CCo-Patch: Continuous Correction with Temporal Scaffolding for Patch-based Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Patch-based Transformer models such as PatchTST have become a dominant paradigm for long-term time series forecasting, but patchification introduces an inter-patch continuity gap: local patterns inside each patch are explicitly encoded, while the transitions between neighboring patches must be inferred implicitly, which can result in incoherent predictions across patch boundaries. We propose CCo-Patch, a continuous correction framework that augments a patch-based backbone with a continuous correction branch (hereafter, the Correction branch). The Correction branch extracts a causal low-frequency scaffold—a reference sequence extracted from the original input—through FFT-based filtering, and fuses it with the main branch via a learnable coefficient, with patch-wise mean-centering and Huber optimization further improving robustness. Across seven long-term forecasting benchmarks and four prediction horizons, CCo-Patch achieves the best MSE in 17 out of 28 settings, competitive with or better than the ICLR 2026 PMDformer on most datasets and horizons, with clear gains on Weather, ECL, and ETTh2, and comparable performance on ETTm2. Continuity-oriented evaluation and a patch-granularity analysis further show that CCo-Patch improves cross-patch continuity and that this advantage is robust across patch lengths. Code is available in the supplementary material.

1 Introduction

Time series forecasting is fundamental in many real-world applications, including weather forecasting, energy management, traffic prediction, and industrial monitoring. Deep learning has become the dominant paradigm (Wang et al., 2024c). Recent studies show that representing long time series as local patches improves both accuracy and efficiency. PatchTST (Nie et al., 2023) segments a long sequence into subseries-level patches as Transformer tokens, preserving local semantics and reducing effective sequence length. These advantages have made patch-based representations an important paradigm for long-term forecasting (Zhang & Yan, 2023; Gong et al., 2024; Zhang et al., 2024).

However, patchification changes how temporal relationships are presented. A time series is intrinsically continuous, and its predictive dynamics are reflected not only within segments but also in the transitions between neighboring segments. In patch-based representations, local patterns are explicitly encoded, while the transitions between neighboring patches must be inferred implicitly (Chen et al., 2026). Although Transformers can learn inter-patch dependencies through attention, there is no explicit structural constraint encouraging these transitions to be coherent. Recent work has explored dynamic patch segmentation (Abeywickrama et al., 2025; Ling et al., 2025) and adaptive patch granularity (Ding et al., 2026) to align patch boundaries with local structure, as well as inter-patch information fusion (Wei et al., 2026). However, these approaches modify the patching strategy or encoding mechanism, rather than providing an explicit continuous reference for cross-patch coherence. We refer to this limitation as the *inter-patch continuity gap*: the discrepancy between predicted and ground-truth transitions across neighboring patches, reflecting how well the model maintains temporal consistency at patch boundaries. We aim to improve *cross-patch continuity* by providing an explicit temporal scaffold. The scaffold offers a structural basis for continuity, but its benefit is realized only when combined with other components, especially the Huber loss. We

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

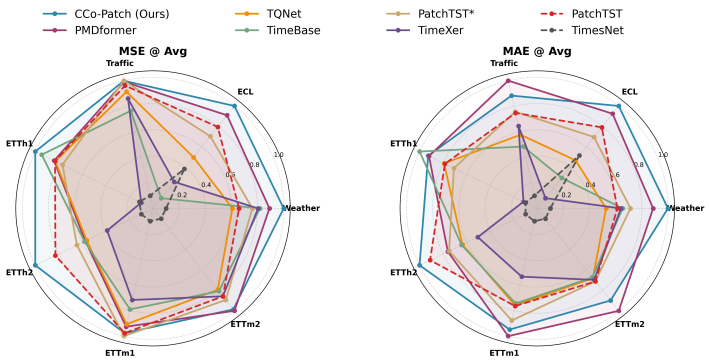

Figure 1: Average MSE and MAE radar comparison over four prediction horizons. For each dataset, values are averaged over the four horizons and min–max normalized across all models to $[0.1, 1.0]$, with a larger radius indicating better performance.

later evaluate continuity using two complementary metrics, IWCE and TRE, as described in the experiments.

This motivates the following question: can an explicit temporal scaffold help patch representations maintain cross-patch continuity? e introduce an additional pathway that provides a low-frequency reference extracted from the original input as the scaffold, complementing patch-based modeling rather than replacing it. This reference helps the model maintain consistent transitions across patches (Xu et al., 2025; Dama et al., 2025), rather than predicting the target or model residuals. Based on this idea, we propose **CCo-Patch**, a Continuous Correction framework that consists of a main branch and a continuous correction branch. The main branch retains the efficient PatchTST backbone and forecasts dominant temporal patterns from patch representations. The continuous correction branch, on the other hand, provides an explicit temporal scaffold to improve cross-patch continuity. We further incorporate two lightweight design choices: patch-wise mean-centering before patch encoding to reduce patch-level bias (Hu et al., 2026; Berthelier et al., 2026), and the Huber loss (Huber, 1964; Volarić et al., 2025) to improve robustness to large forecasting errors.

We evaluate CCo-Patch on seven benchmarks under four horizons (96, 192, 336, 720). It achieves the best MSE in 17 out of 28 settings, and is competitive with or better than the ICLR 2026 PMDformer on most datasets and horizons, while remaining competitive on MAE. Ablation and continuity analyses suggest that CCo-Patch improves cross-patch continuity.

Our contributions are summarized as follows:

- The inter-patch continuity gap is identified and formalized as a structural limitation of patch-based forecasting: local patterns are encoded within patches, while cross-patch transitions must be inferred implicitly.
- CCo-Patch is proposed as a lightweight continuous correction framework that augments a patch-based backbone with a temporal scaffold, without altering how patches are constructed or encoded.
- An asymmetric complementarity is found between the Correction branch and the Huber loss: the Correction branch alone generally improves MSE at the cost of MAE, while the Huber loss improves MAE across datasets and MSE on most datasets; their combination retains the MSE gain without sacrificing MAE.
- Continuity analysis on four datasets shows that CCo-Patch improves cross-patch continuity, with more consistent gains on TRE than on IWCE.

2 Related Work

2.1 Efficient and MLP-based Forecasting

Transformer-based architectures have become a dominant paradigm for time series forecasting. Early studies focused on efficient attention and decomposition, such as Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), and FEDformer (Zhou et al., 2022). PatchTST (Nie et al., 2023) later demonstrated that representing temporal subseries as patches substantially improves efficiency and effectiveness. More recently, MLP-based and linear architectures have emerged as competitive alternatives: DLinear (Zeng et al., 2023) showed that a simple decomposition followed by linear mappings is highly competitive; TiDE (Das et al., 2023) employed nonlinear MLP encoder-decoder mappings; TSMixer (Chen et al., 2023) explored temporal and feature mixing; and TimeMixer (Wang et al., 2024a) extended this paradigm to multiple temporal scales. These approaches focus on temporal mappings rather than explicitly structuring transitions between neighboring segments.

2.2 Patch-based Time Series Forecasting

Patch-based forecasting represents a time series as a sequence of local temporal segments and has become an effective approach for long-horizon prediction (Su et al., 2025). PatchTST (Nie et al., 2023) introduced subseries-level patches as Transformer tokens, preserving local semantic information while reducing the effective sequence length. Subsequent studies extended this paradigm through multi-scale and hierarchical representations. Pathformer (Chen et al., 2024) constructs patches at multiple temporal resolutions and routes information across scales; TimeMixer (Wang et al., 2024a) exploits multi-scale temporal representations through decomposable mixing; TimeXer (Wang et al., 2024b) combines patch-level and global endogenous representations to incorporate exogenous information; and TimeBase (Huang et al., 2025) explores segment-level forecasting and basis representations for efficient long-term prediction.

Despite these advances, recent work has begun to address inter-patch continuity from two directions: adjusting patch boundaries via entropy-guided segmentation (Abeywickrama et al., 2025; Ling et al., 2025; Ding et al., 2026) and enhancing cross-patch interaction within the backbone (Zhang et al., 2024; Wei et al., 2026). These approaches define continuity in terms of patch boundary alignment or cross-patch information flow, and modify either the patching strategy or the encoding mechanism to improve it; but they do not explicitly quantify cross-patch continuity. We instead define cross-patch continuity as the consistency of transitions between neighboring patches, and evaluate it using the continuity metrics defined in the experiments. CCo-Patch retains the original patch-based backbone and introduces a lightweight continuous scaffold as a separate pathway for improving cross-patch continuity.

3 Method

3.1 Overall Framework

As illustrated in Figure 2, CCo-Patch consists of a main branch that inherits the PatchTST backbone and a continuous correction branch that constructs a low-frequency temporal scaffold. The main branch additionally applies mean-centering to the input patches, and the two branches are fused into the final prediction.

Given a multivariate time series $\mathbf{X} \in \mathbb{R}^{L\times C}$ with historical context length L and C variables, the goal is to predict the future sequence $\hat{\mathbf{Y}} \in \mathbb{R}^{H\times C}$ over horizon H. As shown in Figure 3, the input is split into two branches. Taking a single channel as an example, let $\mathbf{X}^{(i)} \in \mathbb{R}^{1\times L}$ denote the input sequence of the i-th channel with a look-back window of length L. The main branch follows the original PatchTST transformation, while the Correction branch processes the input through the proposed pipeline and a lightweight MLP to produce an auxiliary prediction.

Specifically, the main branch applies the standard PatchTST backbone and produces $\hat{\mathbf{Y}}_{\text{main}}$. Before patch encoding, patch-wise mean-centering (Hu et al., 2026; Liu et al., 2025; Wu et al., 2025) is applied to each patch z:

$$\tilde{z} = z - \text{Mean}(z), \tag{1}$$

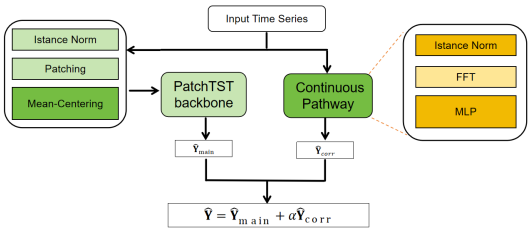

Figure 2: Overview of CCo-Patch. The main branch (left) follows the standard PatchTST backbone, while the continuous correction branch (right) extracts a low-frequency temporal scaffold and produces a complementary prediction. The final forecast combines both branches via a learnable coefficient $\alpha = \tanh(\gamma)$.

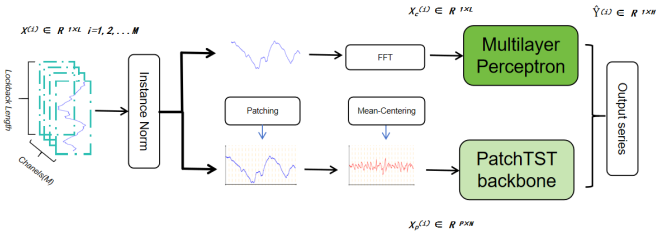

Figure 3: Data flow in CCo-Patch. The input is split into a main branch and a correction branch, which are fused into the final prediction.

where the mean is computed along the temporal dimension within each patch, removing patch-level bias while preserving intra-patch variation. In parallel, the Correction branch extracts a low-frequency temporal scaffold $\mathbf{S}$ and maps it to a complementary prediction $\hat{\mathbf{Y}}_{\mathrm{corr}}$. The two branches are then fused into the final prediction, as detailed in Section 3.3.

The Correction branch provides an explicit temporal scaffold derived from the continuous input sequence. This scaffold complements the patch-based representation by offering a structural reference for organizing neighboring temporal segments, without replacing the patch-based representation or predicting the main branch's forecasting error.

3.2 Continuous Temporal Structure Modeling

Low-frequency components capture the global trend that spans multiple patches; providing this trend as a reference helps the model keep transitions consistent across patch boundaries. The Correction branch operates directly on the original temporal sequence and extracts a slowly varying scaffold before patchification. This scaffold provides a continuous reference for the temporal evolution across neighboring local segments, complementing the patch-level representation learned by the main branch. We refer to it as the *causal low-frequency scaffold*.

Given the normalized input sequence $\mathbf{X}^{\mathrm{norm}} \in \mathbb{R}^{L \times C}$, we extract the scaffold using a causal sliding window of length W, following the principle of causal convolution that prevents future information leakage (Bai et al., 2018). For each time step t, the window $\mathbf{x}_t^{(W)} = [\mathbf{x}_{t-W+1}, \ldots, \mathbf{x}_t]$ is constructed from the current and preceding observations, with replication padding on the left. We then apply a causal FFT-based low-pass filter to suppress rapidly varying components while retaining the slowly varying structure:

$$\mathbf{s}_t = \left[\mathcal{F}^{-1}\left(\mathcal{F}\left(\mathbf{x}_t^{(W)}\right) \odot \mathbf{M}\right)\right][-1], \tag{2}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, and $\mathbf{M}$ is a mixture of two Gaussian masks whose centers are selected per dataset via grid search, with bandwidths fixed to 1 and 2 and mixture weights learned during training(Yi et al., 2024; Wang et al., 2025). Implementation details are provided in Appendix A.2.

The temporal scaffold is then obtained as

$$\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_L] \in \mathbb{R}^{L \times C}. \tag{3}$$

3.3 Continuous Correction and Fusion

Given the temporal scaffold $\mathbf{S}$, the Correction branch employs a lightweight multilayer perceptron to map it to the forecasting horizon:

$$\hat{\mathbf{Y}}_{\text{corr}} = g_\theta(\mathbf{S}), \tag{4}$$

where $g_\theta(\cdot)$ denotes the prediction head of the Correction branch. The final prediction combines the two branches:

$$\hat{\mathbf{Y}} = \hat{\mathbf{Y}}_{\text{main}} + \alpha \hat{\mathbf{Y}}_{\text{corr}}, \qquad \alpha = \tanh(\gamma), \tag{5}$$

where γ is a learnable scalar initialized to zero. This initialization makes the correction contribution vanish initially, allowing the model to start from the PatchTST prediction and progressively incorporate the complementary signal. The Correction branch operates in a channel-independent manner: the scaffold is extracted and mapped to the horizon separately for each variable, with shared MLP weights across channels. This design keeps the Correction branch lightweight. Its parameter count does not scale with the number of variables, which is particularly beneficial for high-dimensional datasets such as Traffic ($C = 862$).

The resulting $\mathbf{S}$ is not an additional forecasting decomposition, and the Correction branch does not explicitly estimate the main branch's point-wise error. Instead, it supplies a low-frequency reference that complements the patch-based representation, helping the main branch maintain consistent transitions across neighboring segments. The final prediction therefore benefits from both the expressive local representation of the main branch and the temporal scaffold provided by the Correction branch.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate CCo-Patch on seven widely used real-world datasets: ECL, Traffic, Weather, ETTh1, ETTh2, ETTm1, and ETTm2. Following standard long-term forecasting protocols, we consider four forecasting horizons, $H \in \{96, 192, 336, 720\}$. Detailed dataset statistics are provided in Table 1.

Baselines. We compare CCo-Patch with representative long-term time series forecasting methods, including TimesNet (Wu et al., 2023), PatchTST (Nie et al., 2023), TimeXer (Wang et al., 2024b), PatchTST* (PatchTST + TimeBase) (Huang et al., 2025), TQNet (Lin et al., 2025), TimeBase (Huang et al., 2025), and PMDformer(Hu et al., 2026). These baselines cover Transformer-based, patch-based, efficient forecasting, and recent state-of-the-art approaches. Among them, PatchTST serves as the primary backbone for evaluating the effectiveness of the proposed continuous correction framework.

Implementation Details. Following PatchTST, we use an input length of $L = 336$ for all datasets and evaluate forecasting horizons of $\{96, 192, 336, 720\}$. For CCo-Patch and the PatchTST backbone, the patch length and stride are fixed to 16 and 8, respectively, consistent with the original PatchTST implementation. The remaining experimental settings follow the general protocol of PatchTST, while the batch size and learning-rate schedule are adjusted for our implementation. For the other baselines, we follow their official configurations and perform a basic hyperparameter search over the patch-related parameters, reporting the best results for each baseline under the same input length of 336.

Our model is implemented in PyTorch and trained using the Adam optimizer. Following the standard formulation of Huber loss, the training objective uses the Huber loss (Appendix A.3) with parameter δ selected from $\{0.05, 0.10, \ldots, 1.00\}$ via a grid search on the validation set. The Gaussian centers of the spectral filter are selected via a grid search over all pairs $(\mu_1, \mu_2) \in \{0, \ldots, 8\}^2$, using two quantitative criteria: the smoothness of the filtered curve and its preservation of the underlying trend.

Table 1: Statistics of the seven long-term time series forecasting datasets used in our experiments.

Dataset	Domain	Variables	Time Points	Train/Val/Test	Interval
ETTh1	Electricity	7	14,400	6:2:2	1 hour
ETTh2	Electricity	7	14,400	6:2:2	1 hour
ETTm1	Electricity	7	57,600	6:2:2	15 min
ETTm2	Electricity	7	57,600	6:2:2	15 min
ECL	Electricity	321	26,304	7:1:2	1 hour
Weather	Weather	21	52,696	7:1:2	10 min
Traffic	Transportation	862	17,544	7:1:2	1 hour

The specific values for δ and the Gaussian centers are reported in Appendix A.2. A sensitivity analysis on Weather (Appendix A.4) shows that $\delta \in [0.10, 0.40]$ yields near-optimal MSE and MAE, confirming that the performance is not sensitive to the exact value of δ within this range. Main results in Table 2 are averaged over five random seeds; all other experiments use a single seed with fixed configurations.

The Correction branch is lightweight and channel-independent: it adds 0.13M–0.21M parameters depending on the horizon ($H = 96$ to 720), independent of the number of variables, with the variation coming from the final linear layer of the correction prediction head (output dimension = H). All experiments are conducted on NVIDIA RTX 4090 GPUs (24GB memory each), and a detailed efficiency comparison is provided in Appendix A.1.

4.2 Main Results

Table 2 summarizes the long-term forecasting results across seven datasets and four prediction horizons. CCo-Patch achieves the best MSE in 17 out of 28 experimental settings, and is competitive with or better than the PMDformer (Hu et al., 2026) on most datasets and horizons. The only tie occurs at ETTm2 with $H = 192$, where CCo-Patch, PMDformer, and PatchTST achieve the same MSE. For MAE, CCo-Patch remains highly competitive with PMDformer, showing comparable or better performance across datasets. Taken together, the results show that CCo-Patch provides a substantial improvement over the original PatchTST backbone, with its strongest advantage reflected in MSE. In addition to accuracy, we evaluate cross-patch continuity using IWCE (transition consistency across forecasting windows) and TRE (fidelity of first-order temporal evolution),which together suggest a modest improvement in cross-patch continuity, with the effect being more consistent on TRE than on IWCE.

4.3 Ablation Study

We conduct ablation experiments on Weather, ECL, ETTh2, and ETTm2 to examine the contribution of the three components in CCo-Patch and their effects on forecasting performance and temporal continuity. For each dataset, the results over the four forecasting horizons ($H \in \{96, 192, 336, 720\}$) are averaged to provide a compact comparison. We consider seven variants: the original PatchTST, PatchTST trained with the Huber loss, PatchTST augmented with the proposed continuous correction branch, PatchTST with patch-wise mean-centering, No-MC (Huber + Correction, without mean-centering), No-FFT (Huber + Correction + Mean-Centering, with the FFT filter removed), and the complete CCo-Patch.

Ablation on forecasting performance. Table 3 compares the forecasting performance of the seven variants. CCo-Patch achieves the lowest average MSE on three of the four datasets. The exception is ETTm2, where No-MC is slightly better, suggesting that mean-centering may be redundant on this dataset. These results indicate that the overall improvement cannot be attributed to any single component in isolation.

The Huber-only variant generally improves MAE and improves MSE on several datasets, as it reduces the influence of large errors that would otherwise dominate the squared loss. In contrast, the Correction alone behaves as a trade-off between the two metrics: it improves MSE by providing a continuous low-frequency reference, but slightly degrades MAE, likely because the smoothed

Table 2: Multivariate long-term forecasting results with an input length of 336. Results are reported as MSE / MAE. Best and second-best results are **bold** and underlined. The Avg row averages over four horizons (96, 192, 336, 720). Ties are counted for all tied models.

Dataset	Horizon	CCo-Patch (Ours)		PMDformer (2026)		TQNet (2025)		TimeBase (2025)		PatchTST* (2025)		TimeXer (2024)		PatchTST (2023)		TimesNet (2023)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	**0.127**	**0.218**	0.133	0.224	0.132	0.228	0.149	0.241	0.132	0.227	0.140	0.242	0.130	0.222	0.136	0.230
	192	**0.145**	**0.235**	0.149	0.239	0.156	0.250	0.162	0.254	0.149	0.242	0.161	0.259	0.148	0.241	0.155	0.247
	336	**0.160**	**0.252**	0.163	0.255	0.171	0.265	0.178	0.269	0.166	0.260	0.180	0.280	0.165	0.259	0.171	0.263
	720	0.195	0.286	**0.190**	**0.281**	0.212	0.304	0.217	0.301	0.206	0.294	0.211	0.305	0.202	0.291	0.219	0.302
	Avg	**0.157**	**0.248**	0.159	0.250	0.168	0.262	0.177	0.266	0.163	0.256	0.173	0.272	0.161	0.253	0.170	0.261
Traffic	96	0.351	0.232	**0.348**	**0.218**	0.363	0.252	0.424	0.281	0.352	0.247	0.373	0.259	0.367	0.251	0.588	0.320
	192	**0.375**	0.244	0.380	**0.235**	0.400	0.278	0.436	0.287	0.378	0.261	0.408	0.279	0.389	0.262	0.611	0.329
	336	**0.391**	0.253	0.396	**0.242**	0.423	0.297	0.451	0.295	0.392	0.268	0.425	0.284	0.398	0.265	0.622	0.340
	720	0.433	0.280	0.434	**0.266**	0.451	0.308	0.478	0.310	**0.427**	0.285	0.485	0.285	0.434	0.287	0.644	0.342
	Avg	0.388	0.252	0.390	**0.240**	0.409	0.284	0.447	0.293	**0.387**	0.265	0.423	0.277	0.397	0.266	0.616	0.333
Weather	96	0.146	0.186	**0.145**	**0.185**	0.156	0.209	0.147	0.199	0.150	0.198	0.148	0.202	0.156	0.205	0.170	0.228
	192	**0.189**	**0.229**	0.190	0.236	0.202	0.250	0.191	0.243	0.191	0.238	0.194	0.244	0.195	0.242	0.215	0.263
	336	**0.237**	**0.270**	0.245	0.280	0.248	0.284	0.242	0.282	0.244	0.280	0.244	0.285	0.250	0.284	0.272	0.301
	720	**0.302**	**0.323**	0.309	0.324	0.322	0.336	0.319	0.336	0.322	0.335	0.315	0.331	0.320	0.335	0.341	0.351
	Avg	**0.219**	**0.252**	0.222	0.256	0.232	0.270	0.225	0.265	0.227	0.263	0.225	0.266	0.230	0.267	0.250	0.286
ETTh1	96	0.370	0.394	0.380	0.395	0.379	0.401	**0.364**	**0.383**	0.380	0.404	0.395	0.416	0.382	0.405	0.420	0.436
	192	0.412	0.417	0.413	0.415	0.409	0.418	**0.402**	**0.406**	0.427	0.434	0.440	0.440	0.414	0.421	0.517	0.499
	336	**0.398**	0.422	0.439	0.434	0.432	0.433	0.427	**0.419**	0.433	0.436	0.491	0.485	0.431	0.435	0.480	0.474
	720	**0.432**	0.453	0.441	**0.444**	0.457	0.473	0.440	0.457	0.460	0.475	0.629	0.578	0.449	0.466	0.532	0.515
	Avg	**0.403**	0.422	0.418	0.422	0.419	0.431	0.408	**0.416**	0.425	0.437	0.489	0.480	0.419	0.432	0.487	0.481
ETTh2	96	**0.262**	0.332	0.277	**0.330**	0.301	0.355	0.299	0.354	0.282	0.341	0.319	0.370	0.274	0.336	0.377	0.420
	192	**0.314**	**0.368**	0.373	0.393	0.358	0.394	0.350	0.387	0.342	0.382	0.372	0.402	0.339	0.379	0.409	0.439
	336	**0.304**	**0.373**	0.382	0.411	0.363	0.405	0.365	0.408	0.373	0.405	0.402	0.427	0.331	0.381	0.416	0.441
	720	**0.371**	**0.419**	0.398	0.428	0.413	0.444	0.411	0.446	0.403	0.437	0.415	0.437	0.379	0.422	0.428	0.454
	Avg	**0.313**	**0.373**	0.358	0.391	0.359	0.400	0.357	0.399	0.350	0.391	0.377	0.409	0.331	0.380	0.408	0.439
ETTm1	96	0.291	**0.335**	0.298	0.338	0.294	0.345	0.308	0.351	**0.288**	0.342	0.308	0.356	0.292	0.343	0.343	0.378
	192	0.334	0.362	0.332	**0.358**	0.335	0.371	0.341	0.370	**0.328**	0.363	0.352	0.383	0.331	0.369	0.459	0.429
	336	0.365	0.382	**0.363**	**0.374**	0.370	0.390	0.376	0.391	0.365	0.385	0.385	0.405	0.365	0.392	0.394	0.413
	720	**0.418**	0.414	0.428	**0.413**	0.427	0.424	0.431	0.421	0.421	0.417	0.430	0.430	0.420	0.425	0.438	0.439
	Avg	0.352	0.373	0.355	**0.371**	0.357	0.383	0.364	0.383	**0.350**	0.377	0.369	0.394	0.352	0.382	0.409	0.415
ETTm2	96	0.165	0.251	**0.162**	**0.244**	0.167	0.255	0.169	0.259	0.166	0.254	0.172	0.260	0.165	0.255	0.184	0.272
	192	**0.220**	0.288	**0.220**	**0.284**	0.228	0.294	0.223	0.293	0.223	0.294	0.228	0.298	**0.220**	0.292	0.244	0.310
	336	0.273	0.323	**0.269**	**0.319**	0.277	0.329	0.276	0.329	0.275	0.329	**0.269**	0.324	0.278	0.329	0.320	0.356
	720	**0.355**	**0.375**	0.360	0.377	0.368	0.387	0.370	0.385	0.362	0.384	0.362	0.381	0.368	0.385	0.390	0.402
	Avg	**0.253**	0.309	**0.253**	**0.306**	0.260	0.316	0.260	0.317	0.257	0.315	0.258	0.316	0.258	0.315	0.285	0.335

PatchTST* denotes PatchTST enhanced with TimeBase as a plug-in module.

scaffold suppresses fine-grained local variations. The mean-centering variant does not consistently improve either MSE or MAE.

The No-MC and No-FFT variants isolate the contributions of mean-centering and the FFT filter. No-MC performs comparably to CCo-Patch on ETTm2 (0.251 vs. 0.253), but is worse on Weather (0.223 vs. 0.218), indicating that mean-centering contributes meaningfully. No-FFT is slightly worse than CCo-Patch on most datasets (e.g., 0.219 vs. 0.218 on Weather), suggesting that the Correction branch retains most of its benefit even without the FFT filter, and that the filter primarily stabilizes training rather than providing the core gain.

The ablation results reveal an asymmetric complementarity between the Correction branch and the Huber loss: the Huber loss improves MAE across datasets and MSE on most datasets, while the Correction branch alone trades MAE for MSE. When combined, the Huber loss compensates for the MAE degradation introduced by the Correction branch, while the Correction branch retains its MSE benefit. This complementarity explains why the complete CCo-Patch achieves the best MSE and MAE simultaneously, even though the Correction alone is not beneficial on both metrics.

Continuity analysis. To further examine the temporal property targeted by CCo-Patch, we evaluate the predictions using two complementary continuity metrics: inter-window continuity error (IWCE) and temporal relative error (TRE).

Table 3: Ablation study of the proposed components. Values are averaged over the four forecasting horizons ($H \in \{96, 192, 336, 720\}$) for each dataset and reported as MSE / MAE. "Correction" denotes the Correction branch. No-MC denotes the variant with Huber and the Correction branch but without mean-centering. No-FFT denotes the variant where the FFT filter is removed and the scaffold is computed directly from the normalized input sequence.

Design				Weather		ECL		ETTh2		ETTm2	
Variant	Huber	Correction	M-C	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PatchTST	×	×	×	0.231	0.266	0.162	0.254	0.319	0.374	0.258	0.338
+ Huber	✓	×	×	0.231	0.255	0.160	**0.246**	0.322	0.373	<u>0.252</u>	**0.307**
+ Correction	×	✓	×	0.223	0.271	0.160	0.257	<u>0.316</u>	0.373	0.257	0.316
+ Mean-Centering	×	×	✓	0.226	0.265	0.165	0.260	0.318	0.378	0.255	0.315
No-MC	✓	✓	×	0.223	<u>0.253</u>	<u>0.157</u>	0.248	0.321	**0.372**	**0.251**	<u>0.308</u>
No-FFT	✓	✓(No-FFT)	✓	<u>0.219</u>	<u>0.253</u>	**0.156**	<u>0.247</u>	0.322	**0.372**	0.253	0.310
CCo-Patch	✓	✓	✓	**0.218**	**0.252**	<u>0.157</u>	0.248	**0.311**	**0.372**	0.253	0.309

IWCE provides a local measure of cross-window temporal transition consistency. For the $(i-1)$-th and i-th forecasting windows, we measure the discrepancy between the predicted and ground-truth transitions from the last point of the preceding window to the first point of the current window:

$$\mathrm{IWCE} = \frac{1}{C(N-1)} \sum_{c=1}^{C} \sum_{i=2}^{N} \left| (\hat{y}_{i,1,c} - \hat{y}_{i-1,H,c}) - (y_{i,1,c} - y_{i-1,H,c}) \right|. \tag{6}$$

Here, N and C denote the number of forecasting windows and variables, and the windows are constructed with a stride of 8. IWCE measures the discrepancy of cross-window transitions between predictions and the ground truth, characterizing window-level consistency rather than point-wise accuracy.

To characterize temporal consistency over the reconstructed sequence, we further employ temporal relative error (TRE). Let

$$\Delta y_{t,c} = y_{t,c} - y_{t-1,c}, \qquad \Delta \hat{y}_{t,c} = \hat{y}_{t,c} - \hat{y}_{t-1,c}, \tag{7}$$

denote the first-order temporal differences of the ground-truth and predicted sequences, respectively. We adopt a global formulation that aggregates temporal differences across all time steps and variables before normalization:

$$\mathrm{TRE} = \frac{\mathrm{mean}_{t,c} \left| \Delta \hat{y}_{t,c} - \Delta y_{t,c} \right|}{\mathrm{mean}_{t,c} \left| \Delta y_{t,c} \right| + \epsilon}. \tag{8}$$

Here, ϵ is a small constant for numerical stability. Unlike channel-wise relative errors, this global formulation avoids unstable ratios for variables with small temporal variations. We refer to this metric as TRE; in the patch-granularity analysis, the same metric is also denoted Global-TRE. Lower TRE indicates that the first-order temporal evolution of the prediction is closer to that of the ground-truth sequence. For each dataset, the four horizon-specific values are averaged, and the results are reported in Table 4.

The continuity results provide some evidence that CCo-Patch improves cross-patch continuity. Compared with the original PatchTST, CCo-Patch achieves a lower average TRE on all four datasets. The improvement is particularly pronounced on ETTm2, where the average TRE decreases from 1.14 to 1.05. CCo-Patch achieves lower IWCE than the original PatchTST on Weather, ECL, ETTh2, and ETTm2, although No-FFT attains the lowest IWCE on Weather. These results indicate that the complete model generally produces temporal transitions and first-order temporal evolution that are closer to those of the ground-truth sequence.

Robustness to patch granularity. We further examine whether the continuity advantage of CCo-Patch is robust to the choice of patch length. We vary the patch length on Weather, ECL, ETTh2, and ETTm2 at $H = 720$, keeping the stride fixed at 8. For Weather, ETTh2, and ETTm2, the patch length ranges from 8 to 42 with a step of 2; for ECL, it ranges from 8 to 44 with a step of 4. We report MSE and Global-TRE for both models.

Table 4: Continuity analysis of different variants. Each entry reports the average value over the four forecasting horizons ($H \in \{96, 192, 336, 720\}$) for each dataset. Lower values indicate better temporal consistency.

Design				Weather		ECL		ETTh2		ETTm2	
Variant	Huber	Correction	M-C	IWCE	TRE	IWCE	TRE	IWCE	TRE	IWCE	TRE
PatchTST	×	×	×	0.357	1.02	0.342	0.832	0.410	1.06	0.207	1.14
+ Huber	✓	×	×	0.358	1.03	<u>0.336</u>	0.818	0.407	1.05	<u>0.202</u>	1.06
+ Correction	×	✓	×	0.359	1.02	0.342	0.832	0.410	1.06	0.206	1.12
+ Mean-Centering	×	×	✓	0.360	**0.995**	0.347	0.836	0.413	1.05	0.207	1.09
No-MC	✓	✓	×	0.359	1.01	**0.335**	<u>0.815</u>	**0.406**	1.05	**0.201**	1.06
No-FFT	✓	✓(No-FFT)	✓	**0.353**	1.00	0.339	**0.814**	**0.406**	**1.04**	0.203	**1.05**
CCo-Patch	✓	✓	✓	<u>0.355</u>	**0.995**	0.339	0.817	0.408	**1.04**	0.204	**1.05**

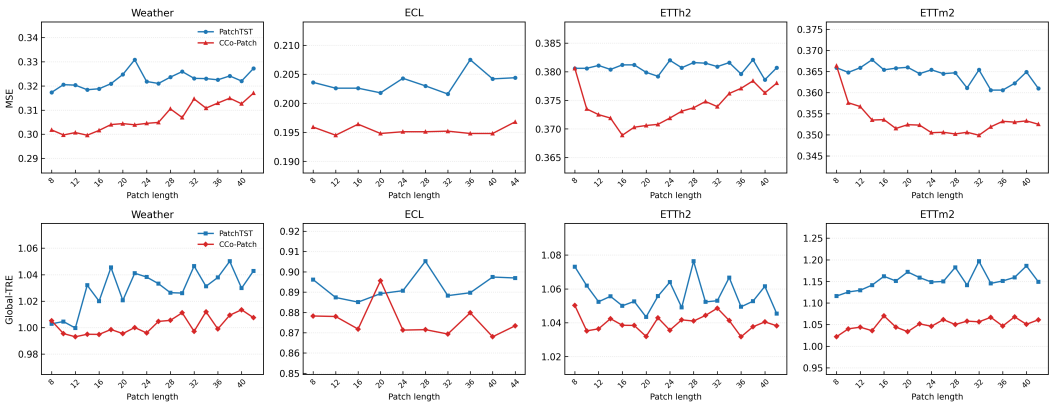

Figure 4: Robustness to patch granularity. MSE (top) and Global-TRE (bottom) as a function of patch length on Weather, ECL, ETTh2, and ETTm2 at $H = 720$, with stride fixed at 8.

As shown in Figure 4, CCo-Patch achieves lower MSE than PatchTST across nearly all patch lengths on all four datasets. The only exception is ETTm2 at patch=8, where the two models perform comparably (0.3664 vs. 0.3659). The advantage of CCo-Patch becomes more pronounced at larger patch lengths, where cross-patch continuity is harder to maintain: on Weather, the MSE gap grows from 0.0155 at patch=8 to 0.0227 at patch=18; on ECL, CCo-Patch maintains a consistent gap of approximately 0.007–0.009 in MSE (3%–5% relative to PatchTST) across all patch lengths. In terms of Global-TRE, CCo-Patch is consistently lower on ETTh2 and ETTm2, and lower on the majority of settings for Weather and ECL. The non-monotonic TRE of PatchTST reflects a trade-off between preserving intra-patch information and over-compressing local variation, whereas CCo-Patch maintains a more stable and lower TRE across the same range.

5 Conclusion

We presented CCo-Patch, a continuous correction framework that augments a patch-based forecaster with an explicit temporal scaffold. The Correction branch extracts a low-frequency scaffold through a causal filter and fuses it with the patch-based prediction, providing a structural reference for neighboring temporal segments. Ablation studies reveal an asymmetric complementarity between the Correction branch and the Huber loss, where the Correction branch contributes the primary MSE gain and the Huber loss preserves MAE. Continuity-oriented evaluation and the patch-granularity analysis further suggest that CCo-Patch improves cross-patch continuity and that this advantage is robust across patch lengths.

The proposed mechanism is conceptually orthogonal to the specific patch-based backbone and may transfer to other patch-based models, but the scaffold can suppress high-frequency variations, and its empirical validation on other backbones is left to future work.

REFERENCES

Sachith Abeywickrama, Emadeldeen Eldele, Min Wu, Xiaoli Li, and Chau Yuen. Entrope: Entropy-guided dynamic patch encoder for time series forecasting, 2025.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018.

Gaspard Berthelier, Tahar Nabil, Etienne Le Naour, Richard Niamke, Samir Perlaza, and Giovanni Neglia. On the role of reversible instance normalization. *arXiv preprint arXiv:2603.11869*, 2026.

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

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

Xingsheng Chen, Deyu Yi, and Siu Ming Yiu. Structured latent space modeling over multi-scale temporal patches for multivariate time series forecasting, 2026.

Srinath Dama, Kevin L. Course, and Prasanth B. Nair. Shifting time: Time-series forecasting with khatri-rao neural operators. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 12402–12435, 2025.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. Timemosaic: Temporal heterogeneity guided time series forecasting via adaptive granularity patch and segment-wise decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20790–20798, 2026. doi: 10.1609/aaai.v40i25.39218.

Zeying Gong, Yujin Tang, and Junwei Liang. Patchmixer: A patch-mixing architecture for long-term time series forecasting, 2024.

Ao Hu, Liangjian Wen, Jiang Duan, Yong Dai, He Yan, Dongkai Wang, Jun Wang, Yukun Zhang, Ruoxi Jiang, and Zenglin Xu. Pmdformer: Patch-mean decoupling information transformer for long-term forecasting. In *International Conference on Learning Representations*, 2026.

Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, and Yang Wang. Timebase: The power of minimalism in efficient long-term time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 26227–26246, 2025.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 37797–37814, 2025.

Jiahao Ling, Xuan Yang, Shimin Gong, and Bo Gu. Eapformer: Entropy-aware patch transformer for multivariate long-term time series forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM)*, 2025.

Kun Liu, Zhongjie Duan, Cen Chen, Yanhao Wang, Dawei Cheng, and Yuqi Liang. Adapatch: Adaptive patch-level modeling for non-stationary time series forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM)*, 2025.

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

Li Su, Xinyi Zuo, Rui Li, Xinyu Wang, Hui Zhao, and Biao Huang. A systematic review for transformer-based long-term series forecasting. *Artificial Intelligence Review*, 58(3), 2025. doi: 10.1007/s10462-024-11044-2.

Tomislav Volarić, Hrvoje Ljubić, Daniel Vasić, and Robert Rozić. Should MSE remain the default benchmarking loss function for long-term time series forecasting? *SN Computer Science*, 6(7): 804, 2025. doi: 10.1007/s42979-025-04341-2.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024a.

Yulong Wang, Yushuo Liu, Xiaoyi Duan, and Kai Wang. Filterts: Comprehensive frequency filtering for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21375–21383, 2025. doi: 10.1609/aaai.v39i20.35438.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guoqing Liu, Mingsheng Long, and Jianmin Wang. Timexer: Empowering transformers for time series forecasting with exogenous variables, 2024b.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024c.

Guang-Yu Wei, Hui-Chuan Huang, Zhi-Qing Zhong, Wen-Long Sun, Yong-Hao Wan, and Ai-Min Feng. Dualmamba: a patch-based model with dual mamba for long-term time series forecasting. *Frontiers of Computer Science*, 20(2):2002315, 2026. doi: 10.1007/s11704-025-41293-5.

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 *International Conference on Learning Representations*, 2023.

Yuemin Wu, Zhongze Wu, Xiu Su, Feng Yang, Hongyan Xu, Xi Lin, Wenti Huang, Shan You, and Chang Xu. Decop: Enhancing self-supervised time series representation with dependency controlled pre-training, 2025. Preprint; introduces dependency-controlled pre-training and involves instance-wise patch normalization.

Huibo Xu, Likang Wu, Xianquan Wang, Haoning Dang, Chun-Wun Cheng, Angelica I. Avilés-Rivero, and Qi Liu. Neutsflow: Modeling continuous functions behind time series forecasting, 2025.

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

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.

Jian Zhang, Lili Guo, Lei Song, Song Gao, Chuanzhu Hao, and Xuzhi Li. Patchtcn: Patch-based transformer convolutional network for times series analysis. In *Proceedings of the 2024 3rd International Symposium on Computing and Artificial Intelligence (ISCAI)*, pp. 1–9, 2024. doi: 10.1145/3711507.3711508.

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

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

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022.

AI USE STATEMENT

The authors used Large Language Models (LLMs) to aid in polishing the writing of this paper and to draft portions of the manuscript. Specifically, LLMs were used for language refinement, improving clarity and readability of the text, and generating initial drafts of certain sections. The authors reviewed, edited, and verified all AI-assisted content, and take full responsibility for the final content of this work, including all text, claims, and artifacts.

A ADDITIONAL IMPLEMENTATION AND EFFICIENCY DETAILS

This appendix provides supplementary details on the CCo-Patch implementation, including efficiency comparisons with the PatchTST backbone, the FFT-based spectral filter, the Huber loss, hyperparameter configurations, and sensitivity analysis.

A.1 EFFICIENCY COMPARISON

Table 5 compares CCo-Patch with the PatchTST backbone on the Weather dataset at $H \in \{96, 192, 336, 720\}$. All measurements are conducted on a single NVIDIA RTX 4090 GPU with a fixed batch size. Inference time is reported as the median over all test batches. Peak GPU memory is measured during inference.

Table 5: Efficiency comparison between PatchTST and CCo-Patch on Weather. Inference time is the median over all test batches. Peak GPU memory is measured during inference.

H	Method	Params (M)	Inference (ms/batch)	Peak GPU (GB)	MSE	MAE
96	PatchTST	0.82	12.90	0.67	0.1499	0.1970
	CCo-Patch	0.95	14.94	0.85	**0.1454**	**0.1856**
192	PatchTST	1.34	12.96	0.68	0.1985	0.2437
	CCo-Patch	1.48	14.67	0.86	**0.1891**	**0.2285**
336	PatchTST	2.11	24.55	1.34	0.2499	0.2841
	CCo-Patch	2.28	29.24	1.69	**0.2370**	**0.2698**
720	PatchTST	4.18	24.93	1.36	0.3189	0.3340
	CCo-Patch	4.39	29.52	1.72	**0.3017**	**0.3232**

The continuous correction branch adds 0.13M–0.21M parameters (5%–16% over PatchTST), depending on the horizon. Inference time and peak GPU memory increase by approximately 13–20% and 0.18–0.36 GB, respectively. This modest overhead reflects the cost of the FFT-based scaffold extraction and the additional MLP head, and is consistent with the lightweight design goal of CCo-Patch.

A.2 FFT IMPLEMENTATION AND HYPERPARAMETER CONFIGURATION

The spectral filter $\mathbf{M}$ is a learnable mixture of two Gaussian masks:

$$\mathbf{M}[i] = w_1 \exp\left(-\frac{(i-\mu_1)^2}{2\sigma_1^2}\right) + w_2 \exp\left(-\frac{(i-\mu_2)^2}{2\sigma_2^2}\right), \tag{9}$$

where w_1, w_2 are obtained via a softmax over two learnable logits, and the bandwidths are fixed to $\sigma_1 = 1$ and $\sigma_2 = 2$. The centers (μ_1, μ_2) are selected per dataset via grid search over $\{0, \ldots, 8\}^2$, using two criteria: smoothness of the filtered curve (mean absolute second-order difference) and

preservation of the underlying trend (correlation between filtered and original curves). For each dataset, we first keep pairs whose trend-preservation score exceeds a threshold, then choose the smoothest among them. The first mask ($\mu_1 = 0$) typically captures the global trend, while the second ($\mu_2 > 0$) captures a dominant low-frequency cycle. With a sliding window of length $W = 96$, a center at frequency 4 corresponds to a daily cycle (24 hours) for hourly data, and a center at frequency 6 corresponds to about a 4-hour cycle for 15-minute data. The causal sliding window length is set to $W = 96$ for all experiments.

Table 6 reports the Huber δ and the selected Gaussian centers for each dataset. The Huber δ is selected from $\{0.05, 0.10, \ldots, 1.00\}$ via a grid search on the validation set.

Table 6: Huber loss parameter δ and Gaussian centers for each dataset.

Dataset	Huber δ	Gaussian centers (μ_1, μ_2)
ETTh1	0.80	(0, 4)
ETTh2	0.90	(4, 5)
ETTm1	0.60	(0, 6)
ETTm2	0.30	(0, 6)
Weather	0.10	(2, 3)
ECL	0.25	(0, 4)
Traffic	0.80	(0, 4)

A.3 Huber Loss

We use the Huber loss (Huber, 1964) during training:

$$\mathcal{L}_\delta(x, \hat{x}) = \begin{cases} \frac{1}{2}(x - \hat{x})^2, & |x - \hat{x}| \le \delta, \\ \delta \cdot \left(|x - \hat{x}| - \frac{1}{2}\delta\right), & |x - \hat{x}| > \delta, \end{cases} \tag{10}$$

where $\delta > 0$ is a threshold that balances quadratic behavior for small errors and linear behavior for large errors.

A.4 Sensitivity to δ

Table 7 reports MSE and MAE of CCo-Patch on Weather for different δ across all four horizons. Smaller δ favors MAE and larger δ favors MSE. For $\delta \in [0.10, 0.40]$, both metrics are near-optimal, indicating low sensitivity to δ. The selected $\delta = 0.10$ lies in this interval; for compactness, the table shows only values up to 0.75, covering the near-optimal range.

Table 7: Sensitivity of MSE / MAE to δ on Weather. Lower values are better.

δ	$H = 96$	$H = 192$	$H = 336$	$H = 720$
0.05	0.1465 / 0.1852	0.1893 / 0.2293	0.2377 / 0.2696	0.3051 / 0.3239
0.10	0.1458 / 0.1853	0.1880 / 0.2280	0.2357 / 0.2686	0.3026 / 0.3231
0.25	0.1453 / 0.1863	0.1887 / 0.2298	0.2351 / 0.2690	0.3021 / 0.3233
0.40	0.1455 / 0.1890	0.1886 / 0.2324	0.2364 / 0.2718	0.3029 / 0.3263
0.60	0.1456 / 0.1911	0.1885 / 0.2342	0.2365 / 0.2740	0.3044 / 0.3295
0.75	0.1457 / 0.1937	0.1884 / 0.2359	0.2375 / 0.2775	0.3038 / 0.3303