LEARNABLE DIFFERENCE-AWARE EMBEDDING: A LIGHTWEIGHT PLUG-AND-PLAY MODULE FOR MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series forecasting supports decision-making in energy, transportation, and other dynamic systems, where models must capture both global dependencies and local temporal changes. However, standard embeddings in representative forecasting backbones do not explicitly encode first- and second-order temporal differences at the initial representation stage. We introduce **Learnable Difference-Aware Embedding (LDAE)**, a lightweight plug-and-play module that constructs channel-wise rate-of-change and curvature features using learnable convolutions initialized with finite-difference stencils. LDAE fuses these derivative cues with the backbone representation at the initial embedding interface, while leaving the backbone's internal temporal modeling unchanged. Across 12 real-world datasets, LDAE improves forecasting accuracy in low-to-moderate-noise, non-stationary settings and transfers across Transformer-, CNN-, and MLP-based backbones. These results demonstrate that explicit derivative cues provide a lightweight and transferable inductive bias for multivariate time series forecasting.

1 INTRODUCTION

Multivariate time series forecasting (MTSF) is essential to real-world decision-making: power grids rely on it to anticipate load peaks, traffic systems use it to mitigate congestion, and data centers depend on it to prevent server overloads (Wu et al., 2021; Zhou et al., 2021). Accurate forecasting requires models to capture not only long-range dependencies and cross-variable interactions, but also non-stationary local dynamics such as channel-specific spikes, asymmetric rate changes, and sharp curvature variations.

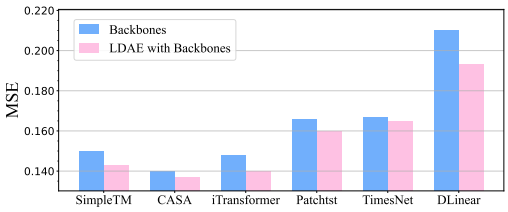


Figure 1: Representative ECL forecasting results for the original backbones and their LDAE-augmented variants. Lower MSE is better.

Recent forecasting backbones have made substantial progress in global dependency modeling, while auxiliary modules have further improved temporal decomposition, frequency modeling, and multi-scale aggregation (Nematirad et al., 2025; Hu et al., 2025; Liu et al., 2025). For the representative backbones considered in this work, first- and second-order temporal differences are not explicit in the standard initial embeddings. Instead, the corresponding rate-of-change and curvature information is typically inferred from raw observations through subsequent transformations (Oreshkin et al., 2019; Challu et al., 2023; Lee et al., 2025; Wu et al., 2023).

This observation motivates us to examine three aspects of existing MTSF pipelines. First, current embedding designs primarily differ in how they handle cross-variable interactions. Models such as xLSTM-Mixer, TimeMixer, and SOFTS introduce interactions among variables to capture shared multivariate structure (Kraus et al., 2024; Wang et al., 2024; Han et al., 2024). PatchTST processes each variable independently using shared embedding and Transformer parameters, whereas iTransformer preserves variable-level tokenization but applies self-attention across variates to model their correlations (Liu et al., 2024; Nie, 2023). Despite these architectural differences, their standard embeddings do not explicitly construct finite-difference features from the input. Second, existing local operators provide flexible mechanisms for extracting temporal patterns. For example, the

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.

multi-period representation and Inception-style convolutions in TimesNet capture complex variations at multiple scales (Wu et al., 2023). Nevertheless, such general-purpose operators are not specifically parameterized as derivative estimators initialized from finite-difference stencils. Third, although deeper attention, convolutional, or patch-based transformations can implicitly learn temporal changes, explicit derivative cues are not directly available to the backbone at its initial representation stage. These observations suggest that derivative-aware embedding can serve as a complementary inductive bias without replacing existing temporal or cross-variable modeling mechanisms. Moreover, because higher-order finite differences may amplify measurement noise, we use first- and second-order differences by default and examine the effect of derivative order in Appendix B.

To investigate this direction, we propose **Learnable Difference-Aware Embedding (LDAE)**, a lightweight, plug-and-play module that extracts and injects channel-wise derivative cues at the embedding stage without modifying the backbone architecture. Appendix C further analyzes embedding-stage injection under the stated assumptions of linear finite-impulse-response filters and band-limited signals.

Figure 1 compares representative forecasting backbones with their LDAE-augmented variants on the ECL dataset. Under the same experimental settings, incorporating LDAE reduces the forecasting MSE across all evaluated backbone families.

Our contributions are threefold:

- We investigate an embedding-stage injection hypothesis for MTSF: for backbones that mix channels, adding learnable derivative cues before that mixing may preserve channel-local temporal information; for channel-independent backbones, the same operation supplies an explicit local temporal prior. We evaluate this choice with four-horizon averages in Table 4 and with complete horizon-specific comparisons in Appendix D.3.
- We instantiate this principle as LDAE, a lightweight, channel-wise, data-adaptive module that models first- and second-order temporal differences through learnable kernels initialized from finite-difference stencils.
- We conduct experiments on 12 real-world datasets across five domains, characterizing improvements in low-to-moderate-noise, non-stationary settings across Transformer-, CNN-, and MLP-based models, as well as settings in which the prior does not help.

2 Related Work

Deep MTSF methods differ in how their embedding architectures represent individual variables and model cross-variable dependencies. Some methods introduce early interactions among variables to capture shared multivariate structure (Zhang et al., 2026; Wang et al., 2024; Han et al., 2024), whereas channel-independent models such as xPatch and PatchTST process each variable separately using shared model parameters (Stitsyuk & Choi, 2025; Nie, 2023). iTransformer follows a distinct design: it represents each variable as a variate token and applies self-attention across these tokens to model multivariate correlations (Liu et al., 2024). It therefore preserves variable-level tokenization but is not channel-independent in the same sense as PatchTST. Across these representative architectures, the initial embedding generally projects raw observations or temporal patches into a latent space without explicitly constructing first- or second-order finite-difference features. Any derivative-related representation is consequently learned through subsequent transformations rather than introduced as an explicit derivative prior. LDAE complements these embedding paradigms by making channel-wise derivative cues directly available to the backbone at the initial representation stage.

Another line of research improves local temporal modeling through signal priors and flexible temporal operators. Signal-prior-based methods exploit temporal decomposition, frequency-domain transformations, or periodic structure (Wu et al., 2021; Zhou et al., 2022; Murad et al., 2025). In particular, TimesNet transforms one-dimensional series into period-aware two-dimensional representations and uses multi-scale Inception blocks, providing a flexible mechanism for modeling intricate temporal variations (Wu et al., 2023). Other methods extract local patterns through patch-based representations, convolutions, or deep residual pathways (Nie, 2023; Luo & Wang, 2024b), while adaptive approaches employ learnable local filters or deformable temporal operators (Cheng et al., 2025; Luo & Wang, 2024a). These methods demonstrate that general-purpose and adaptive op-

erators can capture diverse local patterns. A simpler alternative is to concatenate fixed first- and second-order differences with the input. LDAE instead places the derivative cues at the embedding interface and adapts the initialized filters, normalization/dropout, and learned scales end-to-end; this distinction does not assume that learnable derivative features are universally superior to fixed differencing. LDAE is not distinguished by greater general modeling flexibility; its distinction is explicit derivative semantics: the kernels are initialized from finite-difference stencils and then adapted through end-to-end learning. Relative to the methods compared here, we characterize the contribution as the scoped combination of finite-difference semantics, learnable kernel adaptation, and early injection. Two families come closest but differ in how derivative information enters the model: N-BEATS (Oreshkin et al., 2019) learns trend and seasonality through hierarchical basis expansions without explicit per-channel finite-difference features, whereas Times2D (Nematirad et al., 2025) uses a handcrafted fixed first- and second-order map (FSDH) rather than learnable embedding-stage cues. The theoretical discussion is in Appendix C; the controlled empirical comparison is in Appendix D.1, where replacing only the FSDH branch with LDAE lowers the error on 11 of 12 dataset-horizon pairs.

3 METHOD

3.1 PROBLEM DEFINITION

In MTSF, the goal is to predict future values of multiple channels from their historical observations. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]^\top \in \mathbb{R}^{L \times C}$, where $L \gg 1$ is the lookback window and $C \geq 1$ is the number of channels. We forecast $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_H]^\top \in \mathbb{R}^{H \times C}$, where H is the prediction horizon and each $\mathbf{y}_t \in \mathbb{R}^C$ contains all channel values at future time t. The central challenge is to capture multi-scale dynamics, including abrupt rate changes, curvature inflections, and cross-channel dependencies. As L and H increase, these local cues may be underrepresented in the initial embedding.

3.2 OVERVIEW OF LDAE

LDAE is a lightweight, plug-and-play embedding module for MTSF. As illustrated in Figure 2, it augments the base embedding with local dynamic cues before backbone processing. The module contains three components: (i) a first-order operator that extracts per-channel *rate-of-change* information, (ii) a second-order operator that captures *acceleration/deceleration* and curvature, and (iii) an **EDNBlock** that projects, regularizes, and adaptively scales these features (α_i) before additive fusion with the base embedding.

Three design choices realize this embedding-stage injection hypothesis. First, finite-difference initialization supplies an interpretable derivative prior, which our explicit rate analysis shows already approximates the ideal derivative frequency response within $O(\Omega^3)$ and $O(\Omega^4)$ (Appendix C, Lemma 1). Second, adaptive fusion through learnable scales α_i lets the module down-weight unreliable derivative cues. Third, channel-wise convolution keeps the derivative branch channel-local; replacing it with channel-mixing convolution generally degrades performance, with the DLinear-on-Solar exception reported in Appendix D. Finally, for backbones with cross-channel mixing, LDAE injects the derivative cues before that mixing; for channel-independent backbones, the same interface provides an explicit local temporal prior. These choices make local dynamics available early without requiring deeper layers to reconstruct them implicitly.

3.2.1 ADAPTIVE FIRST- AND SECOND-ORDER OPERATORS VIA CHANNEL-WISE 1D CONVOLUTION

LDAE implements two lightweight derivative operators with grouped 1D convolutions, keeping the derivative branch channel-local. Each channel sequence $\mathbf{x}^{(c)} \in \mathbb{R}^L$ (or $\mathbf{X}_{:,c}$) is processed independently, preventing cross-channel mixing before the backbone embedding. The operators are defined as follows:

Finite-Difference Kernel (Initialized By Central-Difference Stencil) To model the first-order rate of change, we use a learnable finite-difference kernel initialized from the classical central-difference stencil. This initialization can adapt to asymmetric increases and decreases during train-

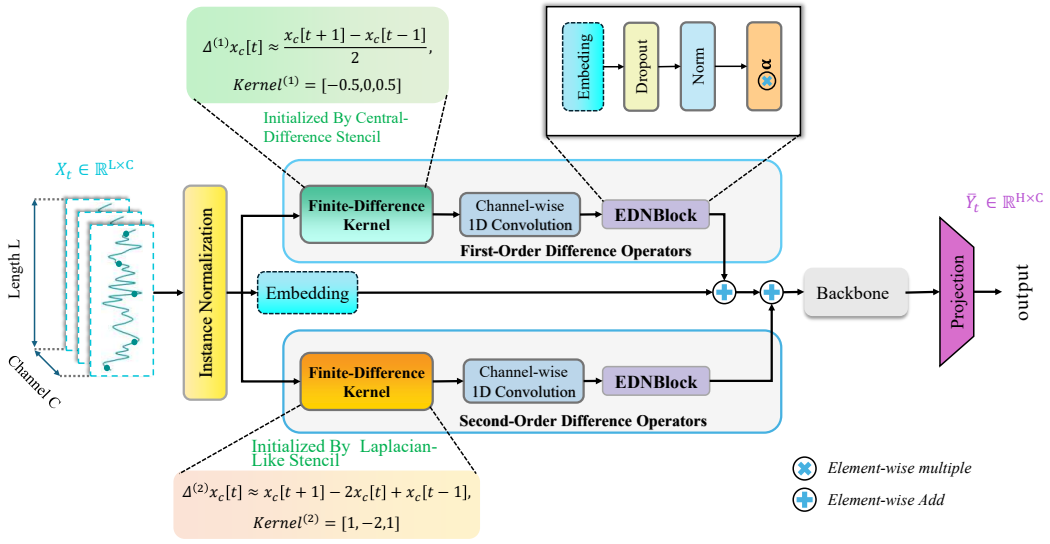


Figure 2: Overall structure of Learnable Difference-Aware Embedding (LDAE). The framework includes two main components: (1) two finite-difference kernels initialized with central-difference and Laplacian-like stencils, respectively, which compute first-order (rate-of-change) and second-order (curvature) derivative features via channel-wise 1D convolution; and (2) EDNBlock, which refines these derivative features and fuses them with the base embedding through additive fusion. LDAE enhances temporal dynamic modeling in MTSF and is plug-and-play with diverse backbones.

ing. The initial kernel is:

$$\mathbf{k}_{\text{init}}^{(1)} = [-0.5,\ 0,\ 0.5], \tag{1}$$

whose discrete approximation of the first-order derivative for sequence $\mathbf{x}^{(c)}$ (per-channel time series) is:

$$\Delta^{(1)}\mathbf{x}^{(c)}[t] \approx \frac{\mathbf{x}^{(c)}[t+1]-\mathbf{x}^{(c)}[t-1]}{2}. \tag{2}$$

This kernel explicitly captures per-channel temporal rate-of-change (velocity-like dynamics).

Finite-Difference Kernel (Initialized By Laplacian-Like Stencil) To model the second-order curvature (acceleration or inflection behavior), we use a finite-difference kernel initialized from a Laplacian-like stencil, also designed to be learnable. The initial kernel is:

$$\mathbf{k}_{\text{init}}^{(2)} = [1,\ -2,\ 1], \tag{3}$$

with its discrete form approximating the second-order derivative as:

$$\Delta^{(2)}\mathbf{x}^{(c)}[t] \approx \mathbf{x}^{(c)}[t+1] - 2\mathbf{x}^{(c)}[t] + \mathbf{x}^{(c)}[t-1]. \tag{4}$$

This stencil effectively detects trend inflections and acceleration/deceleration patterns, providing complementary dynamic information to the first-order operator.

The responses of the two kernels in eq. (1) and eq. (3) are instantiated as the learnable weights

$$\mathbf{W}_{c,:,:}^{(i)} = \mathbf{k}_{\text{init}}^{(i)}, \quad i \in \{1,2\}, \quad c = 1,\dots,C, \tag{5}$$

and the convolutions in eq. (5) directly output $\mathbf{D}^{(i)}$ with these $\mathbf{W}^{(i)}$ as the grouped filters. In the implementation, the first-order branch uses $K_1 =$ diff_kernel_size and the second-order branch uses a fixed $K_2 = 3$; the default is $K_1 = 3$. For $K_1 = 3$, the first-order stencil is exactly $[-0.5, 0, 0.5]$. For larger odd K_1, the implementation places the same two nonzero coefficients at offsets -1 and $+1$, with zeros elsewhere. For computation compatibility with the inverted embedding, $\mathbf{X} \in \mathbb{R}^{L\times C}$ is transposed to $\mathbb{R}^{C\times L}$ before convolution and remains in this channel-first layout for the value projection. The final derivative features are computed as:

$$\mathbf{D}^{(i)} = \text{Conv1D}_{\text{group}}(\mathbf{X}^{\top}; \mathbf{W}^{(i)}), \quad i \in \{1,2\}, \tag{6}$$

where $\mathbf{D}^{(1)} \in \mathbb{R}^{C\times L}$ and $\mathbf{D}^{(2)} \in \mathbb{R}^{C\times L}$ respectively represent the first-order and second-order derivative features before value projection. See Appendix C for the theoretical justification of learnable FIR kernels, approximation guarantees, and practical limitations.

Channel-wise 1D Convolution All derivative operators are implemented via grouped 1D convolution ($\text{groups} = C$). This keeps the derivative branch channel-local, while cross-channel interactions are handled by the downstream backbone. Symmetric padding (padding $= K_i//2$) is used for each odd kernel length K_i, preserving the temporal length L: $L + 2 \cdot (K_i//2) - K_i + 1 = L$. The first-order branch uses the configurable odd length $K_1 \in \{3, 5\}$ in the reported settings, whereas the second-order branch uses $K_2 = 3$. Even kernel lengths are not used. Both grouped convolutions use bias = False, matching the implementation.

Why We Use First- and Second-Order Derivatives by Default. We use first- and second-order derivatives by default because they provide interpretable rate-of-change and curvature features while keeping the module lightweight. Higher-order differences can amplify measurement noise, so their utility is evaluated empirically in Appendix B rather than assumed a priori. The first- and second-order kernels have controlled approximation errors $O(\Omega^3)$ and $O(\Omega^4)$ under the band-limited-signal assumptions in Lemma 1.

3.2.2 EDNBLOCK

The derivative features $\mathbf{D}_1$ and $\mathbf{D}_2$ are processed by EDNBlock to align with the backbone's latent space. Let a batch of input be denoted $\mathcal{X} \in \mathbb{R}^{B\times L\times C}$, where B is the batch size. For each sample, the derivative features $\mathbf{D}_i$ computed by Eq. 6 are collected into tensors $\mathcal{D}_i \in \mathbb{R}^{B\times C\times L}$. Because the evaluated backbone uses an inverted embedding, the same value-embedding matrix V maps the temporal dimension L to the latent dimension d_{model}:

$$\mathcal{Y}_i = \text{Linear}_{L\to d_{\text{model}}}(\mathcal{D}_i) \in \mathbb{R}^{B\times C\times d_{\text{model}}}, \quad i \in \{1, 2\}. \tag{7}$$

The projected features $\mathcal{Y}_i$ undergo Dropout to suppress noise inherent to finite-difference operations. Then LayerNorm is applied to stabilize feature statistics and resolve any distribution mismatch:

$$\bar{\mathcal{Y}}_i = \text{LayerNorm}\big(\text{Dropout}(\mathcal{Y}_i)\big). \tag{8}$$

A learnable scalar α_i balances the contribution of the first- and second-order dynamics, preventing either from dominating the base embedding:

$$\tilde{\mathcal{Y}}_i = \alpha_i \cdot \bar{\mathcal{Y}}_i, \quad i \in \{1, 2\}. \tag{9}$$

In the implementation, α_1 and α_2 are initialized by the configuration parameters `diff_scale` and `diff2_scale`, respectively, and remain learnable during training.

Finally, the scaled derivative features $\tilde{\mathcal{Y}}_1$ and $\tilde{\mathcal{Y}}_2$ are added elementwise to the backbone's base embedding $\mathcal{E}$ (obtained from $\mathcal{X}^\top$ via the same inverted value projection):

$$\mathcal{Y} = \mathcal{E} + \tilde{\mathcal{Y}}_1 + \tilde{\mathcal{Y}}_2. \tag{10}$$

All three tensors share the same shape, enabling seamless additive fusion without modifying the backbone's core structure.

3.3 BACKBONE

A key strength of LDAE is its compatibility with diverse forecasting backbones, including Transformer-based models (Liu et al., 2024; Chen et al., 2025), MLP-based models (Zeng et al., 2023), and CNN-based models (Liu et al., 2026; Wu et al., 2023). Unlike prior derivative-aware or auxiliary-branch modules that require structural modification or joint retraining (Nematirad et al., 2025; Wu et al., 2021), LDAE operates at the embedding stage. For models with a token embedding layer (Liu et al., 2024), it augments the embedding outputs through EDNBlock-refined additive fusion; for backbones without an explicit embedding layer, it is attached after input normalization at the backbone's first representation interface. In both cases, the derivative branch remains channel-local and is projected to the host feature space before additive fusion, while the backbone's internal blocks remain unchanged. Within the tested settings, LDAE is therefore a compatible derivative-aware representation module, not a universal improvement.

4 EXPERIMENTS

4.1 SETUP

We evaluate our method on 12 widely-used real-world datasets across multiple domains, including the ETT series (ETTh1, ETTh2, ETTm1, ETTm2) for energy monitoring (Zhou et al., 2021), ECL (Electricity) for power consumption (Trindade, 2015), Traffic for urban mobility, Weather for meteorological sensing (Wu et al., 2021), Solar for photovoltaic power forecasting (Lai et al., 2018), and PEMS (4 subsets) for short-term traffic flow prediction (Liu et al., 2022). We use a lookback window of 96 and long-term horizons $\{96, 192, 336, 720\}$, and short-term horizons $\{12, 24, 48, 96\}$ for PEMS. We also vary the lookback window $L \in \{48, 96, 192, 336, 720\}$ (Appendix E.1); dataset statistics are in Appendix A.1.

4.2 MAIN RESULTS

We evaluate LDAE as an instantiation of the embedding-stage injection hypothesis on eight long-term benchmarks with $L = 96$ and $H \in \{96, 192, 336, 720\}$. LDAE yields lower MSE in 22 of 32 dataset-horizon comparisons and lower MAE in 22 of 32 comparisons. Table 1 summarizes the four-horizon averages, whereas Appendix F.1 retains the individual horizons. Average errors improve on six of eight datasets. ECL and Weather define the main qualification: MSE is tied and MAE is higher than the VPNet baseline. Gains are clearest on non-stationary datasets such as Solar and ETTm2.

These results support derivative-aware embedding as a candidate inductive bias for MTSF: it makes local dynamic features available at the embedding stage. The benefit varies across datasets and concentrates on low-to-moderate-noise, non-stationary series, while the gains shrink or turn mixed on high-noise data or on already derivative-specialized backbones (Section 5).

Table 1: Average long-term forecasting results across the four prediction lengths, corresponding to Table 18. The best results are **bold**, and the second-best results are underlined. The 1st Count row reports the number of dataset-level averages at which each model attains the best MSE or MAE; tied minima are counted for all tied models.

Dataset	VPNet+LDAE **(Ours)**		VPNet 2026		SimpleTM 2025		CASA 2025		WPMixer 2025		ModernTCN 2024b		iTransformer 2024		PatchTST 2023		TimesNet 2023		DLinear 2023	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.433	**0.427**	0.435	0.428	**0.426**	0.432	0.442	0.440	0.447	0.434	0.435	0.428	0.455	0.448	0.451	0.442	0.461	0.455	0.458	0.451
ETTh2	0.357	**0.384**	0.375	0.394	0.355	0.392	0.381	0.406	0.385	0.408	**0.353**	0.388	0.384	0.407	0.366	0.395	0.400	0.417	0.498	0.479
ETTm1	**0.375**	**0.381**	0.376	0.382	0.382	0.396	0.389	0.400	0.400	0.405	0.388	0.402	0.408	0.413	0.383	0.396	0.412	0.418	0.405	0.409
ETTm2	**0.273**	**0.313**	0.280	0.317	0.276	0.323	0.278	0.323	0.277	0.322	0.281	0.324	0.291	0.335	0.285	0.328	0.294	0.331	0.310	0.366
ECL	**0.164**	0.253	**0.164**	**0.252**	0.174	0.268	0.175	0.268	0.180	0.269	0.212	0.290	0.176	0.267	0.190	0.275	0.199	0.299	0.225	0.318
Traffic	0.449	**0.276**	0.462	0.295	0.448	0.283	0.428	0.277	0.525	0.339	0.625	0.376	**0.422**	0.282	0.470	0.294	0.626	0.333	0.625	0.384
Solar	0.201	**0.209**	0.204	0.212	**0.190**	0.252	0.226	0.263	0.263	0.286	0.485	0.516	0.235	0.262	0.261	0.296	0.276	0.291	0.326	0.397
Weather	**0.239**	0.275	**0.239**	**0.263**	0.240	0.270	0.246	0.274	0.245	0.272	**0.239**	0.272	0.261	0.281	0.258	0.280	0.258	0.285	0.266	0.317
1st Count	4	6	2	2	2	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0

4.2.1 LDAE GENERALITY

To assess LDAE's architectural generality, we integrate it into seven representative forecasting backbones spanning MLP, CNN, and Transformer families (Table 2). LDAE reduces forecasting error in many of the evaluated backbone-dataset settings, with larger gains on low-to-moderate-noise, non-stationary series and smaller or negative gains on high-noise data such as Traffic (full results in Appendix F.2.1). The gains are most pronounced on high-dimensional data such as ECL.

On the lower-dimensional Weather dataset, improvements remain small across the evaluated backbones. Beyond long-horizon forecasting, we evaluate LDAE on short-horizon tasks (Table 3). On PEMS03 and PEMS07, LDAE matches or improves all three representative backbones (CASA, SOFTS, and iTransformer) across the reported metrics, with the largest gains for linear architectures. These results indicate transfer across long- and short-horizon tasks, while the benefit remains backbone-dependent.

4.2.2 INCREASING LOOKBACK WINDOW

Table 2: Average long-horizon errors for seven backbones with and without LDAE on ECL, Solar, and Weather. Lower values are better; bold marks the lower value within each backbone pair. Full results are given in Appendix F.2.1.

Models		SimpleTM 2025		CASA 2025		SOFTS 2024		iTransformer 2024		PatchTST 2023		TimesNet 2023		DLinear 2023	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	Original	0.174	0.268	0.176	0.268	0.175	0.265	0.176	0.267	0.190	0.275	0.199	0.300	0.225	0.318
	+LDAE	**0.167**	**0.261**	**0.168**	**0.263**	**0.170**	**0.262**	**0.170**	**0.263**	**0.189**	**0.274**	**0.194**	**0.293**	**0.208**	**0.295**
Solar	Original	0.190	0.252	0.226	0.263	0.230	0.257	0.235	0.262	0.261	0.296	0.276	0.291	0.326	0.397
	+LDAE	**0.186**	**0.246**	**0.225**	**0.250**	**0.225**	**0.254**	**0.234**	**0.261**	**0.259**	**0.295**	**0.270**	**0.289**	**0.293**	**0.345**
Weather	Original	0.241	0.270	0.246	0.274	0.257	0.279	0.261	0.281	0.258	0.281	0.258	0.285	0.267	0.317
	+LDAE	**0.240**	**0.268**	**0.245**	**0.273**	**0.252**	**0.277**	**0.253**	**0.277**	**0.252**	**0.278**	**0.256**	**0.283**	**0.263**	**0.316**
Total Improvements (vs. Baseline)		3	3	3	3	3	3	3	3	3	3	3	3	3	3

Table 3: Average short-horizon errors on PEMS03 and PEMS07 for three backbones with and without LDAE. Lower values are better; bold marks the lower value within each pair. Full results are given in Appendix F.2.2.

Models		CASA 2025		SOFTS 2024		iTransformer 2024	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	Original	0.152	0.254	0.105	0.213	0.152	0.258
	+LDAE	**0.150**	**0.252**	**0.091**	**0.195**	**0.132**	**0.241**
PEMS07	Original	0.137	0.231	0.092	0.192	0.176	0.266
	+LDAE	**0.135**	**0.231**	**0.073**	**0.165**	**0.169**	**0.260**
Improvements		2	2	2	2	2	2

Table 4: Four-horizon average comparison of LDAE injected before versus after cross-channel mixing in VPNet. Lower errors are better; bold marks the lower average. Full horizon-specific results are reported in Appendix D.3.

Dataset	Embedding		After-mixing	
	MSE	MAE	MSE	MAE
ETTh1	**0.433**	**0.426**	0.435	0.428
ECL	**0.164**	**0.252**	**0.164**	**0.252**
Solar	**0.200**	**0.208**	0.204	0.210
Traffic	0.448	0.275	**0.448**	**0.274**

Prior work shows that increasing the lookback window can degrade Transformer-based forecasting because redundant inputs may diffuse attention (Nie, 2023; Zeng et al., 2023). We therefore evaluate SimpleTM and iTransformer on ECL and ETTm1 with $H = 96$ and $L \in \{48, 96, 192, 336, 720\}$. As L increases, the baselines exhibit larger fluctuations and occasional degradation, whereas LDAE maintains more stable performance across the tested windows. These results suggest that derivative-aware embeddings reduce sensitivity to lookback-window selection, although they do not establish a universal long-context advantage.

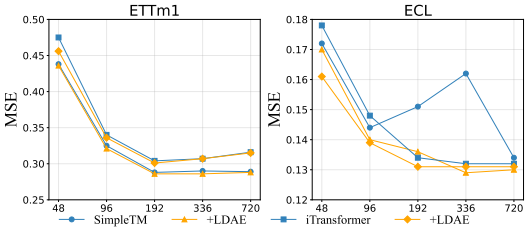


Figure 3: Representative lookback-window results with $H = 96$. The remaining datasets are shown in Appendix E.1.

4.2.3 Stability Analysis

To assess sensitivity to random initialization, we train CASA and CASA+LDAE with three random seeds. As shown in Figure 4, LDAE has a lower mean MSE and a smaller standard deviation in this experiment. This pattern is consistent with, but does not establish, a stabilizing effect of the derivative priors. The exploratory ANOVA and FDR-adjusted comparisons are reported in Appendix F.3.

4.3 Model Analysis

4.3.1 Computational and Memory Efficiency

We examine five design axes in LDAE: derivative-branch removal, initialization, channel locality, kernel learning, and blending-scale learning. As shown in Table 5, the complete model is usually best or near-best across the evaluated datasets and horizons. Full backbone-specific ablations are provided in Appendix D.

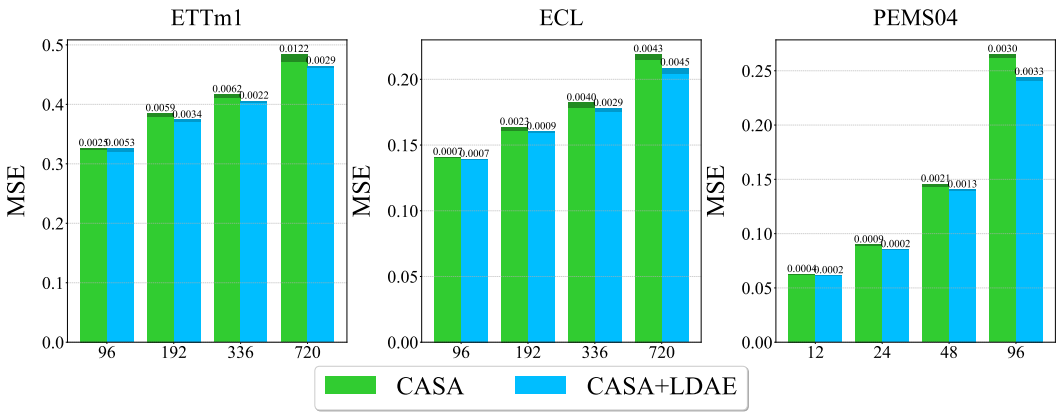


Figure 4: Stability analysis on ETTm1, ECL, and PEMS04 datasets. We report mean MSE (main bars) and standard deviation (error bars) over 3 random seeds.

Table 5: Component ablation for SimpleTM+LDAE across the long-term benchmarks. Lower errors are better; bold marks the best value within each horizon.

Method		SimpleTM+LDAE		w/o 1st-Diff		w/o 2nd-Diff		Random Initialization		Channel-Mixing Conv.		Fixed FD (α learned)		Fixed FD (α fixed)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.319**	**0.359**	0.323	0.362	0.321	0.360	0.321	0.360	0.321	0.360	0.330	0.367	0.330	0.367
	192	**0.359**	**0.379**	0.361	0.381	0.361	0.381	0.361	0.381	0.364	0.385	0.372	0.388	0.372	0.388
	336	**0.389**	**0.402**	0.389	**0.402**	0.392	0.404	0.390	0.403	0.397	0.408	0.402	0.406	0.402	0.406
	720	0.454	**0.437**	0.455	0.438	**0.453**	**0.437**	0.455	0.438	0.468	0.445	0.470	0.444	0.470	0.444
ECL	96	0.143	0.237	0.147	0.241	**0.140**	**0.233**	0.165	0.261	0.168	0.271	0.158	0.250	0.158	0.251
	192	**0.149**	**0.246**	0.151	0.247	0.167	0.265	0.150	**0.246**	0.167	0.271	0.184	0.271	0.185	0.273
	336	**0.172**	**0.266**	0.176	0.269	0.175	0.268	0.182	0.276	0.190	0.289	0.203	0.290	0.203	0.291
	720	**0.202**	**0.293**	0.204	0.295	**0.202**	**0.293**	0.214	0.302	0.228	0.318	0.242	0.321	0.242	0.322
Solar	96	**0.166**	**0.227**	0.179	0.243	0.170	0.229	0.177	0.251	0.180	0.243	0.180	0.248	0.188	0.253
	192	0.181	0.247	**0.180**	0.247	0.181	**0.246**	0.184	0.249	0.200	0.268	0.233	0.301	0.234	0.302
	336	0.192	**0.256**	0.192	**0.256**	**0.191**	0.256	**0.191**	**0.256**	0.217	0.280	0.248	0.309	0.250	0.310
	720	0.203	**0.254**	**0.199**	**0.254**	0.201	0.255	0.211	0.257	0.222	0.277	0.218	0.283	0.219	0.285

We next examine whether the injection location affects performance. Table 4 reports four-horizon averages for two otherwise identical VPNet variants. Embedding-stage injection performs better on ETTh1 and Solar and ties after-mixing injection on ECL. The two variants are comparable on Traffic, where after-mixing injection has slightly lower average MAE. The complete horizon-specific comparison is provided in Appendix D.3 (Table 15). Within this VPNet comparison, these results support using early injection as the default, while the horizon-level evidence remains dataset-dependent.

To assess practical efficiency, we benchmark training time and GPU memory on the high-dimensional ECL dataset. Figure 5 shows that the relative overhead is typically within $\pm 5\%$ for the evaluated backbones. The derivative branch adds an $O(CLK)$ term, linear in channels C, lookback length L, and kernel size K. Thus, LDAE introduces no higher-order asymptotic penalty, although its constant-factor cost is more visible in lightweight backbones. Because runtime and peak memory also depend on implementation details and allocator state, we interpret these values as empirical profiles of the tested configurations rather than as intrinsic hardware-independent guarantees.

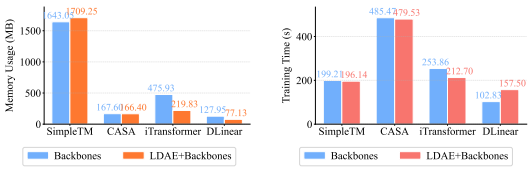


Figure 5: Training time and memory usage on ECL. Lower values indicate better efficiency.

4.3.2 Ablation Study

4.3.3 Token-Level Dynamic Sensitivity Analysis

We rank variables by their mean first-order derivative energy and split tokens into low- and high-variation groups. As shown in Figure 6, LDAE produces stronger responses for high-variation

tokens and weaker responses for low-variation tokens than the baseline embedding, indicating that it amplifies local dynamic cues rather than uniformly boosting all tokens.

4.3.4 KERNEL WEIGHTS VISUALIZATION

Figure 7 visualizes the learned convolution kernels on the Solar dataset. FD-initialized kernels retain a clearly structured pattern after training, whereas random-init kernels appear less structured in the shown visualization (see Appendix E.3).

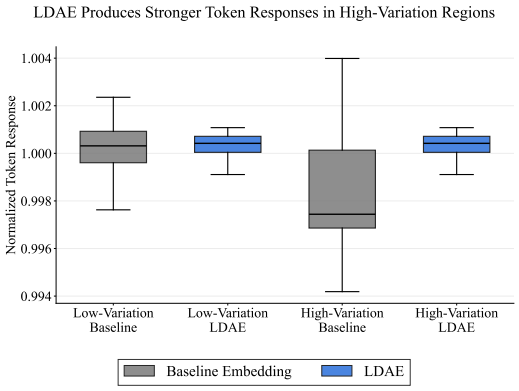


Figure 6: Normalized token-response distributions across low- and high-variation regimes. LDAE yields comparatively stronger responses in high-variation regions.

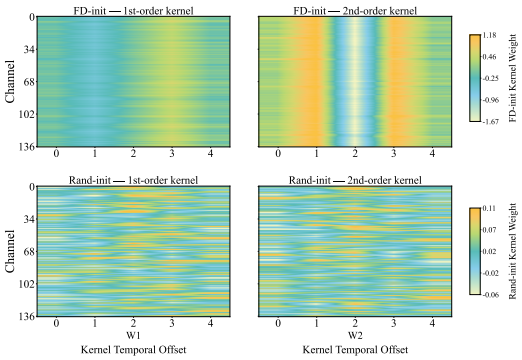


Figure 7: **Learned convolution kernels after full training.** Top: FD initialization preserves structured finite-difference stencils. Bottom: random initialization appears less structured in the shown example.

5 LIMITATIONS

LDAE's benefit is bounded by two consequences of its design. First, finite differencing amplifies high-frequency noise. Gains therefore concentrate on low-to-moderate-noise, non-stationary series and become mixed on high-noise data such as Traffic (Appendix B). Second, the benefit is backbone-dependent: the largest exploratory FDR-adjusted differences occur for SOFTS on ECL, Solar, Weather, and PEMS, whereas gains shrink or become mixed on the ETT series and on SimpleTM and CASA. Adding LDAE to Times2D, which already contains an explicit derivative branch, also produces no consistent gain (Appendix D.2). These boundaries position LDAE as a complementary prior rather than a universal booster and motivate noise-aware injection strength and redundancy detection.

6 CONCLUSION

In this work, we study whether making channel-wise derivative cues explicit at the embedding stage can improve MTSF. LDAE injects learnable first- and second-order differences through kernels initialized from classical stencils, providing a lightweight complementary prior before global aggregation. Its gains concentrate on low-to-moderate-noise, non-stationary series and remain dependent on the backbone. Future work will examine multi-scale operators, probabilistic forecasting, and broader time-series tasks.

AI USE STATEMENT

Large language models (LLMs) were used as assistive tools for English-language editing, improving clarity and concision, organizing the manuscript, checking terminology consistency, debugging LaTeX formatting, and cross-checking the correspondence among the prose, equations, pseudocode, and implementation descriptions. These tools were not used to generate or modify data, produce

or interpret experimental results, design the method or experiments, choose datasets or hyperparameters, formulate mathematical claims, or write proofs. Any AI-assisted wording or formatting suggestions were reviewed and revised by the authors and checked against the implementation, experiment records, equations, and reported results. The authors take full responsibility for the accuracy, originality, and integrity of the final manuscript and disclose this use in accordance with the ICLR policy on LLM use.

Ethics Statement

Our work focuses solely on time-series forecasting and does not raise any foreseeable ethical concerns.

Reproducibility Statement

The paper and appendix provide the information needed to reproduce the reported experiments. In particular, Appendix A.1 describes the public datasets, preprocessing protocol, chronological splits, and forecasting horizons; Appendix A.4 specifies the backbone configurations, optimization settings, evaluation metrics, hardware and software environments, and hyperparameters; and Algorithm 1 gives the implementation procedure for LDAE. The appendix also reports horizon-specific results, ablations, theoretical assumptions, and additional analyses, while the anonymous supplementary materials provide the implementation and configuration files used for the experiments.

References

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, pp. 289–300, 1995.

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

Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. Simpletm: A simple baseline for multivariate time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mingyue Cheng, Jiqian Yang, Tingyue Pan, Qi Liu, Zhi Li, and Shijin Wang. Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 171–180, 2025.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: efficient multivariate time series forecasting with series-core fusion. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: patch-specific spatial-temporal graph filtration for time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Maurice Kraus, Felix Divo, Devendra Singh Dhami, and Kristian Kersting. xlstm-mixer: Multivariate time series forecasting by mixing via scalar memories. *arXiv preprint arXiv:2410.16928*, 2024.

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

Minhyuk Lee, HyeKyung Yoon, and MyungJoo Kang. Casa: Cnn autoencoder-based score attention for efficient multivariate long-term time-series forecasting. *arXiv preprint arXiv:2505.02011*, 2025.

Kun Liu, Renjun Jia, Ruifeng Yang, Xirui Zeng, Yuqi Liang, and Cen Chen. Are global dependencies necessary? scalable time series forecasting via local cross-variate modeling. In *International Conference on Learning Representations*, 2026.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

Qinglong Liu, Cong Xu, Wenhao Jiang, Kaixuan Wang, Lin Ma, and Haifeng Li. Timestacker: A novel framework with multilevel observation for capturing nonstationary patterns in time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2024.

Donghao Luo and Xue Wang. Deformabletst: Transformer for time series forecasting without over-reliance on patching. *Advances in Neural Information Processing Systems*, 37:88003–88044, 2024a.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *The twelfth international conference on learning representations*, pp. 1–43, 2024b.

Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 19581–19588, 2025.

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*, 2025.

Y Nie. A time series is worth 64words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2023.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

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

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

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. *arXiv preprint arXiv:2405.14616*, 2024.

Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2004.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2023.

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*, pp. 11121–11128, 2023.

Xu Zhang, Qitong Wang, Peng Wang, and Wei Wang. Semixer: Semantics enhanced mlp-mixer for multiscale mixing and long-term time series forecasting. In *Proceedings of the ACM Web Conference 2026*, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792639. URL https://doi.org/10.1145/3774904.3792639.

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*, 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 *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A Implementation Details

A.1 Datasets

We evaluate the proposed **LDAE** on twelve publicly available multivariate time-series benchmarks spanning diverse domains including energy, meteorology, and traffic. These benchmarks vary in temporal resolution, sequence length, and channel dimensionality, offering a comprehensive and rigorous testbed for long-term forecasting. Specifically, the Electricity Transformer Temperature (ETT) dataset consists of seven recorded channels from July 2016 to July 2018 (Zhou et al., 2021), divided into four subsets: ETTh1 and ETTh2 (hourly resolution), and ETTm1 and ETTm2 (15-minute resolution). The Weather dataset includes 21 meteorological channels collected every 10 minutes in 2020 from the Max Planck Biogeochemistry Institute (Liu et al., 2022). The ECL dataset provides hourly electricity consumption records of 321 clients (Trindade, 2015). The Traffic dataset reports hourly road occupancy rates from 862 sensors in the San Francisco Bay Area between January 2015 and December 2016 (Wu et al., 2021). The Solar dataset contains 10-minute solar power production data from 137 photovoltaic plants in 2006 (Lai et al., 2018). Lastly, the PEMS dataset includes California traffic flow data at 5-minute resolution. We adopt the four standardized subsets PEMS03, PEMS04, PEMS07, and PEMS08 (Liu et al., 2024).

We follow the standard preprocessing protocol and chronological train-validation-test split adopted by TimesNet (Wu et al., 2023), ensuring no information leakage across temporal partitions. For ETT, Weather, ECL, Solar, PEMS, and Traffic, we evaluate forecasting horizons of $\{96, 192, 336, 720\}$. For PEMS, we use the shorter horizon settings $\{12, 24, 48, 96\}$ as in SCINet (Liu et al., 2022). These datasets jointly cover a wide range of sequence lengths and variate complexities, which are essential for evaluating LDAE's effectiveness in modeling non-stationary local dynamics and channel-wise temporal dependencies. Dataset details are summarized in Table 6.

Table 6: Dataset characteristics and forecasting settings. Sizes are reported as (train, validation, test).

Tasks	Dataset	Dim	Prediction Length	Dataset Size	Frequency	Information
Long-term Forecasting	ETTm1	7	$\{96, 192, 336, 720\}$	$(34465, 11521, 11521)$	15min	Temperature
	ETTm2	7	$\{96, 192, 336, 720\}$	$(34465, 11521, 11521)$	15min	Temperature
	ETTh1	7	$\{96, 192, 336, 720\}$	$(8545, 2881, 2881)$	Hourly	Temperature
	ETTh2	7	$\{96, 192, 336, 720\}$	$(8545, 2881, 2881)$	Hourly	Temperature
	ECL	321	$\{96, 192, 336, 720\}$	$(18317, 2633, 5261)$	Hourly	Electricity
	Traffic	862	$\{96, 192, 336, 720\}$	$(12185, 1757, 3509)$	Hourly	Transportation
	Weather	21	$\{96, 192, 336, 720\}$	$(36792, 5271, 10540)$	10min	Weather
	Solar	137	$\{96, 192, 336, 720\}$	$(36601, 5161, 10417)$	10min	Energy
Short-term Forecasting	PEMS03	358	$\{12, 24, 48, 96\}$	$(15617, 5135, 5135)$	5min	Transportation
	PEMS04	307	$\{12, 24, 48, 96\}$	$(10172, 3375, 3375)$	5min	Transportation
	PEMS07	883	$\{12, 24, 48, 96\}$	$(16911, 5622, 5622)$	5min	Transportation
	PEMS08	170	$\{12, 24, 48, 96\}$	$(10690, 3548, 265)$	5min	Transportation

A.2 Backbone Models

To thoroughly evaluate LDAE's generality and robustness, we employ a diverse set of backbone models from three major architectural families: Transformer-based, CNN-based, and MLP-based forecasting models. These backbones represent state-of-the-art approaches to multivariate time-series modeling, collectively covering global dependency modeling, local pattern extraction, and lightweight temporal mixing. We use publicly available official or open-source implementations and follow the hyperparameter configurations recommended in their respective papers. Unless otherwise stated, all backbones are trained under identical hardware conditions and operate in a non-autoregressive forecasting setting.

Transformer-based Backbones. We include **SimpleTM** (Chen et al., 2025), **iTransformer** (Liu et al., 2024), and **PatchTST** (Nie, 2023)—three representative models built on the attention mechanism. (1) *SimpleTM* adopts a streamlined Transformer encoder optimized for fast training and inference, reducing architectural complexity while preserving long-range dependency modeling capability. (2) *iTransformer* treats each channel as an independent token and applies self-attention over channels, enabling fine-grained modeling of inter-channel correlations through channel-wise token interaction. (3) *PatchTST* tokenizes the input by splitting long sequences into fixed-size patches and performs attention in the patch space, improving scalability for long-context forecasting while maintaining sensitivity to local temporal variations.

CNN-based Backbones. We adopt **VPNet** (Liu et al., 2026), **CASA** (Lee et al., 2025), **ModernTCN** (Luo & Wang, 2024b), and **TimesNet** (Wu et al., 2023) as CNN-based models. (1) *VPNet* models local cross-variate dependencies through variable-wise convolutional mixing, scaling multivariate forecasting without relying on global attention. (2) *CASA* employs a convolutional encoder-decoder architecture with Channel-mixing 1D Convolutions (groups=1) and transposed 1D Convolution, combined with adaptive attention modulation and residual connections, to efficiently capture localized temporal structures in MTSF. (3) *ModernTCN* extends the classical temporal convolutional network with modern architectural elements that stabilize optimization and enlarge the effective receptive field. (4) *TimesNet* maps the original 1D time series into multiple 2D tensors and applies multi-period convolutional operators, enabling joint capture of intra-period and inter-period patterns across different temporal frequencies.

MLP-based Backbones. We further include **WPMixer** (Murad et al., 2025), **SOFTS** (Han et al., 2024), and **DLinear** (Zeng et al., 2023), representing cutting-edge advances in efficient MLP-based architectures. (1) *WPMixer* performs hierarchical temporal decomposition through wavelet projection and mixes cross-frequency components using structured multilayer perceptrons. (2) *SOFTS* enhances temporal modeling via softly-modulated feed-forward layers, improving adaptability to non-stationary local dynamics and fine-grained temporal variations. (3) *DLinear* decomposes each time series into trend and seasonal components and applies dedicated linear predictors to each, achieving competitive accuracy with minimal architectural overhead. Together, these backbones span different temporal modeling paradigms, providing a benchmark for evaluating LDAE's cross-backbone behavior and effectiveness.

A.3 Metric Details

For quantitative assessment of LDAE's performance on long-term MTSF tasks, we employ two standard metrics widely adopted in the time series literature: Mean Squared Error (MSE) and Mean Absolute Error (MAE). These metrics are well-suited for capturing the magnitude of deviation between predicted outputs and ground-truth sequences, aligning with the core goal of accurate long-term forecast estimation.

Formally, let H denote the length of the forecast horizon (i.e., the number of time steps to predict), $\hat{y}_i \in \mathbb{R}^N$ represent the predicted feature vector at the i-th time step (where N is the number of features in the input time series), and $y_i \in \mathbb{R}^N$ denote the corresponding ground-truth feature vector. Averaging over both forecast steps and channels, MSE and MAE are defined as:

$$\mathrm{MSE} = \frac{1}{HN} \sum_{i=0}^{H-1} \|\hat{y}_i - y_i\|_2^2, \qquad \mathrm{MAE} = \frac{1}{HN} \sum_{i=0}^{H-1} \|\hat{y}_i - y_i\|_1. \tag{11}$$

A.4 Implementation Details

We adopt the official or publicly available implementations of all backbone models. Baseline architectural and training hyperparameters are taken from the corresponding papers or official implementations and are not re-tuned against the test split. For each backbone, the original and +LDAE variants use the same optimizer, loss, initial learning rate, learning-rate schedule, batch size, normalization, training budget, and early-stopping rule. We do not retune backbone parameters after adding LDAE. Only the LDAE-specific parameters, namely `use_diff`, `use_diff2`, `diff_kernel_size`, `diff_scale`, `diff2_scale`, and `diff_dropout`, are selected by grid

search on the validation split; the test split is never used for tuning. The search ranges and selected values are listed in Table 7.

For the reported runs, the common optimization setting is Adam with an initial learning rate of 1×10^{-4} and MSE (ℓ_2) training loss, while the backbone-specific schedule, batch size, normalization, and training budget are retained from the corresponding implementation. These settings are applied identically to each original/+LDAE pair. Experiments are conducted using Python 3.10.16 and PyTorch 2.6.0+cu124 on an Ubuntu server equipped with seven Tesla P100-PCIE GPUs (16 GB) and two NVIDIA GeForce RTX 4090 GPUs (24 GB). Each experiment is repeated three times with different random seeds. Unless otherwise specified, we report MSE and MAE as evaluation metrics, where lower values indicate better forecasting performance. LDAE's detailed implementation is presented in Algorithm 1.

Table 7: LDAE configurations used for the reported datasets and prediction horizons.

Datasets	Prediction Horizons	use_diff	use_diff2	diff_kernel_size	diff_scale (Range, Default)	diff2_scale (Range, Default)	diff_dropout (Range, Default)
ETTh1, ETTh2	96, 192, 336, 720	True/False	True/False	3,5	0.08–0.15 (**0.12**)	0.04–0.07 (**0.06**)	0.25–0.35 (**0.30**)
ETTm1, ETTm2	96, 192, 336, 720	True/False	True/False	3,5	0.10–0.18 (**0.15**)	0.05–0.09 (**0.07**)	0.20–0.30 (**0.25**)
Weather	96, 192, 336, 720	True/False	True/False	3,5,7	0.07–0.12 (**0.10**)	0.03–0.05 (**0.04**)	0.30–0.40 (**0.35**)
ECL	96, 192, 336, 720	True/False	True/False	3,5,7	0.14–0.20 (**0.18**)	0.06–0.10 (**0.08**)	0.15–0.25 (**0.20**)
Solar	96, 192, 336, 720	True/False	True/False	3,5,7	0.08–0.14 (**0.12**)	0.04–0.07 (**0.05**)	0.25–0.35 (**0.30**)
Traffic	96, 192, 336, 720	True/False	True/False	3,5,7	0.10–0.16 (**0.14**)	0.05–0.08 (**0.06**)	0.25–0.35 (**0.30**)
PEMS03, PEMS04, PEMS07, PEMS08	12, 24, 48, 96	True/False	True/False	3,5	0.14–0.20 (**0.18**)	0.06–0.10 (**0.08**)	0.30–0.40 (**0.35**)

Key Definitions:

- *use_diff*: Enables first-order finite difference branch (captures rate-of-change dynamics).
- *use_diff2*: Enables second-order finite difference branch (captures curvature/acceleration dynamics).
- *diff_kernel_size*: Window size for first-order difference convolution (smaller = more sensitive to local changes).
- *diff_scale/diff2_scale*: Scaling factors for the derivative branches (balance their contribution to the main embedding).
- *diff_dropout*: Dropout rate for derivative features (reduces overfitting to noisy derivative cues).

Algorithm 1 LDAE-enhanced multi-backbone time series forecasting.

Require: Input time series $X \in \mathbb{R}^{L \times C}$; Backbone type $\mathcal{B}$; LDAE config: use_diff, use_diff2, diff_kernel_size, diff_dropout, diff_scale, diff2_scale.

Ensure: Forecasting output $\hat{Y} \in \mathbb{R}^{H \times C}$

// — 1. Normalization (optional) —
if use_norm **then**
  $\mu, \sigma \leftarrow \text{MeanStd}(X)$
  $X \leftarrow (X - \mu)/\sigma$
end if
// — 2. Base embedding —
$E \in \mathbb{R}^{C \times d_{\text{model}}} \leftarrow \text{InvertedEmbedding}(X)$
The inverted embedding internally permutes X from $L \times C$ to $C \times L$ before the value projection.
// — 3. LDAE branches —
if use_diff **then**
  $D_1 \in \mathbb{R}^{C \times L} \leftarrow \text{ChannelWiseConv1D}(X^\top, \text{kernel} = \text{Stencil}_1(\text{diff_kernel_size}), \text{groups} = C)$
  $\text{Stencil}_1(3) = [-0.5, 0, 0.5]$; for larger odd kernels, retain these coefficients at offsets $-1, +1$
  $E_{d1} \in \mathbb{R}^{C \times d_{\text{model}}} \leftarrow \text{ValueEmbedding}_{L \to d_{\text{model}}}(D_1)$
  $E_{d1} \leftarrow \text{Dropout}(E_{d1}, p = \text{diff_dropout})$
  $E_{d1} \leftarrow \text{LayerNorm}(E_{d1})$
  $E \leftarrow E + \text{diff_scale} \cdot E_{d1}$
end if
if use_diff2 **then**
  $D_2 \in \mathbb{R}^{C \times L} \leftarrow \text{ChannelWiseConv1D}(X^\top, \text{kernel} = [1, -2, 1], \text{groups} = C)$
  $E_{d2} \in \mathbb{R}^{C \times d_{\text{model}}} \leftarrow \text{ValueEmbedding}_{L \to d_{\text{model}}}(D_2)$
  $E_{d2} \leftarrow \text{Dropout}(E_{d2}, p = \text{diff_dropout})$
  $E_{d2} \leftarrow \text{LayerNorm}(E_{d2})$
  $E \leftarrow E + \text{diff2_scale} \cdot E_{d2}$
end if
// — 4. Backbone processing —
$Z \in \mathbb{R}^{C \times d_{\text{model}}} \leftarrow \text{Backbone}_{\mathcal{B}}(E)$
// — 5. Output projection —
$\hat{Y} \in \mathbb{R}^{H \times C} \leftarrow \text{Projector}(Z)$
if use_norm **then**
  $\hat{Y} \leftarrow \hat{Y} \times \sigma + \mu$
end if
return $\hat{Y}$

B On the Role of Derivative Order in Time Series Embedding

The standard LDAE module uses first- and second-order differences, which capture local rate and curvature while retaining a simple interpretation. Higher-order differences are excluded because they can amplify noise.

To test this choice, we add learnable third- and fourth-order branches initialized with $[-1/6, 1/2, -1/2, 1/6]$ and $[1, -4, 6, -4, 1]$, respectively. Across 16 dataset-horizon pairs, the first-plus-second-order configuration has the lowest MSE in 9 pairs and the lowest MAE in 12, compared with 6 and 2 wins for the higher-order alternatives (Table 8). The extra branches also increase GPU memory by 5.8% on average, reaching 17.7% on ETTm1 at horizon 336, with a further 3–5% increase when the fourth-order branch is added (Table 9). The observed signal-to-noise ratio is 8–12 dB lower for third-order than second-order differences. These results support the first-plus-second-order configuration as a practical default, but they constitute exploratory evidence rather than a universal optimality claim.

Table 8: Derivative-order ablation across four datasets and four prediction horizons. The compared branches are learnable and initialized with the finite-difference templates defined in the text; lower errors are better and the best values are bold.

Configuration		1st-Diff		1st+2nd-Diff		3rd-Diff		3rd+4th-Diff	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.321	0.360	**0.319**	**0.359**	0.321	0.360	0.321	0.360
	192	0.361	0.381	**0.359**	**0.379**	**0.359**	**0.379**	0.360	0.381
	336	0.392	0.404	**0.389**	**0.402**	0.392	0.404	**0.389**	0.403
	720	**0.453**	**0.437**	0.454	**0.437**	**0.453**	**0.437**	0.455	0.439
ECL	96	**0.140**	**0.233**	0.143	0.237	0.141	0.235	0.146	0.239
	192	0.167	0.265	**0.149**	**0.246**	0.150	**0.246**	0.155	0.251
	336	0.175	0.268	**0.172**	**0.266**	0.175	0.267	0.177	0.269
	720	0.202	0.293	0.202	0.293	**0.198**	**0.291**	0.209	0.298
Solar	96	0.170	0.229	**0.166**	**0.227**	0.174	0.238	0.177	0.240
	192	0.181	0.246	0.181	0.247	**0.179**	**0.246**	0.184	0.250
	336	**0.191**	**0.256**	0.192	**0.256**	0.194	0.257	0.196	0.296
	720	0.201	0.255	0.203	**0.254**	**0.200**	**0.254**	**0.200**	**0.254**
Weather	96	0.157	0.203	**0.155**	**0.201**	**0.155**	**0.201**	0.158	0.203
	192	0.206	0.249	**0.202**	**0.243**	0.206	0.248	0.206	0.248
	336	**0.261**	**0.289**	0.263	0.290	0.265	0.291	0.271	0.296
	720	0.341	0.341	**0.338**	**0.339**	0.340	0.342	0.340	0.342

Note on even-length kernels. While the main LDAE module uses only odd kernel sizes ($K \in \{3, 5, 7\}$), the ablation study includes a third-order kernel of length $K = 4$. For this even-length kernel, symmetric padding padding $= K//2$ would change the output length. In our implementation, we use asymmetric same-length padding (left padding $= K//2$, right padding $= K//2 - 1$) to preserve the sequence shape. This detail is not discussed in the main paper for brevity, but it ensures the additive fusion remains valid.

Table 9: GPU memory usage (MB) for LDAE variants with different derivative orders.

Dataset	Horizon	1st-Diff	1st+2nd-Diff	3rd-Diff
		Size (MB)	Size (MB)	Size (MB)
ETTm1	96	**27.6**	27.9	28.0
	192	30.7	**28.4**	31.0
	336	34.6	**29.4**	34.6
	720	**53.7**	53.8	95.5
ECL	96	1370	**1200.4**	1406.5
	192	1497	**1323.3**	1528.8
	336	1682	**1519.1**	1713.9
	720	2280	**2120.8**	2308.3
Solar	96	**344.2**	350.1	358.1
	192	**284.3**	290.6	299.3
	336	**374.9**	383.2	390.3
	720	**626.0**	634.1	642.1
Weather	96	49.0	**46.7**	50.0
	192	59.6	**57.2**	60.6
	336	65.8	**65.0**	66.8
	720	**115.4**	**115.4**	115.9

C THEORETICAL ANALYSIS OF LEARNABLE FINITE-DIFFERENCE KERNELS

Note: This appendix separates analytical facts, design intuition, and exploratory evidence. The information-availability discussion is heuristic; the computational and channel-locality statements follow directly from the implementation; the training-stability statement is exploratory empirical evidence; and the FIR approximation result is rigorous only under its stated signal and frequency-response assumptions, guaranteeing existence but not a forecasting-performance improvement or an approximation rate.

This section addresses four key aspects: (1) the operator class represented by LDAE's kernels; (2) their distinction from fixed stencils; (3) why embedding-stage injection may preserve dynamic information; and (4) fundamental limitations.

Our analysis is based on the critical premise that linear FIR filters—including those used in LDAE—can only approximate linear derivative operators. They cannot represent nonlinear derivative expressions (e.g., $(x'(t))^2$ or $x(t) \cdot x''(t)$); nonlinear modeling is delegated to the backbone's subsequent layers.

Let $x[n] = s(n\Delta t)$ denote discrete samples of a smooth signal $s(t)$. In deep learning libraries (e.g., PyTorch's `Conv1d`), the operation is implemented as cross-correlation:

$$(D_w x)[n] = \sum_{k=-(K-1)/2}^{(K-1)/2} w_k \, x[n+k]. \tag{12}$$

This convention must be considered when determining the sign of an asymmetric finite-difference kernel. The classical finite-difference stencils used in LDAE are

$$w_{\text{FD}}^{(1)} = [-0.5,\ 0,\ 0.5], \qquad w_{\text{FD}}^{(2)} = [1,\ -2,\ 1],$$

which correspond to the first- and second-order derivatives, respectively.

Taylor Expansion under Cross-Correlation. Expanding $x[n+k] = s((n+k)\Delta t)$ around $n\Delta t$ gives

$$x[n+k] = \sum_{m=0}^{M} \frac{s^{(m)}(n\Delta t)}{m!}(k\Delta t)^m + R_M(n,k).$$

Substituting this expansion into Eq. 12 yields

$$(D_w x)[n] = \sum_{m=0}^{M} \frac{s^{(m)}(n\Delta t)}{m!} \underbrace{\sum_k w_k (k\Delta t)^m}_{\triangleq \mu_m(w)} + \widetilde{R}_M, \tag{13}$$

where

$$\mu_m(w) = \sum_k w_k (k\Delta t)^m$$

is the m-th discrete moment of the kernel and $\widetilde{R}_M$ is the accumulated remainder. For the moment calculations below, we use the normalized sampling interval $\Delta t = 1$.

Moment constraints for classical stencils. For $w_{\text{FD}}^{(1)}$, with $k \in \{-1, 0, 1\}$,

$$\mu_0 = -0.5 + 0 + 0.5 = 0,$$

and

$$\mu_1 = (-0.5)(-1) + 0(0) + (0.5)(1) = 1.$$

Therefore, the zeroth-order component is removed and the first-order component is retained with positive sign, producing the centered approximation

$$(D_{w_{\text{FD}}^{(1)}} x)[n] = \frac{x[n+1] - x[n-1]}{2}.$$

For $w_{\text{FD}}^{(2)}$,

$$\mu_0 = 1 - 2 + 1 = 0,$$
$$\mu_1 = 1(-1) + (-2)(0) + 1(1) = 0,$$
$$\mu_2 = 1(-1)^2 + (-2)(0)^2 + 1(1)^2 = 2.$$

Thus, the second-order term becomes

$$\frac{s''(n)}{2!}\,\mu_2 = s''(n),$$

under the normalized sampling interval. The corresponding second-order isolation constraints are therefore $\mu_0 = 0$, $\mu_1 = 0$, and $\mu_2 = 2$.

Learnable Kernels: Moment and Frequency Adaptation. Learnable kernels can adjust μ_m to balance approximation bias and noise sensitivity, whereas fixed stencils enforce predetermined low-order constraints and cannot adapt their higher-order moments. In the frequency domain, the discrete-time Fourier transform

$$H(\omega) = \sum_k w_k e^{j\omega k}$$

under the cross-correlation convention should approximate $(j\omega)^r$ within the relevant signal band. Learnable kernels can adaptively shape $H(\omega)$, whereas fixed stencils have a rigid frequency response.

Analytical Intuition and Approximation Results.

Information-availability intuition for derivative cues. Let Z_l be the representation at layer l of a deep network. Because $D(X)$ is a deterministic function of X, the Markov relation $D(X) \to X \to Z_l$ implies, by the data processing inequality,

$$I(D(X); Z_l) \le I(X; Z_l).$$

If derivative information is not explicitly provided at the input, the backbone may need to recover $D(X)$ from internal representations, which can attenuate derivative cues as information is transformed or compressed. LDAE computes $D(X)$ directly from X and fuses it with the base embedding, making derivative cues available to subsequent layers without requiring their implicit reconstruction. This is an information-availability intuition, not a performance theorem: the data-processing inequality alone does not prove a strict advantage for every backbone, so the empirical injection-position ablation provides the relevant evidence.

Proposition 2 (Computational Decoupling). Let $\mathcal{C}_{\text{backbone}}(L)$ denote the computational cost of the backbone, such as $O(L^2 d_{\text{model}})$ for a standard Transformer. LDAE adds an embedding-stage grouped 1D convolution with cost

$$\mathcal{C}_{\text{LDAE}} = O(CLK),$$

where $K = \max(K_1, K_2)$ is a small kernel size. This cost is additive rather than multiplicative. Consequently,

$$\mathcal{C}_{\text{total}} = \mathcal{C}_{\text{backbone}}(L) + O(CLK).$$

For backbones whose computational cost grows superlinearly with L, the additional linear term is asymptotically lower order. For linear-complexity backbones, LDAE preserves linear complexity while adding only a constant-factor overhead determined by C and K.

Proposition 3 (Channel-local derivative branch). Grouped convolutions with $\text{groups} = C$ have weight tensors

$$\mathbf{W}^{(i)} \in \mathbb{R}^{C \times 1 \times K_i},$$

where the c-th group operates only on the c-th input channel. For any distinct channels $c \neq c'$, the derivative feature $\mathbf{D}_c$ depends only on $\mathbf{x}^{(c)}$. Hence,

$$\frac{\partial \mathbf{D}_c}{\partial \mathbf{x}^{(c')}} = \mathbf{0}, \qquad c \neq c',$$

where $\mathbf{0}$ denotes the zero matrix. This zero cross-contamination property follows directly from the definition of grouped convolution.

Empirical observation (Training Stability). In the evaluated settings, finite-difference initialization is associated with lower early-training gradient variance than random initialization. This exploratory observation is reported across multiple datasets and backbones (see Appendix F.3); it is not a formal result and does not establish a universal optimization advantage.

Lemma 1 (Explicit derivative-approximation rate). The central-difference kernel $k^{(1)} = [-0.5,\ 0,\ 0.5]$ and the Laplacian-like kernel $k^{(2)} = [1,\ -2,\ 1]$ have cross-correlation frequency responses

$$H_1(\omega) \ = \ j\sin\omega, \qquad H_2(\omega) \ = \ 2\cos\omega - 2,$$

and satisfy the explicit error bounds

$$\sup_{|\omega|\le\Omega} |H_1(\omega) - j\omega| \ \le \ \frac{\Omega^3}{6}, \qquad \sup_{|\omega|\le\Omega} \left|H_2(\omega) + \omega^2\right| \ \le \ \frac{\Omega^4}{12}.$$

Consequently, on a signal band-limited to Ω, a length-3 kernel already achieves $O(\Omega^3)$ first-order and $O(\Omega^4)$ second-order approximation error; higher-order central-difference stencils of order s reduce the error to $O(\Omega^{2s+1})$.

Proof. By Taylor expansion, $\sin\omega = \omega - \omega^3/6 + \cdots$ and $2\cos\omega - 2 = -\omega^2 + \omega^4/12 - \cdots$. Taylor's theorem with the Lagrange remainder gives $|\sin\omega - \omega| \le |\omega|^3/6$ and $|2\cos\omega - 2 + \omega^2| \le \omega^4/12$, from which the stated bounds follow by taking the supremum over $[-\Omega, \Omega]$. Thus, for low-frequency components (small effective Ω), the length-3 stencil has a controlled approximation error; the kernel-size experiments in Appendix E.2 assess whether larger kernels are useful for richer spectra.

FIR approximation result under stated assumptions. Let

$$G(\omega) = (j\omega)^r, \qquad r \in \{1, 2\},$$

be defined on $[-\Omega, \Omega]$, where $0 < \Omega < \pi$. For every $\varepsilon > 0$, there exists a finite-length real-valued FIR kernel w such that

$$\sup_{|\omega|\le\Omega} |H(\omega) - G(\omega)| < \varepsilon,$$

where

$$H(\omega) = \sum_k w_k e^{j\omega k}$$

is the cross-correlation frequency response. Thus, the ideal first- or second-derivative frequency response can be uniformly approximated arbitrarily well on a compact subband by a finite-length real-valued FIR filter.

Proof. The function $G(\omega)$ is continuous on the compact interval $[-\Omega, \Omega]$. Consider the algebra of trigonometric polynomials

$$\mathcal{A} = \left\{ \sum_{k=-M}^{M} a_k e^{j\omega k} : M \in \mathbb{N},\ a_k \in \mathbb{C} \right\}.$$

This algebra contains the constants, separates points on $[-\Omega, \Omega]$, and is closed under complex conjugation. By the complex Stone–Weierstrass theorem, $\mathcal{A}$ is dense in $C([-\Omega, \Omega], \mathbb{C})$. Therefore, for every $\varepsilon > 0$, there exists a trigonometric polynomial

$$P(\omega) = \sum_{k=-M}^{M} a_k e^{j\omega k}$$

such that

$$\|P - G\|_\infty < \varepsilon.$$

Because G satisfies the conjugate-symmetry relation

$$G(-\omega) = \overline{G(\omega)},$$

we may symmetrize P by defining

$$Q(\omega) = \frac{1}{2}\left(P(\omega) + \overline{P(-\omega)}\right).$$

Then Q also satisfies

$$\|Q - G\|_\infty < \varepsilon$$

and can be written as

$$Q(\omega) = \sum_{k=-M}^{M} w_k e^{j\omega k}$$

with real coefficients w_k. Defining the FIR kernel using these coefficients gives a real-valued kernel of length $K = 2M + 1$ whose frequency response is exactly $Q(\omega)$. Therefore,

$$\sup_{|\omega| \le \Omega} |H(\omega) - G(\omega)| < \varepsilon.$$

The theorem guarantees existence but does not assert a particular approximation rate such as $K = O(\Omega/\varepsilon)$, which would require additional Jackson-type estimates. These approximation statements concern the frequency response of finite-length filters under the stated assumptions; they do not by themselves guarantee improved forecasting accuracy. □

Practical Implication. Small kernels ($K = 3, 5$) can suffice for low-frequency approximation. For signals with richer spectra, such as electricity load, a larger kernel (e.g., $K = 7$) can improve approximation in practice.

Limitations.

- **Nonlinearity:** FIR filters cannot approximate nonlinear derivative expressions. LDAE provides linear derivative features, while the backbone handles nonlinear combinations.
- **Aliasing:** Components above the Nyquist frequency cannot be recovered from the sampled sequence.
- **Kernel Length:** Increasing K increases the parameter count and may cause overfitting, especially on small datasets.

Initializing the kernels with finite-difference stencils provides a structured inductive prior that may facilitate optimization. During training, the kernels adapt to the empirical signal spectrum and noise level; in the reported ablations, finite-difference initialization often outperforms random initialization, but this should be interpreted as exploratory empirical evidence rather than a universal optimization claim.

D Ablation Studies

To isolate the contribution of each component, we evaluate six LDAE variants on SimpleTM, CASA, and DLinear across four prediction horizons: (1) removing the first-order differencing branch, (2) removing the second-order differencing branch, (3) replacing channel-wise 1D convolution with channel-mixing 1D convolution, (4) replacing structured finite-difference (FD) initialization with random initialization, (5) fixing FD kernels while keeping α learnable, and (6) fixing both FD kernels and α. Here, "channel-wise convolution" refers to group-wise convolution with groups $= C$, processing each variable independently without cross-talk; its "channel-mixing" counterpart uses groups $= 1$, allowing kernels to mix derivative features across channels.

Tables 10, 11, and 12 show that the complete module is strongest on the high-dimensional ECL and Solar datasets, although the magnitude of each component's contribution depends on the backbone.

It preserves channel-local derivative information and generally outperforms channel-mixing alternatives across most settings. Notably, replacing channel-wise convolution with channel-mixing convolution degrades MSE by up to 28.5%, with the largest gap observed on ECL-96 using the CASA backbone. On the SimpleTM backbone, performance drops by 12.9% on ECL-720 and 8.4% on Solar-96. These results are consistent with a scoped design hypothesis: channel-local processing at the embedding stage can be useful for derivative feature extraction when early cross-channel mixing would alter channel-specific temporal dynamics.

However, we observe an exception to this trend: for the linear DLinear backbone on the Solar dataset, the channel-mixing variant ("Replace 1D Convolution") consistently achieves lower MSE

than the channel-wise version (e.g., 0.242 vs. 0.270 at horizon 96; see Table 12). This may reflect DLinear's channel-independent linear predictors: because they do not provide an internal cross-channel mixing path, channel-mixing convolution can supply shared local patterns across highly correlated solar channels, whereas channel-wise convolution preserves channel separation. Despite this exception, the advantage of channel-wise convolution remains substantial for Transformer- and CNN-based backbones, which are the primary focus of this work.

Beyond channel-local processing, multi-order finite differences also play an important role. The combination of first- and second-order branches enables LDAE to capture both abrupt changes and smooth curvature variations. On ECL and Solar, removing either branch can lead to performance degradation across multiple horizons. This indicates that multi-order derivative cues provide complementary information for modeling complex non-stationary patterns. For relatively stationary datasets such as Weather, a single finite-difference branch may capture most temporal variations, resulting in smaller performance differences.

Complementing the structural design, the initialization strategy and adaptive α-scaling mechanism further improve LDAE in the reported ablations. Structured FD initialization often outperforms random initialization in these settings, consistent with a possible optimization benefit from the finite-difference prior. Comparisons with the fixed-kernel variants further show that learnable FD kernels and learnable α often provide the strongest results, although several individual settings contain ties or small exceptions. Allowing the model to adjust derivative feature magnitudes and fine-tune the FD kernels therefore offers a useful trade-off across the evaluated data distributions.

While the above components deliver gains across most scenarios, we also observe edge cases that delineate the method's applicability. For channel-independent linear backbones such as DLinear, the gains from adding multiple finite-difference branches are relatively modest and depend on whether the derivative branch is allowed to mix channels. Nevertheless, channel-local convolution remains effective for most nonlinear backbones, supporting the value of preserving per-channel derivative information.

Collectively, these ablation results suggest that channel-local convolution, structured finite-difference initialization, and adaptive multi-order derivative cues with learnable α-scaling can work complementarily in the evaluated settings. The gains across diverse backbones, together with the observed exceptions, characterize both the empirical scope and the applicability boundary of LDAE.

Table 10: Component ablation for SimpleTM+LDAE across the long-term benchmarks. Lower errors are better; bold marks the best value within each horizon.

Method		SimpleTM+LDAE		w/o 1st-Diff		w/o 2nd-Diff		Replace FD-Init		Replace 1D Convolution		Fixed FD (α learn.)		Fixed FD (α fixed)	
Design		1st-Diff 2nd-Diff Channel-wise		w/o 2nd-Diff Channel-wise		1st-Diff w/o Channel-wise		Random Initialization Random Initialization Channel-wise		1st-Diff 2nd-Diff Channel-mixing		1st-Diff 2nd-Diff Channel-wise		1st-Diff 2nd-Diff Channel-wise	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.319**	**0.359**	0.323	0.362	0.321	0.360	0.321	0.360	0.321	0.360	0.330	0.367	0.330	0.367
	192	**0.359**	**0.379**	0.361	0.381	0.361	0.381	0.361	0.381	0.364	0.385	0.372	0.388	0.372	0.388
	336	**0.389**	**0.402**	0.389	**0.402**	0.392	0.404	0.390	0.403	0.397	0.408	0.402	0.406	0.402	0.406
	720	0.454	**0.437**	0.455	0.438	**0.453**	**0.437**	0.455	0.438	0.468	0.445	0.470	0.444	0.470	0.444
ECL	96	0.143	0.237	0.147	0.241	**0.140**	**0.233**	0.165	0.261	0.168	0.271	0.158	0.250	0.158	0.251
	192	**0.149**	**0.246**	0.151	0.247	0.167	0.265	0.150	**0.246**	0.167	0.271	0.184	0.271	0.185	0.273
	336	**0.172**	**0.266**	0.176	0.269	0.175	0.268	0.182	0.276	0.190	0.289	0.203	0.290	0.203	0.291
	720	**0.202**	**0.293**	0.204	0.295	**0.202**	**0.293**	0.214	0.302	0.228	0.318	0.242	0.321	0.242	0.322
Solar	96	**0.166**	**0.227**	0.179	0.243	0.170	0.229	0.177	0.251	0.180	0.243	0.180	0.248	0.188	0.253
	192	0.181	0.247	**0.180**	0.247	0.181	**0.246**	0.184	0.249	0.200	0.268	0.233	0.301	0.234	0.302
	336	0.192	**0.256**	0.192	**0.256**	**0.191**	0.256	**0.191**	**0.256**	0.217	0.280	0.248	0.309	0.250	0.310
	720	0.203	**0.254**	**0.199**	**0.254**	0.201	0.255	0.211	0.257	0.222	0.277	0.218	0.283	0.219	0.285
Weather	96	**0.155**	**0.201**	0.157	0.203	0.157	0.203	0.157	0.203	0.159	0.205	0.165	0.210	0.166	0.210
	192	**0.202**	**0.243**	0.203	0.246	0.206	0.249	0.205	0.248	0.207	0.248	0.214	0.253	0.215	0.253
	336	0.263	0.290	0.262	**0.289**	**0.261**	**0.289**	0.263	0.290	0.263	0.290	0.274	0.296	0.274	0.296
	720	**0.338**	**0.339**	0.339	0.341	0.341	0.341	0.339	0.341	0.342	0.342	0.345	0.342	0.345	0.342

D.1 Comparing Derivative Modules in Times2D: FSDH vs. LDAE

To test LDAE against an explicitly derivative-oriented backbone, we replace Times2D's handcrafted first- and second-order heatmaps (FSDH) with LDAE. Period decomposition, the Transformer encoder, and the prediction head remain unchanged, isolating the effect of the derivative representation.

Table 13 reports the results on three diverse datasets (ETTh1, Weather, and Solar) over four forecasting horizons. In terms of MSE, LDAE improves Times2D in 11 of the 12 dataset-horizon pairs,

Table 11: Component ablation for CASA+LDAE, emphasizing the effect of channel locality and finite-difference initialization. Lower errors are better; bold marks the best value within each horizon.

Method		CASA+LDAE		w/o 1st-Diff		w/o 2nd-Diff		Replace FD-Init		Replace 1D Convolution		Fixed FD (α learn.)		Fixed FD (α fixed)	
Design		1st-Diff 2nd-Diff Channel-wise		w/o 2nd-Diff Channel-wise		1st-Diff w/o Channel-wise		Random Initialization Random Initialization Channel-wise		1st-Diff 2nd-Diff Channel-mixing		1st-Diff 2nd-Diff Channel-wise		1st-Diff 2nd-Diff Channel-wise	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.321**	**0.358**	0.322	0.360	0.323	0.360	0.327	0.362	0.324	0.362	0.325	0.360	0.323	**0.358**
	192	0.374	0.387	0.380	0.391	**0.367**	**0.383**	0.376	0.387	0.368	0.385	0.379	0.391	0.377	0.389
	336	0.400	0.409	**0.399**	**0.406**	0.404	0.411	0.401	0.407	0.405	0.411	0.421	0.417	0.398	0.406
	720	**0.457**	0.446	0.460	0.445	0.460	**0.444**	0.469	0.448	0.458	0.446	0.468	0.451	0.471	0.452
ECL	96	**0.137**	**0.233**	0.138	0.235	0.140	0.235	0.138	0.234	0.176	0.277	**0.137**	**0.233**	0.138	**0.233**
	192	**0.159**	0.254	0.160	**0.253**	0.160	**0.253**	0.165	0.258	0.195	0.290	0.160	0.255	0.158	0.253
	336	**0.173**	**0.269**	0.175	0.272	0.178	0.272	0.175	0.272	0.193	0.291	0.178	0.271	0.183	0.276
	720	**0.201**	**0.296**	0.212	0.303	0.213	0.304	0.215	0.300	0.244	0.336	0.211	0.300	0.216	0.305
Solar	96	**0.192**	**0.233**	0.197	0.237	0.195	0.234	0.195	0.234	0.229	0.272	0.195	0.235	0.195	0.234
	192	**0.226**	**0.220**	0.227	0.261	0.227	0.261	0.230	0.260	0.263	0.293	0.227	0.260	0.226	0.260
	336	0.240	**0.272**	0.240	**0.272**	**0.239**	0.273	0.240	0.273	0.283	0.308	0.241	0.274	0.242	0.274
	720	0.243	**0.275**	0.242	**0.275**	0.242	**0.275**	**0.241**	**0.275**	0.281	0.290	0.241	0.275	0.241	0.275
Weather	96	**0.158**	**0.205**	0.159	**0.205**	0.163	0.210	0.160	0.207	0.159	0.207	0.160	0.206	0.160	0.206
	192	**0.206**	**0.248**	0.209	0.250	0.208	0.250	0.209	0.251	0.208	0.257	0.211	0.251	0.209	0.250
	336	**0.266**	**0.292**	0.267	**0.292**	0.283	0.325	0.275	0.298	0.268	0.294	0.265	0.291	0.265	0.291
	720	0.350	0.346	**0.347**	0.344	**0.347**	**0.343**	**0.347**	0.344	0.348	0.346	0.352	0.349	0.354	0.350

Table 12: Component ablation for DLinear+LDAE, including the channel-mixing exception on Solar. Lower errors are better; bold marks the best value within each horizon.

Method		DLinear+LDAE		w/o 1st-Diff		w/o 2nd-Diff		Replace FD-Init		Replace 1D Convolution		Fixed FD (α learn.)		Fixed FD (α fixed)	
Design		1st-Diff 2nd-Diff Channel-wise		w/o 2nd-Diff Channel-wise		1st-Diff w/o Channel-wise		Random Initialization Random Initialization Channel-wise		1st-Diff 2nd-Diff Channel-mixing		1st-Diff 2nd-Diff Channel-wise		1st-Diff 2nd-Diff Channel-wise	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.333	0.372	0.332	0.372	**0.330**	**0.370**	0.334	0.378	0.334	0.382	0.352	0.381	0.352	0.381
	192	0.369	0.388	0.372	0.388	**0.366**	**0.385**	0.367	0.389	0.371	0.397	0.389	0.396	0.389	0.396
	336	0.402	0.413	0.399	0.405	**0.395**	**0.405**	0.402	0.409	0.404	0.420	0.388	0.394	0.420	0.419
	720	0.458	0.447	0.467	0.450	0.452	**0.437**	**0.451**	**0.437**	0.480	0.471	0.477	0.454	0.477	0.454
ECL	96	0.193	0.276	0.193	**0.272**	0.192	0.276	0.193	0.277	**0.180**	0.284	0.200	0.286	0.201	0.287
	192	0.192	**0.279**	0.193	**0.279**	0.192	**0.279**	0.193	**0.279**	**0.190**	0.295	0.200	0.290	0.201	0.291
	336	0.205	0.295	0.205	0.295	**0.204**	**0.294**	0.205	0.295	0.207	0.314	0.213	0.306	0.214	0.308
	720	0.240	0.328	0.240	0.328	0.294	**0.326**	0.240	0.327	**0.236**	0.333	0.248	0.335	0.249	0.336
Solar	96	0.270	0.350	0.268	0.342	0.277	0.352	0.283	0.370	**0.242**	**0.319**	0.289	0.375	0.290	0.376
	192	0.291	0.343	0.289	0.353	0.288	0.349	0.311	0.388	**0.260**	**0.323**	0.319	0.396	0.321	0.399
	336	0.307	0.347	0.312	0.361	0.309	0.355	0.343	0.404	**0.271**	**0.322**	0.353	0.413	0.354	0.414
	720	0.304	0.340	0.311	0.352	0.307	0.346	0.322	0.368	**0.272**	**0.316**	0.353	0.412	0.356	0.416
Weather	96	0.193	**0.260**	**0.191**	0.265	0.194	0.267	0.195	0.263	**0.191**	0.264	0.203	0.268	0.203	0.268
	192	**0.232**	**0.294**	0.233	0.302	**0.232**	**0.294**	0.237	0.299	0.241	0.305	0.239	0.297	0.239	0.295
	336	**0.282**	**0.330**	0.285	0.343	**0.282**	0.331	0.297	0.355	0.291	0.345	0.286	0.336	0.287	0.334
	720	**0.346**	**0.378**	0.351	0.390	0.356	0.386	0.354	0.390	0.351	0.385	0.347	0.381	0.347	0.379

with Weather at horizon 96 being the only exception. In terms of MAE, LDAE also improves 11 of the 12 pairs, with ETTh1 at horizon 720 being the only exception. The gains are particularly large on Solar (average MSE improvement of 9.9% and MAE improvement of 10.4%) and are consistent on ETTh1 (4.9% MSE and 1.8% MAE on average). On Weather, LDAE achieves MAE reductions across all horizons and improves MSE on three horizons, suggesting that learnable derivative modeling is especially beneficial for non-stationary time series. Overall, these results support the interpretation that learnable finite-difference initialization can strengthen the derivative branch in Times2D under the evaluated settings.

Table 13: Controlled FSDH replacement in Times2D. Only the handcrafted derivative branch is replaced by LDAE; lower errors are better and bold marks the best value.

Method	Horizon	ETTh1		Weather		Solar	
		MSE	MAE	MSE	MAE	MSE	MAE
Original (FSDH)	96	0.381	0.402	**0.163**	0.216	0.233	0.289
	192	0.449	0.435	0.219	0.262	0.270	0.320
	336	0.481	0.448	0.282	0.308	0.303	0.338
	720	0.464	**0.458**	0.345	0.349	0.293	0.330
+LDAE	96	**0.371**	**0.394**	0.169	**0.212**	**0.216**	**0.261**
	192	**0.419**	**0.420**	**0.217**	**0.254**	**0.243**	**0.281**
	336	**0.456**	**0.439**	**0.271**	**0.293**	**0.272**	**0.307**
	720	**0.459**	0.460	**0.342**	**0.341**	**0.258**	**0.295**
Improvements (MSE / MAE)		4/4 / 3/4		3/4 / 4/4		4/4 / 4/4	
Avg. Improvement (MSE / MAE)		4.9% / 1.8%		0.5% / 3.0%		9.9% / 10.4%	

D.2 End-to-End Integration into Times2D

We next add LDAE to the complete Times2D pipeline while retaining its FSDH branch. This end-to-end test asks whether a second derivative pathway remains useful after the backbone already exposes derivative cues. The results on ETTh1 and ETTm1 (Table 14) show no consistent gain: performance degrades on most ETTh1 horizons and improves only at intermediate ETTm1 horizons. The result defines a clear applicability boundary for redundant derivative representations.

Table 14: End-to-end Times2D comparison with and without an additional LDAE branch. Lower errors are better; bold marks the best value.

Method	Horizon	ETTh1		ETTm1	
		MSE	MAE	MSE	MAE
Times2D	96	**0.369**	**0.393**	**0.343**	**0.371**
	192	**0.434**	**0.426**	0.379	0.395
	336	0.473	**0.442**	0.419	0.424
	720	0.481	**0.469**	0.473	0.449
Times2D+LDAE	96	0.379	0.401	0.345	0.374
	192	0.439	0.433	**0.377**	**0.390**
	336	0.473	0.443	**0.413**	**0.418**
	720	**0.480**	0.470	0.473	0.449

D.3 Injection Position: Embedding Stage vs. After Cross-Channel Mixing

To determine whether the benefit of LDAE is specific to the embedding stage, we keep the derivative branch unchanged and move only its injection point. In the embedding-stage variant, the derivative features are added to the patch-encoder output before cross-channel mixing; in the after-mixing variant, they are added to the cross-channel-mixing output immediately before the prediction head.

Both variants use identical finite-difference convolutions, projection, dropout, normalization, and learned scales.

Across the 16 dataset-horizon pairs in Table 15, embedding-stage injection matches or improves upon after-mixing injection in 13 pairs for each metric. The preference is clearest on Solar and ETTh1, whereas the two positions are nearly equivalent on ECL. Traffic is the main exception: after-mixing injection performs better at shorter horizons, while embedding-stage injection is preferable at horizon 720. These horizon-level results provide qualified support for injecting derivative cues before cross-channel mixing.

Table 15: Horizon-specific injection-position ablation on VPNet across four datasets and four prediction horizons. The two variants differ only in whether LDAE is added before or after cross-channel mixing; lower errors are better.

Dataset	Horizon	Embedding-stage injection		After-mixing injection	
		MSE ↓	MAE ↓	MSE ↓	MAE ↓
ETTh1	96	**0.374**	**0.386**	**0.374**	0.387
	192	**0.428**	**0.419**	0.431	0.420
	336	**0.462**	**0.437**	0.465	0.440
	720	**0.469**	**0.464**	0.470	0.465
	Avg	**0.433**	**0.426**	0.435	0.428
ECL	96	**0.134**	**0.224**	0.135	**0.224**
	192	**0.151**	**0.239**	0.152	0.241
	336	**0.167**	**0.257**	**0.167**	**0.257**
	720	0.204	0.291	**0.202**	**0.289**
	Avg	**0.164**	**0.252**	**0.164**	**0.252**
Solar	96	**0.168**	**0.187**	0.171	0.188
	192	**0.201**	**0.205**	0.202	0.206
	336	**0.211**	**0.219**	0.216	0.220
	720	**0.223**	**0.223**	0.229	0.226
	Avg	**0.200**	**0.208**	0.204	0.210
Traffic	96	0.408	**0.258**	**0.407**	**0.258**
	192	0.438	0.271	**0.437**	**0.268**
	336	**0.455**	0.278	**0.455**	**0.275**
	720	**0.493**	**0.295**	0.496	0.296
	Avg	**0.448**	0.275	**0.448**	**0.274**

D.4 Core Performance Gain Attribution: First- and Second-Order Derivative Convolutions

To isolate and quantify the fundamental performance contribution of LDAE, we conduct an extreme ablation study that removes all first- and second-order derivative convolution kernels, retaining only the EDNBlock. This minimal single-variable modification ensures that any observed performance difference can be attributed to the explicit derivative modeling capability, with all other training and inference conditions held constant. Experiments are performed on a representative backbone (SimpleTM) across five benchmark datasets, with results presented in Table 16.

Removing the derivative convolution kernels leads to performance degradation in 19 of the 20 dataset-horizon combinations. The sole exception is Traffic at horizon 96, where EDNBlock alone yields marginally better MSE and MAE. These results indicate that explicit first- and second-order temporal derivative modeling is a major source of LDAE's performance improvement under the evaluated settings.

Table 16: Derivative-branch ablation on SimpleTM: full LDAE versus EDNBlock alone. Lower errors are better; the best values are bold red.

Method		SimpleTM+LDAE		EDNBlock Only	
Metric		MSE	MAE	MSE	MAE
ETTm1	96	**0.319**	**0.359**	0.329	0.366
	192	**0.359**	**0.379**	0.371	0.387
	336	**0.389**	**0.402**	0.403	0.407
	720	**0.454**	**0.437**	0.470	0.444
ECL	96	**0.143**	**0.237**	0.156	0.249
	192	**0.149**	**0.246**	0.180	0.269
	336	**0.172**	**0.266**	0.200	0.288
	720	**0.202**	**0.293**	0.239	0.320
Solar	96	**0.166**	**0.227**	0.233	0.300
	192	**0.181**	**0.247**	0.189	0.259
	336	**0.192**	**0.256**	0.247	0.308
	720	**0.203**	**0.254**	0.219	0.284
Weather	96	**0.155**	**0.201**	0.165	0.209
	192	**0.202**	**0.243**	0.214	0.253
	336	**0.263**	**0.290**	0.273	0.295
	720	**0.338**	**0.339**	0.345	0.340
Traffic	96	0.427	0.280	**0.419**	**0.278**
	192	**0.428**	**0.276**	0.454	0.302
	336	**0.450**	**0.284**	0.472	0.312
	720	**0.482**	**0.304**	0.507	0.334

E Hyperparameter Analysis

E.1 Increasing Lookback Window

We test sensitivity to the lookback window at fixed $H = 96$, using $L \in \{48, 96, 192, 336, 720\}$. Across the datasets shown in Figures 8, 9, and 10, LDAE varies less across window lengths than the baselines. This pattern suggests reduced sensitivity to window selection, not a universal long-context advantage.

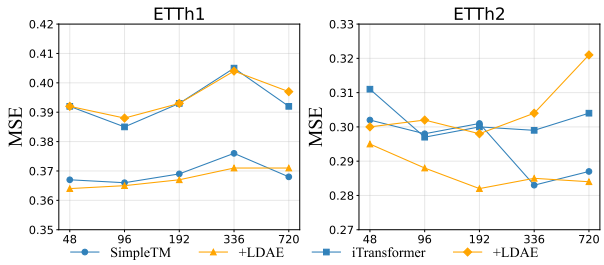


Figure 8: Lookback-window sensitivity on the hourly ETT datasets, ETTh1 and ETTh2.

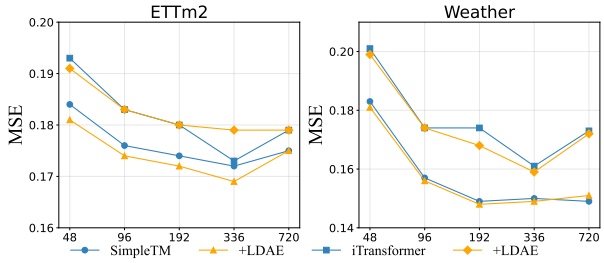


Figure 9: Lookback-window sensitivity on the higher-frequency ETTm2 and Weather series.

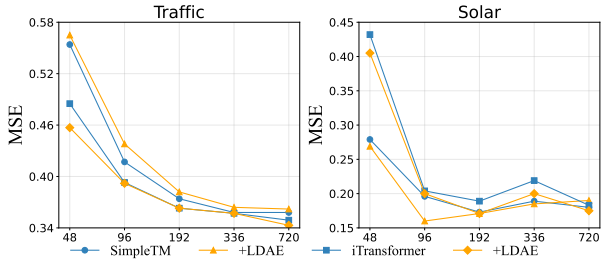


Figure 10: Lookback-window sensitivity under traffic and solar dynamics.

E.2 FINITE-DIFFERENCE KERNEL SIZE

We evaluate kernel sizes $K \in \{3, 5, 7\}$ for LDAE's finite-difference operators. Table 17 summarizes performance across eight dataset-horizon-metric combinations per dataset. The results show dataset-dependent preferences: $K = 3$ is effective for high-frequency or short-cycle data, $K = 7$ is advantageous for low-frequency or long-horizon tasks, and $K = 5$ provides a balanced intermediate choice.

The largest absolute MSE spread is 0.102 on ETTh1 at horizon 96, corresponding to approximately 21% relative to the $K = 3$ result. The largest spreads on Traffic, ECL, and Solar are 0.009, 0.006, and 0.007, respectively. We recommend $K = 3$ for high-frequency or short-term forecasts, $K = 7$ for slow-moving or long-horizon series, and $K = 5$ as a balanced default.

Table 17: Performance of iTransformer combined with LDAE under different finite-difference kernel sizes ($K = 3, 5, 7$). Lower values indicate better forecasting accuracy, and the best results for each horizon are highlighted in **bold**.

Method		Kernel=3		Kernel=5		Kernel=7	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.488	**0.405**	**0.386**	**0.405**	0.388	0.406
	192	0.443	0.437	0.441	0.436	**0.440**	**0.435**
	336	0.489	0.460	0.494	0.462	**0.475**	**0.453**
	720	0.502	0.489	0.528	0.504	**0.500**	**0.487**
ETTh2	96	0.298	0.349	0.302	0.350	**0.295**	**0.347**
	192	0.381	0.399	0.378	**0.397**	**0.377**	0.398
	336	**0.420**	**0.430**	0.421	0.431	0.424	0.432
	720	0.425	0.444	**0.424**	**0.443**	0.427	0.445
ETTm1	96	0.335	0.372	**0.334**	**0.371**	0.335	**0.371**
	192	**0.376**	0.392	**0.376**	0.393	**0.376**	**0.391**
	336	**0.411**	**0.416**	0.413	**0.416**	0.414	0.417
	720	**0.480**	**0.453**	0.482	0.455	0.487	0.457
ETTm2	96	0.184	0.269	0.183	**0.265**	**0.182**	0.266
	192	0.251	0.312	0.252	0.312	**0.249**	**0.305**
	336	0.315	0.351	0.316	**0.350**	**0.313**	**0.350**
	720	0.415	0.407	**0.410**	0.405	**0.410**	**0.404**
ECL	96	0.140	0.234	**0.139**	**0.233**	**0.139**	0.234
	192	0.158	**0.251**	0.159	0.253	**0.157**	**0.251**
	336	0.174	0.268	**0.172**	**0.267**	**0.172**	**0.267**
	720	0.210	0.299	**0.204**	**0.296**	**0.204**	**0.296**
Traffic	96	**0.388**	**0.268**	0.391	0.269	0.393	0.270
	192	**0.417**	**0.277**	**0.417**	0.278	0.420	0.279
	336	**0.438**	**0.284**	**0.438**	0.286	**0.438**	0.286
	720	**0.471**	**0.302**	0.480	0.304	0.478	0.404
Solar	96	0.202	0.236	**0.198**	**0.231**	**0.198**	**0.231**
	192	0.236	0.261	0.237	0.265	**0.230**	**0.257**
	336	0.248	**0.272**	0.248	0.273	**0.247**	**0.272**
	720	0.251	0.276	**0.248**	**0.275**	0.249	**0.275**
Weather	96	0.172	0.213	0.171	0.212	**0.167**	**0.209**
	192	**0.212**	**0.251**	0.214	0.252	0.215	0.254
	336	**0.274**	**0.296**	0.276	0.298	0.275	0.297
	720	0.355	**0.349**	0.354	**0.349**	**0.353**	**0.349**

E.3 KERNEL WEIGHTS VISUALIZATION

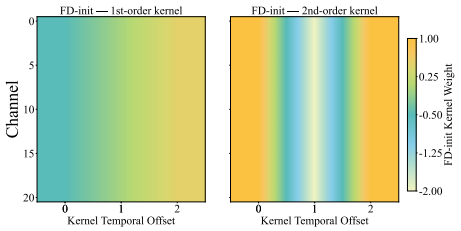
To examine how the initialized filters adapt across signal regimes, we visualize first- and second-order kernels on Weather, Solar, and Traffic. Each figure compares FD-init and random-init before and after training.

On Weather, FD-init kernels remain close to the initial stencils after training, whereas random-init kernels remain less structured in the displayed example. The limited movement is consistent with the smoother dynamics of this dataset.

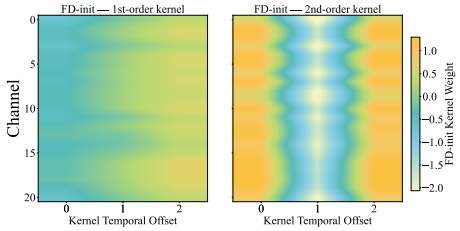
On Solar, the trained FD-init kernels retain their coarse finite-difference pattern but develop finer structure. This qualitative change is consistent with adaptation to the dataset's non-stationary and irregular fluctuations.

On Traffic, the learned filters preserve a coarse finite-difference pattern but show more blurring. This visual pattern is compatible with the observed degradation in some settings, although the visualization alone cannot establish the noise-amplification mechanism.

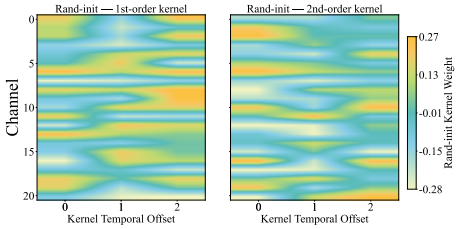
Across the three examples, random-init kernels are less structured than FD-init kernels. These visualizations provide qualitative support for structured initialization, but they do not by themselves establish a causal optimization benefit.



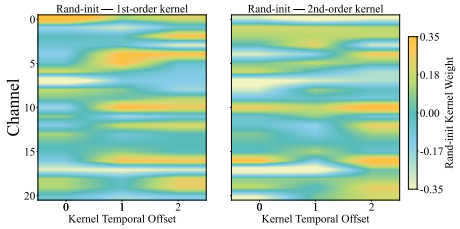
(a) FD-init *before* training



(b) FD-init *after* training



(c) Random-init *before* training



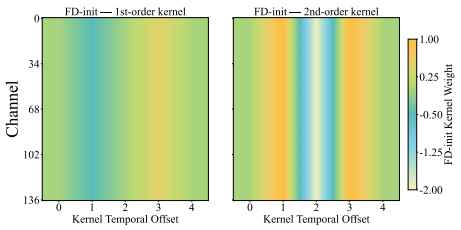
(d) Random-init *after* training

Figure 11: Weather: FD-init kernels remain close to their initial finite-difference stencils after training.

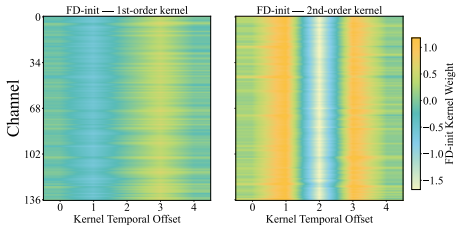
F FULL RESULTS

F.1 LONG-TERM FORECASTING

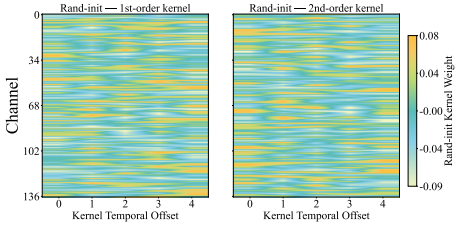
This section records horizon-specific long-term results for eight benchmarks at $L = 96$ and $H \in \{96, 192, 336, 720\}$. The main text reports the averages; Table 18 exposes how those averages vary across datasets and horizons. The table is intended for detailed comparison, not as a second narrative summary.



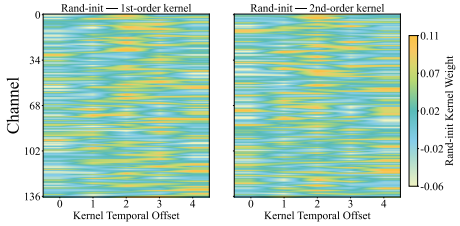
(a) FD-init *before* training



(b) FD-init *after* training

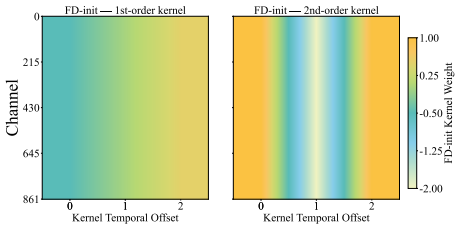


(c) Random-init *before* training

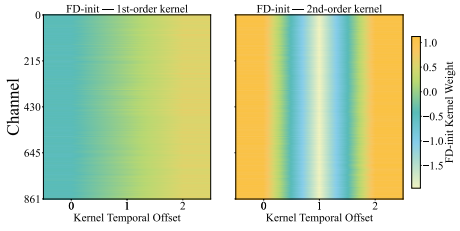


(d) Random-init *after* training

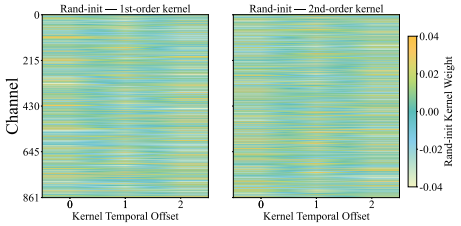
Figure 12: Solar: trained FD-init kernels retain the coarse stencil pattern while developing finer structure.



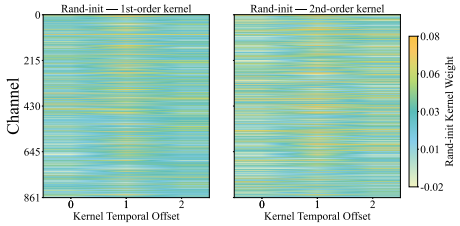
(a) FD-init *before* training



(b) FD-init *after* training



(c) Random-init *before* training



(d) Random-init *after* training

Figure 13: Traffic: FD-init kernels preserve coarse finite-difference structure but show stronger blurring after training.

Table 18: Horizon-specific long-term forecasting results across eight benchmarks. "Avg." is the mean over the four prediction horizons; red bold and blue underlining denote the best and second-best values, respectively. The 1st Count row reports the number of red-bold first-place entries for each model-metric column across all displayed horizons and Avg. rows; tied minima are counted for all tied models.

Model		VPNet+LDAE (Ours)		VPNet 2026		SimpleTM 2025		CASA 2025		WPMixer 2025		ModernTCN 2024b		iTransformer 2024		PatchTST 2023		TimesNet 2023		DLinear 2023	
Year																					
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.374	**0.386**	0.375	0.387	**0.366**	0.393	0.382	0.404	0.386	0.399	0.384	0.396	0.386	0.405	0.394	0.408	0.389	0.412	0.382	0.395
	192	0.428	0.419	0.428	**0.418**	**0.421**	0.421	0.427	0.428	0.428	0.424	0.439	0.425	0.442	0.436	0.445	0.434	0.442	0.443	0.432	0.425
	336	0.462	0.437	0.470	0.444	**0.438**	0.439	0.474	0.451	0.483	0.448	0.442	**0.426**	0.486	0.458	0.484	0.453	0.495	0.471	0.491	0.467
	720	0.469	0.464	**0.467**	**0.463**	0.479	0.474	0.484	0.475	0.490	0.467	0.474	0.464	0.506	0.491	0.480	0.471	0.519	0.494	0.528	0.518
	Avg.	0.433	**0.427**	0.435	0.428	**0.426**	0.432	0.442	0.440	0.447	0.434	0.435	0.428	0.455	0.448	0.451	0.442	0.461	0.455	0.458	0.451
ETTh2	96	0.284	**0.329**	0.286	0.331	0.282	0.338	0.296	0.347	0.298	0.347	**0.281**	0.336	0.299	0.350	0.295	0.344	0.335	0.370	0.329	0.380
	192	0.359	**0.377**	0.372	0.381	**0.357**	0.388	0.375	0.397	0.373	0.396	0.361	0.387	0.381	0.399	0.377	0.393	0.403	0.413	0.431	0.443
	336	0.401	0.409	0.444	0.438	0.366	0.401	0.422	0.433	0.424	0.434	**0.349**	**0.388**	0.422	0.432	0.382	0.410	0.435	0.440	0.458	0.461
	720	**0.385**	**0.419**	0.399	0.425	0.417	0.439	0.430	0.446	0.446	0.456	0.421	0.440	0.432	0.448	0.411	0.433	0.428	0.444	0.773	0.631
	Avg.	0.357	**0.384**	0.375	0.394	0.355	0.392	0.381	0.406	0.385	0.408	**0.353**	0.388	0.384	0.407	0.366	0.395	0.400	0.417	0.498	0.479
ETTm1	96	**0.310**	**0.340**	0.313	0.342	0.321	0.360	0.321	0.360	0.335	0.366	0.322	0.364	0.340	0.375	0.323	0.360	0.333	0.375	0.345	0.373
	192	**0.360**	**0.369**	0.361	0.372	0.362	0.381	0.372	0.388	0.387	0.398	0.368	0.389	0.382	0.396	0.364	0.384	0.408	0.414	0.382	0.392
	336	0.386	0.390	**0.385**	**0.389**	0.390	0.403	0.403	0.407	0.396	0.406	0.398	0.410	0.419	0.419	0.391	0.402	0.415	0.422	0.416	0.417
	720	**0.442**	**0.423**	0.443	0.424	0.454	0.438	0.461	0.445	0.483	0.451	0.462	0.444	0.490	0.462	0.454	0.439	0.492	0.462	0.479	0.456
	Avg.	**0.375**	**0.381**	0.376	0.382	0.382	0.396	0.389	0.400	0.400	0.405	0.388	0.402	0.408	0.413	0.383	0.396	0.412	0.418	0.405	0.409
ETTm2	96	**0.171**	**0.244**	0.176	0.248	0.175	0.258	0.176	0.258	0.174	0.257	0.174	0.256	0.185	0.271	0.178	0.260	0.185	0.262	0.186	0.281
	192	**0.234**	**0.289**	0.236	0.291	0.238	0.298	0.243	0.303	0.239	0.298	0.238	0.298	0.252	0.312	0.249	0.307	0.251	0.306	0.272	0.348
	336	0.297	**0.331**	0.304	0.334	0.298	0.339	**0.296**	0.336	0.302	0.340	0.312	0.346	0.316	0.352	0.309	0.345	0.319	0.348	0.343	0.394
	720	**0.391**	**0.388**	0.404	0.395	0.394	0.396	0.399	0.396	0.393	0.393	0.399	0.394	0.413	0.406	0.404	0.400	0.420	0.407	0.438	0.443
	Avg.	**0.273**	**0.313**	0.280	0.317	0.276	0.323	0.278	0.323	0.277	0.322	0.281	0.324	0.291	0.335	0.285	0.328	0.294	0.331	0.310	0.366
ECL	96	**0.134**	**0.224**	0.135	0.225	0.150	0.245	0.140	0.236	0.150	0.241	0.189	0.270	0.148	0.240	0.166	0.252	0.167	0.271	0.210	0.301
	192	**0.151**	**0.239**	**0.151**	**0.239**	**0.151**	0.248	0.164	0.256	0.162	0.252	0.196	0.276	0.166	0.257	0.174	0.260	0.187	0.289	0.210	0.304
	336	**0.167**	**0.257**	0.168	**0.257**	0.189	0.284	0.179	0.273	0.180	0.270	0.210	0.290	0.179	0.272	0.190	0.277	0.203	0.304	0.223	0.319
	720	0.204	0.291	**0.202**	**0.288**	0.206	0.296	0.219	0.307	0.230	0.313	0.253	0.324	0.210	0.299	0.229	0.310	0.238	0.334	0.257	0.349
	Avg.	**0.164**	0.253	**0.164**	**0.252**	0.174	0.268	0.175	0.268	0.180	0.269	0.212	0.290	0.176	0.267	0.190	0.275	0.199	0.299	0.225	0.318
Traffic	96	0.408	**0.258**	0.419	0.277	0.420	0.270	0.396	0.262	0.521	0.344	0.652	0.390	**0.394**	0.269	0.450	0.286	0.589	0.315	0.649	0.396
	192	0.438	**0.271**	0.437	0.282	0.435	0.277	0.416	0.272	0.507	0.332	0.599	0.363	**0.413**	0.277	0.456	0.287	0.614	0.323	0.599	0.371
	336	0.455	**0.278**	0.480	0.309	0.450	0.284	0.433	**0.278**	0.530	0.336	0.606	0.366	**0.423**	0.283	0.471	0.293	0.638	0.341	0.606	0.374
	720	0.493	**0.295**	0.513	0.314	0.488	0.303	0.467	0.297	0.541	0.344	0.644	0.385	**0.459**	0.301	0.503	0.312	0.664	0.353	0.646	0.395
	Avg.	0.449	**0.276**	0.462	0.295	0.448	0.283	0.428	0.277	0.525	0.339	0.625	0.376	**0.422**	0.282	0.470	0.294	0.626	0.333	0.625	0.384
Solar	96	**0.168**	**0.187**	0.179	0.193	0.180	0.244	0.194	0.234	0.230	0.269	0.452	0.497	0.203	0.235	0.230	0.277	0.241	0.268	0.285	0.372
	192	0.201	**0.205**	0.193	0.210	**0.182**	0.248	0.227	0.269	0.273	0.293	0.478	0.510	0.238	0.262	0.260	0.295	0.268	0.294	0.315	0.393
	336	0.211	**0.219**	0.217	0.222	**0.198**	0.262	0.240	0.273	0.276	0.292	0.513	0.532	0.250	0.275	0.277	0.306	0.288	0.297	0.349	0.413
	720	0.223	**0.223**	0.225	**0.223**	**0.200**	0.254	0.241	0.275	0.272	0.291	0.499	0.523	0.250	0.276	0.276	0.305	0.307	0.306	0.354	0.410
	Avg.	0.201	**0.209**	0.204	0.212	**0.190**	0.252	0.226	0.263	0.263	0.286	0.485	0.516	0.235	0.262	0.261	0.296	0.276	0.291	0.326	0.397
Weather	96	**0.151**	0.239	0.155	**0.193**	0.156	0.201	0.161	0.206	0.163	0.208	0.157	0.205	0.176	0.215	0.178	0.219	0.168	0.219	0.196	0.255
	192	**0.202**	**0.237**	**0.202**	0.240	0.204	0.247	0.208	0.251	0.209	0.249	0.205	0.250	0.226	0.259	0.225	0.259	0.224	0.264	0.239	0.297
	336	0.262	0.285	0.260	**0.283**	0.263	0.289	0.267	0.293	0.266	0.289	**0.254**	0.291	0.283	0.300	0.278	0.298	0.283	0.305	0.283	0.332
	720	0.340	0.338	**0.337**	**0.337**	0.339	0.341	0.349	0.346	0.341	0.342	0.340	0.343	0.359	0.351	0.350	0.346	0.355	0.352	0.348	0.384
	Avg.	**0.239**	0.275	**0.239**	**0.263**	0.240	0.270	0.246	0.274	0.245	0.272	**0.239**	0.272	0.261	0.281	0.258	0.280	0.258	0.285	0.266	0.317
1st Count		17	29	8	12	10	0	1	1	0	0	5	2	5	0	0	0	0	0	0	0

F.2 Full Framework Generality Results

F.2.1 Long-Term Forecasting Generality Results

We next test architectural transfer in long-term forecasting. LDAE is inserted before a backbone's first global or cross-channel interaction when such an interaction stage is present; for linear backbones, it is attached at the corresponding input representation interface. All models use $L = 96$. Tables 19 and 20 report the complete comparisons across eight datasets.

For each backbone-dataset pair, *Count of Wins* records how often +LDAE is no worse than the original model across the four horizons and their average. Win counts are high for most pairs, but Traffic remains a clear exception. The results therefore support transfer across the evaluated architectures while retaining the dataset- and backbone-specific boundary identified above.

Table 19: Long-term generality results on ETTh1–ETTm2 for seven backbones with and without LDAE. Lower errors are better; bold marks the better value within each comparison.

Models			SimpleTM 2025		CASA 2025		SOFTS 2024		iTransformer 2024		PatchTST 2023		TimesNet 2023		DLinear 2023	
Metric			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	Original	96	0.366	0.393	0.382	0.404	0.385	0.402	**0.386**	0.405	0.394	0.408	**0.389**	**0.412**	0.382	0.395
		192	**0.421**	**0.421**	**0.427**	**0.428**	0.441	0.437	0.442	0.436	0.445	0.434	0.442	0.443	0.432	**0.425**
		336	**0.438**	0.439	**0.474**	0.451	0.484	0.457	0.486	0.458	0.484	0.453	0.495	**0.471**	0.491	0.467
		720	0.479	0.474	0.484	0.475	0.495	0.486	0.506	0.491	0.480	0.471	0.519	0.494	0.528	0.518
		Avg	0.426	0.432	0.442	0.440	0.451	0.446	0.455	0.448	0.451	0.442	0.461	0.455	0.458	0.451
	+LDAE	96	**0.365**	**0.391**	**0.380**	**0.404**	**0.385**	**0.401**	0.387	**0.405**	**0.391**	**0.406**	0.403	0.420	**0.381**	**0.395**
		192	0.422	0.422	0.430	0.429	**0.440**	**0.433**	**0.440**	**0.436**	**0.443**	**0.433**	**0.438**	**0.439**	**0.430**	0.428
		336	**0.438**	**0.438**	**0.474**	**0.451**	**0.480**	**0.454**	**0.484**	**0.457**	**0.481**	**0.450**	**0.493**	0.474	**0.474**	**0.447**
		720	**0.470**	**0.467**	**0.476**	**0.470**	**0.490**	**0.479**	**0.502**	**0.489**	**0.476**	**0.469**	**0.489**	**0.478**	**0.496**	**0.488**
		Avg	**0.424**	**0.430**	**0.440**	**0.439**	**0.449**	**0.442**	**0.453**	**0.447**	**0.448**	**0.440**	**0.456**	**0.453**	**0.445**	**0.440**
	Count of Wins		4/5	4/5	4/5	4/5	5/5	5/5	4/5	5/5	5/5	5/5	4/5	3/5	5/5	4/5
ETTh2	Original	96	**0.282**	0.338	**0.296**	**0.347**	0.294	0.346	0.299	**0.350**	0.295	**0.344**	0.335	**0.370**	0.379	0.422
		192	0.357	0.388	0.375	0.397	0.378	0.397	0.381	**0.399**	0.377	**0.393**	0.403	0.413	0.508	0.495
		336	**0.366**	**0.401**	0.422	0.433	**0.424**	**0.434**	**0.424**	**0.432**	**0.381**	**0.409**	**0.435**	**0.440**	0.625	0.558
		720	0.417	0.439	0.430	0.446	**0.424**	**0.442**	0.432	0.448	**0.412**	**0.433**	**0.428**	**0.444**	**0.773**	**0.631**
		Avg	0.356	0.392	0.381	0.406	0.380	0.405	0.384	**0.407**	**0.366**	**0.395**	**0.400**	**0.417**	**0.571**	**0.527**
	+LDAE	96	**0.282**	0.340	**0.296**	**0.347**	**0.291**	**0.346**	**0.298**	**0.350**	**0.294**	**0.344**	**0.332**	0.371	**0.364**	**0.411**
		192	**0.355**	**0.384**	**0.370**	**0.393**	**0.375**	**0.396**	**0.380**	**0.399**	**0.376**	**0.393**	**0.403**	**0.413**	**0.505**	**0.493**
		336	**0.366**	**0.401**	**0.419**	**0.431**	**0.424**	**0.434**	**0.424**	**0.432**	**0.381**	**0.409**	0.465	0.459	**0.619**	**0.555**
		720	**0.416**	**0.438**	**0.427**	**0.444**	**0.424**	**0.442**	**0.430**	**0.447**	**0.412**	**0.433**	0.461	0.466	0.858	0.669
		Avg	**0.355**	**0.391**	**0.378**	**0.404**	**0.379**	**0.404**	**0.383**	**0.407**	**0.366**	**0.395**	0.415	0.427	0.587	0.532
	Count of Wins		4/5	3/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	2/5	1/5	4/5	4/5
ETTm1	Original	96	0.321	0.360	**0.321**	0.360	0.322	0.361	0.340	0.375	0.323	0.360	0.333	0.375	0.345	0.373
		192	0.362	0.381	**0.372**	0.388	**0.372**	**0.389**	0.382	0.396	0.364	**0.384**	0.408	0.414	0.382	0.392
		336	**0.389**	0.403	0.403	**0.409**	0.422	0.418	0.419	0.419	**0.388**	**0.406**	**0.415**	**0.422**	0.416	0.417
		720	**0.454**	0.438	0.461	0.445	0.472	0.450	0.490	0.462	0.454	0.439	0.492	0.462	0.479	0.456
		Avg	0.382	0.396	0.389	**0.400**	0.397	0.405	0.408	0.413	0.383	**0.396**	0.412	0.418	0.406	0.410
	+LDAE	96	**0.319**	**0.359**	**0.321**	**0.358**	**0.319**	**0.357**	**0.334**	**0.371**	**0.320**	**0.359**	0.335	**0.374**	**0.333**	**0.372**
		192	**0.359**	**0.379**	0.374	**0.387**	**0.372**	**0.389**	**0.376**	**0.393**	**0.361**	**0.384**	**0.387**	**0.402**	**0.369**	**0.388**
		336	**0.389**	**0.402**	**0.400**	**0.409**	**0.417**	**0.415**	**0.413**	**0.416**	**0.388**	**0.406**	0.426	0.429	**0.402**	**0.413**
		720	**0.454**	**0.437**	**0.457**	0.446	**0.459**	**0.446**	**0.482**	**0.455**	**0.447**	**0.438**	**0.491**	0.463	**0.458**	**0.447**
		Avg	**0.380**	**0.394**	**0.388**	**0.400**	**0.392**	**0.402**	**0.401**	**0.409**	**0.379**	0.397	**0.410**	0.417	**0.391**	**0.405**
	Count of Wins		5/5	5/5	4/5	4/5	5/5	5/5	5/5	5/5	5/5	4/5	3/5	2/5	5/5	5/5
ETTm2	Original	96	**0.175**	0.258	0.176	**0.258**	0.181	0.263	0.185	0.271	0.178	**0.260**	0.185	**0.267**	**0.186**	**0.281**
		192	0.238	**0.298**	0.243	0.303	0.248	**0.308**	0.252	0.312	0.249	**0.305**	**0.251**	**0.313**	**0.272**	**0.348**
		336	0.298	0.339	**0.296**	**0.337**	0.312	0.345	0.316	0.352	0.309	**0.343**	**0.316**	**0.346**	**0.407**	**0.425**
		720	**0.394**	**0.400**	**0.397**	**0.396**	0.409	**0.405**	0.413	**0.405**	0.404	**0.401**	**0.426**	**0.409**	**0.634**	**0.540**
		Avg	**0.276**	**0.323**	0.279	0.323	0.288	0.330	0.292	0.335	0.285	0.328	**0.294**	**0.331**	**0.310**	**0.367**
	+LDAE	96	**0.175**	**0.257**	**0.175**	**0.258**	**0.180**	**0.263**	**0.184**	**0.269**	**0.177**	**0.260**	0.187	**0.267**	0.208	0.310
		192	**0.237**	**0.298**	**0.196**	**0.235**	**0.248**	**0.308**	**0.251**	**0.312**	**0.245**	**0.305**	0.262	**0.313**	0.301	0.367
		336	**0.296**	**0.337**	**0.296**	**0.337**	**0.309**	**0.345**	**0.315**	**0.351**	**0.306**	**0.343**	**0.316**	**0.346**	**0.407**	**0.425**
		720	0.403	**0.400**	**0.397**	**0.396**	**0.407**	**0.405**	**0.410**	**0.405**	**0.401**	**0.401**	**0.426**	**0.409**	**0.634**	**0.540**
		Avg	0.278	**0.323**	**0.266**	**0.307**	**0.286**	**0.330**	**0.290**	**0.334**	**0.282**	**0.327**	0.298	0.334	0.388	0.411
	Count of Wins		4/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	3/5	5/5	3/5	2/5

Note: Unlike some prior works that adopt dataset-specific lookback windows, we employ a uniform lookback window ($T = 96$) across all datasets to ensure an architecture-agnostic evaluation. Count of Wins denotes the number of prediction lengths (including Avg.) for which +LDAE achieves an error lower than or equal to that of the original baseline.

Table 20: Long-term generality results on ECL, Traffic, Solar, and Weather. Lower errors are better; bold marks the better value within each comparison, and red marks Count-of-Wins entries.

Models			SimpleTM 2025		CASA 2025		SOFTS 2024		iTransformer 2024		PatchTST 2023		TimesNet 2023		DLinear 2023	
Metric			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	Original	96	0.150	0.245	0.140	0.236	0.144	0.235	0.148	0.240	0.166	0.252	0.167	0.271	0.210	0.301
		192	0.151	0.248	0.164	0.256	0.161	0.252	0.166	0.257	0.174	0.260	0.187	0.289	0.210	0.304
		336	0.189	0.284	0.179	0.273	0.180	0.271	0.179	0.272	**0.190**	0.277	0.203	0.304	0.223	0.319
		720	0.206	0.296	0.219	0.307	0.214	0.302	**0.210**	**0.299**	**0.229**	**0.310**	0.238	0.334	0.257	0.349
		Avg	0.174	0.268	0.176	0.268	0.175	0.265	0.176	0.267	0.190	0.275	0.199	0.300	0.225	0.318
	+LDAE	96	**0.143**	**0.237**	**0.137**	**0.233**	**0.139**	**0.231**	**0.140**	**0.234**	**0.160**	**0.249**	**0.165**	**0.269**	**0.193**	**0.276**
		192	**0.149**	**0.246**	**0.159**	**0.254**	**0.158**	**0.250**	**0.158**	**0.251**	**0.170**	**0.259**	**0.185**	**0.287**	**0.192**	**0.279**
		336	**0.172**	**0.266**	**0.173**	**0.269**	**0.176**	**0.270**	**0.174**	**0.268**	0.192	**0.276**	**0.194**	**0.295**	**0.205**	**0.295**
		720	**0.202**	**0.293**	**0.201**	**0.296**	**0.206**	**0.297**	**0.210**	**0.299**	0.235	0.313	**0.230**	**0.321**	**0.240**	**0.328**
		Avg	**0.167**	**0.261**	**0.168**	**0.263**	**0.170**	**0.262**	**0.171**	**0.263**	**0.189**	**0.274**	**0.194**	**0.293**	**0.208**	**0.295**
	Count of Wins		5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	3/5	4/5	5/5	5/5	5/5	5/5
Traffic	Original	96	**0.420**	**0.270**	**0.396**	0.262	**0.376**	0.251	0.394	0.269	**0.450**	**0.286**	0.589	0.315	0.649	0.396
		192	0.435	0.277	**0.416**	0.272	**0.398**	**0.261**	**0.413**	**0.277**	**0.456**	**0.287**	0.614	0.323	0.599	0.371
		336	**0.450**	**0.284**	**0.433**	**0.278**	**0.415**	**0.269**	**0.423**	**0.283**	**0.471**	**0.293**	0.638	0.341	0.606	0.374
		720	0.488	**0.303**	**0.467**	**0.297**	**0.448**	**0.288**	**0.459**	**0.301**	**0.503**	**0.312**	0.664	0.353	**0.646**	0.395
		Avg	0.448	**0.284**	**0.428**	0.277	**0.409**	**0.267**	**0.422**	**0.283**	**0.470**	**0.295**	0.626	0.333	0.625	0.384
	+LDAE	96	0.427	0.280	0.418	**0.259**	0.388	**0.250**	**0.388**	**0.268**	0.486	0.287	**0.587**	**0.314**	**0.644**	**0.391**
		192	**0.428**	**0.276**	0.439	**0.270**	0.427	0.264	0.417	**0.277**	0.491	0.291	**0.613**	**0.322**	**0.598**	**0.367**
		336	**0.450**	**0.284**	0.458	**0.278**	0.438	0.274	0.438	0.284	0.507	0.296	**0.630**	**0.322**	**0.604**	**0.370**
		720	**0.482**	0.304	0.504	0.298	0.476	0.289	0.471	0.302	0.543	0.343	**0.659**	**0.346**	0.651	**0.388**
		Avg	**0.447**	0.286	0.455	**0.276**	0.432	0.269	0.429	**0.283**	0.507	0.304	**0.622**	**0.326**	**0.624**	**0.379**
	Count of Wins		4/5	2/5	0/5	4/5	0/5	1/5	1/5	3/5	0/5	0/5	5/5	5/5	4/5	5/5
Solar	Original	96	0.180	0.244	0.194	0.234	0.201	0.232	0.203	**0.235**	0.230	0.277	0.241	0.268	0.285	0.372
		192	0.182	0.248	0.227	0.269	0.230	0.255	0.238	0.262	0.260	**0.295**	0.268	0.294	0.315	0.393
		336	0.198	0.262	**0.240**	0.273	0.242	0.268	0.250	0.275	0.277	**0.306**	**0.288**	0.297	0.349	0.413
		720	**0.200**	**0.254**	**0.241**	**0.275**	0.245	0.272	0.250	0.276	0.276	0.305	0.307	**0.306**	0.354	0.410
		Avg	0.190	0.252	0.226	0.263	0.230	0.257	0.235	0.262	0.261	0.296	0.276	0.291	0.326	0.397
	+LDAE	96	**0.166**	**0.227**	**0.192**	**0.233**	**0.198**	**0.229**	**0.202**	0.236	**0.229**	**0.275**	**0.236**	**0.267**	**0.270**	**0.350**
		192	**0.181**	**0.247**	**0.226**	**0.220**	**0.228**	**0.254**	**0.236**	**0.261**	**0.258**	**0.295**	**0.259**	**0.283**	**0.291**	**0.343**
		336	**0.192**	**0.256**	**0.240**	**0.272**	**0.236**	**0.266**	**0.248**	**0.273**	**0.276**	**0.306**	**0.288**	**0.295**	**0.307**	**0.347**
		720	0.203	**0.254**	0.243	**0.275**	**0.238**	**0.268**	**0.248**	**0.275**	**0.275**	**0.304**	**0.297**	0.309	**0.304**	**0.340**
		Avg	**0.186**	**0.246**	**0.225**	**0.250**	**0.225**	**0.254**	**0.234**	**0.261**	**0.260**	**0.295**	**0.270**	**0.289**	**0.293**	**0.345**
	Count of Wins		4/5	5/5	4/5	5/5	5/5	5/5	5/5	4/5	5/5	5/5	5/5	4/5	5/5	5/5
Weather	Original	96	0.156	**0.201**	0.161	0.206	0.171	0.212	0.176	0.215	0.178	0.219	0.168	0.219	0.196	**0.255**
		192	0.204	0.247	0.208	0.251	0.222	0.257	0.226	0.259	0.225	0.259	0.224	0.264	0.239	0.297
		336	**0.263**	**0.289**	0.267	0.293	0.279	**0.298**	0.283	0.300	0.278	0.298	0.283	0.305	0.283	0.332
		720	0.339	0.341	**0.349**	**0.346**	0.357	0.350	0.359	0.351	0.350	0.346	**0.355**	**0.352**	0.348	0.384
		Avg	0.241	0.270	0.246	0.274	0.257	0.279	0.261	0.281	0.258	0.281	0.258	0.285	0.267	0.317
	+LDAE	96	**0.155**	**0.201**	**0.158**	**0.205**	**0.163**	**0.206**	**0.172**	**0.213**	**0.171**	**0.216**	**0.167**	**0.218**	**0.193**	0.260
		192	**0.202**	**0.243**	**0.206**	**0.248**	**0.215**	**0.255**	**0.212**	**0.251**	**0.217**	**0.254**	**0.222**	**0.262**	**0.232**	**0.294**
		336	**0.263**	0.290	**0.266**	**0.292**	**0.274**	**0.298**	**0.274**	**0.296**	**0.275**	**0.297**	**0.277**	**0.300**	**0.282**	**0.330**
		720	**0.338**	**0.339**	0.350	**0.346**	**0.354**	**0.349**	**0.355**	**0.349**	**0.345**	**0.345**	0.356	**0.352**	**0.346**	**0.378**
		Avg	**0.240**	**0.268**	**0.245**	**0.273**	**0.252**	**0.277**	**0.253**	**0.277**	**0.252**	**0.278**	**0.256**	**0.283**	**0.263**	**0.316**
	Count of Wins		5/5	4/5	4/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	4/5	5/5	5/5	4/5

F.2.2 SHORT-TERM FORECASTING GENERALITY RESULTS

Table 21 gives the short-term comparisons for five backbones on PEMS03, PEMS04, PEMS07, and PEMS08 at $H \in \{12, 24, 48, 96\}$. The *Count of Wins* column counts horizons, including the average, in which +LDAE is no worse than its original backbone. The pattern is positive but heterogeneous: gains are frequent for Xpatch and CASA, while the remaining backbones show stronger dataset dependence.

Table 21: Short-term generality results on the four PEMS benchmarks at four prediction horizons. Lower errors are better; bold marks the better value within each backbone comparison.

Models			Xpatch 2025		CASA 2025		SOFTS 2024		iTransformer 2024		DLinear 2023	
Metric			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	Original	12	0.084	0.191	**0.068**	**0.174**	0.067	0.171	0.069	0.175	0.108	0.220
		24	**0.126**	0.233	**0.098**	**0.209**	0.085	0.190	0.098	0.209	0.183	0.291
		48	0.219	0.310	0.169	0.278	0.114	0.231	0.166	0.277	0.302	0.392
		96	**0.321**	**0.381**	0.272	0.355	0.156	0.262	0.274	0.370	0.386	0.466
		Avg	0.188	0.279	0.152	0.254	0.105	0.213	0.152	0.258	0.244	0.342
	+LDAE	12	**0.083**	**0.189**	**0.068**	**0.174**	**0.061**	**0.161**	**0.067**	**0.173**	**0.098**	**0.212**
		24	**0.126**	**0.231**	0.099	0.210	**0.076**	**0.179**	**0.095**	**0.206**	**0.154**	**0.272**
		48	**0.218**	**0.308**	**0.165**	**0.273**	**0.099**	**0.206**	**0.141**	**0.255**	**0.269**	**0.379**
		96	0.322	0.383	**0.269**	**0.353**	**0.129**	**0.237**	**0.228**	**0.332**	**0.359**	**0.446**
		Avg	**0.187**	**0.278**	**0.150**	**0.252**	**0.091**	**0.195**	**0.132**	**0.241**	**0.220**	**0.327**
	Count of Wins		**4/5**	**4/5**	**4/5**	**4/5**	**5/5**	**5/5**	**5/5**	**5/5**	**5/5**	**5/5**
PEMS04	Original	12	0.092	0.197	0.075	0.178	0.073	0.175	0.082	0.191	0.111	0.227
		24	0.135	**0.240**	0.097	0.206	0.088	0.193	0.101	0.213	0.183	0.301
		48	0.235	0.320	0.150	0.265	0.110	0.218	0.133	0.247	0.303	0.400
		96	0.351	0.401	**0.198**	**0.308**	0.137	0.245	0.165	0.276	0.385	0.471
		Avg	0.203	0.290	0.130	0.239	0.102	0.207	0.120	0.231	0.245	0.349
	+LDAE	12	**0.091**	**0.196**	**0.061**	**0.159**	**0.071**	**0.172**	**0.079**	**0.187**	**0.105**	**0.221**
		24	**0.134**	**0.240**	**0.085**	**0.188**	**0.082**	**0.186**	**0.096**	**0.211**	**0.170**	**0.290**
		48	**0.215**	**0.302**	**0.140**	**0.243**	**0.097**	**0.204**	**0.124**	**0.241**	**0.290**	**0.394**
		96	**0.314**	**0.368**	0.228	0.320	**0.116**	**0.224**	**0.146**	**0.265**	**0.376**	**0.467**
		Avg	**0.189**	**0.277**	**0.128**	**0.227**	**0.091**	**0.196**	**0.111**	**0.226**	**0.235**	**0.343**
	Count of Wins		**5/5**	**5/5**	**4/5**	**4/5**	**5/5**	**5/5**	**5/5**	**5/5**	**5/5**	**5/5**
PEMS07	Original	12	**0.077**	**0.180**	**0.061**	**0.159**	0.057	0.151	0.066	0.164	0.100	0.218
		24	**0.123**	**0.227**	0.086	0.189	0.085	0.188	0.087	0.190	0.186	0.303
		48	0.214	0.301	0.143	0.246	0.103	0.206	0.229	0.328	0.354	0.443
		96	0.325	0.376	0.261	**0.331**	0.125	0.224	**0.322**	**0.384**	0.466	0.521
		Avg	0.185	**0.271**	0.137	**0.231**	0.092	0.192	0.176	0.266	0.276	0.371
	+LDAE	12	0.078	0.181	**0.061**	0.160	**0.056**	**0.148**	**0.064**	**0.161**	**0.086**	**0.201**
		24	**0.123**	**0.227**	**0.084**	**0.187**	**0.065**	**0.160**	**0.082**	**0.184**	**0.144**	**0.266**
		48	**0.213**	**0.299**	**0.138**	**0.242**	**0.079**	**0.171**	**0.200**	**0.295**	**0.257**	**0.365**
		96	**0.321**	**0.375**	**0.258**	0.338	**0.092**	**0.181**	0.331	0.403	**0.365**	**0.453**
		Avg	**0.184**	**0.271**	**0.135**	**0.231**	**0.073**	**0.165**	**0.169**	**0.260**	**0.213**	**0.321**
	Count of Wins		**4/5**	**4/5**	**5/5**	**3/5**	**5/5**	**5/5**	**4/5**	**4/5**	**5/5**	**5/5**
PEMS08	Original	12	**0.082**	**0.184**	**0.077**	**0.181**	0.074	0.171	0.088	0.193	0.102	0.217
		24	0.123	0.224	0.122	0.227	0.105	0.201	0.136	0.242	0.169	0.287
		48	0.202	0.292	0.200	**0.286**	0.165	0.254	0.240	0.279	0.295	0.399
		96	0.315	0.368	0.381	0.395	0.215	0.254	**0.320**	**0.343**	0.387	0.477
		Avg	0.181	0.267	0.195	0.272	0.139	0.220	0.196	**0.264**	0.238	0.345
	+LDAE	12	**0.082**	0.185	0.078	0.182	**0.072**	**0.170**	**0.087**	**0.192**	**0.097**	**0.213**
		24	**0.121**	**0.223**	**0.118**	**0.224**	**0.094**	**0.192**	**0.131**	**0.239**	**0.160**	**0.281**
		48	**0.196**	**0.287**	**0.193**	0.287	**0.137**	**0.233**	**0.218**	**0.269**	**0.277**	**0.390**
		96	**0.310**	**0.365**	**0.372**	**0.391**	**0.183**	**0.222**	0.366	0.369	**0.370**	**0.469**
		Avg	**0.177**	**0.265**	**0.190**	**0.271**	**0.121**	**0.204**	**0.194**	0.267	**0.226**	**0.338**
	Count of Wins		**5/5**	**4/5**	**4/5**	**3/5**	**5/5**	**5/5**	**4/5**	**3/5**	**5/5**	**5/5**

F.3 Stability Analysis

We assess the empirical stability of the LDAE enhancement using an exploratory finite-sample protocol. Each model configuration is executed 3 times with independent random seeds to characterize run-to-run variation in MSE and MAE, rather than to provide confirmatory population-level inference (Wasserman, 2004). Evaluation covers four canonical horizons for long-term datasets ($H \in \{96, 192, 336, 720\}$) and four horizons for short-term PEMS tasks ($H \in \{12, 24, 48, 96\}$), yielding 12 measurements per model-dataset pair.

We summarize within-horizon variability using the pooled standard deviation:

$$\overline{\mathrm{SD}} = \sqrt{\frac{\sum_{i=1}^{4}\sum_{j=1}^{3}(x_{ij} - \bar{x}_i)^2}{4 \times (3-1)}}, \tag{14}$$

where x_{ij} is the metric observed at repetition j for horizon i, and $\bar{x}_i$ is the horizon-specific mean. This estimator ($\mathrm{df} = 8$) provides a comparable measure of run-to-run variability across prediction lengths and is reported together with the mean in the format "mean (pooled SD)" (Montgomery, 2017).

For exploratory assessment of whether observed mean differences are distinguishable from horizon effects, we fit a blocked linear model and apply Type II ANOVA with prediction length as a blocking factor:

$$y_{ijk} = \mu + \alpha_k + \beta_i + \varepsilon_{ijk},$$

where y_{ijk} denotes the metric for model variant k at horizon i and repetition j, α_k is the model-type effect, and β_i is the block effect. We obtain an F-statistic for the model-type effect and report its p-value. Under the null hypothesis that LDAE has no effect, the p-value is the probability of obtaining a test statistic at least as extreme as the observed statistic. A lower p-value therefore provides stronger evidence against the null hypothesis. All dataset/metric p-values are corrected for multiple comparisons using the Benjamini–Hochberg FDR procedure and evaluated at $\alpha = 0.05$ (Benjamini & Hochberg, 1995).

The exploratory workflow is therefore: (1) collect 3 repetitions per horizon (12 samples per model-dataset pair); (2) compute horizon-specific means $\bar{x}_i$; (3) compute $\overline{\mathrm{SD}}$; and (4) apply the blocked ANOVA and inspect FDR-adjusted differences. We emphasize that $n = 3$ follows common practice and provides only a limited estimate of sampling variability and statistical power. To complement the exploratory comparison, we report a standardized effect size and a 95% confidence interval for each backbone-to-LDAE mean difference. The effect size is the mean difference divided by the pooled standard deviation, $d = (\bar{x}_{\text{base}} - \bar{x}_{\text{LDAE}})/s_{\text{pooled}}$, with $s^2_{\text{pooled}} = (s^2_{\text{base}} + s^2_{\text{LDAE}})/2$, and the 95% interval for the raw mean difference is $\Delta \pm t_{0.975,\mathrm{df}}\, s_{\text{pooled}}\sqrt{2/3}$ with $\mathrm{df} = 8$. For example, the SOFTS ECL MSE reduction of $0.1747 \to 0.1695$ corresponds to $d \approx 3.7$. Because only three repetitions are available, these intervals and adjusted p-values are exploratory and should not be interpreted as definitive evidence.

Tables 22, 23, and 24 summarize the aggregated results. In brief:

1. **Reproducibility.** For most long-horizon experiments, pooled SDs are small (e.g., $\overline{sd}_{\text{MSE}} \approx 0.0015$–$0.0045$ on ETT), indicating low run-to-run variability in the reported runs. This pattern is consistent with, but does not prove, that the observed improvements are not solely artifacts of random initialization.
2. **Selective efficacy.** The effect of LDAE is selective rather than uniform. In this exploratory analysis, the largest FDR-adjusted differences occur for SOFTS on ECL, Solar, Weather, and the four PEMS datasets (e.g., ECL MSE $0.1747 \to 0.1695$, $d = 3.72$, adjusted $p < 0.001$), and for CASA and iTransformer on PEMS04. In contrast, most SimpleTM and CASA long-term gains are small, so the benefit is not uniform across backbones.
3. **Architecture sensitivity.** The effect differs markedly across backbones. SOFTS shows the most consistent FDR-adjusted differences in the reported runs on ECL, Solar, Weather, and PEMS, whereas the SimpleTM and CASA gains on long-term datasets are mostly small and not consistently distinguishable in this exploratory analysis. This indicates that the marginal value of LDAE depends on whether the backbone already captures the local dynamics that LDAE injects.

4. **Dataset-specific performance.** The clearest boundary is Traffic, where the reported FDR-adjusted differences indicate degradation for SOFTS, CASA, and SimpleTM (e.g., SOFTS MSE $0.4093 \rightarrow 0.4321$, $d = -5.68$, adjusted $p < 0.001$), consistent with the noise-amplification limitation of finite differencing. On low-to-moderate-noise, non-stationary datasets such as ECL, Solar, and Weather, the SOFTS differences are positive in the reported runs.
5. **Short-term/high-variance cases.** For short-horizon PEMS tasks, many model-dataset combinations show FDR-adjusted differences in the reported runs. In several high-variance baselines, the absolute improvements are larger. High-variance sequences may contain more abrupt spikes and asymmetric fluctuations—scenarios where LDAE's derivative features can play a more prominent role. This observation is descriptive rather than causal.

In summary, the stability experiments provide an exploratory assessment of run-to-run variation. LDAE shows large FDR-adjusted differences for SOFTS on ECL, Solar, Weather, and the PEMS benchmarks in the reported runs, while its gains on most SimpleTM and CASA long-term settings are small. The clearest boundary is Traffic, where finite differencing amplifies high-frequency noise and the reported results degrade for all three backbones. These results support interpreting LDAE as a complementary prior rather than a universal booster: it appears most useful when the backbone lacks explicit local-dynamics cues and the signal is not dominated by noise.

Table 22: Exploratory stability statistics for the ETT benchmarks. Entries report pooled SDs, standardized effects, confidence intervals, and FDR-adjusted p-values; the sign of d indicates whether LDAE lowers the error. Bold green/red effects mark significant improvements/degradations.

Dataset	Model	MSE (Pooled SD)	MAE (Pooled SD)	MSE d [95% CI]	MAE d [95% CI]	MSE adj p	MAE adj p
ETTh1	SOFTS (Baseline)	0.4514 (0.0046)	0.4455 (0.0035)	-0.42 [-0.021, 0.014]	-0.40 [-0.014, 0.009]	0.475	0.483
	SOFTS+LDAE	0.4552 (0.0123)	0.4480 (0.0081)				
	SimpleTM (Baseline)	0.4271 (0.0035)	0.4320 (0.0023)	+0.61 [-0.004, 0.009]	+0.65 [-0.003, 0.006]	0.337	0.292
	SimpleTM+LDAE	0.4250 (0.0033)	0.4304 (0.0027)				
	CASA (Baseline)	0.4423 (0.0021)	0.4399 (0.0016)	-0.17 [-0.008, 0.006]	-0.32 [-0.006, 0.004]	0.745	0.547
	CASA+LDAE	0.4429 (0.0048)	0.4407 (0.0034)				
ETTh2	SOFTS (Baseline)	0.3803 (0.0041)	0.4047 (0.0028)	-0.51 [-0.015, 0.009]	-0.74 [-0.009, 0.004]	0.450	0.292
	SOFTS+LDAE	0.3836 (0.0080)	0.4073 (0.0040)				
	SimpleTM (Baseline)	0.3550 (0.0027)	0.3915 (0.0025)	-0.93 [-0.006, 0.002]	-0.48 [-0.005, 0.003]	0.120	0.475
	SimpleTM+LDAE	0.3570 (0.0015)	0.3925 (0.0014)				
	CASA (Baseline)	0.3806 (0.0012)	0.4059 (0.0010)	+0.10 [-0.004, 0.005]	+0.15 [-0.003, 0.003]	0.845	0.785
	CASA+LDAE	0.3803 (0.0032)	0.4057 (0.0018)				
ETTm1	SOFTS (Baseline)	0.3972 (0.0063)	0.4044 (0.0028)	+1.05 [-0.004, 0.015]	**+1.16** [-0.002, 0.007]	0.068	**0.037**
	SOFTS+LDAE	0.3918 (0.0037)	0.4017 (0.0017)				
	SimpleTM (Baseline)	0.3830 (0.0006)	0.3963 (0.0005)	+0.57 [-0.003, 0.006]	+0.32 [-0.003, 0.005]	0.292	0.547
	SimpleTM+LDAE	0.3816 (0.0033)	0.3956 (0.0031)				
	CASA (Baseline)	0.3962 (0.0093)	0.4033 (0.0044)	+0.58 [-0.011, 0.021]	+0.57 [-0.005, 0.010]	0.299	0.306
	CASA+LDAE	0.3912 (0.0079)	0.4010 (0.0038)				
ETTm2	SOFTS (Baseline)	0.2874 (0.0040)	0.3304 (0.0031)	+0.08 [-0.006, 0.006]	+0.20 [-0.004, 0.005]	0.871	0.738
	SOFTS+LDAE	0.2871 (0.0024)	0.3299 (0.0013)				
	SimpleTM (Baseline)	0.2799 (0.0031)	0.3246 (0.0015)	+0.25 [-0.006, 0.008]	+0.24 [-0.003, 0.004]	0.713	0.738
	SimpleTM+LDAE	0.2790 (0.0041)	0.3242 (0.0022)				
	CASA (Baseline)	0.2798 (0.0084)	0.3239 (0.0039)	-0.00 [-0.012, 0.012]	-0.30 [-0.007, 0.005]	0.999	0.586
	CASA+LDAE	0.2798 (0.0027)	0.3248 (0.0022)				

Table 23: Exploratory stability statistics for ECL, Traffic, Solar, and Weather. Column definitions and the FDR procedure are given in the preceding stability discussion.

Dataset	Model	MSE (Pooled SD)	MAE (Pooled SD)	MSE d [95% CI]	MAE d [95% CI]	MSE adj p	MAE adj p
ECL	SOFTS (Baseline)	0.1747 (0.0017)	0.2652 (0.0015)	**+3.72** [0.003, 0.008]	**+2.59** [0.001, 0.005]	< 0.001	< 0.001
	SOFTS+LDAE	0.1695 (0.0010)	0.2620 (0.0007)				
	SimpleTM (Baseline)	0.1734 (0.0189)	0.2667 (0.0164)	+0.51 [-0.018, 0.032]	+0.52 [-0.016, 0.028]	0.361	0.361
	SimpleTM+LDAE	0.1665 (0.0026)	0.2606 (0.0024)				
	CASA (Baseline)	0.1729 (0.0036)	0.2668 (0.0023)	+0.74 [-0.004, 0.010]	+0.81 [-0.003, 0.007]	0.292	0.169
	CASA+LDAE	0.1702 (0.0038)	0.2647 (0.0029)				
Traffic	SOFTS (Baseline)	0.4093 (0.0008)	0.2673 (0.0004)	**-5.68** [-0.030, -0.015]	**-1.28** [-0.005, 0.001]	< 0.001	**0.042**
	SOFTS+LDAE	0.4321 (0.0056)	0.2695 (0.0024)				
	SimpleTM (Baseline)	0.4486 (0.0018)	0.2842 (0.0015)	-2.60 [-0.010, -0.002]	**-5.07** [-0.011, -0.005]	0.120	**0.003**
	SimpleTM+LDAE	0.4546 (0.0027)	0.2923 (0.0017)				
	CASA (Baseline)	0.4277 (0.0043)	0.2779 (0.0009)	**-5.61** [-0.039, -0.020]	+0.65 [-0.001, 0.003]	< 0.001	0.361
	CASA+LDAE	0.4571 (0.0060)	0.2773 (0.0011)				
Solar	SOFTS (Baseline)	0.2296 (0.0015)	0.2569 (0.0012)	**+2.16** [0.001, 0.008]	+0.94 [-0.002, 0.007]	**0.001**	0.069
	SOFTS+LDAE	0.2252 (0.0024)	0.2545 (0.0035)				
	SimpleTM (Baseline)	0.1897 (0.0045)	0.2504 (0.0069)	+0.21 [-0.009, 0.011]	+0.11 [-0.012, 0.013]	0.745	0.845
	SimpleTM+LDAE	0.1886 (0.0061)	0.2497 (0.0062)				
	CASA (Baseline)	0.2258 (0.0018)	0.2611 (0.0011)	-0.24 [-0.003, 0.003]	-0.06 [-0.002, 0.002]	0.655	0.904
	CASA+LDAE	0.2262 (0.0014)	0.2611 (0.0005)				
Weather	SOFTS (Baseline)	0.2572 (0.0015)	0.2795 (0.0009)	**+2.56** [0.001, 0.010]	**+1.47** [-0.001, 0.006]	< 0.001	**0.022**
	SOFTS+LDAE	0.2517 (0.0027)	0.2769 (0.0023)				
	SimpleTM (Baseline)	0.2421 (0.0013)	0.2709 (0.0007)	-0.06 [-0.002, 0.002]	-0.46 [-0.003, 0.002]	0.915	0.521
	SimpleTM+LDAE	0.2422 (0.0012)	0.2714 (0.0013)				
	CASA (Baseline)	0.2465 (0.0021)	0.2741 (0.0016)	+0.33 [-0.003, 0.005]	+0.27 [-0.002, 0.003]	0.547	0.628
	CASA+LDAE	0.2459 (0.0020)	0.2737 (0.0013)				

Table 24: Exploratory stability statistics for the short-term PEMS benchmarks. The table applies the same effect-size and FDR-adjusted comparison used for the long-term results.

Dataset	Model	MSE (Pooled SD)	MAE (Pooled SD)	MSE d [95% CI]	MAE d [95% CI]	MSE adj p	MAE adj p
PEMS03	SOFTS (Baseline) SOFTS+LDAE	0.1061 (0.0024) 0.0949 (0.0029)	0.2121 (0.0039) 0.2009 (0.0029)	**+4.24** [0.006, 0.016]	**+3.26** [0.005, 0.018]	< 0.001	< 0.001
	iTransformer (Baseline) iTransformer+LDAE	0.3256 (0.3117) 0.2869 (0.0956)	0.3516 (0.1429) 0.3352 (0.0461)	+0.17 [-0.395, 0.473]	+0.15 [-0.184, 0.216]	0.745	0.761
	CASA (Baseline) CASA+LDAE	0.1525 (0.0022) 0.1512 (0.0009)	0.2550 (0.0012) 0.2536 (0.0010)	+0.76 [-0.002, 0.005]	**+1.28** [-0.001, 0.004]	0.161	**0.038**
PEMS04	SOFTS (Baseline) SOFTS+LDAE	0.1024 (0.0006) 0.0923 (0.0009)	0.2079 (0.0004) 0.1973 (0.0007)	**+13.34** [0.009, 0.012]	**+18.10** [0.010, 0.012]	< 0.001	< 0.001
	iTransformer (Baseline) iTransformer+LDAE	0.1211 (0.0021) 0.1122 (0.0009)	0.2321 (0.0020) 0.2255 (0.0011)	**+5.43** [0.006, 0.012]	**+4.02** [0.003, 0.010]	**0.001**	**0.004**
	CASA (Baseline) CASA+LDAE	0.1392 (0.0024) 0.1316 (0.0022)	0.2338 (0.0026) 0.2287 (0.0023)	**+3.31** [0.003, 0.012]	**+2.10** [0.001, 0.010]	**0.005**	**0.001**
PEMS07	SOFTS (Baseline) SOFTS+LDAE	0.0885 (0.0020) 0.0723 (0.0006)	0.1859 (0.0042) 0.1640 (0.0010)	**+10.89** [0.013, 0.019]	**+7.19** [0.016, 0.028]	< 0.001	< 0.001
	iTransformer (Baseline) iTransformer+LDAE	0.4279 (0.2498) 0.2119 (0.2372)	0.4282 (0.1589) 0.2822 (0.1657)	+0.89 [-0.243, 0.675]	+0.90 [-0.160, 0.452]	0.124	0.120
	CASA (Baseline) CASA+LDAE	0.1392 (0.0024) 0.1339 (0.0055)	0.2338 (0.0026) 0.2311 (0.0043)	+1.26 [-0.003, 0.013]	+0.76 [-0.004, 0.009]	0.059	0.169
PEMS08	SOFTS (Baseline) SOFTS+LDAE	0.1408 (0.0024) 0.1214 (0.0093)	0.2218 (0.0035) 0.2046 (0.0110)	**+2.86** [0.007, 0.032]	**+2.12** [0.002, 0.033]	< 0.001	**0.002**
	iTransformer (Baseline) iTransformer+LDAE	0.2441 (0.0390) 0.2284 (0.0402)	0.3017 (0.0346) 0.2899 (0.0320)	+0.40 [-0.059, 0.090]	+0.35 [-0.051, 0.075]	0.475	0.521
	CASA (Baseline) CASA+LDAE	0.1930 (0.0062) 0.1906 (0.0074)	0.2742 (0.0053) 0.2720 (0.0052)	+0.34 [-0.011, 0.015]	+0.42 [-0.008, 0.012]	0.536	0.475

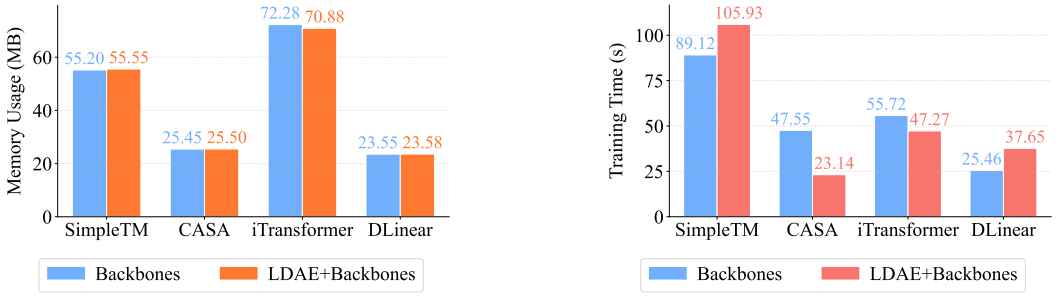
G Efficiency Analysis

We measure training time and peak GPU memory for four representative backbones (SimpleTM, CASA, iTransformer, and DLinear) across seven datasets (Figures 14 and 15). The additional GPU memory is generally modest, including on the high-dimensional Traffic dataset with 862 channels. Training-time changes vary across backbones and datasets: heavier backbones often show small relative changes, whereas lightweight models can exhibit larger percentage increases despite small absolute increments. In a few cases, such as SimpleTM on ECL, the measured training time decreases slightly; because the experiments use a fixed epoch budget, we treat these decreases as runtime variation rather than evidence that LDAE requires fewer optimization steps.

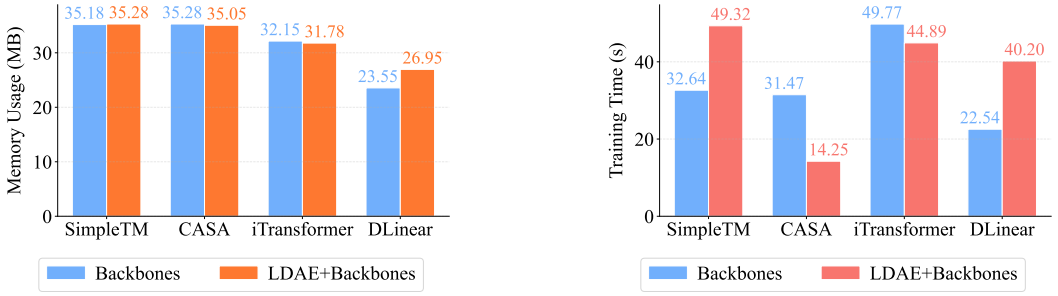
The grouped convolution introduced by LDAE has complexity $O(CLK)$, which is linear in the number of channels C, the lookback length L, and the small kernel size K. It therefore preserves linear scaling with sequence length and adds no higher-order asymptotic penalty. Heavy backbones such as CASA and iTransformer can absorb the extra convolution with relatively small overhead, while lightweight models such as SimpleTM and DLinear may show larger relative time increases despite small absolute increments. Overall, LDAE provides a favorable accuracy–efficiency trade-off with a modest additional computational cost.

H Showcases

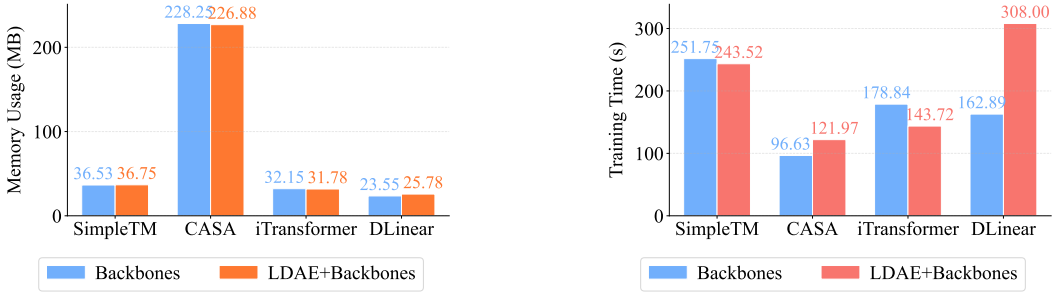
This section presents visualizations of prediction trajectories to compare the forecasting performance of LDAE-augmented models with baseline models. The results are categorized into two groups, with specific visualization rules for time-step display to ensure clarity: Long-term forecasting results of the SimpleTM model on eight datasets (ETTh1, ETTh2, ETTm1, ETTm2, Weather, Solar, ECL, Traffic), using a lookback window of 96 and prediction horizons of 96, 192, 336, and 720 (Figures 16 and 17). For time-step visualization: The ground-truth values, baseline model predictions, and LDAE-augmented model predictions are displayed simultaneously to enable direct and intuitive comparison. Short-term forecasting results of the CASA model on four PEMS datasets (PEMS03, PEMS04, PEMS07, and PEMS08) use a lookback window of 12 and prediction horizons of 12, 24, 48, and 96 (Figure 18). The ground-truth values, baseline predictions, and LDAE-augmented predictions are displayed simultaneously. All plots share a unified time-step axis for consistent comparison across different forecasting scenarios. Additionally, the y-axis ticks are simplified to enhance readability without compromising the integrity of the data representation.



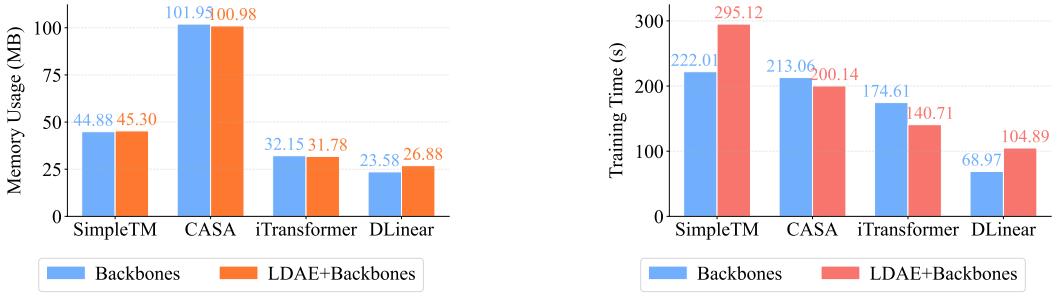
(a) ETTh1



(b) ETTh2

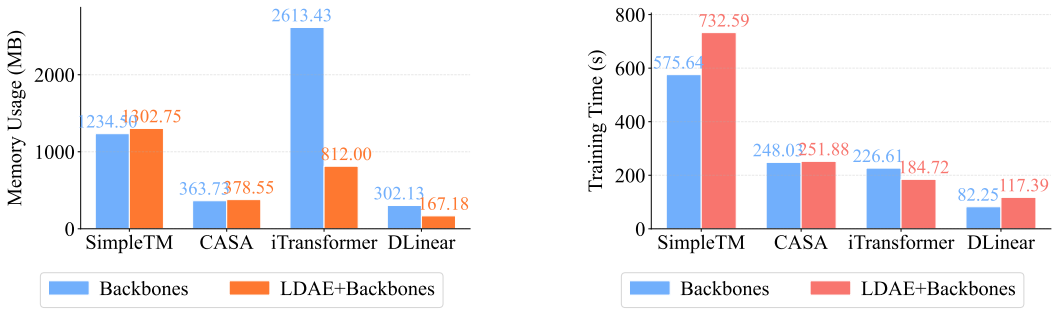


(c) ETTm1

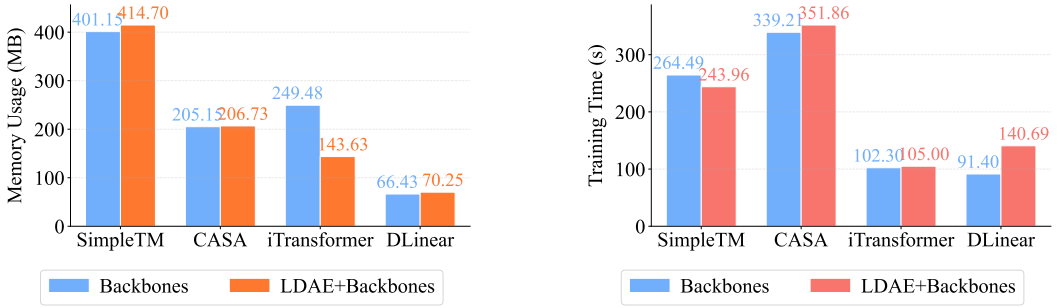


(d) ETTm2

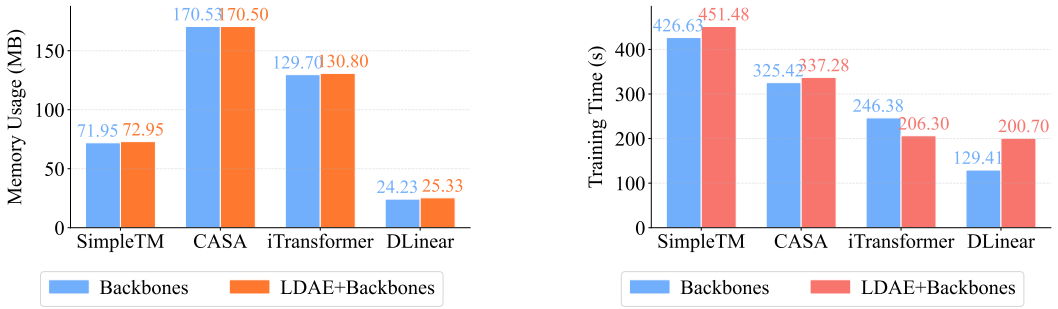
Figure 14: Training time and memory usage of mainstream forecasting backbones and their LDAE-augmented variants on the ETT datasets across multiple prediction horizons. All results are averaged over five random runs. Lower values indicate better efficiency.



(a) Traffic



(b) Solar



(c) Weather

Figure 15: Training time and memory usage of mainstream forecasting backbones and their LDAE-augmented variants on Traffic, Solar, and Weather across multiple prediction horizons. All results are averaged over five random runs. Lower values indicate better efficiency.

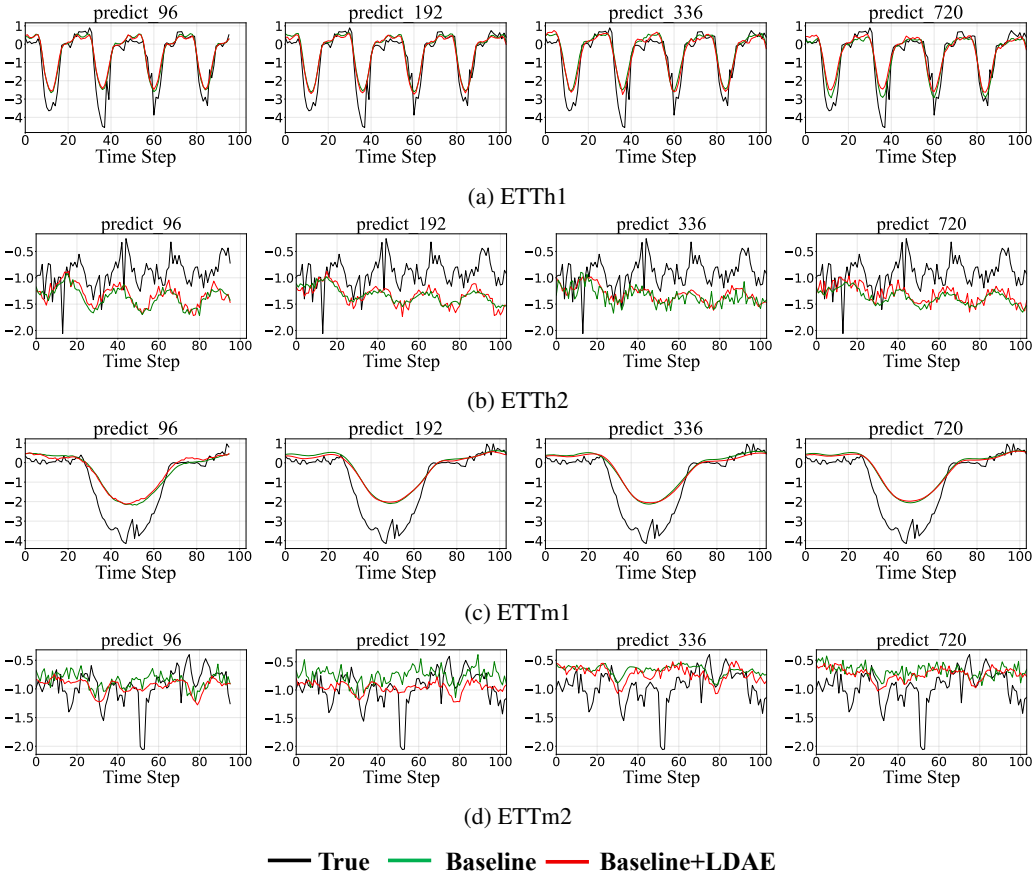


Figure 16: Long-term prediction results of SimpleTM under the lookback window of 96.

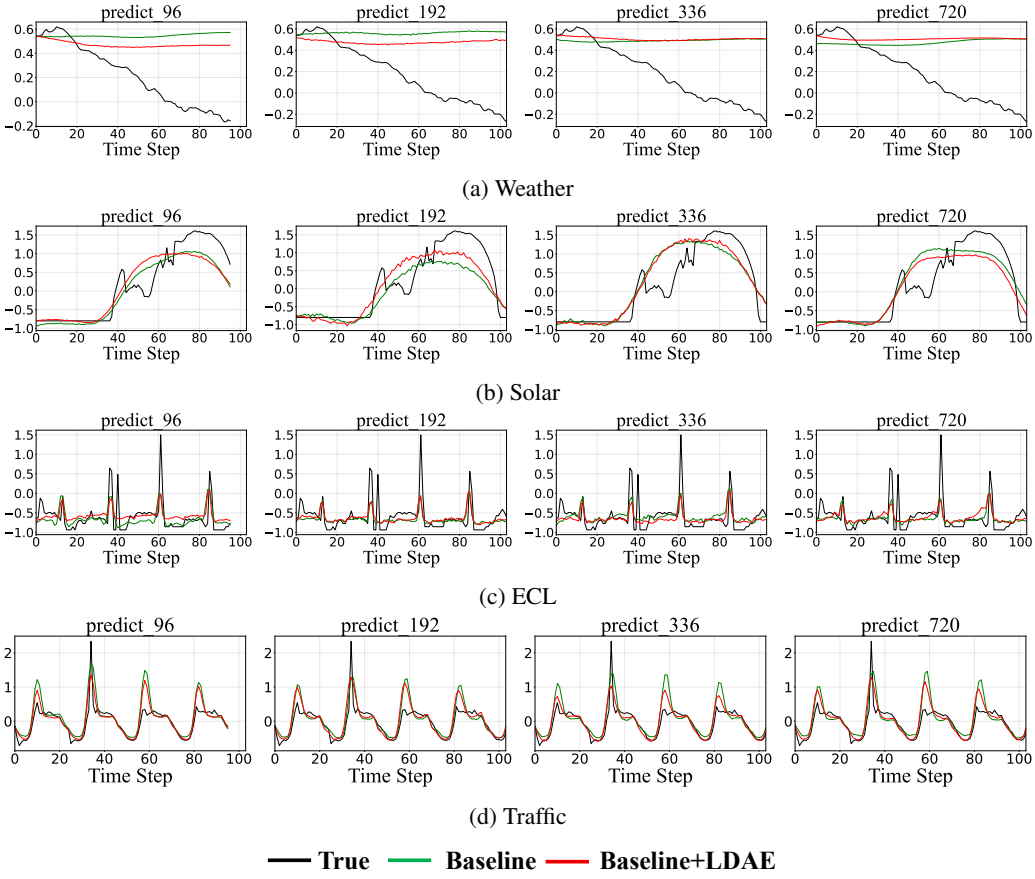


Figure 17: Long-term prediction results of SimpleTM under the lookback window of 96.

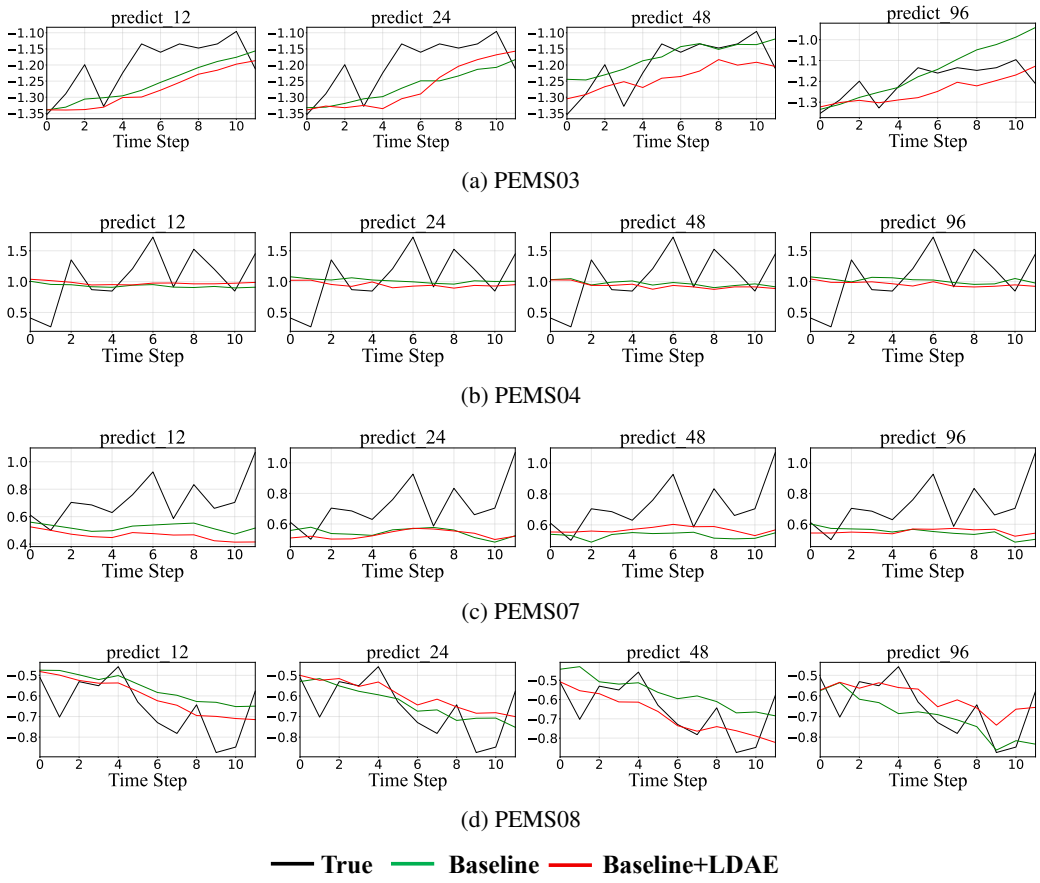


Figure 18: Short-term prediction results of CASA with a lookback window of 12.