CoLaDe: Competitive Latent Compression and Lag-Aware Decoding for Time Series Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Multivariate time series forecasting relies on compact representations that preserve both temporal and inter-variable dependencies. Latent-bottleneck encoder–decoder models achieve this by compressing patch tokens from all variables into a small set of latent tokens and decoding future patches with learnable target queries. However, standard cross-attention does not regulate how latents allocate their attention across the input, which can lead to redundant compression. To address this limitation, this paper proposes Competitive Latent Compression and Lag-Aware Decoding (CoLaDe), which augments latent cross-attention with a signed competition term that reweights attention to tokens according to their relative preference across latents. The competition strength is learned per attention head, which allows latents either to specialize in distinct input regions or to share them. To retrieve historical information, a relative-lag prior explicitly models the temporal distance between each target query and observed patch. A lightweight low-rank correction then adapts this prior for each variable, accommodating heterogeneous temporal dependencies. Extensive evaluations on several real-world benchmark datasets show that CoLaDe achieves superior forecasting performance compared with state-of-the-art baselines.

1 Introduction

Time series forecasting is a fundamental task in machine learning with broad applications, including energy systems, healthcare, finance, and weather prediction (Wang et al., 2024b). In the multivariate setting, the task requires a model to exploit dependencies among variates while preserving the temporal dynamics of each one. Both requirements become harder as the input length and the number of variates grow, because relevant dependencies span more observations and channels.

Recent work has approached this problem from several directions. Variate-oriented designs treat each variate as a token and model cross-variate dependencies through attention (Liu et al., 2024). Subsequent models refine these interactions by aligning tokens across variates (Wang et al., 2024a) or by separating query construction from the input sequence (Lu et al., 2024). Lightweight models capture much of the forecasting signal with far fewer parameters (Ma et al., 2026a). Other methods exploit periodic structure through two-dimensional representations (Nematirad et al., 2025; Dai et al., 2024), period-aware inductive biases (Ma et al., 2025; 2026b), or multiple temporal scales (Wang et al., 2025). A further approach compresses the input into a small set of latent tokens and decodes future values using learnable target queries, avoiding self-attention over all input patches (Lee & Lee, 2025).

This latent-bottleneck design is attractive because the compressed representation is read jointly from all variates, capturing temporal and cross-variate structure in one stage. Two limitations, however, remain. **(i)** Although each latent token normalizes its attention over input patches, the mechanism does not explicitly discourage different latents from attending to the same patches. Several latents may therefore concentrate on dominant regions, leaving other regions weakly represented. **(ii)** The decoder retrieves history using content and learned absolute positional embeddings, but does not explicitly represent the relative lag between a target query and an observed patch. Relative lag may help identify useful recent or recurring patterns, which can differ across variates. We propose Com-

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.

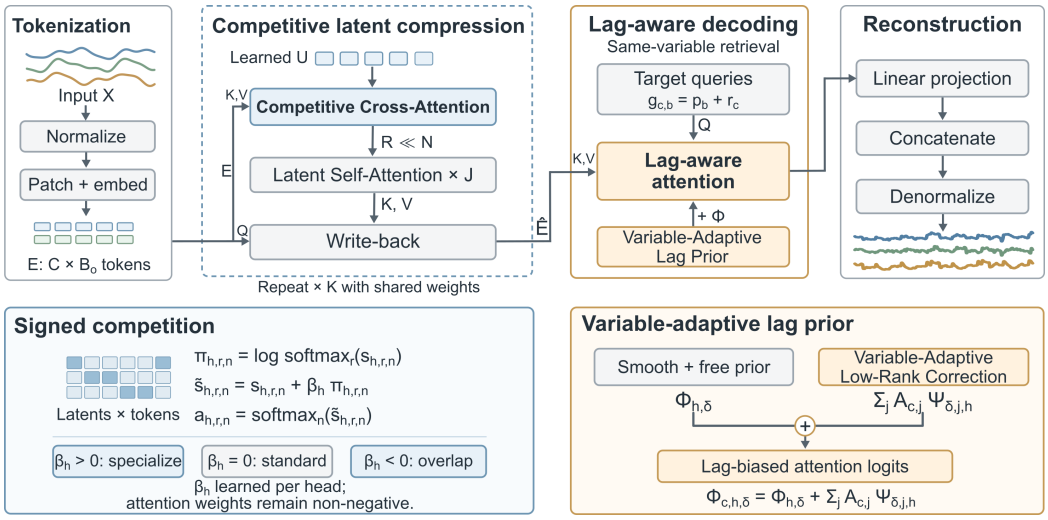

Figure 1: Overview of CoLaDe: competitive latent compression and lag-aware decoding.

petitive Latent Compression and Lag-Aware Decoding (CoLaDe) to address these limitations. For compression, a learned signed coefficient adjusts each latent's attention according to its preference for an input patch relative to the other latents. The coefficient is learned separately for each attention head, allowing the model to learn whether latents should specialize or share input regions. When the coefficient is zero, the mechanism reduces exactly to standard cross-attention. For decoding, an explicit relative-lag prior is added to the query–patch attention logits. It combines a smooth component that generalizes across neighboring lags with a free component that adjusts individual offsets. A low-rank correction adapts the prior to each variate with parameter cost linear in the number of variates. These changes require no additional attention stage or separate forecasting branch. The overall architecture is shown in Figure 1.

Contributions. The main contributions of this work are as follows:

1. We introduce competitive latent compression, which learns how latent tokens share or specialize in representing input patches through a head-specific adjustment to the cross-attention logits.
2. We introduce lag-aware decoding, which combines smooth and offset-specific relative-lag biases with a low-rank, variate-adaptive correction to guide attention to relevant history.
3. Comprehensive experiments on several real-world benchmark datasets demonstrate that CoLaDe achieves state-of-the-art performance in multivariate long-term forecasting, while maintaining competitive computational efficiency.

2 Related Work

Patch-Based and Multivariate Forecasting. Recent forecasting models increasingly construct representations from local temporal segments rather than individual time points. PatchTST (Nie et al., 2023) introduces channel-independent patching to preserve local temporal semantics while reducing sequence length. Crossformer (Zhang & Yan, 2023) explicitly models cross-time and cross-variable dependencies through two-stage attention, whereas iTransformer (Liu et al., 2024) represents entire variable histories as tokens and applies attention across variables. Recent extensions further explore adaptive patch representations: SRSNet (Wu et al., 2026) selectively constructs informative patch spaces, while MCV-PatchTST (Qarni et al., 2026) combines multi-scale patching with lightweight cross-variate attention. These studies highlight the importance of token construction and variable interaction in multivariate forecasting.

Attention and Latent Representation. Recent studies have revisited how attention selects, represents, and compresses temporal information. DeformableTST (Luo & Wang, 2024) employs data-

driven sparse attention to focus on informative temporal locations, while FSatten and SOatten (Wu, 2025) reconsider the representation space in which attention dependencies are learned. TimeXer (Wang et al., 2024b) combines patch-wise and variate-wise attention with global tokens that mediate information exchange across representations. Latent-space modeling has also been explored through LDT (Feng et al., 2024), which compresses high-dimensional multivariate series before probabilistic forecasting, and LNO (Wang & Wang, 2024), which uses cross-attention to map observations into and out of a compact latent space. More closely related to our setting, TimePerceiver (Lee & Lee, 2025) compresses input patches into a fixed set of latent tokens and retrieves information through target queries, while PhaseFormer (Niu et al., 2026) employs compact routers to aggregate and redistribute phase representations. However, these approaches do not explicitly regulate how multiple latent representations allocate overlapping input information.

Structured Temporal Dependencies and Variable Heterogeneity. Another line of research explicitly incorporates temporal structure into the forecasting process. PDF (Dai et al., 2024) exploits periodicity to decouple short- and long-term variations, while Times2D (Nematirad et al., 2025) transforms dominant periodic components and derivative information into two-dimensional representations. FMDP-Transformer (Liu et al., 2026) further combines frequency-guided multi-scale representations with decomposition and patch-based residual modeling. PHAT (Ma et al., 2026b) considers heterogeneous periodic behavior across variables, while THCVformer (Ma et al., 2026c) jointly models temporal dependencies, variable heterogeneity, and cross-variable correlations. These methods demonstrate the benefit of structured temporal modeling, but rely on periodic decomposition, reorganization, or variable-level processing rather than an explicit variable-adaptive lag prior during decoding. In contrast, CoLaDe focuses jointly on information allocation and retrieval within latent-bottleneck forecasting. Competitive latent compression regulates how input patches are distributed across latent representations, while lag-aware decoding explicitly models the relative temporal distance between target and observed patches. A low-rank variable-adaptive correction further allows this temporal prior to vary across variables.

3 METHODOLOGY

Given a multivariable time series $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L] \in \mathbb{R}^{L \times C}$, where L represents the input sequence length and C denotes the number of variables, the forecasting task is to predict the next P time steps $\mathbf{Y} = [\mathbf{x}_{L+1}, \ldots, \mathbf{x}_{L+P}] \in \mathbb{R}^{P \times C}$.

The proposed CoLaDe model consists of two main components. First, *Competitive Latent Compression* maps the observed patch tokens into a compact latent representation while controlling how the latent tokens attend to the input. Second, *Lag-Aware Decoding* retrieves historical information for future patches by considering both content similarity and relative temporal distance. In addition, a lightweight low-rank correction adapts the temporal prior to different variables.

First, the input time series is normalized and divided into non-overlapping temporal patches. Then, the patch tokens from all variables are jointly compressed through the competitive latent mechanism and updated using the compressed information. Finally, target queries attend to the historical tokens of the corresponding variable using lag-aware attention to generate the forecasting output. The latent-bottleneck and query-decoding backbone follows prior encoder–decoder forecasting frameworks (Lee & Lee, 2025), while CoLaDe modifies the attention logits used for compression and decoding. The proposed mechanisms do not introduce a frequency-domain branch or a separate forecasting branch.

3.1 PATCH TOKENIZATION

Given a patch length ρ, the input sequence is divided into $B_o = L/\rho$ observed patches, while the prediction sequence contains $B_f = P/\rho$ target patches. The complete number of patch positions is $B = B_o + B_f$, with observed indices $\mathcal{O} = \{1, \ldots, B_o\}$ and future target indices $\mathcal{T} = \{B_o + 1, \ldots, B\}$. Divisibility is assumed for notational simplicity. Following the normalization–denormalization scheme of RevIN (Kim et al., 2022), each variable is standardized using its statistics over the input sequence. For variable c, the normalized value at time step t is

calculated as

$$\widetilde{x}_{t,c} = \frac{x_{t,c} - \mu_c}{\sigma_c}, \qquad \mu_c = \frac{1}{L}\sum_{t=1}^{L} x_{t,c}, \qquad \sigma_c = \sqrt{\frac{1}{L}\sum_{t=1}^{L}(x_{t,c} - \mu_c)^2 + \epsilon}, \tag{1}$$

where ϵ is a small constant for numerical stability. The same statistics are restored after prediction.

Next, let $\widetilde{\mathbf{x}}_{c,b} \in \mathbb{R}^{\rho}$ denote the normalized patch of variable c at observed patch position $b \in \mathcal{O}$. Each patch is projected into a d-dimensional token and combined with learnable temporal and variable embeddings,

$$\mathbf{e}_{c,b} = \mathbf{W}_{\text{in}}\widetilde{\mathbf{x}}_{c,b} + \mathbf{p}_b + \mathbf{r}_c, \qquad \mathbf{e}_{c,b} \in \mathbb{R}^d, \tag{2}$$

where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d\times\rho}$ denotes the patch projection, $\mathbf{p}_b$ represents the temporal position embedding, and $\mathbf{r}_c$ denotes the variable embedding. The observed tokens are then flattened over variables and patches to obtain $\mathbf{E} \in \mathbb{R}^{N\times d}$ with $N = CB_o$. As a result, any mechanism that reads $\mathbf{E}$ observes temporal and cross-variable structure jointly.

3.2 Competitive Latent Compression

Processing all N patch tokens can be computationally costly and may also introduce redundant information when multiple latent representations focus on similar regions of the input. To address this issue, CoLaDe compresses the observed tokens into $R \ll N$ learnable latent tokens $\mathbf{U} \in \mathbb{R}^{R\times d_u}$ and introduces a signed competition mechanism to regulate how the input information is allocated among them.

Attention Block. First, the attention block used throughout the model is defined as

$$\begin{aligned} \mathbf{H} &= \mathbf{Q} + \text{Drop}\left[\text{Attn}\left(\text{LN}_q(\mathbf{Q}), \text{LN}_k(\mathbf{K}); \mathbf{B}\right)\right], \\ \text{Block}(\mathbf{Q}, \mathbf{K}; \mathbf{B}) &= \mathbf{H} + \text{Drop}\left[\text{FFN}\left(\text{LN}(\mathbf{H})\right)\right], \end{aligned} \tag{3}$$

where $\mathbf{B}$ denotes an optional attention-logit bias. The key and value representations are obtained from the same normalized context $\mathbf{K}$. The FFN contains two linear layers with GELU activation.

Compression, Refinement, and Write-Back. Then, the latent and observed representations are updated as

$$\begin{aligned} \mathbf{U} &\leftarrow \text{Block}_\beta(\mathbf{U}, \mathbf{E}), \\ \mathbf{U} &\leftarrow \text{Block}(\mathbf{U}, \mathbf{U}), \\ \widehat{\mathbf{E}} &\leftarrow \text{Block}(\mathbf{E}, \mathbf{U}), \end{aligned} \tag{4}$$

where Block_β denotes the proposed competitive cross-attention block. The first operation performs competitive compression, the second exchanges information within the compact latent set, and the final operation writes the compressed information back to the observed tokens, yielding the encoded representations $\widehat{\mathbf{E}}$. When $K > 1$, $\widehat{\mathbf{E}}$ becomes the observed-token representation supplied to the next latent round. The operations are repeated using the same modules. As a result, each observed token keeps its temporal and variable identity while incorporating the global information captured through the latent representation.

Signed Competitive Attention. For attention head h, latent r, and input token n, the temperature-scaled content score is

$$\begin{aligned} s_{h,r,n} &= \gamma_h^{\text{cmp}} \frac{\mathbf{q}_{h,r}^\top \mathbf{k}_{h,n}}{\sqrt{d_h}}, \qquad \gamma_h^{\text{cmp}} = \exp(-\eta_h^{\text{cmp}}) > 0, \\ c_{h,r,n} &= \log \frac{\exp(s_{h,r,n})}{\sum_{r'=1}^{R} \exp(s_{h,r',n})}, \end{aligned} \tag{5}$$

where d_h is the dimension of each attention head, η_h^{cmp} denotes the learnable compression log-temperature, and the normalization of $c_{h,r,n}$ is performed across the latent dimension. The competition term is added to the original score using a learnable signed coefficient β_h,

$$\widetilde{s}_{h,r,n} = s_{h,r,n} + \beta_h c_{h,r,n}, \qquad \beta_h \in \mathbb{R}. \tag{6}$$

Finally, the modified scores are normalized over the input-token dimension,

$$a_{h,r,n} = \frac{\exp(\widetilde{s}_{h,r,n})}{\sum_{n'=1}^{N} \exp(\widetilde{s}_{h,r,n'})},$$
$$\mathbf{z}_{h,r} = \sum_{n=1}^{N} a_{h,r,n} \mathbf{v}_{h,n}, \tag{7}$$

where $\mathbf{v}_{h,n}$ denotes the value representation of input token n. When $\beta_h > 0$, tokens with a stronger relative preference for a particular latent receive more emphasis, which encourages a more differentiated allocation. When $\beta_h < 0$, different latents are allowed to concentrate more strongly on similar input regions. When $\beta_h = 0$, the proposed mechanism reduces exactly to standard cross-attention. Learning β_h end-to-end therefore makes the degree of specialization a property of the data rather than a design choice, at a cost of H additional scalars. Although β_h is signed, the final attention weights remain non-negative, since Eq. 7 applies a softmax normalization.

Equivalent Form. Substituting Eq. 5 into Eq. 6 yields an equivalent expression,

$$\widetilde{s}_{h,r,n} = (1 + \beta_h)\, s_{h,r,n} - \beta_h \log \sum_{r'=1}^{R} \exp(s_{h,r',n}), \tag{8}$$

Eq. 8 reveals two effects of the competition coefficient: it changes the weight placed on the latent–token affinity and introduces a token-specific term determined by the total affinity across all latents. Because the latter term varies across input tokens, it remains active under the subsequent token-wise softmax and reweights tokens according to how strongly they are preferred across the latent set. The learnable temperature γ_h^{cmp} and β_h are separately parameterized: the former scales the base logits, whereas the latter controls the sign and strength of competitive reweighting. The complete derivation is provided in Appendix A.

3.3 Lag-Aware Decoding

The importance of a historical patch depends not only on its learned feature representation but also on its temporal distance from the target. To consider this temporal information, CoLaDe introduces an explicit relative-lag prior into the decoder attention. For each target patch $b \in \mathcal{T}$ and variable c, the decoder query is constructed as $\mathbf{g}_{c,b} = \mathbf{p}_b + \mathbf{r}_c$, which reuses the embeddings of Eq. 2 and aligns the positional reference of the decoder with that of the encoder. The queries of variable c attend only to the encoded observed tokens of the same variable. As a result, cross-variable information is captured during the shared latent compression stage, while the final retrieval remains variable-specific.

Relative-Lag Representation. For a target patch $b \in \mathcal{T}$ and an observed patch $b' \in \mathcal{O}$, the relative lag and its normalized value are defined as

$$\delta_{b,b'} = b - b', \qquad \bar{\delta}_{b,b'} = \frac{\delta_{b,b'}}{B-1}. \tag{9}$$

Each lag is represented using its signed distance, absolute distance, and a set of sinusoidal features,

$$\boldsymbol{\phi}(\delta) = \left[\bar{\delta}, |\bar{\delta}|, \sin(2\pi\bar{\delta}), \cos(2\pi\bar{\delta}), \ldots, \sin(2\pi K_f \bar{\delta}), \cos(2\pi K_f \bar{\delta})\right]. \tag{10}$$

Shared Relative-Lag Prior. The lag features are passed through a lightweight nonlinear mapping and combined with a free learnable lookup value,

$$\Phi_{h,\delta} = \underbrace{\mathbf{w}_h^\top \mathbf{m}(\delta)}_{\text{smooth component}} + \underbrace{\tau_{h,\delta}}_{\text{free component}}, \qquad \mathbf{m}(\delta) = \mathrm{GELU}\left(\mathbf{W}_f \boldsymbol{\phi}(\delta)\right), \tag{11}$$

where $\mathbf{m}(\delta) \in \mathbb{R}^{d_f}$ denotes the hidden lag feature and $\tau_{h,\delta}$ is a learnable value for each discrete relative lag. The smooth component captures structure shared among related lags, while the free component allows individual temporal offsets to be adjusted separately. Both shared components are initialized at zero, so the prior is inactive at initialization and is learned only when supported by the forecasting loss.

Variable-Adaptive Low-Rank Correction. Different variables can exhibit different temporal dependencies when they respond at different delays or contain different dominant cycles. Learning a complete lag table independently for every variable, however, would introduce a large number of additional parameters. Instead, CoLaDe adapts the shared prior using a low-rank correction,

$$\Phi_{c,h,\delta} = \Phi_{h,\delta} + \sum_{j=1}^{q} A_{c,j}\Psi_{\delta,j,h}, \qquad \Psi_{\delta,j,h} = [\mathbf{W}_c\mathbf{m}(\delta)]_{j,h}\,, \tag{12}$$

where $\mathbf{A} \in \mathbb{R}^{C\times q}$ contains the variable-specific coefficients, q denotes the correction rank, and $\Psi_{\delta,j,h}$ represents the lag-dependent basis. The basis reuses the hidden feature $\mathbf{m}(\delta)$ already computed in Eq. 11, so the correction requires only $q(C+d_fH)$ additional parameters, which is linear in V and negligible for small q. The coefficient matrix $\mathbf{A}$ is initialized to zero, while $\mathbf{W}_c$ is initialized at a small nonzero scale. Accordingly, all variables begin from the shared temporal prior while the correction can receive gradient from the first training step.

Lag-Aware Attention. Let $\widehat{\mathbf{e}}_{c,b'}$ denote the encoded representation of observed patch b' obtained from Eq. 4. The decoder uses its own head-wise temperature $\gamma_h^{\mathrm{dec}} = \exp(-\eta_h^{\mathrm{dec}}) > 0$. The decoder score and attention weights are calculated as

$$d_{c,h,b,b'} = \gamma_h^{\mathrm{dec}}\left[\frac{(\mathbf{W}_h^Q\mathbf{g}_{c,b})^\top(\mathbf{W}_h^K\widehat{\mathbf{e}}_{c,b'})}{\sqrt{d_h}} + \Phi_{v,h,b-b'}\right], \qquad \alpha_{c,h,b,:} = \mathrm{Softmax}_{b'}\left(d_{c,h,b,:}\right). \tag{13}$$

The target representation is obtained by aggregating the observed value representations,

$$\mathbf{o}_{c,h,b} = \sum_{b'\in\mathcal{O}} \alpha_{c,h,b,b'}\mathbf{W}_h^V\widehat{\mathbf{e}}_{c,b'}. \tag{14}$$

The outputs from all heads are combined and passed through the residual and feed-forward updates in Eq. 3.

Forecast Reconstruction. Let $\widehat{\mathbf{g}}_{c,b} \in \mathbb{R}^d$ denote the decoded target token. A shared linear projection maps each target representation back to a patch,

$$\widehat{\mathbf{x}}_{c,b} = \mathbf{W}_{\mathrm{out}}\widehat{\mathbf{g}}_{c,b}, \qquad \mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{\rho\times d}, \qquad b \in \mathcal{T}. \tag{15}$$

The predicted patches are concatenated along the temporal dimension and the normalization statistics from Eq. 1 are restored to obtain $\widehat{\mathbf{Y}} \in \mathbb{R}^{P\times C}$.

3.4 Training Objective

CoLaDe is trained end-to-end using the mean squared forecasting error over all variables and future time steps,

$$\mathcal{L}_{\mathrm{forecast}} = \frac{1}{CP}\sum_{c=1}^{C}\sum_{t=1}^{P}\left(\widehat{Y}_{t,c} - Y_{t,c}\right)^2. \tag{16}$$

This objective is used to jointly learn the patch representation, competitive latent compression, shared relative-lag prior, variable-specific correction, and forecasting decoder. During training, the architecture can additionally operate on alternative observed and target patch indices while preserving their original temporal identities (Lee & Lee, 2025). This capability is used as a training strategy, while evaluation follows the standard forecasting setting defined in Section 3.1. The hyperparameter settings are reported in Appendix B.

4 Experiments

Datasets. CoLaDe is evaluated on seven widely used real-world benchmark datasets: Weather, Solar-Energy, Electricity, and the four ETT datasets (ETTh1, ETTh2, ETTm1, and ETTm2), consistent with prior forecasting benchmarks (Zhou et al., 2021). These datasets span different domains, sampling frequencies, and numbers of variables, covering both low- and high-dimensional multivariate forecasting settings. Further dataset details are provided in Appendix C.

Baselines. To evaluate the effectiveness of CoLaDe on multivariate time series forecasting, we select nine high-performing baseline models spanning a wide range of architectures. Specifically, we utilize DeformableTST (Luo & Wang, 2024), CARD (Wang et al., 2024a), CATS (Lu et al., 2024), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), TiDE (Das et al., 2023), MixLinear (Ma et al., 2026a), Crisp (Wang et al., 2026), and Times2D (Nematirad et al., 2025).

Setup and Evaluation Metrics. The input sequence length is fixed to $L \in \{96, 384, 768\}$ for all datasets, and prediction horizons are $P \in \{96, 192, 336, 720\}$. Forecasting accuracy is evaluated using mean squared error (MSE) and mean absolute error (MAE), where lower values indicate better performance. Computational efficiency is assessed using multiple metrics: (1) average training time per epoch; (2) peak allocated GPU memory; (3) inference latency per batch; and (4) the number of trainable parameters. All timing metrics are reported in seconds and memory usage in megabytes (MB).

Infrastructure. All experiments are conducted on Linux-based high-performance computing nodes equipped with NVIDIA A100-SXM4 GPUs with 40 GB of memory. The implementation uses PyTorch within Conda-managed environments.

5 RESULTS

Table 1: Multivariate time series forecasting results using the best result selected independently from input lengths $L \in \{96, 384, 768\}$ for each dataset, prediction horizon, model, and metric. Best results are shown in **red**, and second-best results are blue.

Models		CoLaDe (Ours)		MixLinear (2026)		Crisp (2026)		Times2D (2025)		DeformableTST (2024)		CARD (2024)		CATS (2024)		iTransformer (2024)		TiDE (2024)		PatchTST (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.355	0.389	**0.351**	**0.383**	0.382	0.394	0.362	0.395	0.365	0.396	0.377	0.402	0.371	0.395	0.386	0.405	0.438	0.448	0.381	0.410
	192	**0.391**	**0.410**	0.395	0.414	0.428	0.423	0.405	0.423	0.407	0.417	0.404	0.418	0.416	0.422	0.432	0.436	0.479	0.471	0.418	0.432
	336	0.392	**0.411**	0.411	0.434	0.474	0.447	0.423	0.433	**0.391**	0.414	0.428	0.435	0.435	0.432	0.472	0.458	0.491	0.481	0.439	0.452
	720	0.425	**0.442**	**0.423**	0.456	0.479	0.469	0.445	0.464	0.447	0.462	0.437	0.459	0.451	0.460	0.503	0.491	0.499	0.504	0.470	0.481
	Avg	**0.391**	**0.413**	0.395	0.422	0.441	0.433	0.409	0.429	0.403	0.422	0.412	0.429	0.418	0.427	0.448	0.448	0.477	0.476	0.427	0.444
ETTh2	96	**0.268**	**0.332**	0.283	0.340	0.288	0.336	0.271	0.334	0.269	0.334	0.275	0.336	0.276	0.338	0.297	0.346	0.353	0.422	0.274	0.337
	192	0.330	0.380	0.337	0.376	0.360	0.385	0.334	0.378	**0.325**	**0.369**	0.339	0.376	0.339	0.384	0.380	0.400	0.427	0.462	0.334	0.376
	336	**0.311**	0.376	0.356	0.395	0.399	0.421	0.329	0.385	0.319	**0.373**	0.360	0.397	0.368	0.403	0.427	0.432	0.451	0.489	0.356	0.402
	720	**0.376**	**0.422**	0.380	0.423	0.419	0.436	0.379	0.423	0.395	0.431	0.390	0.425	0.408	0.433	0.427	0.445	0.490	0.519	0.396	0.437
	Avg	**0.321**	0.378	0.339	0.384	0.367	0.395	0.328	0.380	0.327	**0.377**	0.341	0.384	0.348	0.390	0.383	0.406	0.430	0.473	0.340	0.388
ETTm1	96	**0.281**	**0.333**	0.322	0.363	0.297	0.340	0.295	0.343	0.292	0.344	0.295	0.342	0.284	0.339	0.314	0.368	0.306	0.341	0.296	0.350
	192	**0.318**	**0.356**	0.350	0.378	0.344	0.368	0.344	0.382	0.326	0.371	0.339	0.371	0.319	0.363	0.349	0.389	0.334	0.364	0.331	0.371
	336	**0.349**	**0.377**	0.385	0.399	0.383	0.392	0.352	0.381	0.358	0.392	0.367	0.387	0.354	0.382	0.387	0.410	0.363	0.384	0.363	0.393
	720	**0.392**	**0.406**	0.439	0.434	0.443	0.430	0.410	0.411	0.417	0.418	0.415	0.414	0.402	0.416	0.461	0.450	0.411	0.408	0.419	0.433
	Avg	**0.335**	**0.368**	0.374	0.394	0.367	0.383	0.350	0.379	0.348	0.381	0.354	0.379	0.340	0.375	0.378	0.404	0.354	0.374	0.352	0.387
ETTm2	96	**0.158**	**0.244**	0.177	0.267	0.163	0.249	0.165	0.258	0.167	0.257	0.164	0.255	0.166	0.260	0.180	0.264	0.174	0.283	0.174	0.259
	192	**0.216**	**0.286**	0.233	0.305	**0.216**	0.292	0.220	0.296	0.227	0.295	0.225	0.298	0.232	0.306	0.240	0.309	0.239	0.322	0.239	0.302
	336	**0.270**	**0.321**	0.284	0.337	0.273	0.330	0.272	0.330	0.278	0.327	0.274	0.327	0.278	0.335	0.299	0.348	0.297	0.361	0.296	0.343
	720	**0.336**	**0.376**	0.365	0.390	0.366	0.387	0.346	0.381	0.347	0.384	0.367	0.383	0.363	0.391	0.397	0.407	0.383	0.423	0.372	0.396
	Avg	**0.245**	**0.307**	0.265	0.325	0.255	0.315	0.251	0.316	0.255	0.316	0.258	0.316	0.260	0.323	0.279	0.332	0.273	0.347	0.270	0.325
Weather	96	0.142	0.194	0.179	0.235	0.146	**0.187**	0.145	0.197	0.144	0.198	0.150	0.208	**0.141**	0.198	0.163	0.214	0.165	0.214	0.149	0.203
	192	**0.188**	0.238	0.223	0.270	0.197	**0.236**	0.191	0.240	0.192	0.238	0.207	0.250	0.189	0.242	0.207	0.251	0.206	0.252	0.192	0.242
	336	**0.238**	0.282	0.267	0.304	0.256	0.284	0.243	0.282	0.242	**0.278**	0.261	0.292	0.239	0.280	0.257	0.293	0.251	0.284	0.242	0.280
	720	0.313	0.333	0.329	0.351	0.322	0.333	0.315	0.335	**0.308**	0.331	0.346	0.346	**0.308**	0.331	0.320	0.340	0.315	0.334	0.312	**0.330**
	Avg	0.220	0.262	0.250	0.290	0.230	**0.260**	0.224	0.264	0.222	0.261	0.241	0.274	**0.219**	0.263	0.237	0.275	0.234	0.271	0.224	0.264
Solar	96	**0.158**	**0.210**	0.212	0.260	0.169	0.212	0.177	0.240	0.167	0.232	0.175	0.243	0.161	0.216	0.176	0.237	0.187	0.267	0.193	0.275
	192	**0.177**	0.228	0.227	0.262	0.186	**0.224**	0.188	0.252	0.185	0.248	0.209	0.262	0.189	0.229	0.196	0.257	0.214	0.288	0.210	0.283
	336	**0.188**	**0.229**	0.240	0.271	0.196	0.231	0.195	0.255	0.193	0.260	0.209	0.272	0.190	0.236	0.206	0.273	0.231	0.311	0.223	0.292
	720	**0.197**	**0.224**	0.240	0.265	0.215	0.238	0.204	0.251	**0.197**	0.263	0.227	0.281	0.198	0.246	0.218	0.275	0.234	0.316	0.236	0.296
	Avg	**0.180**	**0.223**	0.230	0.265	0.192	0.226	0.191	0.250	0.186	0.251	0.205	0.265	0.185	0.232	0.199	0.261	0.217	0.296	0.216	0.287
Electricity	96	**0.124**	**0.218**	0.139	0.231	0.133	0.228	0.126	0.222	0.131	0.231	0.131	0.229	0.125	0.221	0.140	0.240	0.135	0.238	0.129	0.236
	192	**0.143**	**0.237**	0.155	0.251	0.157	0.248	0.145	0.240	0.146	0.249	0.157	0.256	0.147	**0.237**	0.160	0.253	0.151	0.253	0.152	0.247
	336	**0.160**	**0.254**	0.172	0.266	0.177	0.266	0.161	0.255	0.166	0.267	0.169	0.268	0.162	0.256	0.177	0.269	0.172	0.271	0.170	0.271
	720	0.202	0.290	0.201	0.288	**0.164**	**0.285**	0.198	0.293	0.199	0.294	0.200	0.292	0.195	0.286	0.220	0.311	0.207	0.299	0.204	0.296
	Avg	**0.157**	**0.250**	0.167	0.259	0.158	0.257	0.158	0.253	0.161	0.260	0.164	0.261	**0.157**	**0.250**	0.174	0.268	0.166	0.265	0.164	0.263
1st Count		27	24	2	1	2	5	0	0	4	4	0	0	4	2	0	0	0	0	0	1

Table 2: Latent-pathway ablation on ETTh1 with $L = 768$. Results are averaged over three matched seeds per horizon. The lower error in each comparison is shown in bold.

	Full model		Without latent pathway	
Horizon	MSE	MAE	MSE	MAE
96	0.3701	0.3975	**0.3636**	**0.3917**
192	0.4225	0.4260	**0.3993**	**0.4129**
336	**0.3953**	**0.4141**	0.4086	0.4216
720	0.4689	0.4664	**0.4377**	**0.4564**

Table 1 reports the best forecasting results obtained across the three input lengths $L \in \{96, 384, 768\}$ for each model, dataset, prediction horizon, and evaluation metric. CoLaDe achieves consistently strong performance across the evaluated benchmarks. Averaged over the four ETT datasets, CoLaDe reduces MSE and MAE by approximately 3.1% and 2.0%, respectively, compared with the strongest competing model. On Solar, CoLaDe improves the average MSE and MAE by 2.7% and 1.3% over the corresponding second-best results, while matching the best average performance on Electricity. Overall, CoLaDe achieves 27 first-place MSE and 24 first-place MAE results, and attains the best or tied-best average MSE on six of the seven datasets. Full results for each input length $L \in \{96, 384, 768\}$ are provided in Appendix D.

Computational Efficiency. We evaluate parameter count, peak GPU memory, training time, and inference latency across different input lengths, prediction horizons, and numbers of variates. The computational cost of CoLaDe is mainly determined by the latent set size and data dimensionality rather than the prediction horizon: the model size remains nearly constant as the horizon increases and grows linearly with the number of variates, whereas memory and runtime scale with the number of variates and batch size. Detailed measurements and comparisons with representative baselines are provided in Appendix E.

Statistical Analysis and Robustness. To assess the robustness of CoLaDe to random initialization, we retrain the model using five different random seeds and report the mean and standard deviation of each metric in Appendix F. Across the ETT and Weather datasets, the standard deviations remain small, with values at or below 0.015 for MSE and 0.013 for MAE, and below 0.001 in the most stable settings. These variations are substantially smaller than the performance differences from the strongest baselines in Table 1, indicating that the gains of CoLaDe are consistent across random seeds.

5.1 Component Ablation

Latent pathway. We first examine the role of the latent pathway on ETTh1 with input length $L = 768$. Removing this pathway bypasses latent compression, self-refinement, and write-back while retaining the lag-aware decoder. Table 2 reports results averaged over three matched seeds per horizon, with all other settings fixed within each horizon. The full model achieves lower MSE and MAE at $H = 336$, whereas removing the pathway improves both metrics at the other horizons. This pattern holds across all three seeds, indicating that the benefit of latent processing depends on the forecasting configuration.

Competition and lag-aware decoding. We next retain the latent pathway and isolate the two proposed mechanisms on ETTm1 with $L = 768$. Removing competition restores standard cross-attention in the compression stage; removing lag-aware decoding disables both the shared relative-lag prior and the variate-adaptive correction. All variants are trained from scratch using identical hyperparameters and matched random seeds within each horizon, and results are averaged over three seeds per horizon.

Table 3 shows that removing either mechanism individually increases MSE and MAE at every horizon. Removing competition increases average MSE by 1.09%, while removing lag-aware decoding increases it by 2.33%. The largest relative increases occur at $H = 720$, reaching 1.50% and 3.82%, respectively. Removing both mechanisms increases average MSE by 2.52%, although it yields slightly lower MSE at $H = 192$. These results support the contribution of each mechanism within the full ETTm1 configuration.

Table 3: Component ablation on ETTm1 with $L = 768$. Checkmarks indicate enabled components. Results are averaged over three matched seeds per horizon. ΔMSE is the relative increase in horizon-averaged MSE compared with the full model. The lowest error in each column is shown in bold.

Components		MSE by prediction horizon				Average		ΔMSE
Competitive Latent Compression	Lag-Aware Decoding	96	192	336	720	MSE	MAE	(%)
✓	✓	**0.2838**	0.3430	**0.3726**	**0.4127**	**0.3530**	**0.3815**	—
—	✓	0.2859	0.3449	0.3764	0.4189	0.3565	0.3853	+1.09
✓	—	0.2878	0.3492	0.3796	0.4285	0.3613	0.3886	+2.33
—	—	0.2949	**0.3406**	0.3891	0.4231	0.3619	0.3900	+2.52

6 Conclusion

This work introduces CoLaDe, a latent-bottleneck forecasting framework that combines competitive latent compression with lag-aware decoding. Competitive latent compression regulates how input information is allocated across latent tokens, while lag-aware decoding explicitly models the relative temporal distance between target and observed patches. A lightweight variable-adaptive low-rank correction further allows the temporal prior to vary across variables. Extensive experiments on real-world benchmark datasets show that CoLaDe achieves strong forecasting performance while maintaining competitive computational efficiency and stable results across random seeds. These results demonstrate the effectiveness of jointly modeling information allocation and temporal retrieval for multivariate time series forecasting.

AI use statement

We used generative AI tools to assist with manuscript polishing and grammar correction.

Ethics statement

This work evaluates a forecasting model on established benchmark datasets and does not involve human participants. We are not aware of specific ethical concerns arising from the methods or experiments presented in this paper.

Reproducibility statement

The model architecture and its components are described in the main text. The experimental setup, including datasets, evaluation protocol, and implementation details, is provided in the appendix. Additional results and computational measurements are also reported there to support reproduction and comparison.

References

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dp27P5HBBt.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=pCbC3aQB5W.

Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

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. URL https://openreview.net/forum?id=RCeZ063p33.

Jin Liu, M. H. Yahya, Saidatunur Fauzi Saidin, and Li Lu. A frequency-guided multi-scale decomposition and patch transformer for multivariate cryptocurrency time-series forecasting. *Scientific Reports*, 2026. doi: 10.1038/s41598-026-66760-2. URL https://doi.org/10.1038/s41598-026-66760-2.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Jiecheng Lu, Xu Han, Yan Sun, and Shihao Yang. CATS: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=1lDAGDe0UR.

Donghao Luo and Xue Wang. DeformableTST: Transformer for time series forecasting without over-reliance on patching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=B1Iq1EOiVU.

Aitian Ma, Dongsheng Luo, and Mo Sha. Mixlinear: Extreme low resource multivariate time series forecasting with $0.1k$ parameters. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=QUj0KuCumD.

Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. Mofo: Empowering long-term time series forecasting with periodic pattern modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=sbvLts2HqR.

Jiaming Ma, Qihe Huang, Haofeng Ma, Guanjun Wang, Sheng Huang, Zhengyang Zhou, Pengkun Wang, Xu Wang, Binwu Wang, and Yang Wang. PHAT: Modeling period heterogeneity for multivariate time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=lr4RlISR6x.

Shusen Ma, Yun-Bo Zhao, and Yu Kang. Thcvformer: Modeling temporal-dependencies, heterogeneity, and correlations for enhanced time-series forecasting. *Knowledge-Based Systems*, pp. 115964, 2026c.

Reza Nematirad, Anil Pahwa, and Balasubramaniam Natarajan. Times2d: Multi-period decomposition and derivative mapping for general time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19651–19658, 2025.

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

Yiming Niu, Jinliang Deng, and Yongxin Tong. Phaseformer: From patches to phases for efficient and effective time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Lk9SqMQzhX.

Abdul Moiz Qarni, Ihtisham Ul Haq, Abid Iqbal, and Siddig M Elkhider. Mcv-patchtst: multi-scale cross-variate patch transformers for household short-term load forecasting. *Scientific Reports*, 2026.

Binwu Wang, Gaoyun Lin, Jiaming Ma, Qihe Huang, Zhengyang Zhou, Xu Wang, Pengkun Wang, and Yang Wang. Crisp: A spectral-based interaction strategy for multivariate time series forecasting. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=avYLuILgcr.

Shiyu Wang, Jiawei LI, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Ju Shengtong, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1CLzLXSFNn.

Tian Wang and Chuang Wang. Latent neural operator for solving forward and inverse pde problems. In *Advances in Neural Information Processing Systems*, 2024.

Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. CARD: Channel aligned robust blend transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=MJksrOhurE.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=INAeUQ04lT.

Haixiang Wu. Revisiting attention for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21528–21535, 2025.

Xingjian Wu, Xiangfei Qiu, Hanyin Cheng, Zhengyu Li, Jilin Hu, Chenjuan Guo, and Bin Yang. Enhancing time series forecasting through selective representation spaces: A patch perspective. *Advances in Neural Information Processing Systems*, 38:23328–23354, 2026.

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.

APPENDIX

A Derivation of the Equivalent Form

This appendix derives Eq. 8 from Eq. 5 and Eq. 6, and examines the effect of the competition coefficient on the resulting attention weights.

Derivation. The relative preference of input token n for latent token r is defined in Eq. 5 as a log-softmax across the latent dimension. Expanding the logarithm gives

$$\pi_{h,r,n} = \log \exp(s_{h,r,n}) - \log \sum_{r'=1}^{R} \exp(s_{h,r',n}) = s_{h,r,n} - \Lambda_{h,n}, \qquad \Lambda_{h,n} = \log \sum_{r'=1}^{R} \exp(s_{h,r',n}). \tag{17}$$

Where $\Lambda_{h,n}$ denotes the log-sum-exp of the content scores of input token n over all latent tokens. Substituting Eq. 17 into the competitive logit of Eq. 6 yields

$$\widetilde{s}_{h,r,n} = s_{h,r,n} + \beta_h \left(s_{h,r,n} - \Lambda_{h,n}\right) = (1+\beta_h) s_{h,r,n} - \beta_h \Lambda_{h,n}. \tag{18}$$

This expression is equivalent to Eq. 8.

Effect on the Attention Weights. Substituting Eq. 18 into the token-wise softmax of Eq. 7 gives

$$a_{h,r,n} = \frac{\exp[(1+\beta_h) s_{h,r,n}]\, \mathcal{Z}_{h,n}^{-\beta_h}}{\sum_{n'=1}^{N} \exp[(1+\beta_h) s_{h,r,n'}]\, \mathcal{Z}_{h,n'}^{-\beta_h}}, \qquad \mathcal{Z}_{h,n} = \exp(\Lambda_{h,n}) = \sum_{r'=1}^{R} \exp(s_{h,r',n}). \tag{19}$$

Where $\mathcal{Z}_{h,n}$ represents the total affinity received by input token n across the latent set. Eq. 19 shows that the competition introduces the token-dependent factor $\mathcal{Z}_{h,n}^{-\beta_h}$ before the final normalization. When $\beta_h > 0$, this factor down-weights input tokens with larger total affinity across the latent set, which encourages a more differentiated allocation among the latent tokens. When $\beta_h < 0$, this factor has the opposite effect, which allows stronger overlap among the latent representations. When $\beta_h = 0$, Eq. 19 reduces exactly to standard cross-attention. Since the numerator is strictly positive for finite logits and the denominator is the sum of these positive terms over the input tokens, the resulting attention weights are positive and sum to one for any finite β_h.

Relation to the Learnable Temperature. The learnable temperature γ_h^{cmp} and the competition coefficient β_h affect the attention logits in different ways. The temperature applies a positive global scaling to the content scores of an attention head, whereas β_h introduces both the factor $(1+\beta_h)$ on the latent–token affinity and the token-dependent term $-\beta_h \Lambda_{h,n}$. For $\beta_h > -1$, the first effect resembles a positive rescaling of the content scores, but the second effect cannot be reproduced by a temperature because it depends on the total affinity of each input token across the latent set. Moreover, for $\beta_h \leq -1$, the factor $(1+\beta_h)$ cannot in general be represented by the strictly positive temperature parameter. Therefore, the temperature and the competition coefficient provide distinct and independently learned controls over the attention distribution.

B ALGORITHM

Algorithm 1 CoLaDe forward pass.

Require: Multivariate input $\mathbf{X} \in \mathbb{R}^{L\times C}$; observed patch indices $\mathcal{O}$; target patch indices $\mathcal{T} = \{1,\dots,B\} \setminus \mathcal{O}$; patch length ρ; embedding dimension d; number of latent tokens R; latent dimension d_u; number of latent rounds K; number of latent self-refinement blocks J; number of attention heads H; low-rank correction rank q.

1: Apply RevIN to $\mathbf{X}$ and store the per-variable statistics (μ_c, σ_c).
2: Unfold the normalized series into non-overlapping patches $\widetilde{\mathbf{x}}_{c,b} \in \mathbb{R}^{\rho}$ and embed $\mathbf{e}_{c,b} = \mathbf{W}_{\text{in}}\widetilde{\mathbf{x}}_{c,b} + \mathbf{p}_b + \mathbf{r}_c$; flatten over variables and observed patches to obtain $\mathbf{E} \in \mathbb{R}^{(CB_o)\times d}$.
// Compress input into latent tokens
3: Initialize the latent array $\mathbf{U} \leftarrow \mathbf{U}^{(0)} \in \mathbb{R}^{R\times d_u}$; when the input-conditioned option is enabled, add a zero-gated correction $\mathbf{U} \leftarrow \mathbf{U} + \alpha\,\mathrm{Lin}(\overline{\mathbf{E}})$ with a learnable scalar gate α initialized to zero.
4: **for** $k = 1$ to K **do**
5: $\quad\mathbf{U} \leftarrow \mathrm{Block}_{\beta}(\mathbf{U}, \mathbf{E})$ ▷ signed competitive cross-attention (Eq. equation 6)
6: $\quad$**for** $j = 1$ to J **do**
7: $\quad\quad\mathbf{U} \leftarrow \mathrm{Block}(\mathbf{U}, \mathbf{U})$ ▷ latent self-refinement
8: $\quad$**end for**
9: $\quad\mathbf{E} \leftarrow \mathrm{Block}(\mathbf{E}, \mathbf{U})$ ▷ write refined latents back to the observed tokens
10: **end for**
// Lag-aware decoding of target patches
11: Build target queries $\mathbf{g}_{c,b} = \mathbf{p}_b + \mathbf{r}_c$ for $b \in \mathcal{T}$ (or take them from a learned per-variable query bank $\mathbf{Q}^{(0)} \in \mathbb{R}^{C\times B\times d}$ when query sharing is disabled); reshape $\mathbf{E}$ so that each variable's tokens are grouped together.
12: Compute the relative-lag prior $\Phi_{c,h,\delta} = \Phi_{h,\delta} + \sum_{j=1}^{q} A_{c,j}\Psi_{\delta,j,h}$ for all c, h and every lag $\delta = b - b'$ with $b \in \mathcal{T}$, $b' \in \mathcal{O}$, reusing the hidden feature $\mathbf{m}(\delta)$ from Eq. equation 11.
13: $\mathbf{G} \leftarrow \mathrm{Block}(\mathbf{Q}, \mathbf{E}; \Phi)$ ▷ per-variable query–patch attention with the lag bias
// Reconstruct the forecast
14: $\widehat{\mathbf{x}}_{c,b} = \mathbf{W}_{\text{out}}\mathbf{g}_{c,b}$ for $b \in \mathcal{T}$, then concatenate along the temporal axis.
15: Restore the RevIN statistics to obtain $\widehat{\mathbf{Y}} \in \mathbb{R}^{P\times C}$.
16: **return** $\widehat{\mathbf{Y}}$ (and, during training, the auxiliary latent-diversity term $\mathcal{L}_{\text{div}} = \frac{1}{R^2}\sum_{r\neq r'}(\bar{\mathbf{u}}_r^{\top}\bar{\mathbf{u}}_{r'})^2$ computed over unit-normalized latents $\bar{\mathbf{u}}_r$).

Table 4: Settings held fixed across every dataset and horizon. The learning rate is the only optimization setting that varies between benchmarks: it is 1×10^{-4} on Weather and 5×10^{-4} everywhere else.

Setting	Value
Training objective	generalized (resampled observed/target patches), MSE
Optimizer	AdamW, weight decay 0.05
Learning-rate schedule	5-epoch linear warm-up, then cosine annealing
Early stopping	validation loss, patience 10
Lookback L / label length	96,384,768 / 0
Dropout	0.2

C IMPLEMENTATION DETAILS

Dataset Details. The evaluation is carried out on seven publicly available multivariate benchmarks that are commonly used in the long-term forecasting literature: Weather, Solar-Energy, Electricity, and the four ETT collections (ETTh1, ETTh2, ETTm1, and ETTm2) (Zhou et al., 2021). The chosen benchmarks differ substantially in scale, sampling rate, and application domain, which lets us examine CoLaDe under both low- and high-dimensional regimes. In every case each series is split chronologically into training, validation, and test partitions, so that no future information

Table 5: Detailed description of the forecasting benchmarks.

Dataset	Dim	Dataset Size	Series Length	Frequency	Information
ETTm1	7	(34,465, 11,521, 11,521)	{96, 192, 336, 720}	15 min	Temperature
ETTm2	7	(34,465, 11,521, 11,521)	{96, 192, 336, 720}	15 min	Temperature
ETTh1	7	(8,545, 2,881, 2,881)	{96, 192, 336, 720}	1 hour	Temperature
ETTh2	7	(8,545, 2,881, 2,881)	{96, 192, 336, 720}	1 hour	Temperature
Electricity	321	(18,317, 2,633, 5,261)	{96, 192, 336, 720}	1 hour	Electricity
Weather	21	(36,792, 5,271, 10,540)	{96, 192, 336, 720}	10 min	Weather
Solar-Energy	137	(36,601, 5,161, 10,417)	{96, 192, 336, 720}	10 min	Electricity

leaks into the fitting stage. A per-dataset summary is given in Table 5. All models observe an input window of length $L \in \{96, 384, 768\}$ and are asked to produce forecasts over the horizons $P \in \{96, 192, 336, 720\}$.

Baseline Details. CoLaDe is compared against nine competitive forecasters that together cover the main architectural families studied in recent work. The Transformer-based group includes PatchTST (Nie et al., 2023), which tokenizes each channel into patches; iTransformer (Liu et al., 2024), which attends across variates rather than time steps; CARD (Wang et al., 2024a), which aligns tokens across channels; CATS (Lu et al., 2024), which separates query construction from the input sequence; and DeformableTST (Luo & Wang, 2024), which learns data-dependent sparse attention. The periodicity- and structure-oriented group comprises Times2D (Nematirad et al., 2025), which reshapes the signal into a two-dimensional periodic representation. Finally, we include the lightweight model MixLinear (Ma et al., 2026a), which shows how much of the forecasting signal can be recovered with a small parameter budget, together with the recent Crisp (Wang et al., 2026). Wherever possible we reuse the officially released configurations of these models, and otherwise tune them under the same protocol as CoLaDe to ensure a fair comparison.

Metric Details. Forecasting quality is reported with two standard error measures, the mean squared error (MSE) and the mean absolute error (MAE). Given the ground-truth future values $\mathbf{Y} \in \mathbb{R}^{P \times C}$ and the corresponding predictions $\widehat{\mathbf{Y}}$, the two quantities are computed over all horizon steps and variates as

$$\mathrm{MSE} = \frac{1}{PC} \sum_{t=1}^{P} \sum_{c=1}^{C} \left(Y_{t,c} - \widehat{Y}_{t,c} \right)^2, \qquad \mathrm{MAE} = \frac{1}{PC} \sum_{t=1}^{P} \sum_{c=1}^{C} \left| Y_{t,c} - \widehat{Y}_{t,c} \right|. \tag{20}$$

Both metrics are evaluated on the standardized scale used during training, and a lower value indicates a more accurate forecast.

D Full Results

This section presents the complete forecasting results obtained with input lengths $L \in \{96, 384, 768\}$ across all datasets and prediction horizons $H \in \{96, 192, 336, 720\}$.

Table 6: Multivariate time series forecasting results under a fixed input length of $L = 768$ and prediction horizons $H \in \{96, 192, 336, 720\}$. Best results are shown in **red**, and second-best results are blue.

Models		CoLaDe (Ours)		MixLinear (2026)		Crisp (2026)		Times2D (2025)		DeformableTST (2024)		CARD (2024)		CATS (2024)		iTransformer (2024)		TiDE (2024)		PatchTST (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.361	0.394	**0.351**	**0.383**	0.417	0.441	0.364	0.399	0.365	0.402	0.379	0.411	0.371	0.405	0.403	0.430	0.438	0.452	0.381	0.412
	192	**0.395**	**0.419**	**0.395**	0.423	0.459	0.467	0.405	0.423	0.407	0.429	0.409	0.427	0.418	0.439	0.432	0.450	0.482	0.472	0.418	0.432
	336	0.414	0.435	0.411	**0.434**	0.512	0.498	0.423	0.451	**0.407**	0.436	0.428	0.444	0.455	0.466	0.473	0.482	0.506	0.491	0.439	0.453
	720	0.497	0.488	**0.423**	**0.456**	0.599	0.534	0.446	0.467	0.459	0.483	0.472	0.486	0.501	0.490	0.568	0.550	0.513	0.513	0.470	0.482
	Avg	0.417	0.434	**0.395**	**0.424**	0.497	0.485	0.410	0.435	0.410	0.438	0.422	0.442	0.436	0.450	0.469	0.478	0.485	0.482	0.427	0.445
ETTh2	96	**0.268**	**0.333**	0.283	0.340	0.308	0.366	0.277	0.339	0.269	0.335	0.280	0.343	0.276	0.338	0.309	0.361	0.353	0.422	0.274	0.337
	192	0.336	0.388	0.337	0.376	0.360	0.396	0.337	0.381	**0.327**	**0.374**	0.348	0.386	0.339	0.385	0.442	0.444	0.427	0.462	0.334	0.376
	336	**0.320**	**0.376**	0.356	0.395	0.399	0.424	0.329	0.385	0.322	0.379	0.368	0.402	0.368	0.406	0.453	0.457	0.451	0.489	0.356	0.402
	720	**0.377**	0.425	0.380	**0.423**	0.427	0.460	0.402	0.445	0.411	0.448	0.393	0.428	0.436	0.469	0.432	0.464	0.490	0.519	0.396	0.437
	Avg	**0.325**	**0.381**	0.339	0.384	0.374	0.412	0.336	0.388	0.332	0.384	0.347	0.390	0.355	0.400	0.409	0.432	0.430	0.473	0.340	0.388
ETTm1	96	0.289	**0.341**	0.333	0.375	0.321	0.357	0.295	0.358	0.293	0.346	0.295	0.342	**0.286**	0.342	0.314	0.372	0.316	0.349	0.296	0.350
	192	0.324	**0.363**	0.352	0.382	0.365	0.391	0.344	0.383	0.326	0.373	0.341	0.373	**0.319**	0.366	0.349	0.389	0.334	0.364	0.331	0.371
	336	**0.349**	**0.379**	0.391	0.407	0.409	0.414	0.352	0.381	0.358	0.392	0.367	0.387	0.354	0.382	0.387	0.410	0.363	0.386	0.363	0.393
	720	**0.392**	**0.406**	0.439	0.435	0.445	0.439	0.411	0.411	0.419	0.424	0.415	0.419	0.402	0.416	0.461	0.457	0.411	0.408	0.419	0.433
	Avg	**0.339**	**0.372**	0.379	0.400	0.385	0.400	0.351	0.383	0.349	0.384	0.355	0.380	0.340	0.377	0.378	0.407	0.356	0.377	0.352	0.387
ETTm2	96	**0.163**	**0.252**	0.180	0.271	0.170	0.259	0.165	0.258	0.167	0.259	0.164	0.255	0.166	0.263	0.184	0.277	0.174	0.283	0.181	0.270
	192	**0.222**	**0.295**	0.234	0.308	0.232	0.302	0.241	0.312	0.228	0.297	0.225	0.300	0.232	0.308	0.240	0.319	0.239	0.322	0.245	0.316
	336	0.280	0.339	0.284	0.339	0.290	0.341	0.291	0.351	0.282	0.334	**0.274**	**0.327**	0.284	0.341	0.299	0.353	0.300	0.363	0.305	0.353
	720	**0.336**	**0.376**	0.365	0.390	0.388	0.397	0.346	0.381	0.347	0.385	0.392	0.401	0.366	0.392	0.402	0.417	0.383	0.427	0.372	0.399
	Avg	**0.250**	**0.316**	0.266	0.327	0.270	0.325	0.261	0.326	0.256	0.319	0.264	0.321	0.262	0.326	0.281	0.342	0.274	0.349	0.276	0.335
Weather	96	0.142	**0.197**	0.179	0.237	0.158	0.208	0.145	0.200	0.144	0.198	0.150	0.209	**0.141**	0.198	0.163	0.216	0.166	0.214	0.149	0.203
	192	**0.188**	**0.238**	0.223	0.275	0.205	0.250	0.191	0.241	0.193	0.239	0.229	0.271	0.189	0.242	0.207	0.251	0.206	0.253	0.192	0.242
	336	**0.238**	0.284	0.267	0.307	0.256	0.290	0.243	0.282	0.242	0.282	0.261	0.299	0.239	**0.280**	0.257	0.294	0.251	0.284	0.242	**0.280**
	720	0.315	0.335	0.329	0.351	0.322	0.337	0.315	0.336	**0.308**	0.332	0.361	0.358	**0.308**	0.331	0.320	0.340	0.315	0.337	0.312	**0.330**
	Avg	0.221	0.264	0.250	0.293	0.235	0.271	0.224	0.265	0.222	**0.263**	0.250	0.284	**0.219**	**0.263**	0.237	0.275	0.235	0.272	0.224	0.264
Solar	96	**0.158**	**0.212**	0.212	0.260	0.169	**0.212**	0.177	0.240	0.167	0.236	0.175	0.244	0.161	0.221	0.176	0.241	0.187	0.267	0.193	0.275
	192	**0.183**	0.230	0.227	0.262	0.186	**0.225**	0.188	0.252	0.185	0.252	0.212	0.266	0.194	0.233	0.196	0.257	0.214	0.288	0.210	0.283
	336	**0.189**	**0.229**	0.240	0.271	0.196	0.232	0.195	0.255	0.193	0.261	0.221	0.275	0.190	0.249	0.206	0.274	0.231	0.311	0.223	0.292
	720	**0.197**	0.253	0.240	0.265	0.219	0.247	0.204	0.251	**0.197**	0.263	0.231	0.295	0.198	**0.246**	0.218	0.277	0.234	0.316	0.236	0.296
	Avg	**0.182**	0.231	0.230	0.265	0.193	**0.229**	0.191	0.250	0.186	0.253	0.210	0.270	0.186	0.237	0.199	0.262	0.217	0.296	0.216	0.287
Electricity	96	0.127	0.223	0.139	0.233	0.140	0.231	0.132	0.230	0.131	0.235	0.131	0.229	**0.125**	**0.221**	0.140	0.243	0.135	0.238	0.129	0.236
	192	0.148	**0.242**	0.155	0.251	0.159	0.248	0.155	0.255	**0.146**	0.249	0.158	0.257	0.147	**0.242**	0.161	0.256	0.151	0.253	0.152	0.247
	336	**0.165**	**0.261**	0.172	0.266	0.177	0.266	0.175	0.273	0.166	0.268	0.180	0.284	0.169	0.264	0.177	0.279	0.172	0.271	0.170	0.271
	720	0.205	0.297	0.216	0.305	**0.164**	**0.288**	0.198	0.293	0.199	0.294	0.205	0.302	0.195	**0.288**	0.223	0.311	0.212	0.302	0.204	0.296
	Avg	0.161	0.256	0.171	0.264	0.160	0.258	0.165	0.263	0.161	0.262	0.169	0.268	**0.159**	**0.254**	0.175	0.272	0.168	0.266	0.164	0.263
1st Count		20	19	4	5	1	4	0	0	5	2	1	1	7	7	0	0	0	0	0	2

Table 7: Multivariate time series forecasting results under a fixed input length of $L = 384$ and prediction horizons $H \in \{96, 192, 336, 720\}$. Best results are shown in **red**, and second-best results are blue.

Models		CoLaDe (Ours)		MixLinear (2026)		Crisp (2026)		Times2D (2025)		DeformableTST (2024)		CARD (2024)		CATS (2024)		iTransformer (2024)		TiDE (2024)		PatchTST (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.355**	**0.389**	0.398	0.412	0.404	0.413	0.362	0.395	0.369	0.396	0.377	0.402	0.381	0.405	0.400	0.420	0.447	0.448	0.390	0.410
	192	**0.391**	**0.410**	0.402	0.414	0.433	0.438	0.405	0.423	0.410	0.417	0.404	0.418	0.416	0.433	0.441	0.446	0.479	0.471	0.429	0.433
	336	0.392	**0.411**	0.484	0.449	0.516	0.496	0.423	0.433	**0.391**	0.414	0.428	0.435	0.435	0.440	0.472	0.466	0.491	0.481	0.458	0.452
	720	**0.436**	**0.453**	0.466	0.460	0.546	0.526	0.445	0.465	0.447	0.464	0.437	0.459	0.451	0.460	0.533	0.520	0.499	0.504	0.473	0.481
	Avg	**0.394**	**0.416**	0.437	0.434	0.475	0.468	0.410	0.429	0.404	0.423	0.412	0.429	0.421	0.435	0.461	0.463	0.480	0.478	0.438	0.444
ETTh2	96	**0.270**	**0.332**	0.294	0.345	0.326	0.368	0.271	0.334	0.272	0.334	0.275	0.336	0.283	0.346	0.303	0.346	0.368	0.427	0.284	0.341
	192	0.330	0.380	0.345	0.379	0.429	0.428	0.334	0.378	**0.325**	**0.369**	0.339	0.376	0.345	0.384	0.396	0.415	0.438	0.468	0.354	0.384
	336	**0.311**	0.379	0.361	0.396	0.408	0.422	0.329	0.385	0.319	**0.373**	0.360	0.397	0.372	0.406	0.427	0.438	0.465	0.495	0.378	0.408
	720	**0.376**	**0.422**	0.384	0.423	0.436	0.455	0.379	0.423	0.395	0.433	0.390	0.425	0.408	0.443	0.432	0.454	0.495	0.521	0.409	0.439
	Avg	**0.322**	0.379	0.346	0.386	0.400	0.418	0.327	0.379	0.328	**0.377**	0.341	0.384	0.352	0.395	0.390	0.417	0.434	0.474	0.356	0.394
ETTm1	96	**0.281**	**0.333**	0.322	0.363	0.297	0.340	0.295	0.343	0.292	0.344	0.295	0.346	0.284	0.339	0.321	0.368	0.306	0.341	0.305	0.354
	192	**0.318**	**0.356**	0.350	0.378	0.344	0.370	0.344	0.390	0.335	0.371	0.339	0.371	0.323	0.363	0.359	0.389	0.343	0.368	0.345	0.377
	336	**0.349**	**0.377**	0.385	0.399	0.383	0.392	0.364	0.394	0.365	0.392	0.373	0.392	0.356	0.384	0.399	0.412	0.369	0.384	0.377	0.400
	720	**0.398**	**0.409**	0.445	0.434	0.443	0.430	0.410	0.415	0.417	0.418	0.420	0.414	0.411	0.421	0.463	0.450	0.432	0.412	0.432	0.433
	Avg	**0.337**	**0.369**	0.376	0.393	0.367	0.383	0.353	0.385	0.352	0.381	0.357	0.381	0.344	0.377	0.386	0.405	0.356	0.373	0.365	0.391
ETTm2	96	**0.158**	**0.244**	0.177	0.267	0.163	0.252	0.166	0.259	0.170	0.257	0.165	0.256	0.172	0.260	0.181	0.271	0.182	0.288	0.174	0.262
	192	**0.216**	**0.286**	0.233	0.305	**0.216**	0.292	0.220	0.296	0.227	0.295	0.229	0.298	0.234	0.306	0.244	0.313	0.241	0.329	0.239	0.305
	336	**0.270**	**0.321**	0.285	0.337	0.273	0.330	0.272	0.330	0.278	0.327	0.274	0.328	0.278	0.335	0.301	0.349	0.297	0.361	0.296	0.343
	720	**0.360**	**0.381**	0.377	0.392	0.366	0.387	0.377	0.390	0.365	0.384	0.367	0.383	0.363	0.391	0.397	0.407	0.411	0.423	0.382	0.396
	Avg	**0.251**	**0.308**	0.268	0.325	0.254	0.315	0.259	0.319	0.260	0.316	0.259	0.316	0.262	0.323	0.281	0.335	0.283	0.348	0.273	0.327
Weather	96	**0.143**	**0.194**	0.180	0.235	0.152	0.200	0.146	0.197	0.149	0.198	0.155	0.209	0.147	0.199	0.166	0.214	0.165	0.227	0.158	0.207
	192	**0.190**	0.240	0.223	0.270	0.198	0.242	0.191	0.240	0.192	**0.238**	0.207	0.255	0.191	0.242	0.210	0.253	0.213	0.252	0.204	0.248
	336	**0.241**	0.282	0.271	0.304	0.256	0.287	0.243	0.282	0.244	**0.278**	0.263	0.294	0.244	0.280	0.263	0.293	0.264	0.296	0.255	0.286
	720	**0.313**	0.333	0.339	0.352	0.322	0.333	0.315	0.335	0.317	**0.331**	0.355	0.352	0.314	**0.331**	0.335	0.343	0.324	0.334	0.328	0.336
	Avg	**0.222**	0.263	0.253	0.290	0.232	0.265	0.224	0.263	0.226	**0.261**	0.245	0.277	0.224	0.263	0.244	0.276	0.240	0.272	0.236	0.269
Solar	96	**0.163**	**0.210**	0.247	0.274	0.209	0.228	0.201	0.241	0.173	0.241	0.175	0.243	0.172	0.216	0.188	0.242	0.222	0.293	0.207	0.283
	192	**0.177**	0.228	0.273	0.287	0.194	**0.224**	0.224	0.269	0.187	0.248	0.209	0.262	0.189	0.229	0.212	0.263	0.251	0.339	0.231	0.304
	336	**0.188**	0.235	0.284	0.298	0.202	**0.231**	0.248	0.272	0.193	0.260	0.209	0.272	0.197	0.236	0.226	0.275	0.264	0.349	0.249	0.312
	720	**0.199**	0.244	0.278	0.291	0.215	**0.238**	0.255	0.287	0.204	0.272	0.227	0.281	0.203	0.246	0.232	0.282	0.288	0.343	0.250	0.314
	Avg	**0.182**	**0.229**	0.271	0.288	0.205	0.230	0.232	0.267	0.189	0.255	0.205	0.265	0.190	0.232	0.215	0.266	0.257	0.334	0.234	0.303
Electricity	96	**0.124**	**0.218**	0.145	0.239	0.133	0.228	0.126	0.222	0.132	0.231	0.133	0.233	0.128	0.224	0.143	0.240	0.153	0.257	0.148	0.246
	192	**0.143**	**0.237**	0.161	0.261	0.157	0.253	0.145	0.240	0.150	0.250	0.157	0.256	0.147	**0.237**	0.160	0.256	0.166	0.269	0.163	0.256
	336	**0.160**	**0.254**	0.178	0.281	0.178	0.274	0.161	0.255	0.167	0.267	0.169	0.268	0.162	0.256	0.177	0.274	0.178	0.273	0.180	0.276
	720	0.202	0.290	0.222	0.319	**0.193**	**0.285**	0.200	0.298	0.203	0.299	0.200	0.292	0.198	0.286	0.220	0.313	0.207	0.299	0.218	0.306
	Avg	0.161	0.254	0.177	0.275	0.165	0.260	**0.158**	0.254	0.163	0.262	0.165	0.262	0.159	**0.253**	0.175	0.271	0.176	0.275	0.177	0.271
1st Count		31	23	0	0	2	4	1	0	2	7	0	0	0	3	0	0	0	0	0	0

Table 8: Multivariate time series forecasting results under a fixed input length of $L = 96$ and prediction horizons $H \in \{96, 192, 336, 720\}$. Best results are shown in **red**, and second-best results are blue.

Models		CoLaDe (Ours)		MixLinear (2026)		Crisp (2026)		Times2D (2025)		DeformableTST (2024)		CARD (2024)		CATS (2024)		iTransformer (2024)		TiDE (2024)		PatchTST (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.368**	**0.394**	0.399	0.404	0.382	**0.394**	0.376	0.395	0.373	0.396	0.398	0.405	0.371	0.395	0.386	0.405	0.476	0.456	0.414	0.419
	192	**0.417**	**0.421**	0.456	0.440	0.428	0.423	0.435	0.423	0.427	0.427	0.448	0.433	0.426	0.422	0.441	0.436	0.519	0.487	0.460	0.445
	336	0.440	0.429	0.484	0.449	0.474	0.447	0.457	0.439	**0.437**	**0.426**	0.494	0.457	**0.437**	0.432	0.487	0.458	0.558	0.508	0.460	0.466
	720	**0.425**	**0.442**	0.466	0.460	0.479	0.469	0.465	0.464	0.464	0.462	0.483	0.472	0.474	0.461	0.503	0.491	0.587	0.554	0.500	0.488
	Avg	**0.413**	**0.422**	0.451	0.438	0.441	0.433	0.433	0.430	0.425	0.428	0.456	0.442	0.427	0.428	0.454	0.447	0.537	0.505	0.469	0.454
ETTh2	96	0.289	0.339	0.305	0.347	0.288	0.336	0.290	0.338	**0.281**	**0.334**	0.294	0.341	0.287	0.341	0.297	0.349	0.396	0.435	0.302	0.348
	192	0.365	0.385	0.385	0.396	0.363	0.385	0.379	0.393	**0.353**	**0.382**	0.375	0.391	0.361	0.388	0.380	0.400	0.527	0.498	0.388	0.400
	336	0.370	0.398	0.425	0.429	0.406	0.421	0.377	0.407	**0.341**	**0.379**	0.419	0.426	0.374	0.403	0.428	0.432	0.635	0.567	0.426	0.433
	720	**0.395**	**0.423**	0.420	0.438	0.419	0.436	0.412	0.433	0.410	0.431	0.420	0.438	0.412	0.433	0.427	0.445	0.863	0.670	0.431	0.446
	Avg	0.355	0.386	0.384	0.402	0.369	0.395	0.365	0.393	**0.346**	**0.382**	0.378	0.399	0.359	0.391	0.383	0.407	0.607	0.548	0.387	0.407
ETTm1	96	0.320	0.358	0.353	0.373	**0.311**	**0.342**	0.330	0.364	0.316	0.358	0.327	0.359	0.318	0.347	0.334	0.368	0.362	0.378	0.329	0.367
	192	**0.349**	0.371	0.399	0.397	0.357	**0.368**	0.361	0.382	0.354	0.380	0.372	0.381	0.357	0.377	0.377	0.391	0.389	0.402	0.367	0.385
	336	**0.375**	**0.392**	0.436	0.423	0.392	0.393	0.407	0.417	0.379	0.405	0.403	0.401	0.387	0.410	0.426	0.420	0.421	0.423	0.399	0.410
	720	**0.433**	**0.427**	0.496	0.456	0.454	0.432	0.472	0.450	0.443	0.438	0.467	0.438	0.448	0.437	0.491	0.459	0.477	0.460	0.484	0.439
	Avg	**0.369**	0.387	0.421	0.412	0.379	**0.384**	0.393	0.403	0.373	0.395	0.392	0.395	0.378	0.393	0.407	0.410	0.416	0.413	0.387	0.400
ETTm2	96	0.170	0.252	0.189	0.273	**0.168**	**0.249**	0.174	0.262	0.178	0.262	0.176	0.259	0.178	0.261	0.180	0.264	0.203	0.301	0.175	0.259
	192	0.236	**0.293**	0.254	0.311	**0.235**	**0.293**	0.241	0.303	0.243	0.301	0.246	0.306	0.248	0.308	0.250	0.309	0.286	0.353	0.241	0.302
	336	**0.294**	**0.333**	0.316	0.351	0.297	0.335	0.301	0.342	0.310	0.348	0.302	0.343	0.304	0.343	0.311	0.348	0.373	0.418	0.305	0.343
	720	0.395	0.391	0.416	0.406	**0.391**	**0.390**	0.397	0.396	0.400	0.398	0.400	0.404	0.402	0.402	0.412	0.407	0.548	0.513	0.402	0.400
	Avg	0.274	**0.317**	0.294	0.335	**0.273**	**0.317**	0.278	0.326	0.283	0.327	0.281	0.328	0.283	0.329	0.288	0.332	0.355	0.398	0.281	0.326
Weather	96	0.151	0.201	0.200	0.243	**0.146**	**0.187**	0.181	0.221	0.169	0.213	0.160	0.208	0.161	0.207	0.174	0.214	0.191	0.250	0.177	0.218
	192	0.203	0.277	0.245	0.277	**0.197**	**0.236**	0.228	0.260	0.216	0.255	0.207	0.250	0.208	0.250	0.221	0.254	0.236	0.297	0.225	0.259
	336	**0.259**	0.288	0.297	0.313	**0.259**	**0.284**	0.280	0.297	0.271	0.294	0.265	0.292	0.264	0.290	0.278	0.296	0.283	0.323	0.278	0.297
	720	**0.337**	0.339	0.370	0.361	0.338	**0.336**	0.355	0.345	0.347	0.344	0.346	0.346	0.342	0.341	0.358	0.347	0.347	0.378	0.354	0.348
	Avg	0.238	0.276	0.278	0.298	**0.235**	**0.261**	0.261	0.281	0.251	0.276	0.245	0.274	0.244	0.272	0.258	0.278	0.269	0.315	0.259	0.281
Solar	96	**0.189**	**0.211**	0.513	0.446	0.190	0.213	0.236	0.283	0.193	0.232	0.230	0.281	0.216	0.247	0.203	0.237	0.306	0.395	0.234	0.286
	192	**0.224**	**0.231**	0.542	0.457	0.229	0.237	0.278	0.310	0.225	0.261	0.267	0.308	0.247	0.267	0.233	0.261	0.330	0.410	0.267	0.310
	336	0.238	0.264	0.554	0.458	**0.203**	**0.232**	0.291	0.321	0.239	0.267	0.289	0.319	0.267	0.277	0.248	0.273	0.366	0.424	0.290	0.315
	720	0.236	**0.224**	0.526	0.431	**0.216**	0.239	0.290	0.317	0.238	0.265	0.294	0.327	0.296	0.293	0.249	0.275	0.365	0.422	0.289	0.317
	Avg	0.222	0.233	0.534	0.448	**0.209**	**0.230**	0.274	0.308	0.224	0.256	0.270	0.309	0.257	0.271	0.233	0.262	0.336	0.413	0.270	0.307
Electricity	96	**0.134**	**0.228**	0.142	0.231	0.143	0.235	0.175	0.269	0.158	0.255	0.155	0.246	0.149	0.237	0.148	0.240	0.231	0.323	0.181	0.270
	192	**0.151**	**0.244**	0.161	0.266	0.162	0.252	0.184	0.273	0.168	0.265	0.170	0.259	0.163	0.250	0.162	0.253	0.232	0.328	0.188	0.274
	336	**0.169**	**0.263**	0.172	0.269	0.177	0.267	0.199	0.288	0.183	0.280	0.194	0.285	0.180	0.268	0.178	0.269	0.243	0.337	0.204	0.293
	720	0.208	0.298	**0.201**	**0.288**	0.204	0.290	0.235	0.319	0.223	0.313	0.229	0.316	0.219	0.302	0.225	0.317	0.282	0.367	0.246	0.324
	Avg	**0.165**	**0.258**	0.169	0.264	0.171	0.261	0.198	0.287	0.183	0.278	0.187	0.277	0.178	0.264	0.178	0.270	0.246	0.334	0.205	0.290
1st Count		18	17	1	1	12	15	0	0	5	5	0	0	1	0	0	0	0	0	0	0

E COMPUTATIONAL EFFICIENCY AND SCALABILITY

This appendix reports the computational cost of CoLaDe across benchmark datasets and forecasting horizons. The measurements in Table 9 use an input length of $L = 384$ and a single NVIDIA A100-SXM4-40GB. Memory is the peak GPU memory allocated during training; timings are averaged over epochs after the warm-up epoch.

Table 9: Computational cost of CoLaDe. All runs use $L = 384$ on a single A100-40GB, and memory is the peak allocated GPU memory. Hyperparameters are tuned per dataset and per horizon, so the model size varies across rows. The batch size is reduced on Solar and Electricity to fit the available memory, and the memory and latency columns should therefore be read together with the batch size.

Dataset	C	P	Params (M)	Peak allocated (MB)	Batch	Time/epoch (s)	Latency (ms)
ETTh1	7	96	1.1535	831.76	128	2.13	7.24
		192	1.6085	850.22	128	2.28	8.65
		336	3.2639	1401.93	128	2.76	9.31
		720	4.3189	1612.70	128	2.97	11.89
ETTh2	7	96	1.6086	479.62	128	1.81	5.98
		192	1.1544	466.54	128	1.85	6.25
		336	4.0196	1084.70	128	2.78	10.97
		720	6.3862	896.25	128	2.55	10.57
ETTm1	7	96	4.0139	727.18	128	9.42	9.15
		192	4.0159	836.96	128	10.20	10.01
		336	4.0190	1033.96	128	11.83	10.72
		720	4.0280	1594.15	128	15.93	14.69
ETTm2	7	96	4.0143	760.94	128	9.63	9.07
		192	4.5522	864.51	128	10.42	10.10
		336	4.5554	1058.95	128	11.95	11.16
		720	6.3758	1728.12	128	16.67	14.09
Weather	21	96	4.5604	2365.03	128	24.16	19.90
		192	4.5625	2742.06	128	26.78	21.99
		336	4.0262	2870.04	128	29.72	24.24
		720	4.5710	4270.78	128	40.89	32.89
Solar	137	96	4.6171	5315.25	32	174.51	44.14
		192	4.6202	12144.93	64	194.00	97.05
		336	4.6250	14672.01	64	230.14	112.94
		720	4.6377	10800.23	32	323.41	80.00
Electricity	321	96	4.7146	16155.82	64	138.69	132.16
		192	4.7167	18746.23	64	153.83	150.29
		336	4.7198	11407.54	32	195.02	89.69
		720	4.7283	16575.07	32	261.07	123.23

Model Size and Forecasting Horizon. CoLaDe's query-based decoder predicts future patches in parallel without a direct output projection sized to the forecasting horizon. With the main architectural settings fixed, parameter count therefore depends only weakly on the horizon. From $H = 96$ to $H = 720$, it increases by 0.35% on ETTm1, 0.45% on Solar, and 0.29% on Electricity. Direct projection heads, such as those in PatchTST and iTransformer, instead introduce horizon-dependent output parameters. The larger variations in Table 9 reflect per-horizon hyperparameter selection.

Model Size and Number of Variates. The variate-adaptive correction of the lag prior includes parameters that scale linearly with the number of variates C. At $H = 96$, the parameter count is 4.5604 M for Weather ($C = 21$), 4.6171 M for Solar ($C = 137$), and 4.7146 M for Electricity ($C = 321$). These measurements are consistent with modest growth as the number of variates increases by more than an order of magnitude.

Memory and Runtime. Computational requirements generally increase with problem dimensionality and forecasting horizon, while peak memory also depends on batch size. Across the four ETT datasets, peak allocated GPU memory stays below 1.8 GB at every horizon with a batch size of 128. For Electricity ($C = 321$), batch sizes of 32–64 are used and peak allocated memory reaches 18.7 GB.

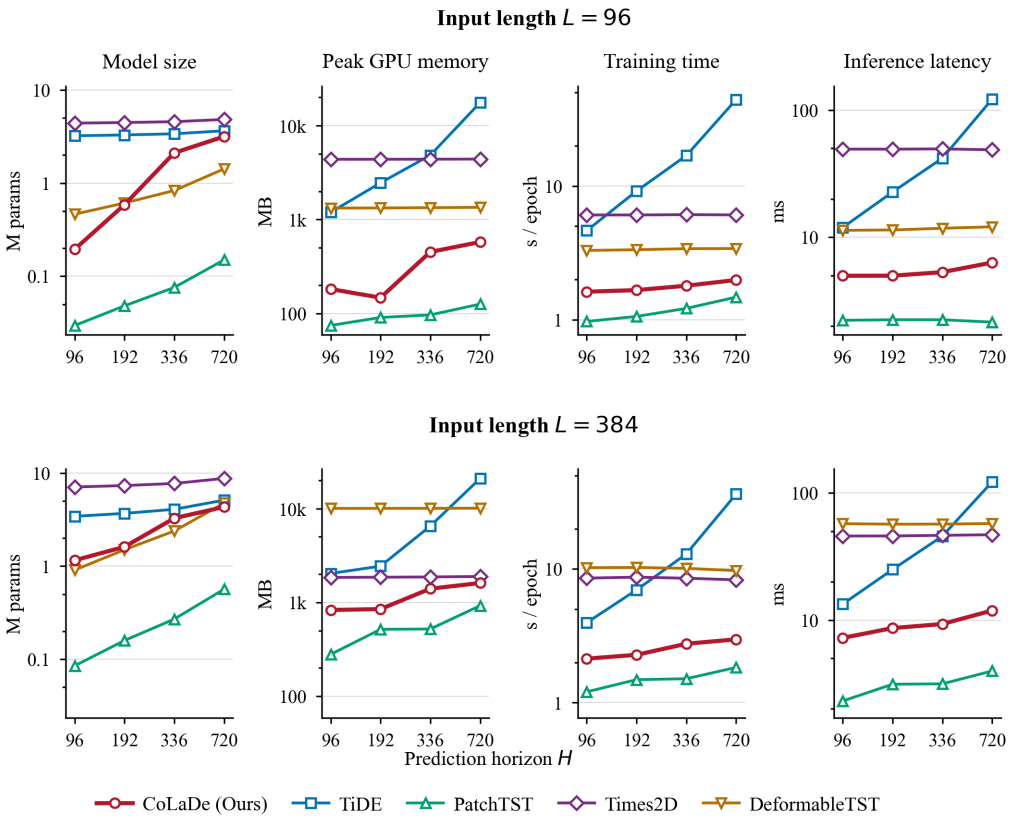

Figure 2: Computational cost on ETTh1 versus prediction horizon. The top and bottom rows use input lengths of $L = 96$ and $L = 384$, respectively. All runs use a batch size of 128 on a single A100-40GB. Memory denotes peak allocated GPU memory, training time excludes the warm-up epoch, and all vertical axes are logarithmic.

Comparison with Representative Baselines

We compare CoLaDe with TiDE (Das et al., 2023), PatchTST (Nie et al., 2023), Times2D (Nematirad et al., 2025), and DeformableTST (Luo & Wang, 2024) on ETTh1 at input lengths $L \in \{96, 384\}$.

At $L = 384$, CoLaDe's memory, training time, and latency rise by factors of 1.9, 1.4, and 1.6 from $H = 96$ to $H = 720$, compared with 10.3, 9.2, and 9.2 for TiDE. At $H = 720$, TiDE requires 20.9 GB and 36.4 seconds per epoch, versus 1.6 GB and 3.0 seconds for CoLaDe. PatchTST has the lowest measured cost, while Times2D and DeformableTST show nearly constant memory and runtime across horizons at higher levels. Thus, CoLaDe's cost grows more slowly than TiDE's. Its varying parameter counts in Figure 2 also reflect horizon-specific hyperparameter selection.

F Statistical Analysis and Robustness

To assess the stability of CoLaDe across random initializations, each configuration is trained repeatedly with different random seeds. From these runs we retain the five seeds that are shared across all reported settings and summarize the resulting forecasts as the mean and standard deviation of the MSE and MAE over those five seeds. Using a fixed set of common seeds keeps the error bars comparable from one dataset and horizon to the next. Table 10 reports these statistics for the input length $L = 384$.

Table 10: Standard deviation of the performance of CoLaDe under a fixed input length of $L = 384$ and prediction lengths $H \in \{96, 192, 336, 720\}$ over five common random seeds. Each entry is written as mean $\pm$ std.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.363±0.005	0.392±0.003	0.278±0.002	0.341±0.004	0.295±0.015	0.346±0.013	0.166±0.006	0.253±0.008
192	0.397±0.005	0.413±0.004	0.336±0.002	0.387±0.004	0.326±0.012	0.361±0.007	0.226±0.004	0.297±0.004
336	0.403±0.008	0.416±0.005	0.316±0.003	0.384±0.002	0.351±0.001	0.376±0.001	0.285±0.008	0.337±0.006
720	0.441±0.003	0.453±0.004	0.385±0.009	0.433±0.008	0.402±0.003	0.410±0.002	0.369±0.005	0.388±0.005

Dataset	Weather		Solar		Electricity	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
96	0.148±0.003	0.203±0.004	0.161±0.001	0.208±0.005	0.122±0.002	0.219±0.004
192	0.194±0.002	0.249±0.003	0.176±0.004	0.226±0.006	0.137±0.005	0.234±0.007
336	0.243±0.000	0.285±0.002	0.186±0.003	0.237±0.003	0.161±0.004	0.258±0.008
720	0.318±0.001	0.339±0.002	0.197±0.005	0.242±0.002	0.199±0.003	0.288±0.001