SHARED-MEMORY AXIS INVERSION FOR LONG-TERM TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate long-term series forecasting (LTSF) involves both temporal and variate axes, whereas efficient sequence models typically model only one axis explicitly. We characterize the resulting resolution–efficiency trade-off in terms of native-step alignment and cross-axis coupling cost, and formulate Step-indexed Completion (SInC) as a design criterion. To realize SInC, we propose DAMIE, which models the two axes sequentially: it first builds shared temporal memory indexed by native time steps, then uses Flip Attention to form a variate-to-time alignment distribution for each target variate and read out its representation from the memory. Across 11 datasets from five benchmark families, DAMIE achieves strong forecasting performance and a favorable accuracy–efficiency trade-off, particularly in settings with many variates. Further analyses indicate that step-indexed cross-axis readout helps target variates select relevant historical information and connect temporal and variate representations. Code is available at `https://anonymous.4open.science/r/ICLR-27-16E1`

1 INTRODUCTION

Multivariate LTSF takes a lookback window $X \in \mathbb{R}^{L\times N}$, namely L past time steps of N coupled variates, and predicts the next P steps $\hat{Y} \in \mathbb{R}^{P\times N}$ (Lim & Zohren, 2021; Wen et al., 2023; Qiu et al., 2024), where P is the forecast horizon. When P is large, accurate forecasting requires identifying the variables and specific historical time steps relevant to a given prediction target (Lim et al., 2021). As illustrated in Fig. 1 (a), predicting variate B may rely on different lags of variate A across samples (e.g., $\ell = 2$ or $\ell = 6$). Such sample-dependent lag behavior naturally arises when effects are delayed (Wu et al., 2021; Liu et al., 2024) and the system is nonstationary (Kim et al., 2022). The key insight is that scalable LTSF should preserve STEP-INDEXED ALIGNMENT for each prediction target. This means the model can assign distinct importance to individual time steps (and source variates) for each target, rather than assuming a single fixed lag works for all samples, or merging multiple steps so the model cannot distinguish which step provides the evidence.

However, preserving such target-specific dependencies requires joint modeling across the temporal and variate axes of the $L \times N$ field. 2D networks can effectively capture local patterns, but often struggle to model long-term dependencies (Yi et al., 2023; Wang et al., 2023a). In contrast, traditional sequence models (Vaswani et al., 2017) operate along a single token axis and therefore cannot directly preserve fine-grained structure along both axes. As shown in Fig. 1(b), single-axis formulations typically encode one axis into token features, sacrificing its explicit resolution (Zhou et al., 2021; Liu et al., 2024). To further model 2D fields, some methods model the two axes separately (Zhang & Yan, 2023), whereas others reorganize the field into one or more 1D sequences of temporal units (Nie et al., 2023; Cai et al., 2026; Wang et al., 2024a). These approaches preserve information along both axes but incur higher computational and memory costs. Therefore, this contradiction between single-axis models and dual-axis data structures raises a trade-off question: *how much fine-grained alignment can be retained under the single-axis budget?*

To address this trade-off, dual-axis dependencies in the $L \times N$ domain must be captured under a *single-axis budget*, precluding the explicit maintenance of a dense $L \times N$ token grid. We therefore model the two axes sequentially, requiring axis transition to satisfy two conditions. First, it must be *step-indexed*, directly addressing native indices along the current explicit axis, such as individual time steps in a time-major view, rather than compression-induced coarse units (Nie et al., 2023).

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



Figure 1: The resolution-efficiency trade-off of the representational mismatch.

Second, it must be *budget-admissible*, avoiding the maintenance or repeated update of a dense two-dimensional token grid (Zhang & Yan, 2023). Sequential modeling need to transfer information from one axis to the next while preserving fine-grained alignment at the native time-step level. To characterize this transition, we introduce *Step-Indexed Completion (*SInC*)*, which represents each transition using three components: an indexable 1D *view* of tokens along the current explicit axis, *transition weights* linking each target representation to indices in this view, and a *readout* that aggregates the indexed information. The indexable view and transition weights preserve step-level access, while the controlled readout keeps the transition within the single-axis budget.

We propose DAMIE, a *dual-axis memory inversion encoder*, a shared-memory axis-inversion design that realizes SInC under *single-axis budget*. DAMIE first encodes the $L \times N$ input into a temporal representation indexed by native time steps, which directly serves as shared memory for all N target variates rather than requiring a standalone memory module. On top of this memory, we model the two axes in sequence and introduce *Flip Attention* to invert the explicit indexing axis from time to variates. As shown in Fig 1 (c), cross-axis information transfer is represented by step-indexed coupling weights over native time indices,without forming an $L \times N$ grid of high-dimensional. Finally, forecasts are decoded from the variate tokens on variate-level features. Contributions are:

- **Characterizing the dual-axis trade-off.** We characterize dual-axis dependency modeling in multivariate LTSF through the trade-off between alignment resolution and cross-axis coupling cost, and use it to identify the limitations of existing modeling strategies.
- **Formulating Step-Indexed Completion.** We introduce *SInC*, which describes cross-axis information transfer through a view–weights–readout interface. Its native-step indexing and budget-admissible coupling constraints turn the trade-off into explicit design criteria.
- **Developing a shared-memory axis-inversion encoder.** We model the temporal and variate axes in sequence using shared temporal memory and Flip Attention, realizing *SInC* through a single native-step cross-axis readout within a single-axis budget.
- **Providing empirical evidence.** The method outperforms multiple Transformer baselines and balances accuracy with efficiency in high-variate settings. Ablation and lookback analyses examine cross-axis readout and the use of longer histories.

2 Problem Formalization

LTSF maps a lookback window $X \in \mathbb{R}^{L \times N}$ to $\hat{Y} \in \mathbb{R}^{P \times N}$. A full-resolution idealization allows each forecast site (p, n) to aggregate the native two-axis history field using data-dependent weights:

$$Z_{p,n}(X) = \sum_{t=1}^{L} \sum_{m=1}^{N} K_{p,n}(t, m; X)\, \psi(X_{t,m}), . \tag{1}$$

where, ψ denotes a feature map, $K_{p,n}(t, m; X)$ is a data-dependent coupling kernel that measures the contribution of the historical site (t, m) to the forecast site (p, n), thereby accommodating dynamic dependencies. However, explicitly materializing an $L \times N$ token grid and repeatedly mixing information across both axes is computationally expensive, as detailed in the appendix E table 12. Therefore, we focus on the practical setting of a "*single-axis budget*," where the model achieves dual-axis modeling by maintaining only a 1-D token set of size $O(L)$ or $O(N)$. Under this constraint, the central question thus becomes: *How can cross-axis information be transferred within a one-axis budget while preserving fine-grained alignment at the native time-step level?*

2.1 STEP-INDEXED COMPLETION

To formalize native-step alignment under a one-axis budget, we introduce *Step-Indexed Completion (SInC)*. Let $\mathcal{I}$ denote the native index set of the explicitly modeled axis, $\mathcal{S}$ the index set of the maintained 1-D view, and $\mathcal{U}$ the target index set on the other axis. For "time-major" completion, $\mathcal{I} = \{1, \ldots, L\}$ and $\mathcal{U} = \{1, \ldots, N\}$ index native time steps and target variates, respectively. The encoder can be represented by the following interface:

$$\textbf{(view)} \quad H = V_\theta(X) \in \mathbb{R}^{|\mathcal{S}| \times D}, \tag{2}$$

$$\textbf{(weights)} \quad \pi_u = \Pi_\phi(X, u) \in \Delta^{|\mathcal{S}|}, \tag{3}$$

$$\textbf{(readout)} \quad z_u = \mathcal{A}(H, \pi_u). \tag{4}$$

where H is an explicitly indexable 1D view over $\mathcal{S}$ and π_u assigns per-variate weights over its indices. A canonical readout is $z_u = \sum_{s \in \mathcal{S}} \pi_u(s)\rho(H_s)$ for a token-wise map ρ (possibly identity).

Definition 1 (Step-Indexed Completion (SInC))**.** A forecaster follows *SInC* if it admits the interface in Eqs. 2–4 and satisfies the following conditions:

(A) Native-step indexing. The view index set $\mathcal{S}$ corresponds to the native index set $\mathcal{I}$, i.e., $\mathcal{S} \equiv \mathcal{I}$ up to relabeling, allowing π_u to assign distinct weights to individual native steps.

(B) Budget-admissible coupling. Let C_{ax} denote the dominant cross-axis interactions needed to compute $\{\pi_u\}_{u \in \mathcal{U}}$ and apply readout. *SInC* requires one-/few-pass coupling: $C_{\text{ax}} = O(|\mathcal{U}| \cdot |\mathcal{S}|)$, i.e., cross-axis transfer scales with $|\mathcal{U}||\mathcal{S}|$ rather than repeated dense mixing over $L \times N$ grid.

Why SInC. Condition (A) excludes coarse indexing, which can conflate distinct lags and lead to non-vanishing error (Thm. A.1 in App A.1), while Condition (B) excludes repeated 2D mixing over an explicit $L \times N$ token grid. Under these constraints, cross-axis alignment is mediated by Eq. 3. Adapting to switching evidence requires π_u to depend on the input X (Thm. A.2 in App A.2), while supporting variate-specific alignment requires it to depend on the target u (Thm. A.3 in App A.3).

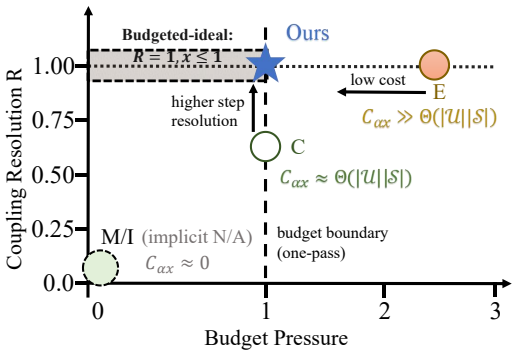

Figure 2: Budget-resolution trade-off.

2.2 DESIGN TRADE-OFFS UNDER SINC

Figure 2 characterizes the trade-off between cross-axis coupling burden and alignment resolution. The horizontal axis measures the coupling burden: $\kappa_{\text{ax}} = \log_2\big(1 + C_{\text{ax}}/(|\mathcal{U}||\mathcal{S}|)\big)$, where C_{ax} counts cross-axis interactions induced by constructing/applying alignment weights (App. B); $\kappa_{\text{ax}} = 1$ represents the budget boundary at which each target-index pair interacts once. The vertical axis measures the alignment resolution $R = |\mathcal{S}|/|\mathcal{I}|$, where $R = 1$ indicates that the alignment weights are defined over individual native steps, while $R < 1$ indicates coarse indexing.

Implicit-mixing backbones (M/I) lack explicit alignment weights, and resolution is marked as N/A. Coarse methods (C) compress the temporal index, reducing the resolution to $R < 1$. Even when their normalized coupling burden remains near the reference level, maintaining Q temporal patches per variate requires NQ high-dimensional tokens and thus scales with both axes. In contrast, 2D-intensive coupling methods (E) can preserve native-step resolution ($R = 1$), but dense or repeated 2D interactions incur a greater coupling burden. Therefore, the shaded region defined by $R = 1$ and $\kappa_{\text{ax}} \leq 1$ marks the desired trade-off, with $(1, 1)$ denoting a full native-step coupling pass.

Definition 2 (Resolution–Efficiency Trade-off Objective)**.** Under a single-axis high-dimensional token budget, the objective is to preserve as much native time-step resolution as possible in cross-axis readout while controlling the cross-axis coupling cost required to perform it.

2.3 INSTANTIATION OF SINC

To achieve cross-axis alignment under the trade-off objectives, KV memory provides a natural foundation: historical representations are encoded into addressable key-value memory (Wang et al.,

2023b), from which different queries retrieve target-specific information:

$$\mathrm{M} = \mathrm{Concat}_{i=1}^{|x|} \left\{ \mathrm{softmax}\left(\frac{Q_i \widetilde{K}_i^\top}{\sqrt{d}} \right) \widetilde{V}_i \right\}. \tag{5}$$

where, $\widetilde{\mathbf{K}}_i$ and $\widetilde{\mathbf{V}}_i$ denote the memory keys and values retrieved for query $\mathbf{Q}_i$, respectively. This mechanism allows different targets to derive specific representations from the same temporal memory. A SInC implementation should satisfy two conditions:

Single-axis budget. Explicitly maintaining L temporal states and N variate representations requires $\Theta(L + N)$ high-dimensional token states, without constructing a $L \times N$ token grid.

Stage-local memory. Memory is constructed and accessed only within the current lookback window and is shared by all target variates in that stage, avoiding cross-stage interference.

3 METHODOLOGY

We instantiate SInC with a time-major design, termed *DAMIE*. Given history X, *DAMIE* (i) builds a shared temporal memory, (ii) completes per-variate representations via one-pass coupling, and (iii) decodes forecasts. We summarize the pipeline as Eq.(6).

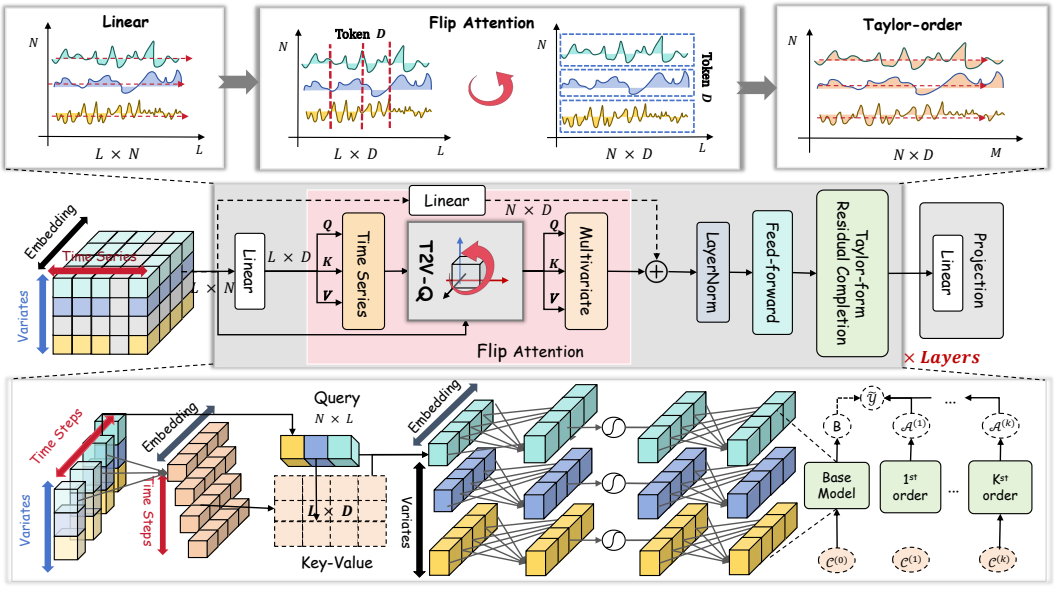

Figure 3: Overview of *DAMIE*: efficient time-to-variate axis inversion.

3.1 OVERALL ARCHITECTURE

Fig. 3 overviews *DAMIE*, and table 1 summarizes tensor shapes. T, V_{var}, and H_{TFRC} correspond to the outputs of FLIP ATTENTION (include TIME SERIES and MULTIVARIATE), TAYLOR-FORM RESIDUAL COMPLETION (TFRC). Given $X \in \mathbb{R}^{B \times L \times N}$, each layer follows the forward path:

$$X\ (B{\times}L{\times}N) \to T\ (B{\times}L{\times}D) \to V_{\text{var}}\ (B{\times}N{\times}D) \to H_{TFRC}\ (B{\times}N{\times}D) \to \hat{Y}\ (B{\times}P{\times}N). \tag{6}$$

Build. We keep the time axis explicitly indexable by projecting each step $X_{t,:} \in \mathbb{R}^N$ into a D-dimensional token and encoding the resulting length-L sequence with a temporal backbone (TIME SERIES). This produces a shared *step-indexed temporal memory* $T \in \mathbb{R}^{B \times L \times D}$, where each slot corresponds to one native time step. When needed by Flip Attention, we use its head-wise key/value form $(K_t, V_t) \in \mathbb{R}^{B \times L \times A \times d}$ with $D = Ad$.

Flip. On top of T, FLIP ATTENTION computes an explicit alignment $\beta \in \mathbb{R}^{B\times A\times N\times L}$ (a distribution over the L time steps for each variate and head) and reads out $V_{\text{var}} \in \mathbb{R}^{B\times N\times D}$ from the shared memory. Under the *SInC* interface (Sec. 2.1), T serves as the *view*, β as the explicit *weights*, and the weighted memory aggregation as the *readout*; importantly, this completion is realized in a one-/few-pass transfer through T without constructing an $L \times N$ token grid.

Table 1: Notation and Tensor shapes in Sec. 3.

Parameter	Explanation
B	batch size
L	lookback length
N	number of variates
P	prediction length
D	model width
A	number of heads
d	per-head width, with $D = Ad$

Completion. The encoded variate representations $S \in \mathbb{R}^{B\times N\times D}$ are refined by the proposed TFRC, which constructs a residual correction from learnable multi-order components weighted by fixed Taylor-form coefficients. The correction is added to the original representations, and the completed representations are mapped by a lightweight prediction head to produce $\hat{Y} \in \mathbb{R}^{B\times P\times N}$.

We next detail the temporal view and the Flip Attention mechanism.

3.2 FLIP ATTENTION AS STEP-INDEXED COMPLETION

Flip Attention is our core mechanism for realizing *SInC*. Starting from the shared *step-indexed temporal memory* built in Sec. 3.1, it produces an explicit *variate-to-time* alignment over the L *native* time steps and reads out variate representations in one-pass cross-axis transfer.

View: Temporal memory bank. For Flip Attention, the *step-indexed temporal memory* T, with one slot per native time step, is projected into $Q_t, K_t, V_t \in \mathbb{R}^{B\times L\times A\times d}$, where $D = Ad$. The pair (K_t, V_t) forms a stage-local time-indexed memory bank that instantiates the *SInC view*, preserving native-step resolution along the explicit time axis while supporting target-specific readout.

Weights: T2V-Q for step-indexed alignment. To make evidence access *variate-conditioned* while keeping the memory indexed by *time*, Flip Attention performs a *query rewrite* from time indexing to variate indexing, which we denote as T2V-Q (time-to-variate queries). Instead of building a separate memory per variate, we keep a shared memory (K_t, V_t) across all variates N, and only *rewrite the queries* so that each variate obtains its own temporal alignment distribution.

Gating coefficients. We first transpose X into the variate-major view $X^\top \in \mathbb{R}^{B\times N\times L}$, which provides a natural variate–time mapping, with each row recording one variate over the L native time steps. Since raw observations do not directly encode temporal relevance, we apply a shared GELU-activated linear projection g (i.e., linear layer activated by GELU) to each variate:

$$\tilde{X} = g(X^\top) \in \mathbb{R}^{B\times N\times L}. \tag{7}$$

For each (b, n), $\tilde{X}_{b,n,:}$ is a length-L coefficient vector indicating which timestamps are useful when assembling the query for variate n; it is *not* required to be a probability distribution.

T2V-Q *rewrite.* Using $\tilde{X}$ as a rewriting kernel, we aggregate the time-indexed query atoms Q_t into one variate-indexed query per (n, a):

$$Q_v = \tilde{X}\, \bar{Q}_t \in \mathbb{R}^{B\times N\times A\times d}, \tag{8}$$

where $\bar{Q}_t$ denotes Q_t viewed as a length-L stack of query atoms and the product represents a weighted sum along the time-step index (equivalently, $Q_v[b, n, a, :] = \sum_{\ell=1}^{L} \tilde{X}[b, n, \ell]\; Q_t[b, \ell, a, :]$). This is the key "flip": the memory (K_t, V_t) remains shared and step-indexed, while the *queries* become variate-indexed, enabling variate-conditioned evidence access without constructing an explicit $L \times N$ token grid.

Step-indexed alignment. We then match Q_v with the temporal keys K_t and normalize over the *native* time-step index to obtain an explicit variate-to-time alignment:

$$\beta = \mathrm{Softmax}_L\left(\frac{\langle Q_v, K_t\rangle}{\sqrt{d}}\right) \in \mathbb{R}^{B\times A\times N\times L}, \tag{9}$$

where $\mathrm{Softmax}_L$ normalizes over the L time steps and produce scores in $\mathbb{R}^{B\times A\times N\times L}$. Thus, $\beta_{b,a,n,:} \in \Delta^L$ defines an explicit native-step alignment distribution.

Readout: Shared-memory axis inversion. Given β, variate representations are obtained by aggregating values V_t over the L time steps within each head, and then merging the A heads ($D = Ad$):

$$V_{\text{var}} = \beta V_t \in \mathbb{R}^{B\times N\times D}, \tag{10}$$

Flip Attention matches the *SInC* interface: the temporal bank (K_t, V_t) is the **view**, the explicit alignment β is the **weights**, and Eq. equation 10 is the **readout**. *Step-indexed.* The view T (hence (K_t, V_t)) has one slot per native time step, and Eq. equation 9 normalizes over $\ell \in \{1, \ldots, L\}$, so $\beta_{:,a,n,:} \in \Delta^L$ assigns weights directly to individual timestamps (no coarse bins). *Budgeted.* Cross-axis interaction occurs only in a single rewrite–align–readout pass (Eqs. equation 8–equation 10), with dominant cost $O(B \cdot A \cdot N \cdot L)$ per layer and no repeated 2D mixing on an $L \times N$ token grid.

3.3 PREDICTION-ORIENTED VARIATE COMPLETION

Flip Attention has read out $V_{\text{var}} \in \mathbb{R}^{B\times N\times D}$ through step-indexed alignment. Subsequently, *Feed-forward Network (FFN)* and *Taylor-form Residual Completion (TFRC)* enhance nonlinear expressiveness and supplement prediction-oriented representation residuals in the aggregated feature space, mitigating potential information loss that may be caused by the cross-axis transfer.

Nonlinear expressiveness. To preserve the original variate information, we first fuse the V_{var} produced by *Flip Attention* with the input variate embeddings X_{var} via a residual connection:

$$\widetilde{V}_{\text{var}} = V_{\text{var}} + X_{\text{var}}, \qquad \widetilde{V}_{\text{var}} \in \mathbb{R}^{B\times N\times D}. \tag{11}$$

A position-wise FFN is then applied independently to each variate representation:

$$H = \text{Norm}\left(\widetilde{V} * \text{var} + \text{Dropout}\left(\sigma\left(\widetilde{V} * \text{var}W_1 + b_1\right)W_2 + b_2\right)\right), \tag{12}$$

where $W_1 \in \mathbb{R}^{D\times D_h}$ and $W_2 \in \mathbb{R}^{D_h\times D}$ are learnable projections, D_h is the hidden dimension, and σ denotes GELU to enhance the variate nonlinear features in $H \in \mathbb{R}^{B\times N\times D}$.

Taylor-form Residual Completion. To complement cross-axis information, we design TFRC to estimate residual information that remains insufficiently represented in the current variate features. Given the FFN output H, TFRC computes K learnable order-specific transformations $C_k(H) \in \mathbb{R}^{N\times D}$, inspired by a truncated Taylor expansion, and combines them into a residual correction:

$$\Delta H = \sum_{k=1}^{K} \omega_k C_k(H), \qquad \Delta H \in \mathbb{R}^{B\times N\times D}. \tag{13}$$

where $\omega_k = \tau^k/k!$ weights the k-th component, with τ controlling successive-order contributions and $k!$ suppressing higher-order corrections. The residual correction then ΔH is added to H, and normalized to obtain the refined variate representation.

$$H_{TFRC} = \text{Norm}\left(H + \Delta H\right) = \text{Norm}\left(H + \sum_{k=1}^{K} \omega_k C_k(H)\right), \qquad H_{TFRC} \in \mathbb{R}^{B\times N\times D}. \tag{14}$$

This residual formulation preserves the principal information in H while learn complementary nonlinear corrections underrepresented in the base representation. The term "Taylor-form" refers to the Taylor-inspired organization of learnable residual transformations for feature completion.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and protocol. We evaluate on 11 datasets from five benchmark families: ETT, PEMS, Weather, Traffic and Electricity (ECL), using the common preprocessing and train/val/test splits from prior work (Wu et al., 2021; Zhou et al., 2021). We fix the lookback length to L=96 and report forecast results. Models are trained with MSE and evaluated by both MSE and MAE. Dataset statistics and preprocessing details are provided in Appendix D.1.

Table 2: Average results with $L = 96$. $P \in \{96, 192, 336, 720\}$ for ETT, Weather, Traffic, and ECL, and $P \in \{12, 24, 48, 96\}$ for PEMS. **Bold** and *italics* denote the best and second-best results.

Dataset	Ours		Patch (Tokens)				$N \times D$				$L \times D$				$N \times L$	
	DAMIE		PatchTST		Crossformer		SOFTS		iTransformer		Autoformer		FEDformer		RLinear	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	*0.451*	**0.445**	0.453	*0.446*	0.529	0.522	0.454	0.447	0.454	0.447	0.496	0.487	**0.440**	0.460	0.489	0.468
ETTh2	**0.362**	**0.395**	0.387	0.407	0.942	0.684	0.383	0.407	*0.382*	*0.406*	0.450	0.459	0.437	0.449	0.402	0.420
ETTm1	0.400	0.407	**0.396**	*0.406*	0.513	0.496	*0.399*	**0.405**	0.408	0.411	0.588	0.517	0.448	0.452	0.414	0.407
ETTm2	**0.284**	**0.326**	*0.287*	*0.330*	0.757	0.610	0.289	0.332	0.291	0.336	0.327	0.371	0.305	0.349	0.291	0.333
Weather	**0.248**	**0.276**	0.259	0.281	0.259	0.315	*0.258*	*0.280*	0.259	0.281	0.338	0.382	0.309	0.360	0.273	0.291
ECL	**0.174**	**0.263**	0.189	0.276	0.244	0.334	*0.177*	*0.264*	0.179	0.271	0.227	0.338	0.214	0.327	0.220	0.300
Traffic	**0.407**	**0.265**	0.481	0.304	0.550	0.304	*0.409*	*0.267*	0.428	0.282	0.628	0.379	0.610	0.376	0.626	0.378
PEMS03	**0.101**	**0.208**	0.180	0.291	0.169	0.281	*0.104*	*0.210*	0.113	0.221	0.667	0.601	0.213	0.327	0.495	0.472
PEMS04	**0.099**	**0.203**	0.195	0.307	0.209	0.314	*0.102*	*0.208*	0.111	0.221	0.610	0.590	0.231	0.337	0.526	0.491
PEMS07	**0.085**	*0.185*	0.211	0.303	0.235	0.315	*0.087*	**0.184**	0.101	0.204	0.367	0.451	0.165	0.283	0.504	0.478
PEMS08	**0.137**	**0.210**	0.280	0.321	0.268	0.307	*0.138*	*0.219*	0.150	0.226	0.814	0.659	0.286	0.358	0.529	0.487

Baselines. Since SInC is instantiated with a Transformer architecture, our baselines primarily consist of Transformer-style forecasters, including: (i) via patch/segment tokenization, (ii) variate-token ($N \times D$) and time-token ($L \times D$) Transformer designs, and (iii) a strong linear baseline operating on the native organization ($N \times L$). To test generality beyond Transformers, we additionally compare against selective-SSM forecasters, MLP-style models, and other interaction-based architectures. A complete baseline list and categorization under our lens are given in Appendix D.2 (Table 7).

Implementation. All experiments are conducted on a single NVIDIA A6000 GPU and follow the training protocol of iTRANSFORMER (Liu et al., 2024). We use the same data splits, input features, and evaluation procedure for all methods, and tune hyperparameters on the validation set. Hyperparameter sensitivity and qualitative forecasting comparisons are provided in Apps. D.4 and D.5.

4.2 MAIN RESULTS

DAMIE achieves the best MSE and MAE simultaneously on ETTh2, ETTm2, Weather, ECL, Traffic, PEMS03, PEMS04, and PEMS08, demonstrating consistent performance across different sampling frequencies, temporal patterns, and data dimensionalities. Its advantage is particularly evident on datasets with relatively many variates: DAMIE obtains the best MSE on Weather, ECL, Traffic, and all four PEMS datasets, together with the best MAE on six of them and the second-best MAE on PEMS07. These results support the central design of DAMIE: a shared step-indexed temporal memory preserves fine-grained temporal evidence, while variate-specific Flip Attention selectively reads this evidence into each target variate. This explicit time-to-variate connection is especially beneficial when diverse variates require different temporal dependencies.

Comparisons across representation families further support this observation. Patch-based models remain competitive on the low-dimensional ETT datasets, but temporal coarsening may obscure native-step information. Variate-token models such as SOFTS and iTransformer perform strongly on ECL, Traffic, and PEMS, yet do not explicitly preserve time-indexed alignment during variate modeling. DAMIE combines both advantages by retaining native-step temporal memory and converting it into variate representations through a single cross-axis readout, without constructing an explicit $L \times N$ token grid. Appendix Table 8 further shows that DAMIE remains competitive with selective-SSM/Mamba and MLP-based forecasters under the same protocol.

Table 3: Horizon-wise results over seven benchmarks.

P	Overall rank		Gain vs. M/I (%)	
	MSE	MAE	MSE	MAE
96	**1.43**	**1.14**	**1.68**	0.56
192	1.86	1.57	0.95	*1.07*
336	*1.57*	1.71	0.30	0.74
720	**1.43**	*1.43*	*1.53*	**1.29**

Performance across forecasting horizons. Table 3 summarizes horizon-wise results over seven benchmarks (ETTh1/2 ETTm1/2, weather, ECL, traffic). "Overall rank" is the average rank across datasets relative to all Transformer-style baselines (lower is better). "Gain vs. M/I" is the averaged relative improvement over the best baseline in {SOFTS, iTransformer, Autoformer, FEDformer} at each horizon (higher is better). DAMIE maintains an average rank below 2.0 for both metrics at every horizon. At $P = 720$, it achieves an average rank of 1.43 and improves the best M/I baseline by 1.53% in MSE and 1.29% in MAE.These sustained gains suggest that preserving step-indexed temporal evidence and explicitly transferring it to variate representations remain effective for long-horizon forecasting.

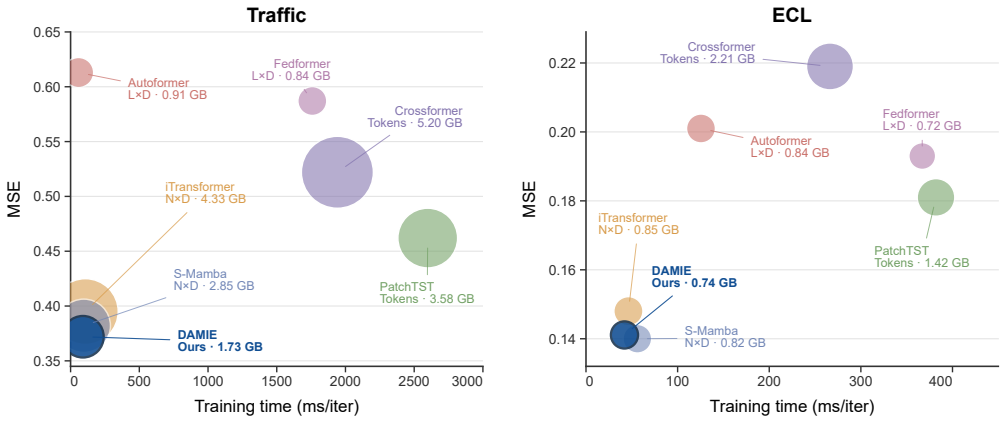

Figure 4: Empirical trade-off on ECL ($N = 321$) and Traffic ($N = 862$) in (L=96, P=96): training time vs. MSE; bubble size indicates peak memory (GB).

4.3 Analyses: Why SInC Matters

We now ask four practical questions that directly correspond to the SInC design.

Q1. Can SInC achieve effective cross-axis modeling within a *single-axis budget*? Sec. 2 requires native-step cross-axis alignment without constructing an $L \times N$ high-dimensional token grid. Fig 4 compares forecasting accuracy, runtime, and GPU memory consumption on ECL and Traffic. DAMIE achieves the best result on both datasets, faster than the iTransformer and uses memory comparable to single-axis encoders, while being more efficient than coarse methods (PatchTST) and explicit dual-axis models (Crossformer). This result is consistent with its $O(L^2 + NL)$ complexity in Table 12: DAMIE completes cross-axis interaction through a single pass.

Takeaway: DAMIE achieves the desired resolution-efficiency trade-off within a *single-axis budget*.

Q2. Is step-indexed cross-axis accessibility effective? Fig 5 presents a controlled lookback analysis on ETTh2, comparing DAMIE with iTransformer over lookback length $L \in \{12, 24, 48, 96, 192, 336\}$. As L increases, iTransformer plateaus beyond $L \approx 96$ and degrades in some long-horizon settings, whereas DAMIE improves more consistently. This contrast supports the effectiveness of step-indexed cross-axis alignment: DAMIE keeps additional history selectively accessible through variate-specific weights instead of compressing it into global variate representations.

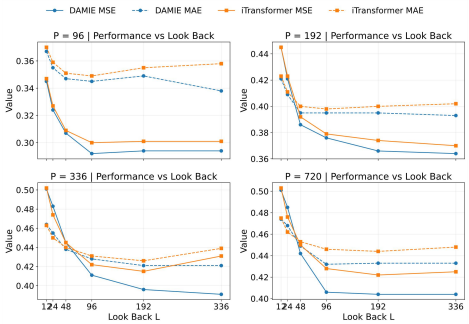

Figure 5: Effect of look-back L on ETTh2.

Takeaway: Step-indexed cross-axis alignment is effective in enabling DAMIE to exploit longer histories more consistently.

Q3. Does *SInC* overcome the limitations of isolated single-axis modeling? Table 4 analysis DAMIE-critical components. TIME-ONLY performs worst, while VARIATE-ONLY also underperforms the *SInC* design, confirming the limitations of isolated single-axis modeling. FLIP (SINC)

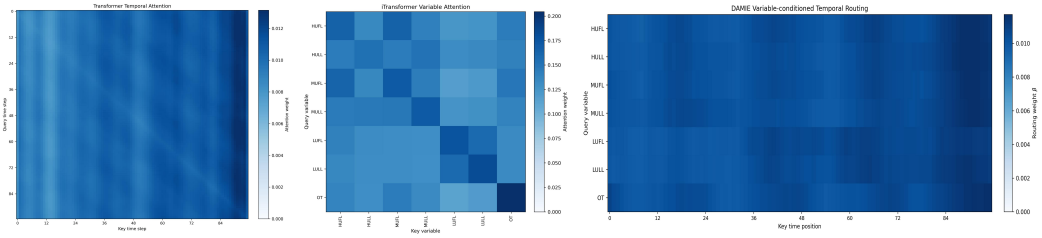

Figure 6: Average last-layer attention weights: Transformer ($L \times L = 96 \times 96$), iTransformer ($N \times N = 7 \times 7$), and DAMIE ($N \times L = 7 \times 96$).

Table 4: Ablation on ETTh2 with $L = 96$ and $P \in \{96, 192, 336, 720\}$. Components are added incrementally, and Δ denotes the improvement over TIME-ONLY.

P	TIME-ONLY		VARIATE-ONLY		FLIP (*SInC*)		FFN		TFRC	
	MSE	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
96	0.351	0.387	0.316	0.366	0.296	0.348	0.294	0.345	**0.288**	**0.341**
192	0.428	0.430	0.399	0.414	0.378	0.405	0.366	0.395	**0.356**	**0.395**
336	0.460	0.463	0.435	0.443	0.415	0.438	0.410	0.429	**0.399**	**0.419**
720	0.494	0.484	0.441	0.455	0.436	0.451	0.412	0.433	**0.406**	**0.430**
Avg	0.433	0.441	0.398	0.420	0.381	0.411	0.371	0.400	**0.362**	**0.396**
Δ	—	—	(↑ 3.5%)	(↑ 2.1%)	(↑ 1.7%)	(↑ 0.9%)	(↑ 1.0%)	(↑ 1.1%)	(↑ 0.8%)	(↑ 0.4%)

gains the largest improvement by explicitly reading step-indexed temporal evidence into variate representations. The FFN and TFRC provide further gains through complementary feature refinement.

Takeaway: SInC drives the primary gain, while FFN and TFRC provide complementary refinement.

Q4. Does *SInC* learn variate-specific step-indexed cross-axis alignments? Fig 6 visualizes the last-layer weights averaged over all attention heads and test samples, reflecting general patterns rather than individual cases. DAMIE preserves the overall variation of temporal attention while forming distinct temporal aggregation profiles for different variates, indicating that each variate selectively reads from the shared memory. Unlike attention restricted to a single axis, DAMIE does not aim to produce sharper weights but explicitly connects the temporal and variate indices. Its relatively smooth distributions facilitate evidence aggregation over a broader temporal range.

Takeaway: SInC learns stable, variate-specific, step-indexed cross-axis alignments.

Summary. The results show that *SInC* achieves the desired resolution-efficiency trade-off within a *single-axis budget*, exploits longer histories consistently. The ablation and weight analyses further identify *SInC* as the primary source of improvement and reveal stable variate-specific, step-indexed alignments, supporting the central claim of Sec. 2.

5 RELATED WORK

LTSF requires long-term temporal modeling, and Transformers have been widely adopted as sequence encoders for LTSF. From the *SInC* perspective, existing Transformer forecasters follow three strategies: *implicit mixing* (M/I) absorbs cross-axis dependencies into hidden representations (Wu et al., 2021; Han et al., 2024; Liu et al., 2024); *coarse-indexed completion* (C) groups multiple native time steps into patches, periods, or other aggregated temporal units, making within-unit steps indistinguishable(Nie et al., 2023; Wang et al., 2024b); and *explicit two-axis recoupling* (E) directly models temporal-variate interactions at higher computational and memory costs(**?**)

Overall, existing methods have not yet achieved the desired trade-off between native step-level alignment and a single-axis computational budget. More details are provided in the appendix C.

6 CONCLUSION

LTSF requires balancing native time-step resolution against the cost of cross-axis modeling. We introduced *SInC* to formalize native-step addressability and cross-axis readout under a single-axis budget, and developed DAMIE to realize these objectives through shared temporal memory and sequential axis modeling. The main experiments show that DAMIE achieves competitive forecasting accuracy across multiple, while remaining computationally efficient even in high-dimensional settings. The ablation study indicates that explicit time-to-variate readout is an important contributor to the gains. The lookback analysis further suggests that, step-indexed memory readout helps the model use longer histories. More broadly, these results suggest that *SInC* may serve as a reusable design principle for efficient long-context forecasters that retain fine-grained cross-axis information transfer without maintaining an expensive $L \times N$ token grid.

AI USE STATEMENT

We used generative AI tools to improve the clarity and readability of the manuscript, review its wording and presentation, and identify potentially relevant literature. The authors evaluated all AI-assisted suggestions, verified the cited sources, and take responsibility for the final content of the paper.

REFERENCES

Xiuding Cai, Xueyao Wang, Yaoyao Zhu, and Yu Yao. Mambats: improved selective state space models for long-term time series forecasting. *Pattern Recognition*, pp. 114536, 2026.

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

Jake Grigsby, Zhe Wang, Nam Nguyen, and Yanjun Qi. Long-range transformers for dynamic spatiotemporal forecasting. *arXiv preprint arXiv:2109.12218*, 2021.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems*, 37: 64145–64175, 2024.

Ao Hu, Liangjian Wen, Jiang Duan, Yong Dai, He Yan, Dongkai Wang, Jun Wang, Yukun Zhang, Ruoxi Jiang, and Zenglin Xu. Pmdformer: Patch-mean decoupling information transformer for long-term forecasting. *arXiv preprint arXiv:2606.26549*, 2026.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 28514–28529, 2025.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4365–4373, 2023.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations (ICLR) 2022*, pp. 1–25, 2022.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences*, 379(2194), 2021.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4): 1748–1764, 2021.

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*, volume 2026, pp. 102794–102817, 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, 2022a.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *International Conference on Learning Representations*, 2022b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In *International conference on learning representations*, volume 2025, pp. 70367–70413, 2025.

Jiaming Ma, Qihe Huang, Haofeng Ma, Guanjun Wang, Sheng Huang, Zhengyang Zhou, Pengkun Wang, Xu Wang, Binwu Wang, and Yang Wang. PHAT: Modeling period heterogeneity for multivariate time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xiaowen Ma, Zhen-Liang Ni, Shuai Xiao, and Xinghao Chen. Timepro: Efficient multivariate long-term time series forecasting with variable-and time-aware hyper-state. In *Forty-second International Conference on Machine Learning*, 2025.

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

Yiming Niu, Jinliang Deng, and Yongxin Tong. Phaseformer: From patches to phases for efficient and effective time series forecasting. In *International Conference on Learning Representations*, volume 2026, pp. 8262–8285, 2026.

Badri N Patro and Vijay S Agneeswaran. Simba: Simplified mamba-based architecture for vision and multivariate time series. *arXiv preprint arXiv:2403.15360*, 2024.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S Jensen, Zhenli Sheng, et al. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proceedings of the VLDB Endowment*, 17(9): 2363–2377, 2024.

Xiangfei Qiu, Xvyuan Liu, Tianen Shen, Xingjian Wu, Hanyin Cheng, Bin Yang, and Jilin Hu. SEER: Transformer-based robust time series forecasting via automated patch enhancement and replacement. In *Forty-third International Conference on Machine Learning*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023a.

Huiqiang Wang, Jieming Shi, and Li Qing. FACT: Fine-grained across-variable convolution for multivariate time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 38626–38652, 2024a.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, volume 2025, pp. 16980–17016, 2025a.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in neural information processing systems*, 36:74530–74543, 2023b.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in neural information processing systems*, 37:469–498, 2024b.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025b.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: a survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 6778–6786, 2023.

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. In *The Eleventh International Conference on Learning Representations*, 2023.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in neural information processing systems*, 36:69638–69660, 2023.

Chengqing Yu, Fei Wang, Zezhi Shao, Tao Sun, Lin Wu, and Yongjun Xu. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 3062–3072, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, pp. 11121–11128, 2023.

Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Multi-resolution time-series transformer for long-term forecasting. In *International conference on artificial intelligence and statistics*, pp. 4222–4230. PMLR, 2024.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

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

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A Formal Support for Step-Indexed Completion

We provide three counterexamples for restricted scalar completion rules that motivate key choices in SInC: distinguishing native steps in the coupling weights, conditioning the weights on the input X, and conditioning them on the target u. These results concern the specified view and readout restrictions; they do not establish necessity for arbitrary forecasting architectures or for the computational budget condition.

We work under squared loss. Let $I = \{1, \dots, L\}$, with $L \geq 3$, and consider a scalar step-indexed view $H_t \in \mathbb{R}$ with readout

$$z = \sum_{t=1}^{L} \pi(t) H_t, \qquad \pi \in \Delta^L, \tag{15}$$

where $\Delta^L = \{\pi \in \mathbb{R}^L_{\geq 0} : \sum_t \pi(t) = 1\}$. Each theorem states its additional restrictions on the view or weights explicitly. The lower bounds apply under those restrictions and are not claimed for richer representations or readouts.

We use i.i.d. *Rademacher* random variables (uniform on $\{\pm 1\}$), denoted by a_1, a_2, s.

A.1 Coarse indexing (patching) induces irreducible loss

A common efficiency tactic is to coarsen the explicit step axis (e.g., patching in time), which effectively merges multiple native steps into one index. We formalize this as a partition of the native index set.

Coarse indexing as a partition. Let $I = \{1, \dots, L\}$ be the native step indices. A coarse (patched/segmented) indexing is a partition

$$I = \bigsqcup_{k=1}^{G} P_k, \qquad G \geq 2, \quad |P_k| \geq 1, \tag{16}$$

and we say the readout is *coarse-indexed* if it cannot distinguish steps within the same block. In the scalar readout equation 15, this means: for any $t, t' \in P_k$, the model is constrained to use identical weights on those steps.

Theorem A.1 (Irreducible loss under equal within-block weights)**.** *Let $I = \{1, \dots, L\}$ with $L \geq 3$ and a partition equation 16 such that some block satisfies $|P_{k^\star}| \geq 2$. Pick two distinct indices $i \neq j$ in the same block $P_{k^\star}$ and pick an index $r \notin P_{k^\star}$ (exists since $G \geq 2$). Let the step-indexed view be identity: $H_t = x_t$. Define the data distribution by*

$$x_i = a_1, \quad x_j = a_2, \quad x_r = s, \quad x_t = 0 \ \ \forall t \notin \{i, j, r\}, \tag{17}$$

where a_1, a_2, s are i.i.d. Rademacher, and define the target by the switching rule

$$y = \begin{cases} x_i, & s = +1, \\ x_j, & s = -1. \end{cases} \tag{18}$$

Then:

1. *There exists a step-indexed rule achieving zero error.*
2. *Any coarse-indexed weighted readout that must assign the same weight to i and j has*

$$\inf_{\text{coarse-indexed } \pi(\cdot)} \mathbb{E}\big[(z - y)^2\big] \ \geq \ \frac{1}{4}. \tag{19}$$

Proof. (1) *Achievability.* Because $s = x_r$ indicates which source is relevant, choose the step-indexed weights $\pi(i) = \mathbb{1}[s = +1]$, $\pi(j) = \mathbb{1}[s = -1]$, and $\pi(t) = 0$ otherwise. Then $z = \pi(i)x_i + \pi(j)x_j = y$ almost surely.

(2) *Lower bound under coarse indexing.*

Coarse indexing forces $\pi(i) = \pi(j) = \alpha$, where $0 \le \alpha \le 1/2$. Let $\gamma = \pi(r) \ge 0$. Weights assigned to the remaining, zero-valued steps do not affect the readout, so

$$z = \alpha(a_1 + a_2) + \gamma s.$$

Consider $E = \{(a_1, a_2) = (-1, +1)\}$, which has probability $1/4$. On E, $a_1 + a_2 = 0$, while the switching rule gives $y = -s$. Consequently,

$$(z - y)^2 = (\gamma s + s)^2 = (\gamma + 1)^2 \ge 1.$$

Therefore,

$$\mathbb{E}[(z - y)^2] \ge \mathbb{P}(E)\, \mathbb{E}[(z - y)^2 \mid E] \ge \frac{1}{4}.$$

□

Theorem A.1 formalizes the step-indexed requirement: once two potentially-relevant native steps are merged (e.g., by patching), there exist tasks where the model suffers non-vanishing error, while step-indexed weighting solves them exactly.

A.2 Switching evidence requires sample-conditioned weights (dependence on X)

Even with step-indexing, switching evidence requires the weights to adapt across samples. The next theorem shows that *sample-independent* (constant) weights cannot represent the interaction needed to switch between sources.

Theorem A.2 (X-conditioning is necessary for switching evidence; arbitrary L)**.** *Let $I = \{1, \dots, L\}$ with $L \ge 3$ and choose distinct indices $i, j, r \in I$. Let $H_t = x_t$ and let (x_t, y) follow the switching construction equation 17–equation 18. Consider the readout equation 15 with* sample-independent *weights, i.e. $\pi \in \Delta^L$ is a fixed constant vector that does not depend on X. Then*

$$\inf_{\pi \in \Delta^L \text{ constant}} \mathbb{E}\big[(z - y)^2\big] \;\ge\; \frac{1}{2}. \tag{20}$$

In contrast, allowing π to depend on X (e.g., on $s = x_r$) achieves zero error.

Proof. Because $x_t = 0$ for all $t \notin \{i, j, r\}$, any constant-weight readout reduces to

$$z = w_1 a_1 + w_2 a_2 + w_3 s, \qquad w_1 = \pi(i),\ w_2 = \pi(j),\ w_3 = \pi(r), \tag{21}$$

and the remaining weights do not contribute. The target can be written as

$$y = \frac{a_1 + a_2}{2} + \frac{s(a_1 - a_2)}{2}. \tag{22}$$

The first term $\frac{a_1 + a_2}{2}$ lies in the linear span of $\{a_1, a_2\}$, but the interaction term $\frac{s(a_1 - a_2)}{2}$ cannot be represented by any linear combination of (a_1, a_2, s) with constant coefficients. Formally, since a_1, a_2, s are independent Rademachers, the set $\{1, a_1, a_2, s, a_1 a_2, a_1 s, a_2 s, a_1 a_2 s\}$ is an orthogonal basis of $L^2(\{\pm 1\}^3)$. Hence the L^2 projection of y onto $\mathrm{span}\{1, a_1, a_2, s\}$ is exactly $\frac{a_1 + a_2}{2}$, attained by $w_1 = w_2 = \frac{1}{2}$ and $w_3 = 0$. Therefore the minimum achievable MSE equals the squared norm of the residual interaction term:

$$\inf_{w_1, w_2, w_3} \mathbb{E}\big[(z - y)^2\big] = \mathbb{E}\left[\left(\frac{s(a_1 - a_2)}{2}\right)^2\right] = \frac{1}{4}\, \mathbb{E}\big[(a_1 - a_2)^2\big]. \tag{23}$$

Now $(a_1 - a_2)^2 = 0$ with probability $1/2$ and equals 4 with probability $1/2$, so $\mathbb{E}[(a_1 - a_2)^2] = 2$ and the minimum MSE is $1/2$, proving equation 20. Finally, the X-dependent step-indexed rule $\pi(i) = \mathbb{1}[s = +1]$, $\pi(j) = \mathbb{1}[s = -1]$ achieves $z = y$ almost surely. □

Theorem A.2 shows that once we require step-indexed alignment, the remaining expressive power lies in $\Pi(X, \cdot)$: switching evidence forces the weight rule to depend on the input.

A.3 VARIATE-CONDITIONED EVIDENCE REQUIRES u-CONDITIONING (DEPENDENCE ON TARGET INDEX)

Finally, ***SInC*** requires different targets (in time-major completion, different variates) to admit different evidence selection rules. We show that if the weights are shared across targets, the joint error must grow with the number of targets.

Theorem A.3 (u-conditioning is necessary for variate-conditioned localization; general N)**.** *Let $S = \{1, \dots, N\}$ and $U = \{1, \dots, N\}$. Let $H_s = a_s$ where $a_1, \dots, a_N$ are i.i.d. Rademachers. Define the N targets by*

$$y_u = a_u, \qquad u \in U. \tag{24}$$

Consider any completion rule of the form equation 15 where the weights are target-independent*, i.e. $\pi_u(X) \equiv \pi(X)$ for all u (equivalently $z_1 = \cdots = z_N = z$ on each sample). Then*

$$\inf_{\textit{target-independent } \pi(\cdot)} \mathbb{E}\left[\sum_{u=1}^{N}(z_u - y_u)^2\right] \;\geq\; N - 1. \tag{25}$$

Proof. Target-independence implies $z_u = z$ for all u on each sample. For any realized $(a_1, \dots, a_N)$, the best scalar z minimizing $\sum_{u=1}^{N}(z - a_u)^2$ is

$$z^\star = \frac{1}{N}\sum_{u=1}^{N} a_u. \tag{26}$$

The resulting minimum joint error equals $\sum_{u=1}^{N}(a_u - \bar{a})^2$ where $\bar{a}$ is the sample mean. Taking expectation and using $\mathrm{Var}(a_u) = 1$ for Rademacher variables yields

$$\mathbb{E}\left[\sum_{u=1}^{N}(a_u - \bar{a})^2\right] = N - 1, \tag{27}$$

which lower-bounds any target-independent rule and proves equation 25. □

Theorem A.3 formalizes why ***SInC*** uses $\Pi(X, u)$ instead of $\Pi(X)$: to model different targets (time-major: different variates), the alignment/readout must be u-conditioned.

A.4 SUMMARY: HOW APPENDIX A SUPPORTS *SInC*

Theorems A.1–A.3 jointly support the three requirements in Sec. 2: (1) coarsening the explicit axis can create irreducible localization loss (Thm. A.1), motivating *step-indexed* completion; (2) switching evidence forces the weight rule to depend on the input X (Thm. A.2); and (3) variate-conditioned evidence forces dependence on the target index u (Thm. A.3). Together, these results justify treating the weight rule $\pi_u = \Pi_\phi(X, u)$ as a first-class design object in ***SInC***.

B EXTRACTION RULES FOR THE *SInC* TRADE-OFF MAP

This appendix specifies how the coordinates in Fig. 2 are obtained from an architecture in a *grounded, reproducible* manner. The map compares method families in terms of (i) *budget pressure* induced by cross-axis coupling operations and (ii) *coupling (alignment) resolution* at which cross-axis weights are defined. Importantly, this map is *not* a wall-clock benchmark: it is an architecture-level accounting of dominant cross-axis interactions under the ***SInC*** lens.

Index sets and what "coupling" means. We use the same index sets as Def. 1. Let $\mathcal{I}$ denote the *native* index set along the explicit axis at *step level*. In the time-major setting, $\mathcal{I} = \{1, \dots, L\}$ (timestamps). In a variate-major setting, $\mathcal{I} = \{1, \dots, N\}$. Let $\mathcal{S}$ be the *indexable* token set on which the model maintains an explicit one-axis view state (Def. 1). Typically $|\mathcal{S}| = O(L)$ (time-major) or $|\mathcal{S}| = O(N)$ (variate-major). If the axis is coarsened by patching/segmenting, $\mathcal{S}$ indexes patches/bins rather than individual steps. Let $\mathcal{U}$ be the set of indices on the *other (non-indexable)* axis that require cross-axis evidence. In time-major completion, $\mathcal{U}$ typically corresponds to variate

Table 5: Template for grounded coordinate extraction in Fig. 2.

Method / Family	$\mathcal{S}$ (coupling index)	$\mathcal{U}$ (queries)	R	C_{ax} (dominant)	κ_{ax}
Ours (*SInC*)	steps ($\lvert\mathcal{S}\rvert = \lvert\mathcal{I}\rvert$)	targets ($\lvert\mathcal{U}\rvert$)	1	$\Theta(\lvert\mathcal{U}\rvert\lvert\mathcal{S}\rvert)$	≈ 1
C (coarse completion; patch size p)	patches ($\lvert\mathcal{S}\rvert = \lvert\mathcal{I}\rvert/p$)	targets ($\lvert\mathcal{U}\rvert$)	$1/p$	$\Theta(\lvert\mathcal{U}\rvert\lvert\mathcal{S}\rvert)$	≈ 1
M/I (implicit mixing)	implicit (no π_u)	–	N/A	≈ 0	≈ 0
E (2D-heavy; repeated 2D mixing)	steps/grid ($R = 1$)	–	1	$\gg \lvert\mathcal{U}\rvert\lvert\mathcal{S}\rvert$	$\gg 1$

targets (or groups), hence $|\mathcal{U}| \approx N$. In variate-major completion, $\mathcal{U}$ corresponds to time positions, hence $|\mathcal{U}| \approx L$. Under *SInC*, cross-axis evidence use is mediated by weights π_u over $\mathcal{S}$ for each $u \in \mathcal{U}$ (e.g., alignment/selection distributions), followed by a readout/transfer through the one-axis view. This is the only component that we attribute to *cross-axis coupling cost* in the map.

Y-axis: coupling (alignment) resolution. The y-axis reports the resolution at which *coupling weights* are defined:

$$R \;=\; \frac{|\mathcal{S}|}{|\mathcal{I}|} \in (0, 1]. \tag{28}$$

Here $|\mathcal{I}|$ is the native step count and $|\mathcal{S}|$ is the number of indices that weights can address. Weights address individual steps, hence $|\mathcal{S}| = |\mathcal{I}|$ and $R = 1$. If patch size is p along the explicit axis, then $|\mathcal{S}| = |\mathcal{I}|/p$ and $R = 1/p$. More generally, any partition/coarsening of $\mathcal{I}$ into $|\mathcal{S}|$ bins yields $R = |\mathcal{S}|/|\mathcal{I}|$. If a method does not expose an explicit weight object π_u over $\mathcal{S}$, we mark the resolution as *implicit N/A* (plotted near $R \approx 0$ for visualization). *Remark (what R is* not*).* R measures the resolution of *cross-axis coupling/alignment* (where weights are defined), not necessarily the temporal tokenization used inside a backbone. This avoids the common confusion that "a time-major backbone always has step tokens": a method can have step tokens for temporal modeling yet still lack step-level *cross-axis* alignment.

X-axis: cross-axis coupling cost and budget pressure. We define C_{ax} as the dominant *cross-axis interaction count* induced by constructing/applying the coupling weights. The x-axis then reports the normalized *budget pressure*

$$\kappa_{\mathrm{ax}} \;=\; \log_2\!\left(1 + \frac{C_{\mathrm{ax}}}{|\mathcal{U}||\mathcal{S}|}\right). \tag{29}$$

The dashed line $\kappa_{\mathrm{ax}} = 1$ corresponds to the one-pass budget boundary $C_{\mathrm{ax}} = |\mathcal{U}||\mathcal{S}|$ (Def. 1). We count cross-axis interactions at the granularity of *pairwise query–token interactions* between indices in $\mathcal{U}$ and $\mathcal{S}$. Concretely, any module that computes scores/weights between $u \in \mathcal{U}$ and $s \in \mathcal{S}$ (e.g., dot-product attention scores, similarity kernels, alignment logits) contributes $\Theta(|\mathcal{U}||\mathcal{S}|)$ interactions per pass. What is included/excluded.

- **Included in C_{ax}:** operations whose dominant term scales with interactions across the two axes, i.e., coupling between $\mathcal{U}$ and $\mathcal{S}$ (constructing weights, cross-axis transfer/readout).
- **Excluded from C_{ax}:** backbone costs that are purely within the indexable axis (e.g., time-only self-attention over $\mathcal{S}$, temporal convolutions/MLPs), since the map isolates the *coupling* burden.

Extraction rules for common coupling patterns We provide minimal rules sufficient to place method families in Fig. 2. In each case, we assume the dominant term is counted and constant factors are ignored.

Rule 1 (one-pass alignment over $\mathcal{S}$). If a method constructs weights $\{\pi_u\}_{u\in\mathcal{U}}$ by computing scores over all $u \times s$ pairs once, then

$$C_{\mathrm{ax}} = \Theta(|\mathcal{U}||\mathcal{S}|) \quad \Rightarrow \quad \kappa_{\mathrm{ax}} \approx 1.$$

This covers standard one-pass attention-style alignment and one-pass transfer/readout.

Rule 2 (repeated cross-axis passes). If cross-axis alignment/transfer is performed for K layers/passes (e.g., iterative refinement), then

$$C_{\mathrm{ax}} = \Theta(K|\mathcal{U}||\mathcal{S}|) \quad \Rightarrow \quad \kappa_{\mathrm{ax}} = \log_2(1 + K).$$

Rule 3 (coarse/patch alignment). If weights are defined on patch/bin indices along the explicit axis, then $|\mathcal{S}| < |\mathcal{I}|$ and

$$R = \frac{|\mathcal{S}|}{|\mathcal{I}|} < 1.$$

If alignment is still one-pass over these coarse indices, then $C_{\mathrm{ax}} = \Theta(|\mathcal{U}||\mathcal{S}|)$ and $\kappa_{\mathrm{ax}} \approx 1$ while $R < 1$ (the "coarse completion" region).

Rule 4 (implicit mixing without explicit weights). If a method uses cross-axis information only implicitly via channel mixing or fixed, non-auditable projections (without constructing an explicit π_u), we set $C_{\mathrm{ax}} \approx 0$ for the map and mark R as implicit N/A. This places the family in the low-cost / low-auditability region.

Rule 5 (explicit $L \times N$ grid with dense/repeated 2D mixing). If a method instantiates an explicit two-axis token grid and applies dense/repeated 2D interactions (e.g., multiple layers of self-attention over LN tokens), then the resulting coupling far exceeds the one-pass *SInC* boundary. We conservatively record this family as

$$C_{\mathrm{ax}} \gg |\mathcal{U}||\mathcal{S}| \quad \Rightarrow \quad \kappa_{\mathrm{ax}} \gg 1,$$

and typically $R = 1$ if the grid preserves step indices.

Family assignment criteria. Table 5 instantiates the coordinate extraction for representative methods. Each row records: (i) the coupling index set $\mathcal{S}$ (and thus R), (ii) the query set $\mathcal{U}$, (iii) the dominant cross-axis interaction term C_{ax} by the rules above, and (iv) the resulting x. We group baselines into four families under the *SInC* lens: **M/I**: no explicit distribution over native steps is instantiated (alignment not auditable at step level); **C**: coarse step indexing $R < 1$ (e.g., patch/segment weights); **E**: explicit $L \times N$ token grid with dense 2D mixing; **Ours**: time-major *SInC* with $R = 1$ and one-/few-pass coupling $C_{\mathrm{ax}} = O(NL)$.

C Scalable Multivariate Long-Term Time-Series Forecasting

Multivariate LTSF requires modeling both long-term temporal dynamics and inter-variate dependencies. CNN/GCN-based encoders typically propagate information over explicit $L \times N$ fields or variable graphs, effectively capturing local temporal patterns and variable relationships (Yi et al., 2023; Wang et al., 2023a) . However, long-term modeling requires enlarged receptive fields or deeper networks, while the computational cost grows with both temporal and variate scales. Transformers are better suited to long-term modeling along a single token axis, but usually absorb the other axis into feature mixing or projection (Vaswani et al., 2017), creating a trade-off between single-axis efficiency and fine-grained cross-axis alignment. Based on whether cross-axis coupling is explicit, the temporal resolution of its indexing set, and the number of cross-axis interaction passes, existing Transformer forecasters can be divided into the following three categories.

Implicit mixing (M/I). Time-major Transformers, such as Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), and FEDformer(Zhou et al., 2022), mix variate information within the feature dimensions of temporal tokens, whereas iTransformer embeds the complete history of each variate into a variate token (Liu et al., 2024; Han et al., 2024). Both designs explicitly preserve only one indexing axis and compress the other into hidden representations, attention, or projection. They therefore do not expose target-conditioned coupling weights over native time steps. Under the *SInC* framework, their step-level alignment resolution is not directly defined and is marked as N/A.

Coarse-indexed completion (C). Coarse-indexed methods reduce the number of tokens using temporal units such as patches, segments, scales, periods, or phases. Representative methods include Pyraformer (Liu et al., 2022b), PatchTST (Nie et al., 2023), TimeXer (Wang et al., 2024a), MTST (Zhang et al., 2024), PMDformer (Hu et al., 2026), PHAT (Ma et al., 2026), SEER (Qiu et al., 2026), and PhaseFormer (Niu et al., 2026). Their primary interactions are defined over aggregated or reorganized temporal indices rather than all native time steps. Because multiple native observations may share the same temporal unit, these methods cannot independently address positions within that unit, resulting in an effective alignment resolution below the native temporal resolution.

Explicit two-axis recoupling (E). Explicit two-axis methods directly restore interactions between time and variates. Spacetimeformer (Grigsby et al., 2021) jointly constructs temporal-variate tokens, Crossformer performs successive cross-time and cross-variate attention (Zhang & Yan, 2023), and PDFormer (Jiang et al., 2023), DSformer similarly combine temporal and node- or variate-wise module (Yu et al., 2023). Although Crossformer also uses segmented representations, its dominant mechanism is repeated two-axis interaction and is therefore categorized as E. These methods retain richer two-axis structure but require explicit two-dimensional states or multiple cross-axis coupling passes, causing their computational and memory costs to grow with both axes.

Non-Transformer encoders. Beyond Transformers, DLinear (Zeng et al., 2023), TiDE (Das et al., 2023), TimeMixer (Wang et al., 2024a), SCINet (Liu et al., 2022a), TimesNet (Wu et al., 2023), TimeMixer++ (Wang et al., 2025a), and TimeKAN (Huang et al., 2025) model temporal patterns using linear mappings, MLPs, convolutions, decomposition, or KANs (Liu et al., 2025), while S-Mamba citewang2025mamba, SiMBA (Patro & Agneeswaran, 2024), MambaTS (Cai et al., 2026), and TimePro(Ma et al., 2025) employ Mamba for long-term modeling. These methods mainly encode cross-axis dependencies implicitly or operate over non-native temporal units such as multiple scales, periods, and patches, and can therefore be viewed as complementary instances of M/I or C. VPNet (Liu et al., 2026) and FACT (Wang et al., 2026) apply convolutions over two-dimensional variate fields and can be regarded as non-Transformer instances of E. In contrast, the two-dimensional representations in TimesNet and TimeMixer++ arise from temporal reorganization rather than temporal–variate grids and thus do not belong to E. Our appendix experiments include selected representative methods as complementary comparisons; other methods primarily target specific temporal priors or multiscale modeling and are therefore not exhaustively evaluated.

Relation to SInC. Overall, M/I maintains single-axis efficiency by representing cross-axis dependencies implicitly, C reduces computation by lowering temporal indexing resolution, and E preserves fine-grained structure at a higher two-axis cost. SInC targets the trade-off between native step-level alignment and a single-axis computational budget: it preserves all native temporal indices and performs completion through a single step-index cross-axis readout, without materializing an explicit temporal–variate token grid.

D Experimental Details

D.1 Datasets and preprocessing

We conduct experiments on six widely used real-world multivariate forecasting benchmarks, including **ETTh1**, **ETTh2**, **ETTm1**, **ETTm2**, **Weather**, and **ECL**. The ETT benchmarks (Electricity Transformer Temperature) are collected from electricity transformers and contain 7 variables; **ETTh1/ETTh2** are recorded at an hourly rate, while **ETTm1/ETTm2** are recorded every 15 minutes. **Weather** consists of 21 meteorological indicators (e.g., temperature and humidity) sampled every 10 minutes. **ECL** records hourly electricity consumption from 321 clients. **Traffic** contains hourly road occupancy rates measured by 862 sensors on freeways in the San Francisco Bay Area. The four **PEMS** datasets contain traffic network measurements from California, aggregated at 5-minute intervals. Dataset statistics and the forecasting horizons used in our experiments are summarized in Table 6.

D.2 Baselines

Baseline family assignment and the *SInC*-based taxonomy used in Fig. 2 and Table 7 follow App. B. We compare against Transformer-style forecasters, including PatchTST (Nie et al., 2023), Crossformer (Zhang & Yan, 2023), SOFTS (Han et al., 2024), iTransformer (Liu et al., 2024), Autoformer (Wu et al., 2021), and FEDformer (Zhou et al., 2022). We also include linear and non-Transformer models: RLinear (Li et al., 2023); selective-SSM/Mamba forecasters S-Mamba (Wang et al., 2025b), SiMBA (Patro & Agneeswaran, 2024), and TimePro (Ma et al., 2025); MLP-style models TiDE (Das et al., 2023) and DLinear (Zeng et al., 2023); and structured interaction architectures TimesNet (Wu et al., 2023) and SCINet (Liu et al., 2022a).

Table 6: Dataset statistics and Prediction Length used in our experiments.

Dataset	Channels	Granularity	Time Steps	Dataset Size (Train/Val/Test)	Prediction Length
ETTh1, ETTh2	7	1 hour	17,420	(8545, 2881, 2881)	$\{96, 192, 336, 720\}$
ETTm1, ETTm2	7	15 mins	17,420	(34465, 11521, 11521)	$\{96, 192, 336, 720\}$
Weather	21	10 mins	52,696	(36792, 5271, 10540)	$\{96, 192, 336, 720\}$
ECL	321	Hourly	26,304	(18317, 2633, 5261)	$\{96, 192, 336, 720\}$
Traffic	862	Hourly	17,544	(12185, 1757, 3509)	$\{96, 192, 336, 720\}$
PEMS03	358	5min	26,209	(15617, 5135, 5135)	$\{12, 24, 48, 96\}$
PEMS04	307	5min	16,992	(10172, 3375, 3375)	$\{12, 24, 48, 96\}$
PEMS07	883	5min	28,224	(16911, 5622, 5622)	$\{12, 24, 48, 96\}$
PEMS08	170	5min	17,856	(10690, 3548, 3548)	$\{12, 24, 48, 96\}$

Table 7: Baseline models organized under *SInC* lens. We annotate the explicit view/token axis and corresponding family: **M/I** (implicit mixing), **C** (coarse completion), and **E** (2D-heavy coupling).

Family (*SInC*)	Explicit view / tokens	Representative models
Transformer-style (one-axis tokenization)		
C	Patch tokens ($N \times S$)	PatchTST (Nie et al., 2023)
E	Grid / staged coupling ($N \times S$)	Crossformer (Zhang & Yan, 2023)
M/I (var-token)	Variate tokens ($N \times D$)	SOFTS (Han et al., 2024), iTransformer (Liu et al., 2024)
M/I (time-token)	Time tokens ($L \times D$)	Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022)
Non-Transformer backbones		
Linear baseline	Native field ($N \times L$)	RLinear (Li et al., 2023)
SSM / Mamba	Variate tokens ($N \times D$)	S-Mamba (Wang et al., 2025b), SiMBA (Patro & Agneeswaran, 2024), TimePro (Ma et al., 2025)
MLP-based	Feature mixing (no explicit token axis)	TiDE (Das et al., 2023), DLinear (Zeng et al., 2023)
Other interactions	Hierarchical / multi-scale designs	TimesNet (Wu et al., 2023), SCINet (Liu et al., 2022a)

D.3 Additional Results for Main Experiments

This appendix provides (i) fully checkable per-horizon results for the main baselines under the unified protocol (fixed lookback L=96; horizons $P \in \{96, 192, 336, 720\}$), and (ii) the exact definition used to compute the horizon-wise summary reported in Sec. 4.2. For interpretive discussion under the *SInC* lens, see Secs. 2.1 and 4.2 and the taxonomy in App. B.

D.3.1 Per-horizon results on Transformer-style baselines

Table 9 reports MSE/MAE for representative Transformer-style baselines and an $N \times L$ linear baseline at each horizon. Best and second-best are marked per row (fixed dataset and P) for each metric, following the rule in the table caption. Overall, DAMIE is consistently competitive across horizons and achieves best/second-best on a majority of dataset–horizon pairs.

Table 8: Multivariate long-term forecasting results with L=96 and $P \in \{96, 192, 336, 720\}$ by non-Transformer backbones. Results are averaged over all prediction lengths.

Dataset	Ours		Mamba-based						MLP-Base				Others			
	DAMIE		S-Mamba		SiMBA		TimePro		TiDE		DLinear		TimesNet		SCINet	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	*0.451*	*0.445*	0.459	0.453	0.472	0.453	**0.438**	**0.438**	0.541	0.507	0.456	0.452	0.458	0.450	0.747	0.647
ETTh2	**0.362**	**0.400**	0.381	0.406	0.428	0.444	*0.377*	*0.403*	0.611	0.550	0.559	0.515	0.414	0.427	0.954	0.723
ETTm1	*0.400*	0.407	*0.400*	*0.406*	0.444	0.466	**0.391**	**0.400**	0.419	0.419	0.403	0.407	*0.400*	*0.406*	0.485	0.481
ETTm2	*0.284*	**0.326**	0.289	0.332	0.315	0.353	**0.281**	*0.326*	0.358	0.404	0.350	0.401	0.291	0.333	0.571	0.537
Weather	**0.248**	*0.276*	0.253	*0.277*	0.257	0.279	*0.252*	0.279	0.271	0.320	0.265	0.317	0.259	0.287	0.288	0.314
ECL	*0.174*	**0.263**	0.178	0.272	0.201	0.278	**0.173**	*0.264*	0.251	0.344	0.212	0.300	0.192	0.295	0.193	0.296

Table 9: Detailed Transformer-based forecasting results with L=96, $P \in \{96, 192, 336, 720\}$ for ETT, Weather, Traffic, and ECL, and $P \in \{12, 24, 48, 96\}$ for PEMS. **Bold** and *italics* denote the best and second-best results.

Dataset	P	Ours		Patch (Tokens)				$N \times D$				$L \times D$				$N \times L$	
		DAMIE		PatchTST		Crossformer		SOFTS		iTransformer		Autoformer		FEDformer		RLinear	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	*0.382*	**0.398**	0.394	0.406	0.423	0.448	0.384	*0.403*	0.384	*0.403*	0.449	0.459	**0.376**	0.419	0.435	0.435
	192	*0.438*	**0.433**	0.440	*0.435*	0.471	0.474	0.444	0.441	0.443	0.437	0.500	0.482	**0.420**	0.448	0.481	0.460
	336	0.483	0.465	0.491	0.462	0.570	0.546	*0.482*	**0.455**	0.485	*0.458*	0.521	0.496	**0.459**	0.465	0.520	0.478
	720	*0.502*	*0.485*	**0.487**	**0.479**	0.653	0.621	0.504	0.490	0.503	0.491	0.514	0.512	0.506	0.507	0.519	0.498
	Avg	*0.451*	**0.445**	0.453	*0.446*	0.529	0.522	0.454	0.447	0.454	0.447	0.496	0.487	**0.440**	0.460	0.489	0.468
ETTh2	96	**0.288**	**0.341**	0.302	*0.348*	0.745	0.584	*0.297*	*0.348*	0.300	0.349	0.346	0.388	0.358	0.397	0.319	0.364
	192	**0.356**	**0.388**	0.388	0.400	0.877	0.656	*0.376*	*0.397*	0.379	0.398	0.456	0.452	0.429	0.439	0.403	0.413
	336	**0.399**	**0.419**	0.426	0.433	1.043	0.731	*0.415*	0.432	0.422	*0.431*	0.482	0.486	0.496	0.487	0.441	0.445
	720	**0.406**	**0.430**	0.431	0.446	1.104	0.763	0.442	0.452	*0.428*	*0.446*	0.515	0.511	0.463	0.474	0.444	0.456
	Avg	**0.362**	**0.395**	0.387	*0.407*	0.942	0.684	0.383	0.407	*0.382*	*0.406*	0.450	0.459	0.437	0.449	0.402	0.420
ETTm1	96	**0.322**	*0.363*	0.329	0.365	0.404	0.426	*0.325*	**0.361**	0.345	0.377	0.505	0.475	0.379	0.419	0.355	0.376
	192	0.381	0.397	0.380	0.394	0.450	0.451	**0.375**	**0.389**	*0.378*	*0.392*	0.553	0.496	0.426	0.441	0.391	0.392
	336	*0.417*	*0.412*	**0.399**	**0.410**	0.532	0.515	0.424	0.416	0.418	0.417	0.621	0.537	0.445	0.459	0.424	0.415
	720	0.478	0.456	*0.475*	**0.453**	0.666	0.589	**0.472**	*0.454*	0.491	0.459	0.671	0.561	0.543	0.490	0.487	*0.450*
	Avg	0.400	0.407	**0.396**	*0.406*	0.513	0.496	*0.399*	**0.405**	0.408	0.411	0.588	0.517	0.448	0.452	0.414	0.407
ETTm2	96	**0.175**	**0.261**	0.184	0.264	0.287	0.366	*0.182*	*0.262*	0.183	0.269	0.255	0.339	0.203	0.287	0.187	0.271
	192	**0.242**	**0.298**	*0.246*	*0.306*	0.414	0.492	0.249	0.307	0.253	0.315	0.281	0.340	0.269	0.328	0.251	0.310
	336	*0.313*	**0.343**	**0.308**	*0.346*	0.597	0.542	*0.313*	0.351	0.315	0.351	0.339	0.372	0.325	0.366	0.312	0.348
	720	**0.408**	**0.402**	*0.409*	**0.402**	1.730	1.042	0.415	0.406	0.412	0.407	0.433	0.432	0.421	0.415	0.412	*0.403*
	Avg	**0.284**	**0.326**	*0.287*	*0.330*	0.757	0.610	0.289	0.332	0.291	0.336	0.327	0.371	0.305	0.349	0.291	0.333
Weather	96	*0.163*	**0.211**	0.177	0.218	**0.158**	0.230	0.171	*0.212*	0.175	0.214	0.266	0.336	0.217	0.296	0.192	0.232
	192	*0.211*	**0.251**	0.225	*0.259*	**0.206**	0.277	0.224	*0.259*	0.228	0.259	0.307	0.367	0.276	0.336	0.241	0.270
	336	**0.271**	**0.296**	0.278	**0.297**	**0.272**	0.335	0.281	0.301	0.278	0.300	0.359	0.395	0.339	0.380	0.295	0.308
	720	**0.350**	**0.347**	*0.354*	*0.348*	0.398	0.418	0.356	0.351	0.358	0.351	0.419	0.428	0.403	0.428	0.364	0.352
	Avg	**0.248**	**0.276**	0.259	*0.281*	0.259	0.315	*0.258*	*0.280*	0.259	*0.281*	0.338	0.382	0.309	0.360	0.273	0.291
ECL	96	**0.141**	**0.233**	0.164	0.251	0.219	0.314	*0.143*	**0.233**	0.148	0.240	0.201	0.317	0.193	0.308	0.201	0.281
	192	*0.163*	*0.251*	0.173	0.262	0.231	0.322	**0.160**	**0.249**	0.166	0.258	0.222	0.334	0.201	0.315	0.207	0.286
	336	**0.178**	**0.267**	0.190	0.279	0.246	0.337	**0.178**	*0.269*	**0.178**	*0.269*	0.231	0.338	0.214	0.329	0.215	0.298
	720	**0.217**	**0.301**	0.230	0.313	0.280	0.363	*0.225*	*0.305*	*0.225*	0.317	0.254	0.361	0.246	0.355	0.257	0.333
	Avg	**0.174**	**0.263**	0.189	0.276	0.244	0.334	*0.177*	*0.264*	0.179	0.271	0.227	0.338	0.214	0.327	0.220	0.300
Traffic	96	*0.378*	**0.250**	0.462	0.295	0.522	0.290	**0.376**	*0.251*	0.395	0.268	0.613	0.388	0.587	0.366	0.649	0.389
	192	**0.396**	**0.258**	0.466	0.296	0.530	0.293	*0.398*	*0.261*	0.417	0.276	0.616	0.382	0.604	0.373	0.601	0.366
	336	**0.412**	*0.271*	0.482	0.304	0.558	0.305	*0.415*	**0.269**	0.43	0.283	0.622	0.337	0.621	0.383	0.609	0.369
	720	**0.445**	**0.283**	0.514	0.322	0.589	0.325	*0.447*	*0.287*	0.467	0.302	0.660	0.408	0.626	0.382	0.647	0.387
	Avg	**0.407**	**0.265**	0.481	0.304	0.550	0.304	*0.409*	*0.267*	0.428	0.282	0.628	0.379	0.610	0.376	0.626	0.378
PEMS03	12	*0.065*	*0.170*	0.099	0.216	0.090	0.203	**0.064**	**0.165**	0.071	0.174	0.272	0.385	0.126	0.251	0.126	0.236
	24	**0.082**	**0.182**	0.142	0.259	0.121	0.240	*0.083*	*0.188*	0.093	0.201	0.334	0.440	0.149	0.275	0.246	0.334
	48	**0.108**	**0.219**	0.211	0.319	0.202	0.317	*0.114*	*0.223*	0.125	0.236	1.032	0.782	0.227	0.348	0.551	0.529
	96	**0.146**	**0.261**	0.269	0.370	0.262	0.367	*0.156*	*0.264*	0.164	0.275	1.031	0.796	0.348	0.434	1.057	0.787
	Avg	**0.101**	**0.208**	0.180	0.291	0.169	0.281	*0.104*	*0.210*	0.113	0.221	0.667	0.601	0.213	0.327	0.495	0.472
PEMS04	12	**0.072**	**0.171**	0.105	0.224	0.098	0.218	*0.074*	*0.176*	0.078	0.183	0.424	0.491	0.138	0.262	0.138	0.252
	24	**0.085**	**0.190**	0.153	0.275	0.131	0.256	*0.088*	*0.194*	0.095	0.205	0.459	0.509	0.177	0.293	0.258	0.348
	48	**0.107**	**0.212**	0.229	0.339	0.205	0.326	*0.110*	*0.219*	0.120	0.233	0.646	0.610	0.270	0.368	0.572	0.544
	96	**0.133**	**0.241**	0.291	0.389	0.402	0.457	*0.135*	*0.244*	0.150	0.262	0.912	0.748	0.341	0.427	1.137	0.820
	Avg	**0.099**	**0.203**	0.195	0.307	0.209	0.314	*0.102*	*0.208*	0.111	0.221	0.610	0.590	0.231	0.337	0.526	0.491
PEMS07	12	*0.058*	*0.153*	0.095	0.207	0.094	0.200	**0.057**	**0.152**	0.067	0.165	0.199	0.336	0.109	0.225	0.118	0.235
	24	**0.069**	**0.168**	0.150	0.262	0.139	0.247	*0.073*	*0.173*	0.088	0.190	0.323	0.420	0.125	0.244	0.242	0.341
	48	*0.097*	*0.199*	0.253	0.340	0.311	0.369	**0.096**	**0.195**	0.110	0.215	0.390	0.470	0.165	0.288	0.562	0.541
	96	**0.116**	**0.215**	0.346	0.404	0.396	0.442	*0.120*	*0.218*	0.139	0.245	0.554	0.578	0.262	0.376	1.096	0.795
	Avg	**0.085**	*0.185*	0.211	0.303	0.235	0.315	*0.087*	**0.184**	0.101	0.204	0.367	0.451	0.165	0.283	0.504	0.478
PEMS08	12	**0.071**	**0.167**	0.268	0.232	0.165	0.214	*0.074*	*0.171*	0.079	0.182	0.436	0.485	0.173	0.273	0.133	0.247
	24	**0.099**	**0.194**	0.224	0.281	0.215	0.260	*0.104*	*0.201*	0.115	0.219	0.467	0.502	0.210	0.301	0.249	0.343
	48	*0.170*	*0.239*	0.321	0.354	0.315	0.355	**0.164**	0.253	0.186	**0.235**	0.966	0.733	0.320	0.394	0.569	0.544
	96	**0.209**	**0.246**	0.408	0.417	0.377	0.397	*0.211*	*0.253*	0.221	0.267	1.385	0.915	0.442	0.465	1.166	0.814
	Avg	**0.137**	**0.212**	0.280	0.321	0.268	0.307	*0.138*	*0.219*	0.150	0.226	0.814	0.659	0.286	0.358	0.529	0.487

D.3.2 PER-HORIZON RESULTS ON NON-TRANSFORMER BACKBONES

Tables 8 and 10 provide the averaged (across horizons) comparison and the full per-horizon breakdown for non-Transformer forecasters (SSM/Mamba-style, MLP-style, and structured interaction models) under the same protocol. We use the same per-row marking rule as in Table 9. DAMIE re-

Table 10: Detailed results on non-Transformer baselines with L=96 and $P \in \{96, 192, 336, 720\}$.

Dataset	P	Ours		Mamba-based						MLP-Base				Others			
		DAMIE		S-Mamba		SiMBA		TimePro		TiDE		DLinear		TimesNet		SCINet	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	*0.382*	**0.398**	0.387	0.406	0.432	0.416	**0.375**	*0.398*	0.479	0.464	0.386	0.400	0.384	0.402	0.654	0.599
	192	0.438	0.433	0.442	0.439	0.451	**0.422**	**0.427**	*0.429*	0.525	0.492	0.437	0.432	*0.436*	0.429	0.719	0.631
	336	0.483	0.465	0.490	0.466	*0.477*	0.462	**0.472**	**0.450**	0.565	0.515	0.481	*0.459*	0.491	0.469	0.778	0.659
	720	*0.502*	*0.485*	0.515	0.501	0.531	0.515	**0.476**	**0.474**	0.594	0.558	0.519	0.516	0.521	0.500	0.836	0.699
	Avg	**0.452**	**0.442**	0.459	0.453	0.472	0.453	*0.438*	*0.438*	0.541	0.507	0.456	0.452	0.458	0.450	0.747	0.647
ETTh2	96	**0.288**	**0.341**	0.298	0.349	0.338	0.370	*0.293*	*0.345*	0.400	0.440	0.333	0.387	0.340	0.374	0.707	0.621
	192	*0.356*	**0.388**	0.377	0.398	0.410	0.465	*0.367*	*0.394*	0.528	0.509	0.477	0.476	0.402	0.414	0.860	0.689
	336	**0.399**	**0.419**	0.423	0.434	0.481	0.441	*0.419*	*0.431*	0.643	0.571	0.594	0.541	0.452	0.452	1.000	0.744
	720	**0.406**	**0.430**	*0.426*	*0.443*	0.486	0.501	0.427	0.445	0.874	0.679	0.831	0.657	0.462	0.468	1.249	0.838
	Avg	**0.362**	**0.395**	0.381	0.406	0.428	0.444	*0.377*	*0.403*	0.611	0.550	0.559	0.515	0.414	0.427	0.954	0.723
ETTm1	96	**0.322**	**0.363**	0.338	0.365	0.357	0.381	*0.326*	*0.364*	0.364	0.387	0.345	0.372	0.338	0.375	0.418	0.438
	192	0.381	0.397	*0.377*	*0.393*	0.451	0.466	**0.367**	**0.383**	0.398	0.404	0.380	0.389	0.374	0.387	0.439	0.450
	336	0.417	0.412	0.410	0.415	0.483	0.511	**0.402**	**0.409**	0.428	0.425	0.413	0.413	*0.410*	*0.411*	0.490	0.485
	720	0.478	0.456	0.474	0.451	0.485	0.507	**0.469**	**0.446**	0.487	0.461	*0.474*	0.453	0.478	*0.450*	0.595	0.550
	Avg	*0.400*	0.407	*0.400*	*0.406*	0.444	0.466	**0.391**	**0.400**	0.419	0.419	0.403	0.407	*0.400*	*0.406*	0.485	0.481
ETTm2	96	**0.175**	*0.261*	0.182	0.266	0.251	0.332	*0.178*	**0.260**	0.207	0.305	0.193	0.292	0.187	0.267	0.286	0.377
	192	**0.242**	**0.298**	0.250	0.309	0.287	0.312	*0.242*	*0.303*	0.290	0.364	0.284	0.362	0.249	0.309	0.399	0.445
	336	0.313	0.343	0.312	0.349	**0.303**	**0.337**	*0.303*	*0.342*	0.377	0.422	0.369	0.427	0.321	0.351	0.637	0.591
	720	*0.408*	*0.402*	0.412	0.404	0.420	0.431	**0.400**	**0.399**	0.558	0.524	0.554	0.522	*0.408*	0.403	0.960	0.735
	Avg	*0.284*	**0.326**	0.289	0.332	0.315	0.353	**0.281**	*0.326*	0.358	0.404	0.350	0.401	0.291	0.333	0.571	0.537
Weather	96	**0.163**	0.211	0.166	0.210	0.187	**0.192**	*0.165*	*0.209*	0.202	0.261	0.196	0.255	0.172	0.220	0.173	0.223
	192	**0.211**	**0.251**	0.217	*0.256*	*0.212*	0.265	0.217	0.258	0.242	0.298	0.237	0.296	0.219	0.261	0.245	0.285
	336	**0.271**	*0.296*	0.276	*0.296*	*0.272*	**0.288**	*0.273*	0.299	0.287	0.335	0.283	0.335	0.280	0.306	0.321	0.338
	720	*0.350*	*0.348*	0.351	**0.345**	0.360	0.369	0.352	0.350	0.351	0.386	**0.345**	0.381	0.365	0.359	0.414	0.410
	Avg	**0.248**	**0.276**	0.253	*0.277*	0.257	0.279	*0.252*	0.279	0.271	0.320	0.265	0.317	0.259	0.287	0.288	0.314
ECL	96	*0.141*	**0.233**	**0.140**	0.237	0.167	0.261	0.143	0.237	0.237	0.329	0.197	0.282	0.168	0.272	0.169	0.273
	192	*0.163*	**0.251**	**0.162**	0.257	0.187	*0.253*	0.164	*0.253*	0.236	0.330	0.196	0.285	0.184	0.289	0.182	0.286
	336	*0.178*	**0.267**	0.182	0.279	0.199	0.284	**0.173**	*0.269*	0.249	0.344	0.209	0.301	0.198	0.300	0.200	0.304
	720	*0.217*	*0.301*	0.228	0.314	0.250	0.314	**0.212**	**0.298**	0.284	0.373	0.245	0.333	0.220	0.320	0.222	0.321
	Avg	*0.174*	**0.263**	0.178	0.272	0.201	0.278	**0.173**	*0.264*	0.251	0.344	0.212	0.300	0.192	0.295	0.193	0.296

mains top-tier across datasets and horizons, indicating that the gains are not specific to Transformer baselines.

D.3.3 Definition of the horizon-wise summary used in the main text

Sec. 4.2 reports a compact "Gain vs M/I" horizon summary. We define it as follows so it is reproducible.

Reference set. For each dataset and horizon P, let $\mathcal{M}/\mathcal{I}$ denote the set of Transformer baselines assigned to the **M/I** family under the taxonomy in Table 7. Let $e_{\mathrm{ref}}(P) = \min_{m \in \mathcal{M}/\mathcal{I}} e_m(P)$ be the best (lowest-error) score among $\mathcal{M}/\mathcal{I}$ at horizon P, computed separately for each metric (MSE/MAE).

Gain vs M/I (per horizon). Let $e_{\mathrm{ours}}(P)$ denote our score at horizon P. We compute the relative improvement $\Delta(P) = \frac{e_{\mathrm{ref}}(P) - e_{\mathrm{ours}}(P)}{e_{\mathrm{ref}}(P)}$ for each dataset and each horizon P, and then average $\Delta(P)$ over the six datasets. This produces the "Gain vs M/I" curves reported in the main text.

The full per-horizon tables (Tables 9 and 10) are the primary evidence; the summary provides a compact view of horizon-dependent improvements relative to the M/I reference family defined by App. B.

D.4 Training Hyper-parameter Sensitivity (ETTh2)

Figure 7 reports the sensitivity of DAMIE to several training/configuration choices. (**a**) Varying the encoder depth shows a clear *diminishing-return* trend: moving from 1 to 2 encoder layers consistently reduces the error for all horizons, while deeper stacks (3–5) provide no further benefit and can slightly degrade performance, suggesting that moderate depth is sufficient under this setting. (**b**) For the model width, a medium embedding size yields the best overall trade-off: d_{model}=128

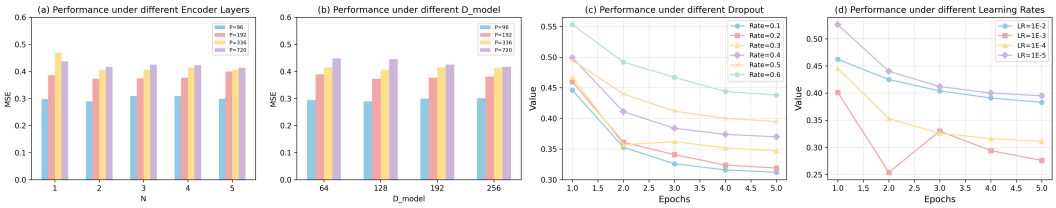

Figure 7: Training hyper-parameter sensitivity on ETTh2. (a) Performance under different encoder layers. (b) Performance under different d_{model}. (c) Performance under different dropout rates. (d) Performance under different learning rates.

Table 11: Forecasting results for different orders.

P	Order=0		Order=1		Order=2		Order=3	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.294	0.345	0.291	0.344	0.288	0.341	0.289	0.343
192	0.366	0.395	0.370	0.400	0.356	0.388	0.361	0.392
336	0.410	0.429	0.408	0.430	0.399	0.419	0.401	0.422
720	0.412	0.433	0.406	0.434	0.406	0.430	0.410	0.428

achieves the lowest or near-lowest MSE across horizons, whereas larger widths (192/256) do not consistently improve and may over-parameterize. (**c**) For dropout, smaller rates converge faster and to lower values: dropout in the range 0.1–0.2 performs best, while heavier dropout (0.4–0.6) noticeably slows optimization and leaves a higher final error. (**d**) For the learning rate, LR=10^{-3} offers the best convergence and final value among the tested options; LR=10^{-2} is comparatively unstable/less optimal, and LR=10^{-5} converges too slowly. We adopt a *moderate* encoder depth and width with *light* regularization (dropout) and LR=10^{-3} as the default configuration.

Across the four prediction lengths, Order$=2$ achieves the lowest MSE and MAE for $P \in \{96, 192, 336\}$. At $P = 720$, it ties for the lowest MSE, while its MAE is slightly higher than that of Order$=3$. We therefore set Order$=2$ as the default, since increasing the order further yields no consistent improvement.

D.5 Qualitative Forecasting Comparison

Figure 8 visualizes forecasts on four datasets (ETTm1, ETTh2, Weather, and ECL). Overall, DAMIE produces trajectories that better align with the ground-truth evolution: its forecasts preserve the *global trend direction* and maintain more stable alignment in oscillatory regimes, while exhibiting smaller high-frequency fluctuations (i.e., less over-reaction to local noise).

ETTm1/ETTh2. For the ETT benchmarks, the target series contains pronounced periodic patterns with slowly varying envelopes. DAMIE tracks the phase and amplitude more consistently across the prediction window, indicating that the model can leverage long-range temporal evidence without accumulating large phase drift. In contrast, the baseline more frequently shows mild phase mismatch and amplitude distortion, which typically compounds as the horizon extends.

Weather. The Weather case highlights trend fidelity: DAMIE follows the overall rising/falling tendency in the future segment with smaller systematic bias. The baseline is more prone to deviating from the correct trend direction in the later part of the forecast, leading to a visible divergence from the ground truth.

ECL. ECL is high-dimensional and noisy, where strong cross-variate coupling alone is insufficient for robust long-horizon forecasting. DAMIE yields smoother predictions that remain consistent with the future trajectory, suggesting that its memory-inversion routing supports *both* cross-variate coupling and temporal trend inference, rather than trading one for the other.

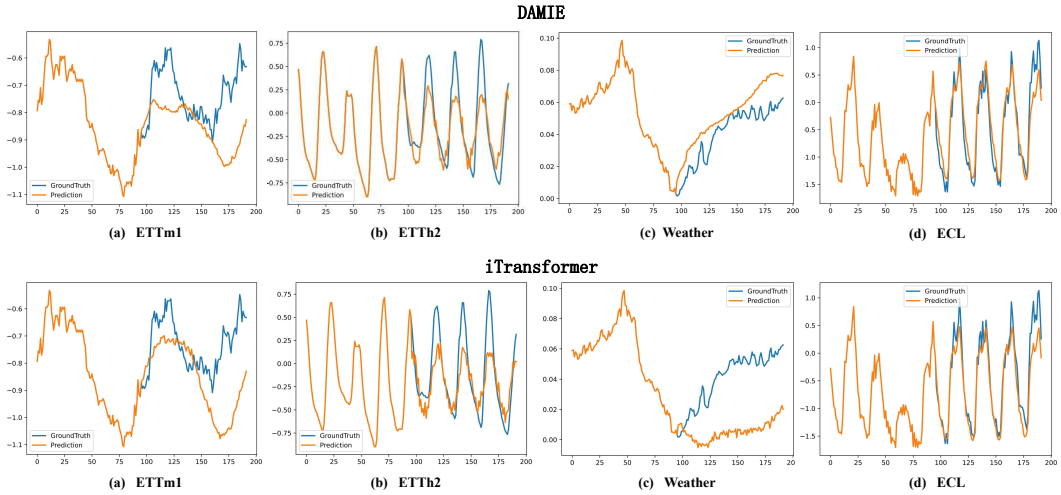

Figure 8: Qualitative forecast results on four datasets (ETTm1, ETTh2, Weather and ECL). DAMIE better matches ground-truth trajectories, especially in long-horizon trend fidelity and stable oscillation alignment, L=96.

Table 12: Asymptotic per-layer complexity and propagation views for representative LTSF frameworks. L is the lookback length and N is the number of variables. $S = \lceil L/p \rceil$ denotes the number of patches/segments with patch length p. M denotes the feature-time length after Taylor expansion in DAMIE (typically $M > L$), and it is *not* an attention token axis.

Method	Propagation	Complexity
iTransformer (Var-token)	$L{\times}N \to N{\times}D$	$\mathcal{O}(N^2)$
S-Mamba (Var-token)	$L{\times}N \to N{\times}D$	$\mathcal{O}(N)$
Transformer (Time-token)	$L{\times}N \to L{\times}D$	$\mathcal{O}(L^2)$
Autoformer (Time-token)	$L{\times}N \to L{\times}D$	$\mathcal{O}(L \log L)$
FEDformer (Time-token)	$L{\times}N \to L{\times}D$	$\mathcal{O}(L)$
PatchTST (Patch-token)	$L{\times}N \to N{\times}S \to N{\times}D$	$\mathcal{O}(NS^2)$
Crossformer (2D Grid-token)	$L{\times}N \to (N{\times}S) \to (N{\times}S)$	$\mathcal{O}(NS^2 + SN^2)$
DAMIE (Time+Var, Flip)	$L{\times}N \to L{\times}D \to N{\times}D$	$\mathcal{O}(L^2 + NL)$

E Complexity Analysis

Table 12 summarizes the dominant *per-layer* complexity of representative long-horizon time series forecasting (LTSF) frameworks under standard implementations. We report only the leading interaction term and omit linear projections, feed-forward blocks, and normalization, as they do not affect the asymptotic scaling in long-history or high-dimensional regimes. We denote by L the lookback length and by N the number of variables. For patch/segment methods, $S = \lceil L/p \rceil$ is the number of patches/segments with patch length p. For DAMIE, M is the length of the feature–time axis after Taylor expansion (typically $M > L$); importantly, M is *not* an attention token axis.

Backbone-dependent interaction axis. Time-token backbones propagate along the temporal axis ($L \times N \to L \times D$) and incur $O(L^2)$ attention. Var-token backbones expose variables as tokens ($L{\times}N \to N{\times}D$), resulting in $O(N^2)$ interactions for attention-based designs (e.g., iTransformer), whereas state-space var-token propagation can scale linearly with the number of variables (e.g., S-Mamba with $O(N)$).

Patch/segment and factorized 2D interaction. Patch-based models first reduce the temporal axis to S segments per variable and then aggregate to $N \times D$, yielding $O(NS^2)$ complexity under a

channel-independent setting (PatchTST). Grid-token models operate on an $N \times S$ token grid and factorize interactions across the two axes, leading to $O(NS^2 + SN^2)$ (Crossformer).

Frequency-domain time-token variants. Autoformer replaces dense attention with FFT-based auto-correlation and achieves $O(L \log L)$ complexity. FEDformer performs frequency-sparse interactions and can reach $O(L)$ when the number of retained modes is fixed. Despite favorable asymptotic scaling, frequency-domain pipelines may increase wall-clock time due to FFT/correlation overheads, additional tensor reshaping, and memory traffic.

DAMIE. DAMIE follows $L \times N \rightarrow L \times D \rightarrow N \times D \rightarrow M \times N$: it first builds a shared temporal memory, then performs a single cross-axis transfer via Flip Attention, and finally applies a Taylor-order local expansion to form the feature–time view. With an attention-based temporal backbone, the dominant per-layer complexity is $O(L^2 + NL)$, where $O(L^2)$ comes from temporal self-attention and $O(NL)$ from one-shot variable-to-time routing/aggregation; the Taylor expansion is linear-time (up to a constant order) and does not change the leading term. This provides explicit cross-axis routing without constructing an LN-token 2D grid or performing multi-round alternating updates.