LEARNING FROM HISTORICAL OVERLAP ERRORS: ADAPTIVE POINTWISE REWEIGHTING FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term time series forecasting models are commonly trained with sliding windows, so nearby input windows predict many of the same future timestamps. Standard training optimizes these overlapping forecasts independently, leaving the errors of preceding forecasts unused as timestamp-specific evidence for determining the optimization priority of current predictions. We propose Historical Overlap Reliability Weighting (HORW), a model-agnostic method that compares each prediction point's current error with a statistical reference derived from historical prediction errors of preceding overlapping windows for the same target timestamp, and uses the resulting error gap to adjust the optimization emphasis on each prediction point. Specifically, after each training epoch, HORW holds the epoch-end model parameters fixed during a gradient-free sequential pass over the training set to construct a historical error memory. For each prediction point in the next epoch, it uses the larger of the mean and median errors from preceding windows forecasting the same target timestamp as a conservative reference. When the current error exceeds this reference, HORW maps the normalized gap to a point-wise loss weight. A single signed scalar coefficient specifies the response to this gap: its sign determines whether the corresponding point is relatively emphasized or attenuated, while its magnitude controls the response strength. HORW preserves the original model architecture, applies to diverse forecasting objectives, and adds no inference-time cost. Experiments on seven datasets with three architecturally diverse forecasting models show improvements in the vast majority of evaluated settings, with additional gains when HORW is combined with auxiliary training objectives. Ablations further support the value of timestamp-aligned historical references over global historical statistics and current-error references.

1 INTRODUCTION

Time series forecasting plays an important role in applications such as energy management, traffic planning, finance, weather analysis, and healthcare (Qiu et al., 2026; Lee & Lee, 2025; Kraus et al., 2025). Recent progress has been driven by both forecasting architectures, including linear models (Zeng et al., 2023; Yue et al., 2025), convolutional models (Liu et al., 2022; Wu et al., 2023), and Transformer-based models (Nie et al., 2023; Liu et al., 2024), and auxiliary training objectives such as DB Loss, PS Loss, and FreDF (Qiu et al., 2025; Kudrat et al., 2025; Wang et al., 2025), which incorporate temporal or structural information beyond point-wise prediction errors.

Most forecasting models are trained with sliding windows that convert a long series into fixed-length input-output pairs (Zeng et al., 2023; Wu et al., 2023; Nie et al., 2023). Although this setup supports efficient mini-batch optimization and exposes models to diverse local contexts, nearby input windows predict many of the same future timestamps. Standard point-wise training treats these overlapping predictions independently, discarding a potentially useful source of supervision: the errors made by preceding windows on the same target timestamps. For a current prediction point, these errors provide direct evidence of how well the same target has been forecast under earlier input contexts, which can be used to determine its relative optimization priority.

AliO (Park et al., 2025) exploits overlapping forecasts by aligning their output segments through pairwise auxiliary objectives. Its Regression Pulling mechanism encourages less accurate predic-

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.

tions to move toward their more accurate counterparts. However, even the more accurate prediction in a pair may have a substantial error, and pairwise relative accuracy alone does not establish how the current error compares with accumulated performance across preceding views. This motivates a different use of overlap information: constructing a historical performance reference to determine how strongly each current prediction point should be optimized. Specifically, we ask whether the errors of *all* preceding overlapping forecasts for the same target timestamp can be accumulated as historical evidence and used to adapt the optimization strength of the current prediction, while preserving the original backbone architecture and single-window mini-batch training procedure.

To this end, we propose Historical Overlap Reliability Weighting (HORW), a model-agnostic method that uses timestamp-specific historical error evidence to determine how strongly each prediction point is optimized. After each training epoch, HORW freezes the resulting model and performs a gradient-free sequential pass over the training set to construct a historical error memory. For each current prediction point in the next epoch, HORW aggregates the errors of all available preceding windows forecasting the same target timestamp, using the larger of their mean and median as a conservative reference. When the current error exceeds this reference, HORW converts the normalized gap into a point-wise loss weight. A single signed scalar coefficient β specifies how optimization responds to the gap: its sign determines whether the corresponding point is relatively emphasized or attenuated, while its magnitude controls the response strength. Using a single frozen model snapshot ensures that the historical errors are computed under the same parameter state, while the reference remains fixed throughout the next epoch. HORW introduces no trainable parameters and preserves the original single-window mini-batch training procedure. It applies to different forecasting objectives without modifying backbone architectures or adding inference-time cost. Our main contributions are summarized as follows:

- We identify that sliding-window training leaves timestamp-specific historical error evidence from preceding overlapping forecasts underused, and use it to adaptively reweight the optimization of individual prediction points.
- We propose HORW, a model-agnostic method that aggregates the errors of all preceding overlapping forecasts for each target timestamp and uses a cross-epoch historical reference to adaptively reweight the original point-wise forecasting objective.
- Experiments on 7 datasets and 3 diverse forecasting backbones show that HORW improves the average MSE in 20 of 21 backbone–dataset combinations and matches the remaining one. HORW also improves forecasting accuracy when applied to composite training objectives that combine the original point-wise forecasting loss with auxiliary losses, demonstrating its broad compatibility.

2 Related Work

2.1 Long-term Time Series Forecasting Models and Objectives

Existing long-term time series forecasting models can be broadly grouped into linear, convolution-based, and Transformer-based approaches. Linear models such as DLinear (Zeng et al., 2023) use simple mappings to model trend and seasonal components, while achieving strong performance with low computational cost. Convolution-based models capture local, periodic, and multi-scale temporal patterns through temporal or two-dimensional convolutions, as exemplified by SCINet (Liu et al., 2022) and TimesNet (Wu et al., 2023). Transformer-based methods (Zhou et al., 2021; Wu et al., 2021; Liu et al., 2025; Zhang & Yan, 2023) use attention mechanisms to model long-range temporal dependencies and variable relations. Most forecasting models are trained with point–wise losses such as MSE, which measure the distance between predictions and ground-truth values at each time step. To capture temporal properties beyond point–wise errors, recent studies have developed more structured forecasting objectives. FreDF introduces frequency-domain supervision to account for dependencies among future labels in direct multi-step forecasting (Wang et al., 2025). DBLoss decomposes the forecasting horizon into seasonal and trend components and applies separate weighted losses to them (Qiu et al., 2025). PS Loss divides predictions and targets into patches and compares their local correlation, variance, and mean, thereby incorporating local structural similarity into training (Kudrat et al., 2025). These studies improve forecasting through model design or objective design, but generally treat each sliding-window sample as an independent training instance.

2.2 Overlapping Forecasts and Output Alignment

Sliding-window forecasting naturally produces multiple predictions for the same target timestamps from lagged input windows. AliO (Park et al., 2025) addresses output alignment by penalizing

discrepancies between overlapping predictions in the time and frequency domains. Its Regression Pulling mechanism uses ground-truth errors to direct less accurate predictions toward their more accurate counterparts within each pair. Although the more accurate prediction can provide a useful alignment target, it may still have a substantial error. Pairwise relative accuracy alone does not establish how the current error compares with the accumulated forecasting performance of preceding windows on the same target. HORW explicitly constructs this historical error reference. For each prediction point, it aggregates errors from all available preceding windows forecasting the same timestamp and variable, using a frozen model snapshot. It then maps the normalized gap between the current error and this reference to a point-wise loss weight. Thus, whereas AliO promotes agreement between overlapping outputs, HORW uses accumulated historical performance to determine the relative optimization emphasis of current prediction points. In addition, computing AliO's alignment losses requires predictions from temporally related windows to be jointly available during training. HORW preserves the backbone's original single-window mini-batch sampling procedure by retrieving historical references from a separately constructed error memory. It retains the backbone configuration and introduces a coefficient β to control weighting strength and direction.

2.3 Adaptive and Robust Loss Reweighting

Adaptive and robust forecasting methods assign different optimization priorities to samples according to signals such as abrupt changes, anomaly likelihood, or the distribution of running losses. For example, Local Discrepancy Density reweighting reduces the influence of abrupt temporal changes and emphasizes representative temporal patterns (Park et al., 2023). RobustTSF focuses on learning robust forecasting models from anomaly-contaminated training data (Cheng et al., 2024), while loss-weighted sampling methods adjust sampling probabilities to reduce the effects of extremely easy or difficult samples (You et al., 2024). Dynamic loss weighting has also been applied to time-series forecasting to adaptively balance forecast accuracy and forecast stability objectives during training (Caljon et al., 2026). Although these methods also modify the training process, their purpose and supervision signals differ fundamentally from those of HORW. Robust reweighting and loss-weighted sampling determine whether an entire sample should receive more or less attention based on its temporal characteristics, anomaly status, or loss magnitude, whereas dynamic loss weighting adjusts the relative contributions of multiple training objectives. In contrast, HORW does not identify anomalies, alter the sampling distribution, or balance multiple loss components. Instead, it asks a different question for each prediction point: compared with multiple preceding windows that forecast the same target timestamp, is the current error above an error level that the model has already demonstrated to be achievable? HORW then uses this cross-window historical evidence to modulate the point-wise optimization strength of the original forecasting objective.

3 Method

3.1 Problem Formulation

Let $\boldsymbol{x}_t \in \mathbb{R}^C$ denote the observation at timestamp t, where C is the number of variables. Given an input length L and a forecast horizon H, unit-stride sliding-window sampling constructs the i-th input sequence as $\mathbf{X}_i = [\boldsymbol{x}_i, \ldots, \boldsymbol{x}_{i+L-1}] \in \mathbb{R}^{L\times C}$, with the corresponding future target $\mathbf{Y}_i = [\boldsymbol{x}_{i+L}, \ldots, \boldsymbol{x}_{i+L+H-1}] \in \mathbb{R}^{H\times C}$. A forecasting model f_θ predicts $\hat{\mathbf{Y}}_i = f_\theta(\mathbf{X}_i)$.

Sliding-window sampling produces multiple forecasts for the same target timestamp. For a current window i, the prediction at horizon position h corresponds to timestamp $i+L+h-1$. A preceding window $i-s$ predicts the same timestamp at horizon position $h+s$:

$$\hat{Y}_{i,h,c} \quad \longleftrightarrow \quad \hat{Y}_{i-s,h+s,c}, \qquad 1 \le s \le \min(H-h, i-1). \tag{1}$$

Here, $\hat{Y}_{i,h,c}$ denotes the prediction of variable c at horizon position h from window i, and the arrow indicates a shared target timestamp rather than equal predictions. Thus, a prediction point can have up to $H-h$ preceding forecasts of the same target. These forecasts use different input contexts and forecast positions, and their errors may differ.

Conventional point-wise training minimizes the average loss over window–target pairs:

$$\mathcal{L}_{\text{base}} = \frac{1}{NHC} \sum_{i=1}^{N} \sum_{h=1}^{H} \sum_{c=1}^{C} \ell_{\text{base}} \left(\hat{Y}_{i,h,c}, Y_{i,h,c} \right), \tag{2}$$

where N denotes the number of sliding-window samples in the training set and ℓ_{base} denotes the original point-wise forecasting loss.

The overlap relation in Eq. (1) provides multiple views of forecasting performance for the same target timestamp. However, the conventional objective in Eq. (2) optimizes every window–target pair independently: it neither compares the current error with the errors previously obtained for the same target timestamp nor uses such cross-window evidence to determine the relative optimization priority of the current prediction point. The central problem addressed in this work is therefore how to convert these naturally available preceding-view errors into a reliable point-wise weighting signal without altering the original forecasting targets or backbone architecture.

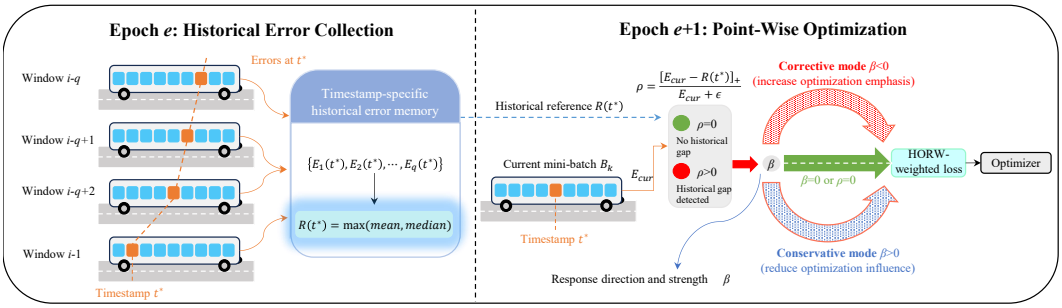

Figure 1: **Overview of HORW.** At the end of epoch e, HORW uses the frozen model to collect historical errors and construct references from preceding windows forecasting the same target timestamp and variable. During epoch $e+1$, each current error is compared with its corresponding reference to obtain a normalized gap ρ. A zero gap produces no additional point-specific weighting signal. For positive gaps, the sign of β determines the response direction: larger ρ yields greater relative weight when $\beta < 0$ and smaller relative weight when $\beta > 0$. The normalized weights modulate the original point-wise forecasting loss without changing the backbone architecture or single-window mini-batch training procedure.

3.2 Overview of HORW

To address the above problem, HORW reuses the errors of preceding overlapping forecasts as timestamp-specific historical evidence for point-wise loss weighting. As illustrated in Figure 1, HORW consists of two stages: historical error collection and point-wise optimization. At the end of each training epoch, HORW freezes the resulting model and evaluates it sequentially on the training set to collect prediction errors. For each prediction point, it constructs a conservative reference using the larger of the mean and median errors from all preceding windows forecasting the same timestamp and variable. These references remain fixed throughout the next epoch.

HORW then compares each current error with its corresponding reference and maps the normalized positive gap to a point-wise weight. The gap quantifies how much the current error exceeds its historical reference, while the sign and magnitude of the signed coefficient β determine the response direction and strength, respectively. After batch-wise normalization, the weights modulate the original point-wise forecasting loss, while preserving the backbone architecture and single-window mini-batch training procedure.

HORW incurs additional training-time computation from the epoch-end gradient-free memory refresh and the evaluation-mode forward passes used to compute current weighting signals. Validation and test-time inference remain unchanged. Detailed computational-complexity analysis and empirical efficiency results are provided in Appendices A and G, respectively.

3.3 Historical Error Memory from Preceding Windows

Denote by $\theta^{(e)}$ the final model parameters after epoch e. At the end of this epoch, HORW freezes $\theta^{(e)}$ and performs a sequential inference pass over the complete training set in evaluation mode without gradient computation. For the prediction at horizon position h and variable c of window i, we record an element-wise error

$$E^{(e)}_{i,h,c} = d\left(\hat{Y}^{(e)}_{i,h,c}, Y_{i,h,c}\right), \tag{3}$$

where $\hat{\mathbf{Y}}_i^{(e)} = f_{\theta^{(e)}}(\mathbf{X}_i)$ and $d(\cdot,\cdot)$ denotes an element-wise discrepancy measure. In all experiments, we match the discrepancy function to the original point-wise forecasting component: $d(a,b) = (a-b)^2$ for MSE-based objectives and $d(a,b) = |a-b|$ for MAE-based objectives.

Under the indexing convention in Section 3.1, $\tau(i,h) = i+L+h-1$ denotes the absolute timestamp predicted at horizon position h of window i. For a current prediction point (i,h,c), we consider only predictions from preceding windows:

$$\mathcal{P}_{i,h} = \{(j,r) \,|\, 1 \le j < i,\ 1 \le r \le H,\ \tau(j,r) = \tau(i,h)\}\,. \tag{4}$$

This restriction includes only temporally preceding windows, irrespective of their processing order during mini-batch training.

If $\mathcal{P}_{i,h} = \emptyset$, no historical reference is used for that point. Under unit-stride sampling, this occurs for the first training window or the terminal forecast position $h = H$. Other early windows may have fewer preceding views. When $\mathcal{P}_{i,h} \neq \emptyset$, HORW summarizes the corresponding historical errors as

$$R_{i,h,c}^{(e)} = \max\left(\mathrm{Mean}_{(j,r)\in\mathcal{P}_{i,h}} E_{j,r,c}^{(e)},\ \mathrm{Median}_{(j,r)\in\mathcal{P}_{i,h}} E_{j,r,c}^{(e)}\right). \tag{5}$$

We use the larger of the mean and median as a conservative reference. The mean captures the aggregate error level across preceding forecasts, whereas the median reflects their typical error level. Therefore, a point is identified as lagging behind historical performance only when its current error exceeds both statistics.

3.4 Historical Reliability Weighting

During epoch $e+1$, the historical reference $R^{(e)}$ remains fixed, whereas the model parameters are updated batch by batch. HORW performs an additional gradient-free forward pass in evaluation mode using the current model parameters and computes

$$E_{i,h,c}^{\mathrm{cur}} = d\left(\mathrm{sg}\left(\hat{Y}_{i,h,c}^{\mathrm{eval}}\right), Y_{i,h,c}\right), \tag{6}$$

where $\hat{Y}_{i,h,c}^{\mathrm{eval}}$ denotes the evaluation-mode prediction and $\mathrm{sg}(\cdot)$ denotes stop-gradient. This prediction is used only to compute the weighting signal; the forecasting loss is computed from a separate training-mode prediction for backpropagation. For valid historical references, we define a normalized cross-window performance gap as

$$\rho_{i,h,c} = \frac{\left[E_{i,h,c}^{\mathrm{cur}} - R_{i,h,c}^{(e)}\right]_+}{E_{i,h,c}^{\mathrm{cur}} + \epsilon}, \tag{7}$$

where $[z]_+ = \max(z,0)$ and $\epsilon = 10^{-8}$. We set $\rho_{i,h,c} = 0$ for points without preceding references. Thus, only points whose current error is larger than the conservative historical reference receive a nontrivial weighting signal.

The historical gap $\rho_{i,h,c}$ quantifies the extent to which the current error exceeds its timestamp-specific historical reference, but does not by itself prescribe a unique optimization response. HORW therefore uses the signed coefficient β together with $\rho_{i,h,c}$ to determine the point-wise weight:

$$\tilde{w}_{i,h,c} = \begin{cases} (1+\beta\rho_{i,h,c})^{-1}, & \beta > 0, \\ 1+|\beta|\rho_{i,h,c}, & \beta < 0, \\ 1, & \beta = 0. \end{cases} \tag{8}$$

A large gap may indicate room for corrective learning, in which case $\beta < 0$ assigns the point greater relative weight; alternatively, it may reflect cross-window difficulty or instability, in which case $\beta > 0$ attenuates the point's relative influence. When $\beta = 0$, HORW recovers the original training objective.

The weights are normalized within each batch,

$$w_{i,h,c} = \frac{\tilde{w}_{i,h,c}}{\max\left(\mathrm{Mean}(\tilde{\mathbf{w}}),\ \epsilon\right)}, \qquad \epsilon = 10^{-8}. \tag{9}$$

Here, $\mathrm{Mean}(\tilde{\mathbf{w}})$ is taken over all prediction points in the current batch. Provided that this mean is at least ϵ, the normalized weights have unit mean. Normalization preserves their relative ordering.

3.5 HORW-WEIGHTED FORECASTING OBJECTIVE

Let $\ell_{\text{base}}(\hat{Y}_{i,h,c}, Y_{i,h,c})$ denote the original point-wise forecasting loss, where $\hat{Y}_{i,h,c}$ is obtained from the training-mode forward pass. The HORW-weighted forecasting component is

$$\mathcal{L}_{\text{HORW}} = \frac{1}{|\Omega|} \sum_{(i,h,c)\in\Omega} w_{i,h,c}\, \ell_{\text{base}} \left(\hat{Y}_{i,h,c}, Y_{i,h,c} \right), \tag{10}$$

where Ω denotes all prediction points in the current batch. The historical references and weights are treated as constants during backpropagation, so the weighting mechanism introduces no additional gradient path.

For point-wise objectives such as MSE and MAE, ℓ_{base} is respectively the squared or absolute prediction error. For training objectives that combine a point-wise forecasting loss with an auxiliary or regularization term, HORW applies historical weighting only to the point-wise forecasting component while preserving the auxiliary or regularization term unchanged.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate HORW on 7 widely used multivariate time-series forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, ILI, Exchange, and Weather. Detailed descriptions of the datasets, such as the train/validation/test splits, are provided in Appendix B.

Backbones. We select 3 representative time-series forecasting models with diverse architectures to comprehensively evaluate HORW. Specifically, we include the Transformer-based model PatchTST (Nie et al., 2023), the linear model DLinear (Zeng et al., 2023), and the hybrid MLP–CNN model xPatch (Stitsyuk & Choi, 2025).

Implementation details. We use the original implementations of the models from their respective repositories. When applying HORW, we retain all original model hyperparameters, dataset-specific experimental settings, and random seeds, and tune only the HORW coefficient β. HORW is used only during training and does not modify the model architecture or inference procedure. The effects of restricting the weighting direction and eliminating configuration-specific selection of β are examined in Appendix K. All experiments are implemented in PyTorch on a server equipped with three NVIDIA RTX 4080 GPUs (16 GB each), with each run using one GPU. Additional implementation details are provided in Appendix C.

4.2 MAIN RESULTS

Table 1 reports the primary forecasting results averaged over all prediction horizons. These experiments follow the random-seed settings of the original backbone implementations, with identical seeds used for each backbone and its HORW-enhanced counterpart. Under this protocol, HORW improves the average MSE in 20 of the 21 backbone–dataset combinations and matches the remaining one (xPatch on ETTm2). The MAE results exhibit a broadly similar pattern.

For DLinear, HORW improves the average MSE on all seven datasets, with particularly notable gains on Exchange (0.369 to 0.250), ETTh2 (0.469 to 0.400), and ILI (2.347 to 2.175). PatchTST also benefits on all seven datasets under the primary protocol; for example, its average MSE decreases from 1.580 to 1.439 on ILI and from 0.419 to 0.413 on ETTh1. For xPatch, HORW improves six datasets and preserves the performance on ETTm2, reducing the average MSE on ETTh1 from 0.444 to 0.420 and on ILI from 1.383 to 1.313. Overall, these primary results show that HORW improves or preserves forecasting accuracy across diverse backbone architectures.

As a supplementary robustness analysis, we additionally repeat the experiments using random seeds 2021, 2023, and 2025 and report the mean and standard deviation in Appendix E. Across these runs, HORW improves or matches the average MSE in 19 of the 21 backbone–dataset combinations and the average MAE in 20 of 21, although the MSE results on ILI are less stable across seeds. These additional experiments assess sensitivity to random initialization and are reported separately from the primary results in Table 1, which follow the respective official backbone protocols.

To further characterize how HORW operates, we examine historical-reference coverage and raw weighting-signal activation, defined by $\rho > 0$. Across the evaluated configurations whose selected

checkpoints use historical memory, valid references cover 95.7%–99.9% of prediction points, and 23.7%–40.5% of all prediction points have $\rho > 0$. These results show that historical references are broadly available, while the raw weighting signal is selectively activated when the current error exceeds its corresponding historical reference.

Detailed per-horizon results, coverage and activation statistics, and forecasting visualizations are provided in Appendices D.1, H, and F, respectively. Efficiency results are reported in Appendix G.

Table 1: Long-term forecasting results. All values are averaged over four prediction horizons: $\{24, 36, 48, 60\}$ for ILI and $\{96, 192, 336, 720\}$ for all other datasets. "Ori" denotes the original backbone without HORW. The better result in each pair is highlighted in **bold**.

Models	DLinear				PatchTST				xPatch			
	Ori		+HORW		Ori		+HORW		Ori		+HORW	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.445	0.454	**0.428**	**0.438**	0.419	0.432	**0.413**	**0.430**	0.444	0.438	**0.420**	**0.431**
ETTh2	0.469	0.463	**0.400**	**0.422**	0.330	**0.379**	**0.327**	0.380	0.342	0.383	**0.339**	**0.381**
ETTm1	0.359	0.381	**0.357**	**0.379**	0.353	0.382	**0.350**	**0.380**	0.352	0.372	**0.348**	**0.369**
ETTm2	0.283	0.345	**0.270**	**0.330**	0.258	0.315	**0.255**	**0.314**	**0.252**	**0.308**	**0.252**	**0.308**
ILI	2.347	1.089	**2.175**	**1.041**	1.580	0.852	**1.439**	**0.800**	1.383	0.718	**1.313**	**0.702**
Exchange	0.369	0.418	**0.250**	**0.358**	0.394	0.426	**0.379**	**0.417**	0.364	0.403	**0.360**	**0.401**
Weather	0.247	0.300	**0.245**	**0.297**	0.229	0.265	**0.228**	**0.264**	0.221	0.251	**0.219**	**0.249**

4.3 Comparison with AliO

AliO (Park et al., 2025) is the most closely related overlapping-window optimization method to HORW. Table 2 compares AliO and HORW using the same DLinear backbone and training protocol. Both methods are applied only during training, while validation, early stopping, and model selection are based on the original forecasting MSE. For AliO, we perform grid search over the method-specific hyperparameter ranges reported in its original paper and select the best hyperparameters for model construction.

HORW achieves lower average MSE and MAE than AliO on 5 of the 7 datasets: ETTh2, ETTm1, ETTm2, ILI, and Exchange. The largest gains are observed on Exchange, where the average MSE decreases from 0.420 to 0.250 and the MAE decreases from 0.437 to 0.358. HORW also reduces the average MSE from 0.455 to 0.400 on ETTh2 and from 2.289 to 2.175 on ILI. AliO obtains slightly better results on ETTh1 and Weather.

Table 2: Comparison between AliO and HORW on DLinear. All values are averaged over 4 prediction horizons. Detailed per-horizon results are provided in Appendix D.2.

Model	DLinear			
	+AliO		+HORW	
Metric	MSE	MAE	MSE	MAE
ETTh1	**0.425**	**0.437**	0.428	0.438
ETTh2	0.455	0.454	**0.400**	**0.422**
ETTm1	0.358	0.380	**0.357**	**0.379**
ETTm2	0.278	0.341	**0.270**	**0.330**
ILI	2.289	1.074	**2.175**	**1.041**
Exchange	0.420	0.437	**0.250**	**0.358**
Weather	**0.244**	**0.296**	0.245	0.297

Overall, under the unified training protocol, HORW is the more effective cross–window optimization method in most settings. Although AliO is marginally better on ETTh1 and Weather, HORW achieves substantially larger gains on Exchange, ETTh2, and ILI. These results indicate that aggregating timestamp-specific errors from multiple preceding views provides a more informative optimization signal than pairwise output alignment for determining the point-wise optimization strength.

Table 3: Effect of adding HORW to DLinear objectives with auxiliary losses. All results are averaged over 4 prediction horizons. Detailed per-horizon results are provided in Appendix D.3.

Models	DLinear											
	AliO		+HORW		PS Loss		+HORW		FreDF		+HORW	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.425	0.437	**0.422**	**0.433**	0.434	0.444	**0.422**	**0.433**	0.423	0.436	**0.420**	**0.433**
ETTh2	0.455	0.454	**0.375**	**0.408**	0.409	0.423	**0.404**	**0.420**	0.413	0.431	**0.402**	**0.423**
ILI	2.289	1.074	**2.261**	**1.068**	2.174	1.044	**2.087**	**1.013**	2.082	1.005	**2.058**	**0.998**
Exchange	0.420	0.437	**0.249**	**0.355**	0.379	0.416	**0.366**	**0.407**	0.369	0.417	**0.251**	**0.358**

4.4 Improving Forecasting Objectives with Auxiliary Losses

We evaluate, using DLinear, the gains from adding HORW to 3 composite objectives that combine the original point-wise forecasting loss with AliO (Park et al., 2025), PS Loss (Kudrat et al., 2025), and FreDF (Wang et al., 2025), respectively. We first tune each auxiliary method over its reported hyperparameter range and then fix its configuration while tuning only the HORW coefficient β. Table 3 shows that HORW improves both MSE and MAE in all 12 method–dataset combinations. Representative average-MSE reductions include AliO from 0.455 to 0.375 on ETTh2 and from 0.420 to 0.249 on Exchange, PS Loss from 2.174 to 2.087 on ILI, and FreDF from 0.369 to 0.251 on Exchange. These results demonstrate that HORW is complementary to diverse auxiliary objectives.

4.5 Ablation Study

We evaluate the components of HORW on DLinear by reversing the weighting direction, retaining only the latest preceding view, using the mean or median alone, replacing the timestamp-specific historical reference with global or current-error references, shuffling the timestamp correspondence of the reference, and using a same-snapshot comparison.

The global-reference variant uses the maximum of the channel-wise mean and median squared errors pooled over all training windows and forecast positions in the previous epoch's error bank. The current-error variant instead computes these statistics across samples and forecast positions in the current mini-batch, using the current model in evaluation mode. Both variants retain HORW's original reference-validity mask and weighting function. The shuffled-timestamp variant jointly permutes the historical mean and median references across valid windows within each forecast position, using the same permutation across channels. This preserves the reference distributions and validity mask while disrupting their correspondence with the current target timestamps. For the same-snapshot variant, both the current-point errors used for weighting and the preceding-view reference errors are computed using the same frozen model at the end of epoch e. The resulting point-wise ratios ρ and unnormalized weights remain fixed throughout epoch $e+1$, while weight normalization is performed within each current mini-batch. The supervised loss is still computed using the model being optimized. All variants retain the backbone hyperparameters and training settings of full HORW. Except for the opposite-direction and same-snapshot variants, each ablation independently selects β from the same search space as full HORW using validation MSE for each dataset–horizon configuration. The opposite-direction variant negates the coefficient selected for full HORW while preserving its magnitude, whereas the same-snapshot variant retains that coefficient without retuning. These two variants therefore provide comparisons at the full-model-selected coefficient; the remaining variants assess component changes after independent coefficient selection.

Table 4 shows that full HORW achieves lower MSE and MAE than the opposite-direction variant on all seven datasets. This demonstrates the importance of the selected direction at the evaluated coefficient magnitudes. Compared with the independently tuned latest-only variant, full HORW achieves lower MSE on six datasets and matches it on ETTm1. It also improves MAE on five datasets, matches it on ETTh1, and is slightly worse on ETTm1. These results support aggregating multiple preceding views rather than relying exclusively on the nearest preceding view.

The benefits of timestamp-specific historical references vary across datasets. Full HORW achieves lower MSE than both the global-reference and current-error variants on ETTh2, ETTm2, ILI, and Exchange. However, the global reference achieves lower MSE on ETTh1, and the current-error reference achieves lower MSE on ETTh1 and ETTm1. Both variants also obtain lower MAE on several datasets. Because these variants change the reference pool as well as its timestamp specificity, the shuffled-timestamp variant provides a more targeted comparison of timestamp correspondence. Full HORW achieves lower MSE than this variant on five datasets and lower MAE on four, while the shuffled variant performs better on both metrics on ETTh1 and ILI. Together, these results support the usefulness of timestamp alignment in most evaluated MSE comparisons, particularly on Exchange and ETTh2, without establishing a uniform advantage across all datasets and metrics.

Full HORW achieves lower MSE than the same-snapshot variant on five datasets and matches it on ETTm1 and Weather. It also achieves lower MAE on six datasets and matches it on Weather. At the coefficient selected for full HORW, these results support combining a frozen historical reference with a current-error signal that is recomputed using the updated model during optimization, rather than fixing the point-wise ratios at the memory-construction snapshot. However, this comparison changes both the model-state relationship between the two errors and the freshness of the weighting signals, and the same-snapshot coefficient is not retuned. It therefore does not isolate the contribution of cross-epoch model-state differences alone.

Finally, full HORW improves MSE over the mean-only variant on ETTh2, ETTm1, and ILI, matches it on ETTh1, Exchange, and Weather, and performs slightly worse on ETTm2. Compared with the median-only variant, full HORW improves MSE on four datasets, matches it on Weather, and performs worse on ETTm2 and ILI. The median-only variant also achieves the best ILI MAE among the compared variants. Thus, neither statistic is uniformly preferable. The combined $\max(\text{mean}, \text{median})$ reference requires the current error to exceed both the average and median historical error levels, offering a competitive combination across datasets rather than a uniformly superior choice.

We also provide an additional experiment in Appendix I to distinguish HORW from generic difficulty-based weighting.

Table 4: Ablation study of HORW on DLinear. OWD: opposite weight direction; CER: current-error reference; STR: shuffled-timestamp reference; SSH: same-snapshot HORW. All results are averaged over 4 prediction horizons. Detailed per-horizon ablation results are provided in Appendix D.4.

Models	DLinear																	
	Full HORW		OWD		Latest only		Mean only		Median only		Global reference		CER		STR		SSH	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.428	0.438	0.454	0.463	0.429	0.438	0.428	0.438	0.429	0.441	**0.425**	**0.434**	0.426	0.435	0.427	0.436	0.470	0.472
ETTh2	**0.400**	**0.422**	0.475	0.478	0.437	0.449	0.404	0.432	0.403	0.431	0.427	0.437	0.422	0.435	0.423	0.439	0.466	0.455
ETTm1	0.357	0.379	0.362	0.383	0.357	0.378	0.358	0.380	0.359	0.382	0.357	**0.374**	**0.356**	**0.374**	0.358	0.378	0.357	0.380
ETTm2	0.270	**0.330**	0.325	0.376	0.279	0.341	**0.268**	**0.330**	**0.268**	**0.330**	0.271	0.335	0.274	0.334	0.273	0.334	0.291	0.348
ILI	2.175	1.041	2.400	1.102	2.264	1.064	2.179	1.042	**2.159**	**1.034**	2.194	1.041	2.192	1.039	2.166	1.035	2.328	1.080
Exchange	**0.250**	0.358	0.383	0.432	0.356	0.407	**0.250**	**0.357**	0.253	0.358	0.366	0.414	0.367	0.416	0.368	0.416	0.375	0.424
Weather	**0.245**	0.297	0.249	0.303	0.246	0.299	**0.245**	0.297	**0.245**	0.297	**0.245**	**0.292**	**0.245**	0.292	0.246	0.299	**0.245**	0.297

5 Sensitivity to the HORW Coefficient β

We evaluate the sensitivity of HORW to β on DLinear using ILI and ETTh2, as shown in Figure 2. Most short- and medium-horizon configurations remain stable across a broad range of coefficients, indicating that HORW does not require precise tuning in these settings. Longer horizons exhibit larger variations, but they also obtain substantially greater improvements over the unweighted objective. In particular, multiple positive coefficients consistently improve the ILI-60 and ETTh2-720 results over $\beta = 0$, with the best performance obtained at $\beta = 3$. These observations indicate that the preferred weighting strength is configuration dependent, while useful performance gains generally persist over a nontrivial interval rather than at a single isolated coefficient.

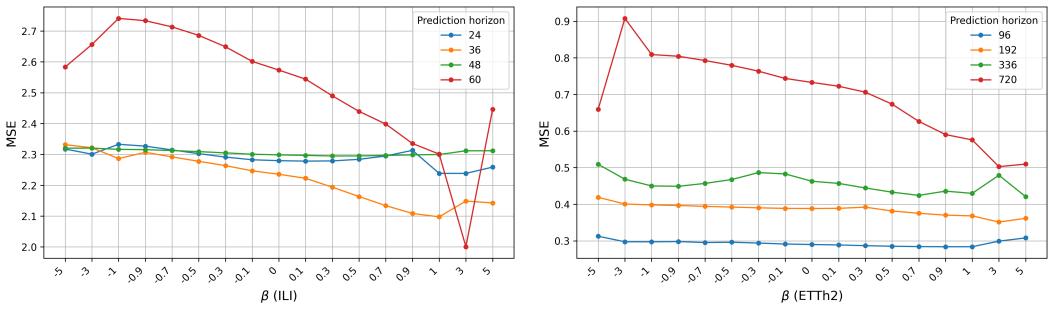

Figure 2: Sensitivity of DLinear+HORW to the coefficient β. Each curve corresponds to one prediction horizon, and lower MSE is better. $\beta = 0$ recovers the unweighted forecasting objective.

6 Conclusion

This paper introduced Historical Overlap Reliability Weighting (HORW) for long-term time-series forecasting. HORW is motivated by the observation that sliding-window training produces multiple preceding forecasts for the same target timestamp, whose error evidence is typically discarded by conventional point-wise optimization. By collecting these errors with the frozen final model of each epoch, HORW constructs timestamp-specific historical references and uses them to adaptively reweight the original point-wise forecasting loss in the subsequent epoch.

Experiments across diverse forecasting backbones, datasets, and training objectives demonstrate that this timestamp-specific historical relative-error signal improves forecasting performance in most evaluated cases. HORW introduces no trainable parameters and requires tuning only a single additional coefficient, β. It preserves the original backbone architecture and experimental configuration, while incurring additional training-time overhead from historical-memory construction; validation and test-time inference remain unchanged. An efficiency-oriented variant that refreshes the memory every two epochs reduces the number of memory-construction operations by approximately half while retaining improvements over the original backbone on most datasets, without retuning β; detailed results are provided in Appendix J. More broadly, our results suggest that accumulated overlap errors provide a useful source of supervision for determining the optimization priority of individual forecasting points and that the refresh frequency offers a practical trade-off between reference freshness and training efficiency.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on experimental design and to assist in code development and debugging. We did not use generative AI tools to generate synthetic datasets, formulate mathematical claims or proofs, assist in proof writing, clean or reformat datasets, or conduct qualitative or thematic data analysis, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to assist in translating, drafting, and editing portions of the manuscript for clarity and readability. We reviewed all AI-assisted work: experimental suggestions were independently evaluated by the authors, AI-assisted code was inspected and tested through the reported experiments, and AI-assisted text was verified, revised, and approved by the authors. All experimental results were produced using the authors' implementations and checked through the reported experimental procedures. We take responsibility for the final content of this work, including any text, claims, code, results, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main text specifies HORW through its formulation and training procedure. The appendix provides the experimental details needed to reproduce the study, including dataset descriptions, backbone configurations, hyperparameter settings, and complete results. The code will be released upon acceptance.

REFERENCES

Daan Caljon, Jeff Vercauteren, Simon De Vos, Wouter Verbeke, and Jente Van Belle. Using dynamic loss weighting to boost improvements in forecast stability. *International Journal of Forecasting*, 42(2):344–358, 2026. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2025.07.002. URL https://www.sciencedirect.com/science/article/pii/S0169207025000615.

Hao Cheng, Qingsong Wen, Yang Liu, and Liang Sun. Robusttsf: Towards theory and design of robust time series forecasting with anomalies. In *International Conference on Learning Representations*, volume 2024, pp. 5787–5813, 2024.

Maurice Kraus, Felix Divo, Devendra Singh Dhami, and Kristian Kersting. xLSTM-Mixer: Multivariate time series forecasting by mixing via scalar memories. In *Advances in Neural Information Processing Systems*, volume 38, pp. 6547–6581, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/09e38101e74a89129ccd0d0756ed36b3-Paper-Conference.pdf.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31841–31859, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kudrat25a.html.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pp. 95–104, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210006. URL https://doi.org/10.1145/3209978.3210006.

Jaebin Lee and Hankook Lee. TimePerceiver: An encoder-decoder framework for generalized time-series forecasting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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, 2022. doi: 10.52202/068431-0421.

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, pp. 39815–39840. PMLR, 2025. URL https://proceedings.mlr.press/v267/liu25cb.html.

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*, volume 2024, pp. 11116–11140, 2024.

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

Junwoo Park, Jungsoo Lee, Youngin Cho, Woncheol Shin, Dongmin Kim, Jaegul Choo, and Edward Choi. Deep imbalanced time-series forecasting via local discrepancy density. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2023*, 2023.

Kwangryeol Park, Jaeho Kim, and Seulki Lee. Alio: Output alignment matters in long-term time series forecasting. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 119563–119608, 2025. doi: 10.52202/085713-3992.

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. DBLoss: Decomposition-based loss function for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 38, pp. 27741–27768, 2025. doi: 10.52202/085713-0936.

Xiangfei Qiu, Yuhan Zhu, Zhengyu Li, Xingjian Wu, Bin Yang, and Jilin Hu. DAG: A dual correlation network for time series forecasting with exogenous variables. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=jntRRU4zcb.

Artyom Stitsyuk and Jaesik Choi. xPatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20601–20609, 2025.

Hao Wang, Lichen Pan, Yuan Shen, Zhichao Chen, Degui Yang, Yifei Yang, Sen Zhang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. FreDF: Learning to forecast in the frequency domain. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4A9IdSa1ul.

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, pp. 22419–22430, 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*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Jiang You, Arben Cela, René Natowicz, Jacob Ouanounou, and Patrick Siarry. Robust time series forecasting with non-heavy-tailed gaussian loss-weighted sampler. *arXiv preprint arXiv:2406.13871*, 2024. URL https://arxiv.org/abs/2406.13871.

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. In *Advances in Neural Information Processing Systems*, volume 38, pp. 60573–60632, 2025. doi: 10.52202/085713-2025.

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.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

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 AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

A Computational Cost

Let N, H, and C denote the number of training windows, forecast horizon, and number of variables, respectively. Let T_{base} denote the cost of an original training epoch and T_{forward} the cost of a forward-only pass over the training set. The per-epoch cost of HORW is

$$T_{\text{HORW}} = T_{\text{base}} + T_{\text{forward}} + T_{\text{compare}} + O(NHC) + T_{\text{median}}, \tag{11}$$

where T_{compare} accounts for the additional evaluation-mode forward passes used to compute current errors during training. The $O(NHC)$ term covers error recording, prefix-mean construction, and point-wise weighting.

The current implementation computes exact preceding-view medians using batched masked tensors. For $N \geq H$, this involves $O(NH^2C)$ candidate entries across the training set; the resulting cost T_{median} depends on the underlying median-reduction implementation. GPU batching reduces execution overhead but does not eliminate this quadratic dependence on the forecast horizon in candidate processing.

The persistent error and reference memories require $O(NHC)$ storage. Median construction additionally uses $O(bH^2C)$ temporary storage, where b is the internal block size. The implementation selects b using a nominal 64 MiB budget for the median-input tensor, with a minimum block size of one; this budget does not bound total peak GPU memory.

All additional computation and storage are used only during training. HORW leaves the inference architecture unchanged and requires no historical memory during validation or testing. Empirical efficiency results are reported in Appendix G.

B Dataset description

All experiments are conducted on 7 publicly available multivariate time-series forecasting datasets. Their detailed statistics are summarized in Table 5. The ETT datasets contain electricity-transformer measurements, where ETTh1 and ETTh2 are sampled hourly and ETTm1 and ETTm2 are sampled every 15 minutes (Zhou et al., 2021). The ILI [1] dataset records weekly influenza-like illness ratios and is used to evaluate short-horizon forecasting. The Exchange dataset contains daily exchange rates of eight countries (Lai et al., 2018). The Weather dataset consists of 21 meteorological variables recorded at 10-minute intervals (Wu et al., 2021). We follow the standard chronological train/validation/test splits used in prior long-term forecasting studies (Zhou et al., 2021; Wu et al., 2021; Liu et al., 2024).

Table 5: Dataset description."Dim." denotes the number of variables.

Dataset	Dim	Forecast length	Domain	Frequency	Train / validation / test
ETTh1	7	96, 192, 336, 720	Electricity	Hourly	8545/2881/2881
ETTh2	7	96, 192, 336, 720	Electricity	Hourly	8545/2881/2881
ETTm1	7	96, 192, 336, 720	Electricity	15 mins	34465/11521/11521
ETTm2	7	96, 192, 336, 720	Electricity	15 mins	34465/11521/11521
ILI	7	24,36,48,60	Health	Weekly	617/74/170
Exchange	8	96, 192, 336, 720	Exchange Rates	Daily	5120/665/1422
Weather	21	96, 192, 336, 720	Weather	10 mins	36792/5271/10540

[1] https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

C IMPLEMENTATION DETAILS

C.1 BACKBONE CONFIGURATIONS

We retain the original backbone configurations, training hyperparameters, and dataset-specific input-length settings without modification. For DLinear (Zeng et al., 2023) and PatchTST (Nie et al., 2023), the input length is 336 for ETT, Exchange, and Weather, and 104 for ILI. For xPatch (Stitsyuk & Choi, 2025), we follow its original dataset-specific input lengths: 512 for ETTh1, ETTh2, and ETTm2; 336 for ETTm1; 96 for Exchange; 720 for Weather; and 104 for ILI.

All data are z-score normalized using a `StandardScaler` fitted exclusively on the training split. The fitted normalization statistics are then applied unchanged to the validation and test splits.

Since the official xPatch [2] repository provides multiple experimental scripts, we use `xPatch_fair` as the backbone configuration for all xPatch experiments.

C.2 HORW HYPERPARAMETER SELECTION

HORW introduces a single signed scalar hyperparameter, β. Its sign determines whether larger historical gaps receive greater or smaller relative weights, while its magnitude controls the strength of this response. For each backbone–dataset–horizon configuration, we select β from the following candidate set based on validation MSE:

$$\{-5.0, -3.0, -1.0, -0.9, -0.7, -0.5, -0.3, -0.1, 0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0, 3.0, 5.0\}.$$

Table 6 summarizes the directions of the selected coefficients across the 28 dataset–horizon configurations for each backbone. Both positive and negative coefficients are selected, and their relative frequencies differ substantially across backbones. DLinear more frequently selects attenuation, whereas PatchTST and xPatch more frequently select emphasis. This distribution indicates that no single response direction uniformly dominates across backbone and forecasting configurations. The four cases in which $\beta = 0$ for xPatch also show that HORW can recover the original objective when reweighting does not improve validation performance.

When applying HORW, we retain all original backbone hyperparameters without modification and tune only the HORW coefficient β. Additional experiments restricting its direction or fixing its value are reported in Appendix K.

Table 6: Distribution of the validation-selected β directions across the 28 dataset–horizon configurations for each backbone.

Backbone	$\beta < 0$	$\beta = 0$	$\beta > 0$
DLinear	8	0	20
PatchTST	22	0	6
xPatch	16	4	8

C.3 INTEGRATING HORW WITH POINT-WISE OBJECTIVES AND THEIR AUXILIARY-LOSS EXTENSIONS

Because HORW reweights only the original point-wise forecasting loss, it can be used to further improve training objectives that combine a point-wise forecasting loss with an auxiliary or regularization term (such as AliO (Park et al., 2025), PS Loss (Kudrat et al., 2025), and FreDF (Wang et al., 2025)).

Specifically, the combined objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{HORW}} + \mathcal{L}_{\mathrm{aux}},$$

where $\mathcal{L}_{\mathrm{HORW}}$ is the HORW-reweighted original forecasting loss, and $\mathcal{L}_{\mathrm{aux}}$ denotes the method-specific auxiliary or regularization objective, such as the PS Loss, the time- and frequency-domain terms in AliO, or the FreDF regularization term, together with their corresponding coefficients.

[2] https://github.com/stitsyuk/xPatch

For these methods, we first determine their optimal method-specific hyperparameters using the ranges reported in their respective papers. We then fix these hyperparameters and search only for the HORW coefficient β.

D Complete Experimental Results

D.1 Detailed Long-Term Forecasting Results

Table 7 provides the complete long-term forecasting results for every prediction horizon, complementing the averaged results in Table 1. The detailed results show that the improvements are not restricted to a single horizon. For example, HORW reduces the DLinear MSE on Exchange from 0.333 to 0.269 at horizon 336 and from 0.898 to 0.491 at horizon 720. On ETTh2, it reduces the MSE from 0.463 to 0.421 at horizon 336 and from 0.733 to 0.510 at horizon 720.

Table 7: Complete long-term forecasting results at each prediction horizon. "Ori" denotes the original backbone without HORW. The "Avg" row reports the average over four prediction horizons. The better result in each pair is highlighted in **bold**.

Model		DLinear				PatchTST				xPatch			
		Ori		+HORW		Ori		+HORW		Ori		+HORW	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.384	0.405	**0.380**	**0.399**	0.382	0.405	**0.381**	**0.404**	**0.369**	**0.397**	0.370	0.401
	192	0.443	0.450	**0.405**	**0.413**	**0.414**	**0.421**	**0.414**	**0.421**	0.424	0.426	**0.409**	**0.423**
	336	0.447	0.448	**0.434**	**0.439**	0.431	0.436	**0.422**	**0.432**	0.450	0.437	**0.427**	**0.431**
	720	0.504	0.515	**0.495**	**0.499**	0.449	0.466	**0.435**	**0.461**	0.533	0.494	**0.476**	**0.472**
	Avg	0.445	0.454	**0.428**	**0.438**	0.419	0.432	**0.413**	**0.430**	0.444	0.438	**0.420**	**0.431**
ETTh2	96	**0.290**	**0.353**	0.308	0.358	0.275	**0.337**	**0.274**	**0.337**	0.275	0.333	**0.270**	**0.331**
	192	0.388	0.422	**0.362**	**0.392**	**0.338**	**0.378**	**0.338**	**0.378**	0.337	0.375	**0.334**	**0.373**
	336	0.463	0.473	**0.421**	**0.434**	0.329	**0.379**	**0.321**	0.385	0.365	0.398	**0.361**	**0.395**
	720	0.733	0.606	**0.510**	**0.502**	0.379	0.422	**0.375**	**0.420**	**0.391**	**0.426**	**0.391**	**0.426**
	Avg	0.469	0.463	**0.400**	**0.422**	0.330	**0.379**	**0.327**	0.380	0.342	0.383	**0.339**	**0.381**
ETTm1	96	0.301	0.345	**0.296**	**0.340**	0.292	0.343	**0.290**	**0.341**	0.289	0.332	**0.287**	**0.329**
	192	0.336	**0.366**	**0.335**	**0.366**	0.336	0.371	**0.332**	**0.369**	0.330	0.357	**0.327**	**0.355**
	336	0.372	0.389	**0.371**	**0.388**	0.366	0.392	**0.365**	**0.389**	0.364	0.381	**0.362**	**0.379**
	720	0.427	**0.423**	**0.426**	**0.423**	0.418	0.424	**0.413**	**0.420**	0.426	0.417	**0.418**	**0.413**
	Avg	0.359	0.381	**0.357**	**0.379**	0.353	0.382	**0.350**	**0.380**	0.352	0.372	**0.348**	**0.369**
ETTm2	96	0.172	0.267	**0.167**	**0.262**	0.165	0.255	**0.164**	**0.253**	**0.159**	**0.245**	**0.159**	**0.245**
	192	0.237	0.314	**0.228**	**0.300**	**0.220**	**0.292**	0.221	0.293	**0.217**	**0.286**	0.218	**0.286**
	336	**0.295**	0.359	0.298	**0.351**	0.278	**0.329**	**0.276**	**0.329**	**0.275**	**0.325**	**0.275**	**0.325**
	720	0.427	0.439	**0.388**	**0.407**	0.368	0.385	**0.362**	**0.383**	**0.357**	**0.377**	**0.357**	**0.377**
	Avg	0.283	0.345	**0.270**	**0.330**	0.258	0.315	**0.255**	**0.314**	**0.252**	**0.308**	**0.252**	**0.308**
ILI	24	2.280	1.061	**2.238**	**1.036**	1.584	0.840	**1.329**	**0.745**	1.334	0.699	**1.167**	**0.653**
	36	2.235	1.059	**2.149**	**1.053**	1.442	0.831	**1.400**	**0.809**	1.329	0.683	**1.293**	**0.678**
	48	**2.298**	1.079	2.312	**1.077**	1.685	0.853	**1.459**	**0.795**	1.358	0.706	**1.301**	**0.697**
	60	2.573	1.157	**2.000**	**0.997**	1.608	0.886	**1.570**	**0.852**	1.512	0.785	**1.491**	**0.783**
	Avg	2.347	1.089	**2.175**	**1.041**	1.580	0.852	**1.439**	**0.800**	1.383	0.718	**1.313**	**0.702**
Exchange	96	0.085	0.209	**0.083**	**0.205**	**0.095**	**0.215**	**0.095**	**0.215**	**0.081**	**0.197**	**0.081**	**0.197**
	192	0.162	0.296	**0.156**	**0.292**	0.213	0.330	**0.202**	**0.324**	0.175	0.296	**0.170**	**0.293**
	336	0.333	0.441	**0.269**	**0.387**	0.395	0.457	**0.351**	**0.431**	0.343	0.421	**0.333**	**0.415**
	720	0.898	0.725	**0.491**	**0.547**	0.872	0.703	**0.868**	**0.698**	**0.857**	**0.698**	**0.857**	**0.698**
	Avg	0.369	0.418	**0.250**	**0.358**	0.394	0.426	**0.379**	**0.417**	0.364	0.403	**0.360**	**0.401**
Weather	96	0.174	0.233	**0.172**	**0.229**	0.152	0.200	**0.150**	**0.198**	**0.146**	**0.185**	**0.146**	**0.185**
	192	0.218	0.278	**0.214**	**0.271**	0.197	0.243	**0.195**	**0.241**	0.192	0.230	**0.188**	**0.228**
	336	0.263	0.314	**0.262**	**0.313**	**0.248**	**0.282**	**0.248**	0.283	0.240	0.270	**0.238**	**0.268**
	720	**0.332**	**0.374**	**0.332**	**0.374**	**0.319**	**0.335**	**0.319**	**0.335**	0.307	0.320	**0.303**	**0.316**
	Avg	0.247	0.300	**0.245**	**0.297**	0.229	0.265	**0.228**	**0.264**	0.221	0.251	**0.219**	**0.249**

D.2 Detailed Comparison with AliO

Table 8 presents the complete per-horizon results corresponding to the averaged comparison in Table 2.

The detailed results further clarify the comparison between AliO and HORW. On Exchange, HORW outperforms AliO at all four prediction horizons in both MSE and MAE. The largest gain occurs at horizon 720, where the MSE decreases from 1.098 to 0.491. On ETTh2, HORW achieves better results at horizons 192, 336, and 720, reducing the MSE from 0.690 to 0.510 at horizon 720.

The results on ILI, ETTh1, and Weather are more horizon-dependent. On ILI, HORW achieves lower MSE at horizons 24, 36, and 60, whereas AliO performs better at horizon 48. In particular,

HORW substantially improves the horizon-60 result, reducing the MSE from 2.391 to 2.000 and the MAE from 1.124 to 0.997. On ETTh1 and Weather, neither method consistently dominates across all prediction horizons, resulting in only small differences in their average results.

Table 8: Complete comparison between AliO and HORW on DLinear at each prediction horizon. "Avg" denotes the average over four prediction horizons. Lower values are better, and the better result in each pair is highlighted in **bold**.

Model		DLinear			
		+AliO		+HORW	
Metric		MSE	MAE	MSE	MAE
ETTh1	96	**0.370**	**0.392**	0.380	0.399
	192	0.407	0.417	**0.405**	**0.413**
	336	0.439	0.441	**0.434**	**0.439**
	720	**0.483**	**0.497**	0.495	0.499
	Avg	**0.425**	**0.437**	0.428	0.438
ETTh2	96	**0.287**	**0.350**	0.308	0.358
	192	0.367	0.404	**0.362**	**0.392**
	336	0.475	0.476	**0.421**	**0.434**
	720	0.690	0.587	**0.510**	**0.502**
	Avg	0.455	0.454	**0.400**	**0.422**
ETTm1	96	0.298	0.342	**0.296**	**0.340**
	192	**0.334**	**0.364**	0.335	0.366
	336	0.374	0.392	**0.371**	**0.388**
	720	**0.426**	**0.422**	**0.426**	0.423
	Avg	0.358	0.380	**0.357**	**0.379**
ETTm2	96	**0.167**	**0.261**	**0.167**	0.262
	192	0.230	0.310	**0.228**	**0.300**
	336	0.306	0.365	**0.298**	**0.351**
	720	0.409	0.427	**0.388**	**0.407**
	Avg	0.278	0.341	**0.270**	**0.330**
ILI	24	2.255	1.053	**2.238**	**1.036**
	36	2.235	**1.051**	**2.149**	1.053
	48	**2.275**	**1.067**	2.312	1.077
	60	2.391	1.124	**2.000**	**0.997**
	Avg	2.289	1.074	**2.175**	**1.041**
Exchange	96	0.086	0.207	**0.083**	**0.205**
	192	0.168	0.304	**0.156**	**0.292**
	336	0.326	0.438	**0.269**	**0.387**
	720	1.098	0.797	**0.491**	**0.547**
	Avg	0.420	0.437	**0.250**	**0.358**
Weather	96	0.175	0.235	**0.172**	**0.229**
	192	0.216	0.274	**0.214**	**0.271**
	336	**0.261**	**0.311**	0.262	0.313
	720	**0.325**	**0.363**	0.332	0.374
	Avg	**0.244**	**0.296**	0.245	0.297

D.3 Detailed Per-Horizon Results for HORW with Composite Forecasting Objectives

Table 9 presents the complete per-horizon results corresponding to the averaged compatibility results in Table 3. The detailed results show that the benefits of HORW can vary across forecasting horizons, while remaining complementary to the underlying auxiliary objectives.

On ETTh2, HORW yields clear improvements at the longer horizons for all three objectives. For example, at horizon 720, the MSE decreases from 0.690 to 0.465 for AliO, from 0.587 to 0.568 for PS Loss, and from 0.594 to 0.554 for FreDF. On ILI, HORW consistently improves PS Loss and FreDF at all four horizons. When combined with AliO, HORW improves the results at horizons 36, 48, and 60, while slightly degrading the horizon-24 result.

The Exchange results further show that HORW can yield substantial long-horizon gains. In particular, adding HORW to AliO reduces the MSE at horizon 720 from 1.098 to 0.459, and adding HORW to FreDF reduces it from 0.898 to 0.491. These results support the conclusion that HORW provides complementary reliability weighting across different auxiliary-loss formulations.

D.4 Detailed Ablation Results

Table 10 reports the complete per-horizon results corresponding to Table 4. Except for the opposite-direction and same-snapshot variants, each ablation independently selects β using validation MSE

Table 9: Complete per-horizon results for adding HORW to DLinear objectives with auxiliary losses. "Avg" denotes the average over 4 prediction horizons. Lower values are better, and the better result within each method pair is highlighted in **bold**.

Model		DLinear											
		AliO		+HORW		PS Loss		+HORW		FreDF		+HORW	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.370**	**0.392**	**0.370**	**0.392**	0.381	0.402	**0.368**	**0.390**	0.380	0.402	**0.373**	**0.396**
	192	**0.407**	0.417	**0.407**	**0.416**	**0.402**	**0.411**	0.403	0.412	0.402	**0.411**	**0.403**	**0.411**
	336	**0.439**	**0.441**	0.440	**0.441**	**0.437**	**0.438**	0.440	0.440	**0.438**	**0.440**	**0.438**	**0.440**
	720	0.483	0.497	**0.472**	**0.483**	0.515	0.523	**0.476**	**0.492**	0.472	0.490	**0.468**	**0.486**
	Avg	0.425	0.437	**0.422**	**0.433**	0.434	0.444	**0.422**	**0.433**	0.423	0.436	**0.420**	**0.433**
ETTh2	96	**0.287**	**0.350**	0.298	0.357	**0.279**	**0.337**	0.280	**0.337**	**0.279**	**0.339**	0.281	**0.339**
	192	0.367	0.404	**0.356**	**0.386**	**0.352**	0.387	**0.352**	**0.386**	**0.354**	0.392	**0.354**	**0.388**
	336	0.475	0.476	**0.381**	**0.410**	0.416	0.435	**0.414**	**0.432**	0.423	0.445	**0.419**	**0.438**
	720	0.690	0.587	**0.465**	**0.480**	0.587	0.533	**0.568**	**0.523**	0.594	0.549	**0.554**	**0.527**
	Avg	0.455	0.454	**0.375**	**0.408**	0.409	0.423	**0.404**	**0.420**	0.413	0.431	**0.402**	**0.423**
ILI	24	**2.255**	**1.053**	2.257	1.054	2.125	0.992	**2.108**	**0.986**	2.109	0.989	**2.100**	**0.986**
	36	2.235	**1.051**	**2.232**	**1.051**	2.072	1.002	**2.060**	**0.995**	2.023	0.984	**1.998**	**0.975**
	48	2.275	1.067	**2.264**	**1.062**	2.110	1.035	**2.026**	**1.009**	2.039	1.006	**2.037**	**1.005**
	60	2.391	1.124	**2.290**	**1.105**	2.387	1.147	**2.155**	**1.060**	2.158	1.040	**2.097**	**1.027**
	Avg	2.289	1.074	**2.261**	**1.068**	2.174	1.044	**2.087**	**1.013**	2.082	1.005	**2.058**	**0.998**
Exchange	96	0.086	0.207	**0.083**	**0.206**	**0.078**	**0.199**	**0.078**	**0.199**	**0.086**	**0.208**	0.087	**0.208**
	192	0.168	0.304	**0.157**	**0.286**	**0.156**	**0.290**	**0.156**	**0.290**	0.157	0.295	**0.156**	**0.290**
	336	0.326	0.438	**0.298**	**0.398**	0.336	0.436	**0.281**	**0.401**	0.333	0.441	**0.269**	**0.387**
	720	1.098	0.797	**0.459**	**0.528**	**0.947**	**0.737**	**0.947**	**0.737**	0.898	0.725	**0.491**	**0.547**
	Avg	0.420	0.437	**0.249**	**0.355**	0.379	0.416	**0.366**	**0.407**	0.369	0.417	**0.251**	**0.358**

over the same search space as full HORW. The opposite-direction variant negates the coefficient selected for full HORW, while the same-snapshot variant retains it without retuning.

Reversing the selected weighting direction generally degrades performance, with several large differences at long horizons. For example, the MSE increases from 0.491 to 0.898 on Exchange at horizon 720, from 0.510 to 0.660 on ETTh2 at horizon 720, and from 2.000 to 2.656 on ILI at horizon 60. The effect is not uniform: on ETTm2 at horizon 96, the opposite direction matches the MSE of full HORW and slightly improves MAE. These results demonstrate sensitivity to the weighting direction at the coefficient magnitudes selected for full HORW.

The latest-only reference remains competitive at several horizons. It improves MSE from 0.380 to 0.370 on ETTh1 at horizon 96 and from 0.308 to 0.284 on ETTh2 at horizon 96. However, it performs worse in several longer-horizon settings: the MSE increases from 0.421 to 0.479 on ETTh2 at horizon 336, from 0.510 to 0.611 on ETTh2 at horizon 720, and from 0.491 to 0.898 on Exchange at horizon 720. Thus, the nearest preceding view can be sufficient in some configurations, while aggregating multiple preceding views provides substantial benefits in others.

The global and current-error references also outperform full HORW in some settings. Both achieve an MSE of 0.289 on ETTh2 at horizon 96, compared with 0.308 for full HORW. On ETTm2 at horizon 336, their MSE values are 0.285 and 0.290, respectively, compared with 0.298 for full HORW. Conversely, full HORW performs better on ETTh2 at horizon 720 (0.510 versus 0.610 and 0.588) and on Exchange at horizon 720 (0.491 versus 0.898 for both variants). The usefulness of timestamp-specific historical references therefore depends on the forecasting configuration; their advantage is not uniform across horizons.

The shuffled-timestamp reference further examines the role of timestamp correspondence. It jointly permutes the historical mean and median references across valid windows within each forecast position, using the same permutation across channels. This preserves the reference distributions and validity mask while disrupting their correspondence with the current target timestamps. Shuffling increases MSE from 0.421 to 0.467 on ETTh2 at horizon 336, from 0.269 to 0.333 on Exchange at horizon 336, and from 0.491 to 0.898 on Exchange at horizon 720. However, the shuffled variant improves both metrics on ILI at horizons 24, 36, and 48, before performing worse at horizon 60. These results support the value of timestamp alignment in several configurations, while also showing that alignment does not yield a consistent advantage at every horizon.

The same-snapshot variant computes both sides of the error comparison using the frozen epoch-e model. The resulting point-wise ratios ρ and unnormalized weights remain fixed throughout epoch $e + 1$, while normalization is performed within each current mini-batch. The supervised loss is

still computed using the model being optimized. Its MSE increases from 0.495 to 0.599 on ETTh1 at horizon 720, from 0.510 to 0.650 on ETTh2 at horizon 720, and from 2.000 to 2.526 on ILI at horizon 60. Nevertheless, it remains close to full HORW on ETTm1 and Weather and improves MSE from 2.312 to 2.219 on ILI at horizon 48. At the coefficient selected for full HORW, the overall results favor updating the current-error signal during optimization. This comparison does not, however, separately identify the effects of model-state differences and signal freshness.

Finally, neither the mean–median combination nor either individual statistic is uniformly best. The mean-only reference improves MSE from 0.510 to 0.502 on ETTh2 at horizon 720 and from 0.298 to 0.290 on ETTm2 at horizon 336. The median-only reference improves both metrics on ILI at horizons 24, 36, and 60, but performs worse at horizon 48. These results show that the stricter $\max(\text{mean}, \text{median})$ reference involves a configuration-dependent trade-off: it requires agreement between the two historical error summaries before activating reweighting, but does not guarantee better performance than either summary alone.

Table 10: Complete ablation results of HORW on DLinear at each prediction horizon. "Avg" denotes the average over four prediction horizons. Lower values are better, and the best result in each row is highlighted in bold.

Model		DLinear																	
		Full HORW		OWD		Latest only		Mean only		Median only		Global reference		CER		STR		SSH	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.380	0.399	0.391	0.413	**0.370**	**0.390**	0.380	0.399	0.379	0.398	0.375	0.393	0.375	0.393	0.376	0.396	0.387	0.406
	192	0.405	0.413	0.442	0.449	0.406	0.414	0.405	0.413	**0.404**	0.413	0.408	0.417	0.407	0.416	**0.404**	**0.410**	0.443	0.449
	336	0.434	0.439	0.481	0.471	**0.433**	**0.432**	0.434	0.439	0.436	0.440	0.441	0.435	0.442	0.436	0.442	0.439	0.452	0.454
	720	0.495	0.499	0.505	0.519	0.507	0.515	0.495	0.499	0.499	0.513	**0.478**	**0.493**	0.480	0.494	0.486	0.499	0.599	0.577
	Avg	0.428	0.438	0.454	0.463	0.429	0.438	0.428	0.438	0.429	0.441	**0.425**	**0.434**	0.426	0.435	0.427	0.436	0.470	0.472
ETTh2	96	0.308	0.358	0.313	0.375	**0.284**	0.349	0.289	0.352	0.294	0.357	0.289	**0.347**	0.289	**0.347**	0.293	0.352	0.316	0.367
	192	**0.362**	**0.392**	0.419	0.440	0.373	0.412	0.368	0.406	0.372	0.409	0.368	0.402	0.367	0.402	0.377	0.412	0.412	0.427
	336	**0.421**	**0.434**	0.509	0.507	0.479	0.476	0.457	0.469	0.429	0.451	0.442	0.451	0.444	0.452	0.467	0.467	0.487	0.473
	720	0.510	0.502	0.660	0.589	0.611	0.561	**0.502**	**0.500**	0.516	0.505	0.610	0.546	0.588	0.539	0.553	0.524	0.650	0.556
	Avg	**0.400**	**0.422**	0.475	0.478	0.437	0.449	0.404	0.432	0.403	0.431	0.427	0.437	0.422	0.435	0.423	0.439	0.466	0.455
ETTm1	96	**0.296**	0.340	0.310	0.352	0.300	0.343	0.300	0.344	**0.296**	0.340	0.298	0.339	0.298	**0.337**	0.300	0.343	0.297	0.341
	192	0.335	0.366	0.336	0.367	0.335	0.366	0.335	0.366	0.342	0.375	**0.334**	**0.362**	**0.334**	**0.362**	0.334	0.364	0.335	0.366
	336	0.371	0.388	0.374	0.391	**0.369**	0.384	0.372	0.388	0.372	0.388	0.370	**0.382**	0.370	**0.382**	0.371	0.387	0.372	0.389
	720	0.426	0.423	0.428	0.423	**0.424**	0.418	0.426	0.423	0.426	0.423	**0.424**	**0.415**	**0.424**	0.416	0.425	0.420	0.426	0.423
	Avg	0.357	0.379	0.362	0.383	0.357	0.378	0.358	0.380	0.359	0.382	0.357	**0.374**	**0.356**	**0.374**	0.358	0.378	0.357	0.380
ETTm2	96	0.167	0.262	0.167	0.260	0.166	0.258	0.167	0.262	0.167	0.262	0.170	0.264	**0.164**	**0.252**	0.164	0.254	0.172	0.267
	192	0.228	0.300	0.245	0.327	0.231	0.308	0.229	**0.299**	0.227	0.301	0.231	0.311	**0.225**	0.300	0.228	0.306	0.244	0.320
	336	0.298	0.351	0.335	0.394	0.296	0.360	0.290	0.351	0.291	0.352	**0.285**	**0.346**	0.290	0.353	0.289	0.350	0.305	0.362
	720	0.388	0.407	0.553	0.523	0.425	0.438	0.388	0.407	**0.387**	**0.405**	0.399	0.419	0.418	0.433	0.411	0.427	0.443	0.443
	Avg	0.270	**0.330**	0.325	0.376	0.279	0.341	**0.268**	**0.330**	**0.268**	**0.330**	0.271	0.335	0.274	0.334	0.273	0.334	0.291	0.348
ILI	24	2.238	1.036	2.300	1.060	2.247	1.047	2.241	1.037	2.230	1.031	2.221	1.026	2.220	1.025	**2.182**	**1.013**	2.259	1.044
	36	2.149	1.053	2.321	1.081	2.160	1.036	2.158	1.056	**2.110**	1.045	2.143	1.020	2.149	1.020	2.124	**1.012**	2.306	1.081
	48	2.312	1.077	2.320	1.089	2.212	1.056	2.315	1.078	2.358	1.088	**2.132**	**1.033**	2.139	1.035	2.150	1.060	2.219	1.052
	60	2.000	0.997	2.656	1.180	2.436	1.117	2.004	0.998	**1.938**	**0.972**	2.282	1.084	2.261	1.077	2.208	1.054	2.526	1.143
	Avg	2.175	1.041	2.400	1.102	2.264	1.064	2.179	1.042	**2.159**	**1.034**	2.194	1.041	2.192	1.039	2.166	1.035	2.328	1.080
Exchange	96	0.083	0.205	0.110	0.237	**0.080**	**0.200**	0.083	0.205	0.084	0.205	0.081	0.203	0.081	0.204	0.083	0.208	0.116	0.241
	192	0.156	0.292	0.164	0.299	0.161	0.295	0.156	0.292	0.161	0.296	**0.153**	**0.286**	0.155	0.291	0.158	0.291	0.162	0.295
	336	**0.269**	**0.387**	0.360	0.466	0.285	0.408	**0.269**	**0.387**	0.272	0.387	0.333	0.441	0.333	0.442	0.333	0.441	0.324	0.434
	720	0.491	0.547	0.898	0.725	0.898	0.725	**0.490**	**0.546**	0.495	0.545	0.898	0.725	0.898	0.725	0.898	0.725	0.898	0.725
	Avg	**0.250**	0.358	0.383	0.432	0.356	0.407	**0.250**	**0.357**	0.253	0.358	0.366	0.414	0.367	0.416	0.368	0.416	0.375	0.424
Weather	96	**0.172**	0.229	0.178	0.238	0.174	0.233	**0.172**	0.230	**0.172**	0.230	0.173	**0.227**	0.173	0.228	0.174	0.233	**0.172**	0.230
	192	**0.214**	0.271	0.223	0.285	0.218	0.278	**0.214**	0.271	**0.214**	0.272	**0.214**	**0.265**	0.215	**0.265**	0.217	0.277	0.214	0.273
	336	0.262	0.313	0.264	0.316	0.263	0.315	0.262	0.314	0.262	0.314	**0.259**	0.302	**0.259**	**0.301**	0.262	0.313	0.261	0.312
	720	0.332	0.374	0.332	0.374	**0.328**	**0.368**	0.332	0.374	0.332	0.374	0.332	0.374	0.332	0.374	0.332	0.374	0.332	0.374
	Avg	**0.245**	0.297	0.249	0.303	0.246	0.299	**0.245**	0.297	**0.245**	0.297	**0.245**	**0.292**	**0.245**	0.292	0.246	0.299	**0.245**	0.297

E Robustness Across Random Seeds

To evaluate robustness to random initialization and stochastic training, we repeat the experiments for the original backbones and HORW using three random seeds: 2021, 2023, and 2025. For each backbone–dataset–horizon configuration, the HORW coefficient β is selected once using the validation results under the random seed specified by the original backbone implementation and then transferred unchanged to all three seed runs. All other model configurations are also kept fixed, with only the random seed changed. For each seed, MSE and MAE are first averaged over the four prediction horizons; Table 11 reports the mean and sample standard deviation of these seed-level averages.

Across the three backbones and seven datasets, HORW improves or matches the average MSE in 19 of the 21 backbone–dataset combinations and the average MAE in 20 combinations. For xPatch, HORW improves or matches both metrics on all seven datasets. PatchTST shows the same pattern for MAE and improves or matches MSE on six datasets, while DLinear improves both metrics on six datasets. The improvements are particularly consistent on Exchange, where HORW reduces the DLinear MSE from 0.376 ± 0.014 to 0.251 ± 0.005, and on ETTh1, where the xPatch MSE decreases from 0.440 ± 0.003 to 0.420 ± 0.004.

Most configurations exhibit small variation across seeds, indicating that the overall gains are not tied to a single initialization. ILI is the main exception: with DLinear, HORW increases the MSE from 2.293 ± 0.053 to 2.418 ± 0.213 and exhibits greater variability; with PatchTST, the MSE changes slightly from 1.592 ± 0.104 to 1.600 ± 0.140, although MAE improves from 0.850 ± 0.022 to 0.846 ± 0.041. Overall, the multi-seed results confirm that the performance advantage of HORW persists across most evaluated configurations, while also revealing increased sensitivity on ILI.

F Qualitative Case Study

Figure 3 presents a forecasting case on Exchange with prediction horizon 720. The original DLinear prediction exhibits a pronounced downward bias and progressively diverges from the ground truth. DLinear+HORW mitigates this bias and is closer to the target trajectory over the displayed forecasting interval. This qualitative observation is consistent with the large quantitative gain of HORW on Exchange reported in Table 1.

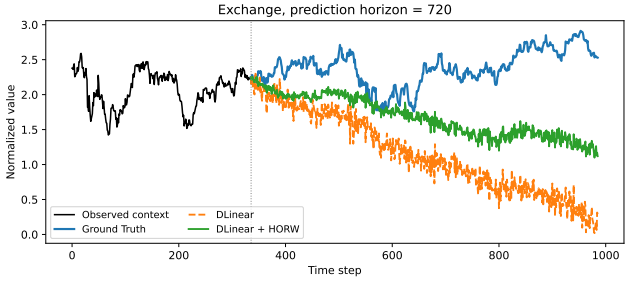

Figure 3: Forecasting visualization on Exchange with prediction horizon 720. For readability, only the first 650 steps of the forecasting horizon are shown.

G Efficiency Analysis

Table 12 compares the training and inference efficiency of DLinear with and without HORW. On ETTh1 at horizon 720, HORW increases the training time per epoch from 2.893 to 3.457 seconds and requires an additional 7.828 seconds for historical-memory construction, resulting in a total of 11.285 seconds per epoch. On ETTh2 at the same horizon, the total epoch time increases from 2.913 to 13.048 seconds. On ILI, the total epoch time with HORW ranges from 1.493 to 1.573 seconds across the four prediction horizons.

Peak GPU memory usage also increases with HORW. On ETTh1 and ETTh2, it ranges from 78.194 MB at horizon 96 to 330.758 MB at horizon 720, compared with 19.317–28.877 MB for the original backbone. On ILI, peak GPU memory remains below 36 MB across all evaluated horizons. HORW adds no trainable parameters, and the reported inference latencies remain below 0.035 milliseconds per sample.

Overall, although HORW introduces additional training time and memory costs, it is conceptually simple, easy to implement, and effective in improving forecasting performance.

Table 11: Forecasting performance over three random seeds. Results are reported as the mean and sample standard deviation. For each seed, MSE and MAE are first averaged over four prediction horizons. Lower values are better, and the better mean result in each pair is highlighted in **bold**.

Models	DLinear				PatchTST				xPatch			
	Ori		+HORW		Ori		+HORW		Ori		+HORW	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.447 ± 0.004	0.456 ± 0.004	**0.439 ± 0.011**	**0.447 ± 0.010**	0.415 ± 0.003	**0.429 ± 0.002**	**0.412 ± 0.002**	**0.429 ± 0.001**	0.440 ± 0.003	0.438 ± 0.001	**0.420 ± 0.004**	**0.432 ± 0.001**
ETTh2	0.446 ± 0.020	0.453 ± 0.009	**0.434 ± 0.030**	**0.438 ± 0.015**	0.330 ± 0.000	**0.380 ± 0.001**	**0.327 ± 0.000**	**0.380 ± 0.000**	0.342 ± 0.000	0.384 ± 0.001	**0.341 ± 0.002**	**0.383 ± 0.001**
ETTm1	0.359 ± 0.001	0.381 ± 0.001	**0.358 ± 0.001**	**0.380 ± 0.001**	0.352 ± 0.001	0.382 ± 0.001	**0.350 ± 0.000**	**0.380 ± 0.000**	0.353 ± 0.001	0.372 ± 0.001	**0.350 ± 0.001**	**0.370 ± 0.001**
ETTm2	0.293 ± 0.010	0.350 ± 0.005	**0.273 ± 0.004**	**0.334 ± 0.005**	0.257 ± 0.000	0.315 ± 0.000	**0.256 ± 0.000**	**0.314 ± 0.000**	**0.253 ± 0.001**	**0.309 ± 0.000**	**0.253 ± 0.001**	**0.309 ± 0.000**
ILI	**2.293 ± 0.053**	**1.074 ± 0.013**	2.418 ± 0.213	1.110 ± 0.061	**1.592 ± 0.104**	0.850 ± 0.022	1.600 ± 0.140	**0.846 ± 0.041**	1.384 ± 0.021	0.715 ± 0.011	**1.339 ± 0.036**	**0.708 ± 0.014**
Exchange	0.376 ± 0.014	0.418 ± 0.008	**0.251 ± 0.005**	**0.360 ± 0.003**	0.391 ± 0.008	0.424 ± 0.003	**0.385 ± 0.008**	**0.420 ± 0.003**	0.368 ± 0.004	0.404 ± 0.001	**0.362 ± 0.008**	**0.401 ± 0.004**
Weather	0.245 ± 0.001	0.297 ± 0.002	**0.244 ± 0.001**	**0.295 ± 0.002**	**0.229 ± 0.000**	0.266 ± 0.001	**0.229 ± 0.001**	**0.265 ± 0.001**	0.222 ± 0.001	0.252 ± 0.001	**0.221 ± 0.002**	**0.251 ± 0.002**

Table 12: Training and inference efficiency of DLinear with and without HORW. Training time, memory-construction time, and total time are reported per epoch. Inference latency is measured in milliseconds per test sample. Peak GPU memory is measured during training. HORW adds no trainable parameters and does not modify the inference architecture.

Dataset	Pred. len	Train time / epoch (s)		Memory build / epoch (s)		Total time / epoch (s)		Extra trainable params	Inference (ms/sample)		Peak GPU memory (MB)	
		Ori	+HORW	Ori	+HORW	Ori	+HORW	+HORW	Ori	+HORW	Ori	+HORW
ETTh1	96	2.5873	6.2533	0	4.2297	2.5873	10.4830	0	0.0130	0.0198	19.3174	78.1938
	192	2.6717	5.0315	0	4.4097	2.6717	9.4411	0	0.0232	0.0120	20.6797	118.7598
	336	2.7584	3.3743	0	3.5068	2.7584	6.8811	0	0.0126	0.0124	22.7217	180.1187
	720	2.8929	3.4572	0	7.8280	2.8929	11.2852	0	0.0127	0.0115	28.8770	330.7578
ETTh2	96	2.5415	3.4133	0	2.5299	2.5415	5.9432	0	0.0305	0.0167	19.3174	78.1938
	192	2.7071	3.2868	0	2.6493	2.7071	5.9361	0	0.0128	0.0119	20.6797	118.7598
	336	2.7067	6.6521	0	4.6846	2.7067	11.3367	0	0.0204	0.0122	22.7217	180.1187
	720	2.9125	5.1478	0	7.8998	2.9125	13.0476	0	0.0113	0.0340	28.8770	330.7578
ILI	24	0.8761	0.7422	0	0.7578	0.8761	1.5000	0	0.0152	0.0142	17.0020	26.9165
	36	0.8886	0.7340	0	0.7588	0.8886	1.4927	0	0.0146	0.0145	17.1089	35.7598
	48	0.8885	0.7448	0	0.7674	0.8885	1.5121	0	0.0158	0.0163	17.2129	35.5229
	60	0.8978	0.7827	0	0.7900	0.8978	1.5727	0	0.0165	0.0193	17.3198	35.5059

H Mechanism Analysis

Table 13 summarizes the historical-memory coverage and activation behavior of HORW on DLinear. Excluding the Weather-720 case, for which the selected checkpoint is obtained before the memory becomes available, the preceding-view memory covers 95.657% to 99.858% of forecast positions. This high coverage shows that the preceding-only reference is available for nearly all forecasting positions.

Despite this broad availability, HORW activates only 23.678% to 40.450% of positions. Thus, HORW does not uniformly reweight all positions with available history; instead, it selectively adjusts positions whose current errors exceed their historical references. The mean normalized gap among activated points ranges from 0.284 to 0.565, indicating that the activated positions exhibit non-negligible historical reliability differences.

The mean active weights further reflect the adaptive weighting behavior of HORW. For example, the active weights on ETTh2 are below one across all horizons, whereas the Exchange-96 setting has a mean active weight of 1.595. These results show that HORW can either suppress or emphasize the contribution of activated positions, depending on the selected weighting direction and the magnitude of the historical gap.

I Comparison with Generic Difficulty-Based Reweighting

To determine whether HORW's gains can be reproduced by generic difficulty-based reweighting, we compare it with point-wise online hard example mining (OHEM) on DLinear. Within each mini-batch and variable, OHEM selects a specified proportion of prediction points with the largest current errors and optimizes the original forecasting loss over the selected points. We select the OHEM ratio from $\{0.1, 0.25, 0.5, 0.75\}$ according to validation MSE for each dataset–horizon configuration. Unlike HORW, OHEM relies solely on current error magnitude and uses neither preceding overlapping forecasts nor timestamp-specific historical references.

As shown in Table 14, HORW achieves lower average MSE and MAE than OHEM on all seven datasets. Across the 28 dataset–horizon configurations, HORW obtains lower MSE in 25 configurations. For MAE, HORW performs better in 25 configurations, matches OHEM in two, and performs worse in one. The remaining OHEM advantages are limited to MSE on ETTh2 at horizon 96, ETTm2 at horizon 336, and ILI at horizon 48, and to MAE on ETTh2 at horizon 96. These results show that HORW outperforms a validation-tuned OHEM baseline in the large majority of configurations, supporting the value of timestamp-specific historical evidence beyond generic emphasis based solely on current error magnitude.

J Effect of Memory-Refresh Frequency

Standard HORW refreshes its historical error memory after every training epoch, which keeps the reference synchronized with the latest model but introduces additional memory-construction cost.

Table 13: Historical-memory coverage and activation statistics of HORW on DLinear. Statistics are accumulated over training batches in the epoch corresponding to the validation-selected checkpoint. Coverage rate denotes the proportion of all prediction points with a valid historical reference, and activation rate denotes the proportion of all prediction points with $\rho > 0$. "Mean active ρ" and "Mean active weight" report the mean normalized gap ρ and normalized weight w, respectively, over activated points. For Weather at horizon 720, the selected checkpoint precedes the first memory refresh, so active-only statistics are not applicable.

DLinear					
Dataset	Pred. len	Coverage rate	Activation rate	Mean active ρ	Mean active weight
ETTh1	96	98.946%	33.253%	0.320	1.192
	192	99.467%	38.825%	0.352	0.942
	336	99.690%	37.843%	0.284	1.400
	720	99.848%	37.579%	0.416	0.827
ETTh2	96	98.946%	27.846%	0.511	0.421
	192	99.467%	28.957%	0.513	0.427
	336	99.690%	30.820%	0.524	0.426
	720	99.848%	23.678%	0.498	0.412
ETTm1	96	98.955%	39.800%	0.318	1.168
	192	99.476%	40.450%	0.347	1.020
	336	99.699%	38.341%	0.346	1.099
	720	99.858%	38.810%	0.361	1.103
ETTm2	96	98.955%	36.084%	0.390	1.116
	192	99.476%	29.113%	0.488	0.548
	336	99.699%	30.656%	0.488	0.554
	720	99.858%	24.198%	0.519	0.516
ILI	24	95.657%	31.623%	0.435	0.594
	36	97.032%	35.472%	0.461	0.586
	48	97.725%	35.869%	0.413	0.621
	60	98.141%	32.707%	0.449	0.585
Exchange	96	98.938%	27.417%	0.363	1.595
	192	99.458%	24.584%	0.442	0.773
	336	99.681%	27.679%	0.561	0.503
	720	99.838%	33.051%	0.565	0.523
Weather	96	98.956%	33.181%	0.337	0.891
	192	99.476%	34.177%	0.375	0.850
	336	99.700%	36.091%	0.368	0.937
	720	0.000%	0.000%	nan	nan

To examine whether a less frequent update remains effective, we evaluate a Refresh-2 variant that constructs the first memory after epoch 1 and subsequently refreshes it every two epochs. The selected β and all other experimental settings are kept identical to those of standard HORW.

As shown in Table 15, Refresh-2 achieves lower average MSE and MAE than standard HORW on ETTh2 and matches both metrics on ETTm1. It also matches standard HORW in average MSE on Weather and in average MAE on ILI. Standard HORW performs better on ETTh1, ETTm2, and Exchange, as well as on ILI MSE and Weather MAE. Compared with the original DLinear model, Refresh-2 improves average MSE on five of the seven datasets and average MAE on six.

These results show that historical references can remain informative across consecutive epochs, allowing the number of memory-construction operations to be reduced by approximately half while retaining forecasting improvements in most datasets. However, the degradation on ETTh1 and the variation across individual horizons indicate that the effect of less frequent refreshing is dataset-dependent. We therefore use epoch-wise refreshing as the default HORW configuration and regard Refresh-2 as an optional efficiency-oriented variant.

K Effect of Restricting β Selection

To determine whether HORW's improvements rely on unrestricted selection of both the direction and magnitude of β, we evaluate three restricted variants on DLinear. Attenuation-only HORW restricts the search space to positive coefficients, $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0, 3.0, 5.0\}$, whereas Emphasis-only HORW restricts it to the corresponding negative coefficients. For both variants, the coefficient magnitude is selected for each dataset–horizon configuration according to validation MSE. Fixed-β HORW instead uses $\beta = 1.0$ for every dataset and prediction horizon, without configuration-specific coefficient selection. All other model hyperparameters and experimental settings remain identical to those of full HORW.

Table 14: Detailed comparison between HORW and generic point-wise online hard example mining (OHEM) on DLinear. OHEM selects a proportion of prediction points with the largest current errors within each mini-batch and variable, with the selection ratio chosen from $\{0.1, 0.25, 0.5, 0.75\}$ according to validation MSE for each dataset–horizon configuration. The best result for each setting is highlighted in **bold**.

Model		DLinear			
		+OHEM		+HORW	
Metric		MSE	MAE	MSE	MAE
ETTh1	96	0.384	0.405	**0.380**	**0.399**
	192	0.413	0.422	**0.405**	**0.413**
	336	0.447	0.449	**0.434**	**0.439**
	720	0.506	0.516	**0.495**	**0.499**
	Avg	0.438	0.448	**0.428**	**0.438**
ETTh2	96	**0.290**	**0.353**	0.308	0.358
	192	0.389	0.423	**0.362**	**0.392**
	336	0.463	0.473	**0.421**	**0.434**
	720	0.733	0.606	**0.510**	**0.502**
	Avg	0.469	0.464	**0.400**	**0.422**
ETTm1	96	0.301	0.345	**0.296**	**0.340**
	192	0.336	**0.366**	**0.335**	**0.366**
	336	0.372	0.389	**0.371**	**0.388**
	720	0.427	**0.423**	**0.426**	**0.423**
	Avg	0.359	0.381	**0.357**	**0.379**
ETTm2	96	0.172	0.267	**0.167**	**0.262**
	192	0.238	0.314	**0.228**	**0.300**
	336	**0.295**	0.359	0.298	**0.351**
	720	0.427	0.440	**0.388**	**0.407**
	Avg	0.283	0.345	**0.270**	**0.330**
ILI	24	2.284	1.062	**2.238**	**1.036**
	36	2.240	1.060	**2.149**	**1.053**
	48	**2.306**	1.081	2.312	**1.077**
	60	2.546	1.146	**2.000**	**0.997**
	Avg	2.344	1.087	**2.175**	**1.041**
Exchange	96	0.084	0.208	**0.083**	**0.205**
	192	0.162	0.296	**0.156**	**0.292**
	336	0.334	0.443	**0.269**	**0.387**
	720	0.814	0.706	**0.491**	**0.547**
	Avg	0.349	0.413	**0.250**	**0.358**
Weather	96	0.174	0.234	**0.172**	**0.229**
	192	0.218	0.279	**0.214**	**0.271**
	336	0.263	0.315	**0.262**	**0.313**
	720	0.333	0.375	**0.332**	**0.374**
	Avg	0.247	0.301	**0.245**	**0.297**

As shown in Table 16, Attenuation-only HORW improves the average MSE and MAE over the original DLinear objective on six of the seven datasets and matches it on ETTm1. At the reported precision, it also matches full HORW on ETTh2, ETTm2, ILI, and Weather, while remaining competitive on the other datasets. This demonstrates that HORW remains broadly effective even when its response is restricted to attenuation.

Emphasis-only HORW also improves both average MSE and MAE on ETTh1, ETTh2, ETTm1, and ETTm2. On Exchange and Weather, it matches the baseline average MSE at the reported precision while slightly improving MAE, whereas it does not improve the ILI results. Notably, Emphasis-only HORW matches full HORW on ETTm1. Thus, either response direction can independently provide useful improvements, although the attenuation-only variant is more consistently effective across the evaluated datasets.

Finally, fixing $\beta = 1.0$ across all datasets and prediction horizons still improves the average MSE and MAE on six of the seven datasets, with ETTm1 being the only exception. Although this fixed-coefficient variant is generally weaker than full HORW, its broad improvements show that HORW's gains cannot be explained solely by configuration-specific hyperparameter selection. Overall, the restricted variants establish that bidirectional validation-based selection is an adaptation mechanism for heterogeneous forecasting configurations rather than a prerequisite for HORW to be effective.

Table 15: Effect of reducing the historical-memory refresh frequency on DLinear. Standard HORW refreshes the historical memory after every epoch, whereas "Refresh-2" refreshes it after epochs 1, 3, 5, and so forth, reusing each memory for two consecutive training epochs. "Ori" denotes the original DLinear model. "Avg" reports the average over four prediction horizons. Lower values are better, and the best result in each row is highlighted in **bold**.

Model		DLinear					
		Ori		+HORW		+HORW (Refresh-2)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.384	0.405	**0.380**	0.399	**0.380**	**0.398**
	192	0.443	0.450	**0.405**	**0.413**	**0.405**	**0.413**
	336	0.447	0.448	**0.434**	**0.439**	0.440	0.442
	720	0.504	0.515	**0.495**	**0.499**	0.599	0.577
	Avg	0.445	0.454	**0.428**	**0.438**	0.456	0.458
ETTh2	96	**0.290**	**0.353**	0.308	0.358	0.305	0.357
	192	0.388	0.422	**0.362**	**0.392**	0.373	0.406
	336	0.463	0.473	0.421	0.434	**0.398**	**0.418**
	720	0.733	0.606	0.510	0.502	**0.473**	**0.480**
	Avg	0.469	0.463	0.400	0.422	**0.387**	**0.415**
ETTm1	96	0.301	0.345	**0.296**	**0.340**	**0.296**	0.341
	192	0.336	**0.366**	**0.335**	**0.366**	**0.335**	**0.366**
	336	0.372	0.389	**0.371**	**0.388**	**0.371**	**0.388**
	720	0.427	**0.423**	**0.426**	**0.423**	**0.426**	**0.423**
	Avg	0.359	0.381	**0.357**	**0.379**	**0.357**	**0.379**
ETTm2	96	0.172	0.267	**0.167**	**0.262**	**0.167**	**0.262**
	192	0.237	0.314	**0.228**	**0.300**	0.251	0.321
	336	**0.295**	0.359	0.298	**0.351**	0.303	0.353
	720	0.427	0.439	**0.388**	**0.407**	0.419	0.432
	Avg	0.283	0.345	**0.270**	**0.330**	0.285	0.342
ILI	24	2.280	1.061	**2.238**	**1.036**	2.268	1.066
	36	2.235	1.059	2.149	1.053	**1.951**	**0.934**
	48	2.298	1.079	2.312	1.077	**2.219**	**1.055**
	60	2.573	1.157	**2.000**	**0.997**	2.410	1.109
	Avg	2.347	1.089	**2.175**	**1.041**	2.212	**1.041**
Exchange	96	0.085	0.209	**0.083**	**0.205**	**0.083**	**0.205**
	192	0.162	0.296	**0.156**	**0.292**	0.174	0.310
	336	0.333	0.441	**0.269**	**0.387**	0.283	0.405
	720	0.898	0.725	**0.491**	**0.547**	0.505	0.559
	Avg	0.369	0.418	**0.250**	**0.358**	0.261	0.370
Weather	96	0.174	0.233	**0.172**	**0.229**	0.173	0.232
	192	0.218	0.278	**0.214**	**0.271**	**0.214**	0.272
	336	0.263	0.314	0.262	0.313	**0.261**	**0.311**
	720	**0.332**	**0.374**	**0.332**	**0.374**	**0.332**	**0.374**
	Avg	0.247	0.300	**0.245**	**0.297**	**0.245**	0.298

Table 16: Effect of restricting the selection freedom of β on DLinear. Full HORW selects both the sign and magnitude of β for each dataset–horizon configuration using validation MSE. Attenuation-only and Emphasis-only HORW restrict the search to $\beta > 0$ and $\beta < 0$, respectively, while retaining validation-based magnitude selection. Fixed-β HORW uses $\beta = 1.0$ for every configuration. Results are averaged over four prediction horizons, and the best result in each row is shown in **bold**.

Model		DLinear									
		Ori		+HORW		Fixed $\beta = 1$		Attenuation-only HORW		Emphasis-only HORW	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.384	0.405	**0.380**	**0.399**	0.391	0.413	0.385	0.406	**0.380**	**0.399**
	192	0.443	0.450	**0.405**	**0.413**	0.420	0.428	**0.405**	**0.413**	0.424	0.430
	336	0.447	0.448	**0.434**	**0.439**	0.467	0.463	0.448	0.449	**0.434**	**0.439**
	720	0.504	0.515	**0.495**	**0.499**	0.498	0.501	**0.495**	**0.499**	0.499	0.513
	Avg	0.445	0.454	**0.428**	**0.438**	0.444	0.451	**0.433**	**0.442**	0.434	0.445
ETTh2	96	0.290	0.353	0.308	0.358	**0.284**	**0.346**	0.308	0.358	0.291	0.355
	192	0.388	0.422	**0.362**	**0.392**	0.368	0.406	**0.362**	**0.392**	0.389	0.423
	336	0.463	0.473	**0.421**	**0.434**	0.430	0.448	**0.421**	**0.434**	0.449	0.469
	720	0.733	0.606	**0.510**	**0.502**	0.576	0.544	**0.510**	**0.502**	0.660	0.589
	Avg	0.469	0.463	**0.400**	**0.422**	0.415	0.436	**0.400**	**0.422**	0.447	0.459
ETTm1	96	0.301	0.345	**0.296**	**0.340**	0.310	0.352	0.302	0.345	**0.296**	**0.340**
	192	0.336	**0.366**	**0.335**	**0.366**	0.341	0.371	0.336	0.367	**0.335**	**0.366**
	336	0.372	0.389	**0.371**	**0.388**	0.378	0.394	0.372	0.389	**0.371**	**0.388**
	720	0.427	**0.423**	**0.426**	**0.423**	0.434	0.428	0.427	**0.423**	**0.426**	**0.423**
	Avg	0.359	0.381	**0.357**	**0.379**	0.366	0.386	0.359	0.381	**0.357**	**0.379**
ETTm2	96	0.172	0.267	**0.167**	0.262	0.168	0.259	**0.167**	**0.260**	**0.167**	0.262
	192	0.237	0.314	**0.228**	**0.300**	0.230	0.310	**0.228**	**0.300**	0.234	0.315
	336	0.295	0.359	0.298	**0.351**	**0.291**	**0.351**	0.298	**0.351**	0.297	0.361
	720	0.427	0.439	**0.388**	**0.407**	0.398	0.417	**0.388**	**0.407**	0.427	0.440
	Avg	0.283	0.345	**0.270**	**0.330**	0.272	0.334	**0.270**	**0.330**	0.281	0.344
ILI	24	2.280	1.061	**2.238**	**1.036**	2.238	1.043	**2.238**	**1.036**	2.282	1.061
	36	2.235	1.059	**2.149**	**1.053**	2.097	1.018	**2.149**	**1.053**	2.247	1.062
	48	**2.298**	1.079	2.312	**1.077**	2.299	1.077	**2.312**	**1.077**	2.300	1.080
	60	2.573	1.157	**2.000**	**0.997**	2.301	1.093	**2.000**	**0.997**	2.583	1.166
	Avg	2.347	1.089	**2.175**	**1.041**	2.234	1.058	**2.175**	**1.041**	2.353	1.092
Exchange	96	0.085	0.209	**0.083**	**0.205**	0.091	0.216	0.085	0.209	**0.083**	**0.205**
	192	0.162	0.296	**0.156**	**0.292**	0.156	0.292	**0.156**	**0.292**	0.161	0.296
	336	0.333	0.441	**0.269**	**0.387**	0.303	0.422	**0.269**	**0.387**	0.335	0.443
	720	0.898	0.725	**0.491**	**0.547**	0.898	0.725	**0.491**	**0.547**	0.898	0.725
	Avg	0.369	0.418	**0.250**	**0.358**	0.362	0.414	**0.250**	0.359	0.369	0.417
Weather	96	0.174	0.233	**0.172**	**0.229**	0.172	0.232	**0.172**	**0.229**	0.175	0.234
	192	0.218	0.278	**0.214**	**0.271**	0.214	0.272	**0.214**	**0.271**	0.218	0.274
	336	0.263	0.314	**0.262**	**0.313**	0.264	0.317	**0.262**	**0.313**	0.263	0.314
	720	**0.332**	**0.374**	**0.332**	**0.374**	0.332	0.374	**0.332**	**0.374**	**0.332**	**0.374**
	Avg	0.247	0.300	**0.245**	**0.297**	0.246	0.299	**0.245**	**0.297**	0.247	0.299