE	0.295	**0.292**	0.298	0.307	0.309	0.302	0.299	0.362
	336	MSE	0.295	**0.287**	0.301	0.314	0.311	0.305	0.298	0.369
		MAE	**0.329**	0.330	0.340	0.345	0.348	0.343	0.340	0.427
	720	MSE	0.393	**0.386**	0.397	0.412	0.412	0.402	0.391	0.554
		MAE	**0.388**	0.390	0.396	0.402	0.407	0.400	0.396	0.522

D.2 Stability Across Random Seeds

To assess the sensitivity of STEP to random initialization, we additionally evaluate three independent runs with different random seeds for each dataset and forecast horizon, using $L = 96$ and $H \in \{96, 192, 336, 720\}$. Table 11 reports the mean and sample standard deviation across the three runs. Overall, the results are consistent across random seeds.

Table 11: STEP forecasting results over three independent runs with different random seeds and $L = 96$. MSE and MAE are reported as mean ± sample standard deviation; H denotes the forecast horizon.

Dataset	H	MSE	MAE	Dataset	H	MSE	MAE
Electricity	96	0.1287±0.0009	0.2217±0.0007	ETTh1	96	0.3578±0.0002	0.3815±0.0001
	192	0.1456±0.0003	0.2386±0.0007		192	0.4129±0.0010	0.4154±0.0002
	336	0.1571±0.0002	0.2545±0.0002		336	0.4558±0.0007	0.4403±0.0004
	720	0.1843±0.0018	0.2860±0.0025		720	0.4776±0.0014	0.4834±0.0004
Traffic	96	0.3806±0.0010	0.2483±0.0003	ETTh2	96	0.2882±0.0010	0.3372±0.0005
	192	0.4083±0.0006	0.2599±0.0002		192	0.3699±0.0015	0.3892±0.0010
	336	0.4284±0.0004	0.2691±0.0001		336	0.4112±0.0024	0.4232±0.0009
	720	0.4615±0.0003	0.2889±0.0002		720	0.4174±0.0013	0.4360±0.0009
Weather	96	0.1497±0.0014	0.1968±0.0026	ETTm1	96	0.3074±0.0014	0.3438±0.0011
	192	0.1955±0.0011	0.2464±0.0024		192	0.3521±0.0010	0.3667±0.0006
	336	0.2432±0.0018	0.2877±0.0040		336	0.3841±0.0007	0.3922±0.0006
	720	0.3134±0.0005	0.3444±0.0010		720	0.4503±0.0002	0.4324±0.0003
Solar	96	0.1787±0.0022	0.2002±0.0016	ETTm2	96	0.1703±0.0026	0.2514±0.0034
	192	0.2086±0.0091	0.2356±0.0035		192	0.2350±0.0012	0.2967±0.0020
	336	0.2308±0.0056	0.2547±0.0025		336	0.2973±0.0025	0.3309±0.0015
	720	0.2326±0.0015	0.2488±0.0009		720	0.3945±0.0013	0.3886±0.0007

E Backbone Enhancement

We consider two ways of enhancing existing forecasting models, serving different purposes. Training-based integration incorporates STEP into existing backbones to improve their forecasting accuracy. Inference-time enhancement considers a more constrained setting in which the backbone remains frozen and matched reruns differ only in the supplied forecast-start value. This confines the intervention to a single, explicit point while leaving the original backbone architecture and learned parameters unchanged.

E.1 Training-based Integration

We integrate STEP at each backbone's native variate-level representation interface using either module insertion or encoder replacement, while retaining the remaining forecasting pathway.

Let $\mathbf{Z} \in \mathbb{R}^{C \times d_b}$ denote the representation at the selected backbone interface, where d_b is its feature width. We augment this representation with projected historical differences and apply the common STEP transformation $\mathcal{S}$:

$$\mathcal{D}(\mathbf{X}) = \sum_{k=1}^{K_b} E_k\big((\Delta^k \mathbf{X})^\top\big)\,, \qquad \widetilde{\mathbf{Z}} = \mathcal{S}(\mathbf{Z} + \mathcal{D}(\mathbf{X}))\,. \tag{11}$$

Here, each E_k maps the corresponding difference order to the backbone feature width d_b, and $K_b \in \{1, 2, 3\}$ is selected using validation performance following the same configuration procedure as STEP. The transformation $\mathcal{S}$ applies the shift, variable interaction, and subspace interaction introduced in Section 3 while preserving the representation shape. For decomposed or multiscale inputs, $\mathbf{X}$ denotes the history of the corresponding branch or scale. Normalization follows the backbone's own convention. The transformed representation is then returned to the backbone through its native interface.

For module insertion, the native processing is retained and STEP transforms an intermediate representation. For encoder replacement, STEP operates at the native encoder interface and replaces that encoder while leaving the remaining forecasting components unchanged. Thus, the two strategies differ in their integration interface, while the same core STEP transformation is used across backbones. Table 12 specifies the strategy and interface for each backbone.

Table 12: Integration interfaces used for training-based STEP enhancement. A common STEP transformation is integrated through backbone-specific representation interfaces.

Backbone	Strategy	STEP interface
DLinear	Module insertion	After seasonal and trend decomposition, before the native prediction heads
PatchTST	Module insertion	After the native patch encoder, before the prediction head
iTransformer	Encoder replacement	Replaces the native encoder after inverted embedding
TimeMixer	Module insertion	At each normalized scale, before native scale encoding
Client	Encoder replacement	Replaces the native cross-variable encoder after RevIN
TQNet	Module insertion	After temporal-query aggregation, before the native input projection
GTR	Module insertion	After GTR aggregation, before the native input projection

After either integration, the remaining native forecasting components produce the initial forecast $\widetilde{\mathbf{Y}}$. Both strategies then use a common residual refinement form:

$$\widehat{\mathbf{Y}} = \widetilde{\mathbf{Y}} + W_{\text{ref}}[\mathbf{X}_b; \widetilde{\mathbf{Y}}], \qquad W_{\text{ref}} \in \mathbb{R}^{H\times(L+H)}. \tag{12}$$

Here, $\mathbf{X}_b \in \mathbb{R}^{L\times C}$ denotes the history sequence supplied at the backbone-specific refinement interface, following the backbone's native normalization convention. The backbone and STEP components are trained jointly using the objective in Eq. (7).

E.2 Inference-Time Enhancement

Inference-time enhancement targets a different use case: improving an established forecasting system through a localized and controlled intervention. Relative to the matched-rerun control, only the supplied forecast-start value is modified, while the backbone remains frozen.

Let f be a frozen forecasting backbone and g an independently trained STEP model. For history $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_L]$, we form

$$\mathbf{X}^{+}(\mathbf{a}) = [\mathbf{x}_2, \ldots, \mathbf{x}_L, \mathbf{a}]. \tag{13}$$

The control uses $\mathbf{a} = f(\mathbf{X})_1$, retaining the backbone's own forecast-start error; STEP uses $\mathbf{a} = g(\mathbf{X})_1$, and GT start uses $\mathbf{a} = \mathbf{Y}_1$, removing this error. All supplied values are expressed on the backbone input scale. All other inputs and backbone parameters remain fixed, and time features, where used, follow the corresponding timestamps.

When reporting the fraction of ground-truth-start improvement recovered by STEP, we use

$$\frac{\text{MSE}_{\text{Control}} - \text{MSE}_{\text{STEP}}}{\text{MSE}_{\text{Control}} - \text{MSE}_{\text{GT start}}}.$$

The first output of $f(\mathbf{X}^{+})$ corresponds to original forecast position 2, so positions 1 to 95 of the resulting forecast align with original positions 2 to 96. We therefore evaluate original positions 2 to 96. Because Control and STEP differ only in the supplied forecast-start value, their difference measures the subsequent forecasting response to forecast-start correction.

For window n, we define forecast-start influence as

$$s_n = \text{MSE}^{\text{Control}}_{n,2:96} - \text{MSE}^{\text{GT start}}_{n,2:96}. \tag{14}$$

For each backbone and dataset, windows are divided into four approximately equal quartiles according to s_n, with Q1 containing the lowest influence and Q4 the highest. Ties follow original window order.

E.3 Comparison with δ-Adapter

We further compare STEP with δ-Adapter (Liang et al., 2026), a recent approach for enhancing existing forecasting models. Since the published δ-Adapter results and our experiments use separately trained iTransformer backbones, their original forecasting errors differ. We therefore compare the

improvement of each method relative to its corresponding original backbone. Table 13 reports results on Electricity with $L = 96$ and $H \in \{96, 192, 336, 720\}$. The δ-Adapter results are taken from the original paper, while the STEP results follow our experimental setting.

Table 13: Comparison with published δ-Adapter results on Electricity using iTransformer ($L = 96$). Since the two studies use separately trained original backbones, reductions are computed relative to the corresponding Original within each implementation.

	δ-Adapter				Our results		
H	Original	+Ada-X	+Ada-Y	Best reduction	Original	+STEP	Reduction
96	.163	.160	.157	3.7%	.148	.132	**10.8%**
192	.175	.173	.167	4.6%	.162	.147	**9.3%**
336	.193	.189	.182	5.7%	.178	.158	**11.2%**
720	.231	.226	.218	5.6%	.225	.189	**16.0%**

In these reported results, STEP shows relative MSE reductions of 9.3%–16.0%, compared with 3.7%–5.7% for the reported Ada-X and Ada-Y variants.

F Ablation Details

F.1 Component Ablations and Parameter Matching

The component ablations in Table 4 use $L = H = 96$. The *w/o all modules* variant jointly removes the historical-difference embedding, learned feature shifts, shifted variable interaction, subspace interaction, and joint history–forecast refinement, while keeping the remaining forecasting pathway unchanged. The STEP objective is retained. Removing the shift sets the learned offsets to zero while keeping the remaining components unchanged. Removing shifted variable interaction disables the variable-interaction module across learned-shift heads, while removing subspace interaction disables the corresponding subspace-interaction module. Removing refinement evaluates and trains the initial forecast directly. Removing the STEP objective replaces Eq. (7) with MSE, while removing the change term and exponential weighting sets $\mu = 0$ and $\lambda = 0$, respectively.

For the capacity-matched comparison in Figure 5, we replace STEP-specific components with residual FFN blocks while matching the number of prediction parameters. Each FFN block uses layer normalization and two linear layers with GELU activation, and its hidden width is adjusted so that the parameter gap from STEP remains below 0.1%. Table 14 reports the resulting parameter counts.

Table 14: Prediction parameter counts for STEP and the capacity-matched FFN control ($L = H = 96$). Gap denotes the absolute relative difference from STEP.

Dataset	STEP	FFN control	Gap (%)
Electricity	19,089,305	19,088,598	0.0037
Traffic	12,946,304	12,945,128	0.0091
Weather	66,857	66,918	0.0912
Solar	12,881,458	12,881,858	0.0031

For each window, STEP gain is the control MSE minus STEP MSE over positions 2 to 96. Forecast-start influence and Q1–Q4 grouping are computed from the control model under the Control and GT-start conditions following Eq. (14), without using STEP predictions. MSE is evaluated over positions 2 to 96 for this analysis.

G Learned Mechanisms and Forecasting Effects

G.1 Learned Mechanisms

Figure 6(a) visualizes the Electricity model with $L = H = 96$. The shift distributions show initialization and the three trained encoder layers. Learned shifts spread from zero to nearby offsets, with distributions that vary across layers. The absolute refinement weights concentrate around the junction between history and forecast and draw information from both the observed history and the initial forecast.

G.2 Differentiated Effects of the Two Design Parts

Figure 6(b) compares STEP with two separately trained, parameter-matched FFN controls on Electricity with $L = 96$ and $H = 720$. For the initial forecast design, the control replaces the variable- and subspace-interaction modules in both the original and shifted branches with FFNs, retaining the remaining components and the STEP objective. For the transition design, the control replaces joint refinement with an FFN acting on the initial forecast and uses MSE loss. This comparison evaluates refinement and its training objective together.

We evaluate all three models on the same 4,541 test windows. For each control, the relative MSE reduction is $100(\mathrm{MSE}_{\mathrm{control}} - \mathrm{MSE}_{\mathrm{STEP}})/\mathrm{MSE}_{\mathrm{control}}$. MSE is averaged over windows and variables at step 1, and additionally over steps 2 to H for later positions. Here, $H = 720$ throughout panel (b).

The initial forecast design reduces MSE by 2.5% at step 1 and 11.0% over later positions. The transition design reduces MSE by 16.9% and 6.2%, respectively. These effects are consistent with broader gains over later positions from the initial forecast design and a stronger effect at the forecast start from the transition design.

Error bars are 95% confidence intervals from 5,000 paired circular-block bootstrap samples with 720 consecutive test windows per block. The same sampled window indices are applied to STEP and both controls, and each relative reduction is recomputed from the resampled MSEs. The interval endpoints are the 2.5th and 97.5th percentiles.

G.3 Grouping and Confidence Intervals

Forecast-start influence and Q1 to Q4 follow the procedure in Appendix E.2, using positions 2 to 96 for the capacity-matched analysis. For the capacity-matched analysis, group assignments are computed from the FFN control model under the Control and GT-start conditions and remain fixed when comparing STEP with the control. For each window n, the STEP gain G_n is the control MSE minus the STEP MSE over positions 2 to 96. The mean gain within group Q is

$$G_Q = \frac{1}{|Q|} \sum_{n \in Q} G_n,$$

and $G_{Q4} - G_{Q1}$ measures the difference in STEP gain between the highest and lowest forecast-start influence groups.

Confidence intervals in Figure 5 are estimated using 5,000 paired circular block bootstrap samples with 168 consecutive forecast windows per block. Each block preserves the temporal ordering of consecutive windows and uses circular wrapping at the end of the evaluation sequence. The same resampled indices are applied to STEP and the control, and the 2.5th and 97.5th percentiles of the bootstrap distribution define the 95% confidence interval.

H Computational Cost

H.1 Parameters, Inference Time, and GPU Memory

Figure 9 reports parameter count, inference latency, and peak GPU memory on Weather and Electricity under a common long horizon setting with $L = H = 720$. All models are evaluated on one NVIDIA L40S using single precision and batch size one. After 30 warm-up iterations, latency is measured over 200 forward passes using CUDA events with inputs resident on the GPU; peak allocated GPU memory is recorded during the same evaluation.

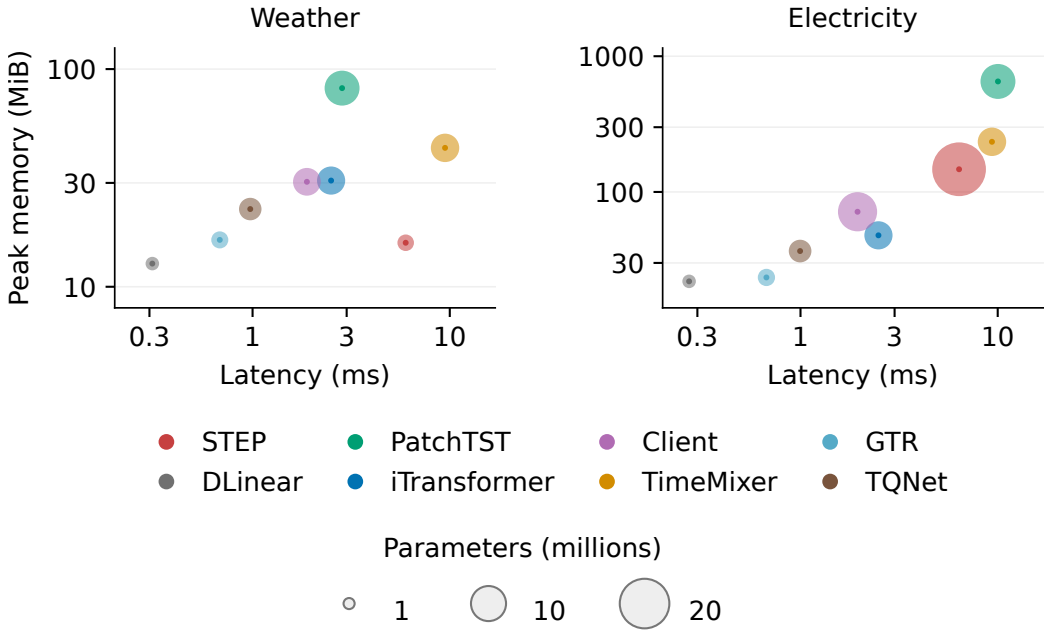


Figure 9: Computational cost on Weather and Electricity with $L = H = 720$. Axes show median inference latency and peak allocated GPU memory on logarithmic scales; bubble area represents parameter count.

STEP shows different computational profiles across the two datasets. On Weather, it uses 15.93 MiB of peak memory with 1.66 million parameters and a median latency of 5.98 ms. On Electricity, it uses 146.97 MiB with 21.59 million parameters. Figure 9 provides the corresponding comparisons with the evaluated baselines.

I VISUALIZATION

I.1 BACKBONE ENHANCEMENT

Figures 10 through 13 provide qualitative examples of training-based and inference-time STEP enhancement across the seven backbones on Electricity, Traffic, Weather, and Solar. Each row shows two forecasting windows under both enhancement settings. Values use training-set standardization.

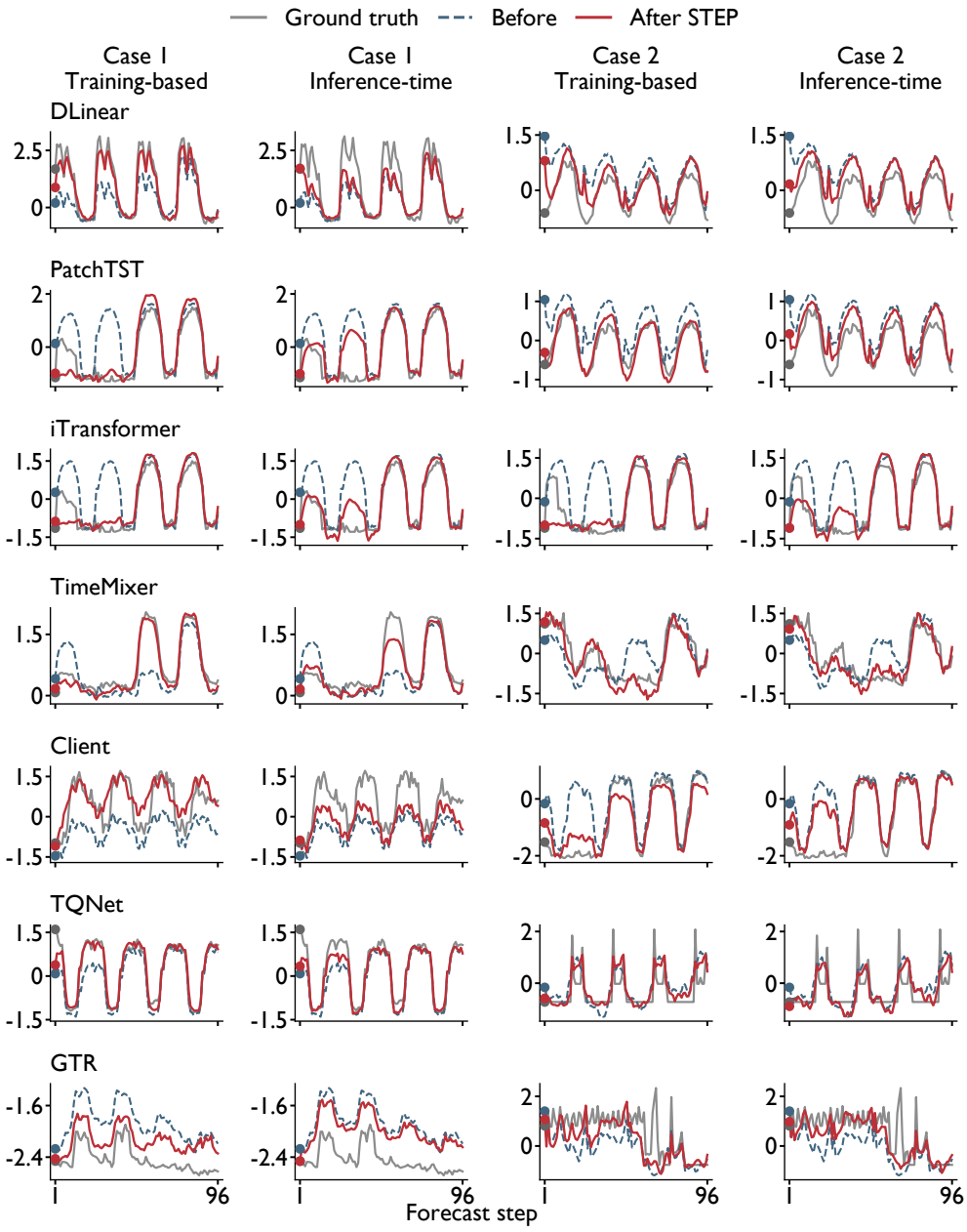


Figure 10: Qualitative examples of STEP backbone enhancement on Electricity ($L = H = 96$). Each row shows two forecasting windows, each under training-based and inference-time enhancement. Training-based panels compare the backbone with its STEP-integrated version; inference-time panels compare matched reruns that differ only in the supplied forecast-start value while the backbone remains frozen.

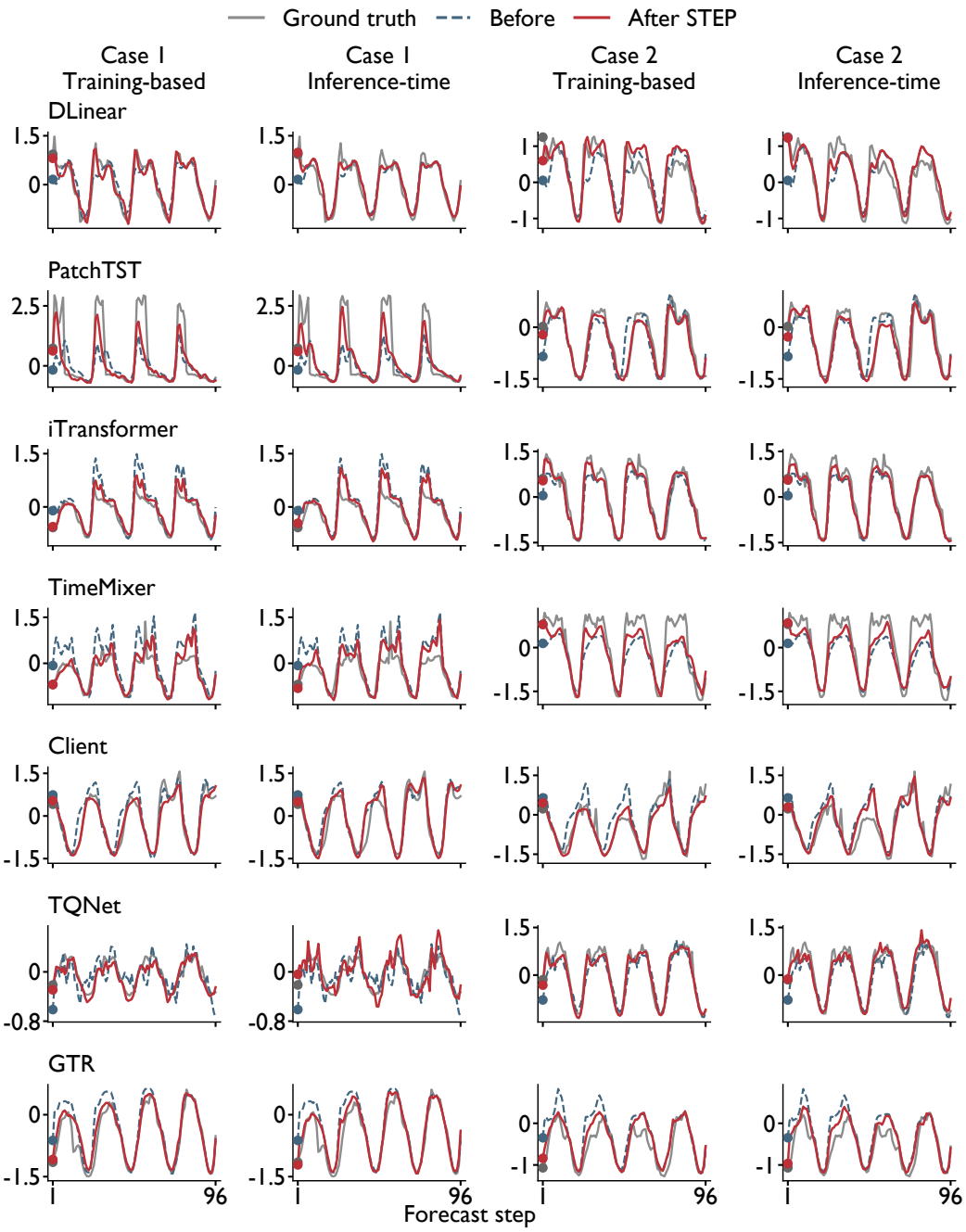


Figure 11: Qualitative examples of STEP backbone enhancement on Traffic ($L = H = 96$). Each row shows two forecasting windows, each under training-based and inference-time enhancement. Training-based panels compare the backbone with its STEP-integrated version; inference-time panels compare matched reruns that differ only in the supplied forecast-start value while the backbone remains frozen.

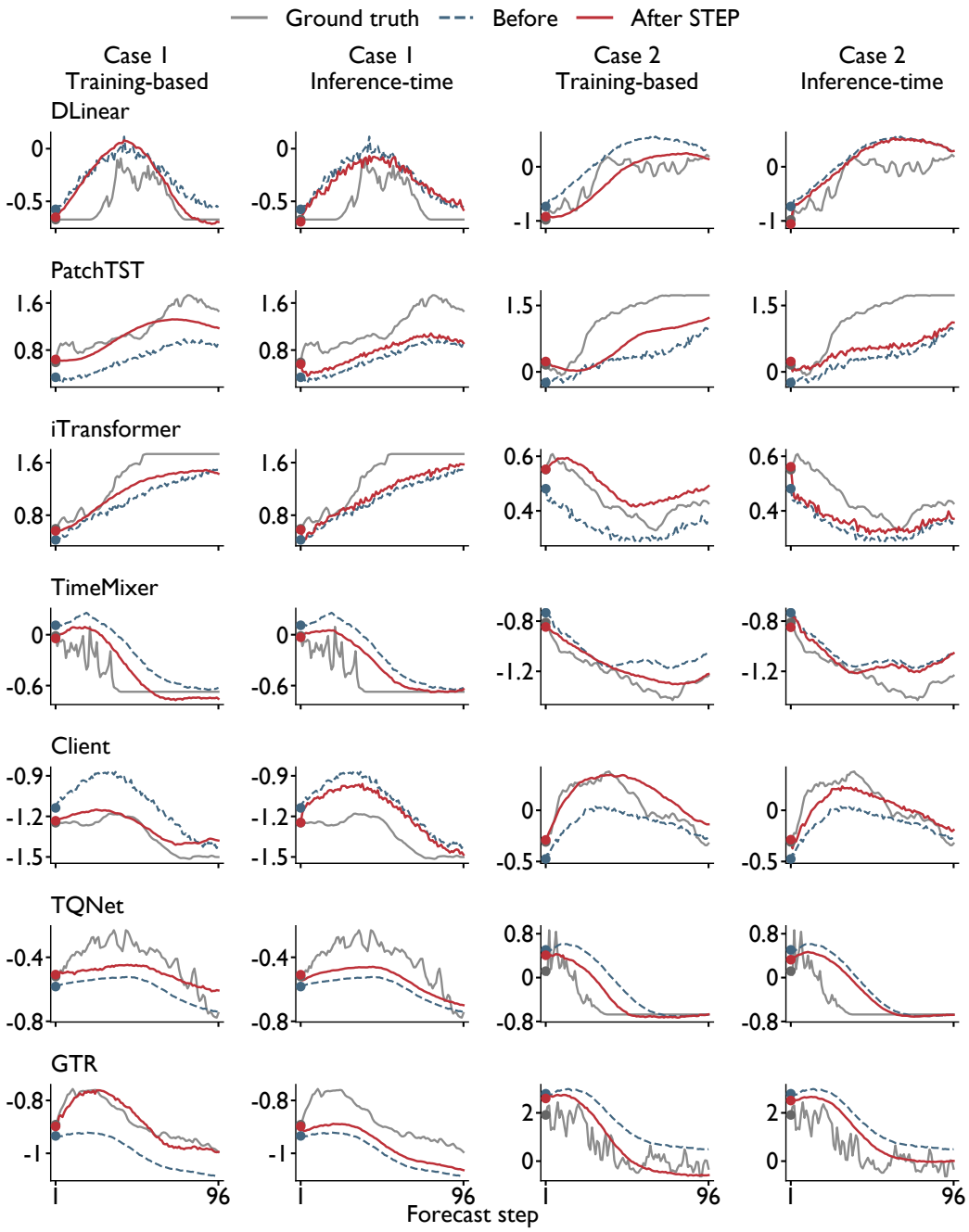


Figure 12: Qualitative examples of STEP backbone enhancement on Weather ($L = H = 96$). Each row shows two forecasting windows, each under training-based and inference-time enhancement. Training-based panels compare the backbone with its STEP-integrated version; inference-time panels compare matched reruns that differ only in the supplied forecast-start value while the backbone remains frozen.

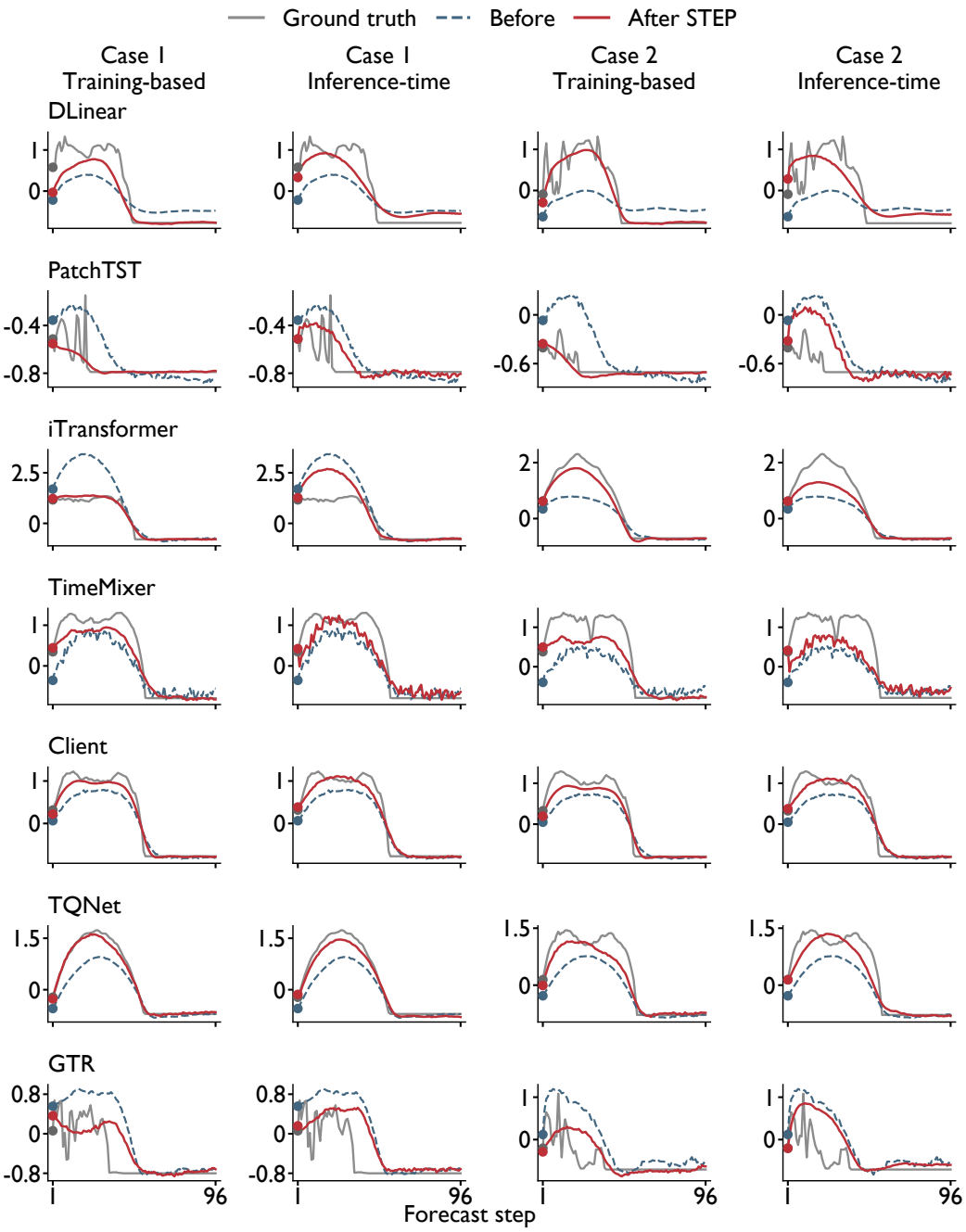


Figure 13: Qualitative examples of STEP backbone enhancement on Solar ($L = H = 96$). Each row shows two forecasting windows, each under training-based and inference-time enhancement. Training-based panels compare the backbone with its STEP-integrated version; inference-time panels compare matched reruns that differ only in the supplied forecast-start value while the backbone remains frozen.

I.2 Capacity-Matched Ablation

Figure 14 provides additional qualitative comparisons between STEP and the capacity-matched FFN control across the eight datasets.

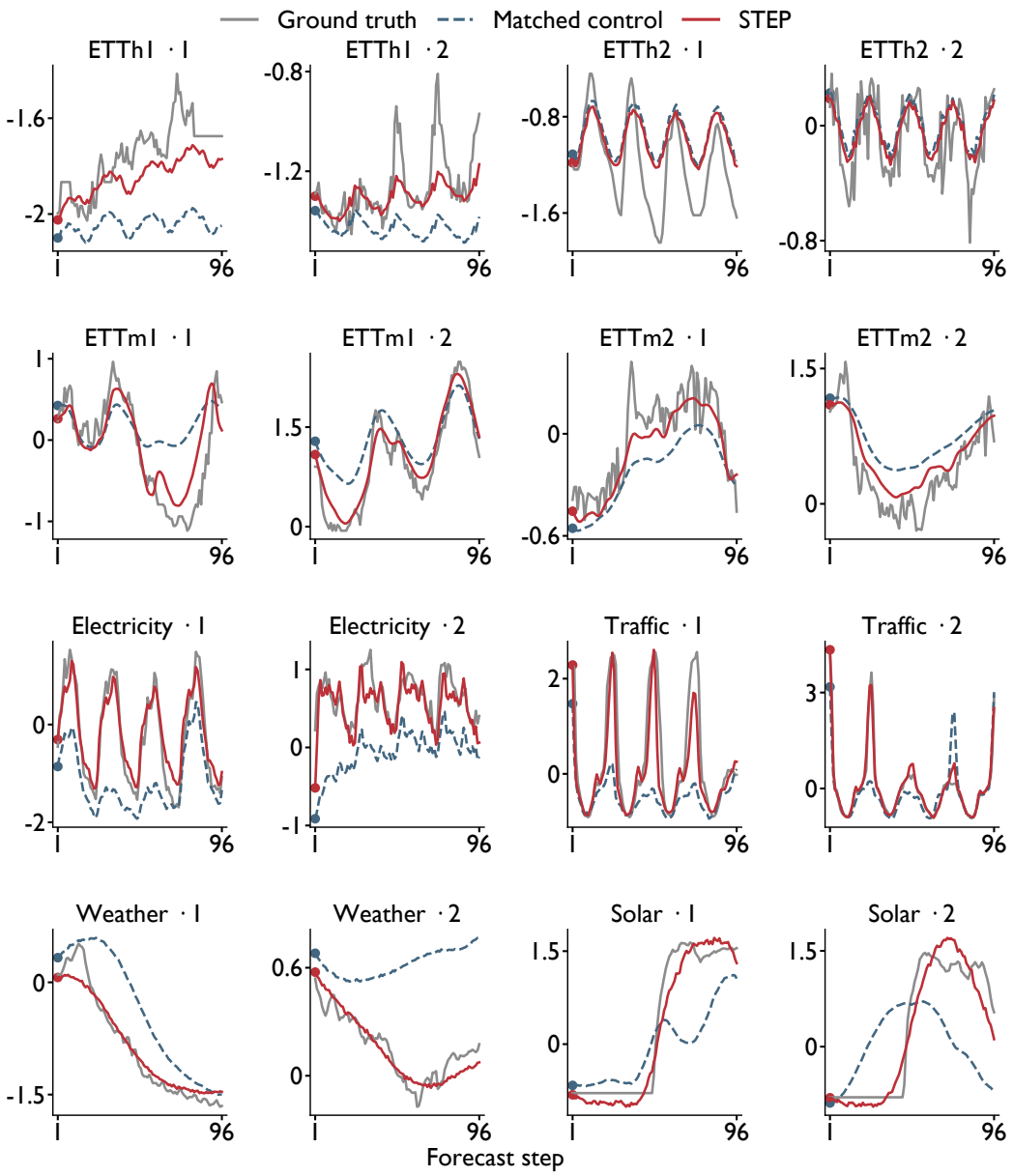


Figure 14: Qualitative forecast comparisons between STEP and the capacity-matched FFN control on eight datasets ($L = H = 96$), with two examples per dataset.

J Additional Inference-Time Enhancement Results

J.1 Enhancement on the Full Test Set

Table 15 reports inference-time forecast-start enhancement averaged over all test windows, following the protocol in Appendix E.2. STEP also improves forecasting performance overall on the full test set, with larger gains in Q4. This pattern is consistent with STEP's emphasis on the forecast start: larger gains are observed in windows with higher forecast-start influence.

Table 15: Inference-time forecast-start enhancement on the full test set. For $L = H = 96$, MSE covers steps 2–96. Control, STEP, and GT supply the frozen backbone with its own prediction, STEP's prediction, or the ground-truth first future value, respectively. Bold/underline: best/second-best distinct value; darker blue: lower MSE within each triple.

	Electricity			Traffic			Weather			Solar		
Backbone	Control	STEP	GT	Control	STEP	GT	Control	STEP	GT	Control	STEP	GT
DLinear	.1970	.1953	**.1949**	.6575	.6513	**.6509**	.2000	.1980	**.1972**	.2920	.2863	**.2854**
PatchTST	.1686	.1673	**.1666**	.4508	.4482	**.4471**	.1810	.1777	**.1774**	.2283	.2276	**.2248**
iTransformer	.1492	.1484	**.1474**	.4006	.4000	**.3975**	.1795	.1771	**.1764**	.2071	.2028	**.2001**
TimeMixer	.1543	.1532	**.1525**	.4924	.4888	**.4877**	.1622	.1614	**.1603**	.2255	.2181	**.2176**
Client	.1433	.1426	**.1417**	.4380	.4364	**.4343**	.1677	.1658	**.1649**	.1999	.1989	**.1972**
TQNet	.1354	.1351	**.1342**	.4159	.4154	**.4130**	.1588	.1576	**.1565**	.1781	.1760	**.1757**
GTR	.1355	.1351	**.1342**	.4313	.4297	**.4281**	.1571	.1560	**.1550**	.1948	.1944	**.1935**

J.2 Comparison with Untouched One-shot Forecasts

As an additional sanity check, we compare the inference-time enhancement with the backbone's untouched one-shot forecast on the same original forecast positions 2 to 96. For a frozen backbone f and STEP model g, One-shot evaluates $[f(X)]_{2:96}$, whereas STEP rerun evaluates $[f(X^{+}([g(X)]_1))]_{1:95}$, aligned with the original forecast positions 2 to 96. Q4 membership is fixed according to Appendix E.2 and is not selected using the one-shot versus STEP-rerun gains reported here.

Table 16: Comparison with untouched one-shot forecasts on Q4 ($L = H = 96$). MSE covers original forecast positions 2 to 96. One-shot denotes the untouched forecast; STEP rerun denotes the STEP-enhanced rerun. Gain reports the relative MSE reduction (%); positive values indicate improvement. Bold and darker blue mark the lower MSE within each pair. Relative gains are computed from unrounded MSE values.

	Electricity			Traffic			Weather			Solar		
Backbone	One-shot	STEP rerun	Gain	One-shot	STEP rerun	Gain	One-shot	STEP rerun	Gain	One-shot	STEP rerun	Gain
DLinear	.2122	**.2081**	+1.92	.6581	**.6442**	+2.12	.2689	**.2661**	+1.03	.4030	**.3961**	+1.71
PatchTST	.1962	**.1939**	+1.14	.4658	**.4613**	+0.97	.2226	**.2202**	+1.10	**.2558**	.2618	−2.34
iTransformer	.1818	**.1808**	+0.50	**.4343**	.4349	−0.14	.2274	**.2247**	+1.16	**.2653**	.2680	−1.00
TimeMixer	.1799	**.1779**	+1.08	.4927	**.4862**	+1.33	.1964	**.1949**	+0.79	.2436	**.2367**	+2.85
Client	.1737	**.1729**	+0.48	.4643	**.4635**	+0.18	.2103	**.2093**	+0.48	.2689	**.2687**	+0.08
TQNet	.1566	**.1561**	+0.36	.4377	**.4370**	+0.18	.2056	**.2046**	+0.48	.2542	**.2490**	+2.04
GTR	.1570	**.1563**	+0.41	.4543	**.4518**	+0.55	.2052	**.2040**	+0.56	.2611	**.2590**	+0.78

Table 17: Comparison with untouched one-shot forecasts on the full test set ($L = H = 96$). MSE covers original forecast positions 2 to 96. One-shot denotes the untouched forecast; STEP rerun denotes the STEP-enhanced rerun. Gain reports the relative MSE reduction (%); positive values indicate improvement. Bold and darker blue mark the lower MSE within each pair. Relative gains are computed from unrounded MSE values.

	Electricity			Traffic			Weather			Solar		
Backbone	One-shot	STEP rerun	Gain	One-shot	STEP rerun	Gain	One-shot	STEP rerun	Gain	One-shot	STEP rerun	Gain
DLinear	.1960	**.1953**	+0.36	.6529	**.6513**	+0.25	.1993	**.1980**	+0.67	.2879	**.2863**	+0.58
PatchTST	.1677	**.1673**	+0.27	.4493	**.4482**	+0.23	.1795	**.1777**	+0.98	**.2268**	.2276	−0.36
iTransformer	.1485	**.1484**	+0.09	**.3996**	.4000	−0.11	.1784	**.1771**	+0.75	**.2024**	.2028	−0.22
TimeMixer	.1536	**.1532**	+0.26	.4897	**.4888**	+0.18	.1621	**.1614**	+0.47	.2195	**.2181**	+0.62
Client	.1428	**.1426**	+0.12	.4365	**.4364**	+0.03	.1668	**.1658**	+0.58	.1992	**.1989**	+0.15
TQNet	.1352	**.1351**	+0.08	**.4152**	.4154	−0.04	.1584	**.1576**	+0.51	.1778	**.1760**	+0.99
GTR	.1353	**.1351**	+0.10	.4302	**.4297**	+0.11	.1568	**.1560**	+0.52	.1957	**.1944**	+0.69

The inference-time enhancement yields lower subsequent MSE than the untouched one-shot forecast in 25/28 Q4 settings and 24/28 full-test settings.

We compute 95% paired circular-block bootstrap confidence intervals using 5,000 samples with 168 consecutive test windows per block, applying the same resampled indices to both forecasts. For each bootstrap sample, we recompute the aggregate MSE of both forecasts and their relative reduction,

$$100\,\frac{\mathrm{MSE}_{\text{One-shot}} - \mathrm{MSE}_{\text{STEP rerun}}}{\mathrm{MSE}_{\text{One-shot}}}.$$

The 2.5th and 97.5th percentiles define the 95% confidence interval. For Q4, the membership defined in Appendix E.2 is held fixed. Circular blocks are sampled from the full test timeline, and only resampled indices belonging to this fixed Q4 set contribute to the Q4 statistic. The confidence intervals for the relative MSE reduction lie above zero in 21/28 Q4 settings and 22/28 full-test settings.

K Limitation and Future Work

Our controlled analyses show that subsequent forecasts respond systematically to corrections in the supplied forecast start across diverse forecasting backbones. A remaining question is how to identify, before intervention, which forecasting models or prediction windows have the greatest potential to benefit from such localized corrections. The current analysis characterizes this potential retrospectively through forecast-start influence, but does not yet provide a prospective diagnostic for estimating it from the forecasting system itself. Future work could therefore develop a model-agnostic identification framework that predicts forecast-start influence and selectively applies correction where larger subsequent improvements are expected.