STATE-CONDITIONED INVERTIBLE CHANNEL DEPENDENCIES FOR MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series (MTS) forecasting requires jointly capturing inter-series dependencies and intra-series temporal evolutions. Although inverted Transformers (e.g., iTransformer) and Graph Neural Networks can capture arbitrary pairwise inter-channel dependencies, they typically suffer from unconstrained $\mathcal{O}(N^2)$ pairwise attention maps that are prone to overfitting spurious cross-channel correlations. Furthermore, aggregating multiple channels into a weighted sum risks destructive signal cancellation and over-smoothing across distinct channels. To address these limitations, we propose a **S**tate-conditioned **I**nvertible **C**hannel **D**ependencies **Net**work (`SICDNet`) that comprises two dedicated components: Structured Invertible Channel Mixer (`SICM`) for inter-series correlation modeling and the Spectral-Gated Modulator (`SGM`) for intra-series feature extraction. `SICM` establishes pairwise channel interactions on latent features using a dyadic pairing schedule, which adopts a multi-stage, multi-round cascade with cyclic node-index shifts to cover more cross-channel pairs. Specifically, `SICM` groups the latent representations and executes Givens rotations within each paired channels of each group, and the rotation angles are conditioned dynamically based on pairwise energy. `SICM` is energy-preserving and mathematically invertible, enabling the extraction of robust inter-channel correlations. Complementarily, `SGM` combines static harmonic priors with data-dependent dynamic spectral gating and dual-domain gated fusion, adaptively capturing global periodic patterns. Theoretical analysis demonstrates that `SICDNet` reduces the inter-series mixing complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ while maintaining channel-independent linear scaling in intra-series spectral processing, outperforming dense attention and graph estimation architectures.

1 INTRODUCTION

Multivariate time series forecasting (MTSF) plays a foundational role in financial market modeling, weather forecasting, traffic flow control, and smart grid management (Shao et al., 2025). A successful forecasting model must concurrently resolve two orthogonal challenges: (i) characterizing *intra-series temporal dynamics* (e.g., non-stationary trends, multi-frequency periodicities (Ma et al., 2025a), and local fluctuations (Wang et al., 2025)), and (ii) capturing *inter-series (i.e., inter-channel/variable) cross-dependencies* (e.g., physical topology, cross-sensor lag correlations, and structural interactions).

Recent advances in deep MTS forecasting have explored two main design paradigms for inter-channel dependency modeling. The first paradigm relies on Spatio-Temporal Graph Neural Networks (ST-GNNs) and adaptive graph learning (Wu et al., 2020; Huang et al., 2023; Shang et al., 2024; Wu et al., 2025; Li et al., 2025b; Ma et al., 2025b). These models dynamically estimate pairwise adjacency matrices $\mathbf{A} \in \mathbb{R}^{N \times N}$ to propagate messages across N variates (typically parameterized as $\mathbf{A} = \text{Softmax}(\mathbf{E}_1 \mathbf{E}_2^\top)$). While effective at imposing relational priors, explicit graph estimation introduces $\mathcal{O}(N^2)$ computational overhead, remains vulnerable to noisy cross-series correlations (Ma et al., 2025b; Yu et al., 2026), and suffers from over-smoothing during deep message passing (Guo et al., 2022), which aggregates multiple channels into a weighted sum. Formally expressed as $\text{Output}_i = \sum_{j=1}^{N} \mathbf{A}_{i,j} \mathbf{V}_j$, where the updated representation of node i is derived by summing the features of all neighbors $\mathbf{V}_j$ multiplied by their respective weights $\mathbf{A}_{i,j}$, this convex combination process risks destructive signal cancellation. A detailed elaboration is provided in Appendix A.2.

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.

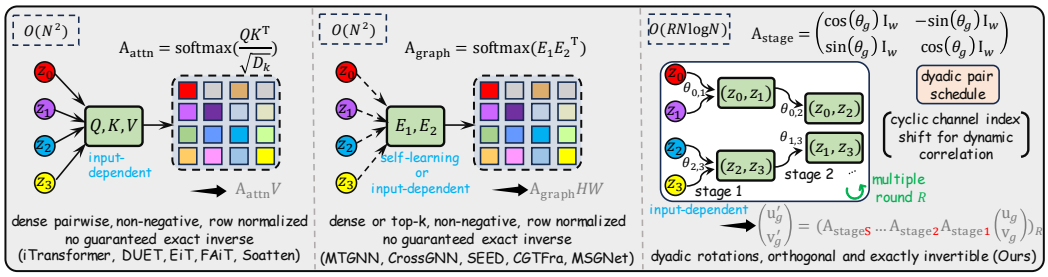


Figure 1: Comparison of inter-channel dependency modeling paradigms. Left: Inverted token encoding based on self-attention. Middle: Self-learned or input-adaptive graph embedding approaches based on GNNs. Right: Our proposed SICM, which utilizes multi-round dyadic Givens rotations.

The second paradigm, epitomized by Inverted Transformers (iTransformer) (Liu et al., 2024), inverts the tokenization axis by mapping the entire historical series of each variable into an individual token $\mathbf{h}_i \in \mathbb{R}^D$, subsequently applying multi-head self-attention across channels (*via* $\text{Softmax}(\mathbf{QK}^\top/\sqrt{D})\mathbf{V}$ with $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times D}$). Although this inverted structure leverages global receptive fields and circumvents temporal token degradation, standard softmax self-attention lacks norm conservation, is non-invertible, and scales quadratically as $\mathcal{O}(N^2)$. Furthermore, dense unconstrained attention tends to overfit spurious cross-channel correlations under noisy regimes (Yu et al., 2026; Xiong et al., 2026).

To overcome the limitations of the aforementioned paradigms, where compressing multiple channels into a weighted sum often leads to signal cancellation (see Appendix A.2), irreversible information loss, quadratic $\mathcal{O}(N^2)$ computational complexity, and susceptibility to noisy dependencies, we propose the SICDNet, a unified framework that integrates dual modeling capabilities: capturing inter-series interactions and intra-series dynamics via the dedicated SICM and SGM modules, respectively. A comparison of inter-channel dependency modeling paradigms is illustrated in Figure 1. Rather than collapsing all channels into a single merged pathway, SICM systematically routes and exchanges pairwise information between channels. It identifies interacting channel pairs via a dyadic pairing schedule and adopts a multi-stage, multi-round strategy with cyclic node-index shifts to cover broader cross-series interactions. Specifically, SICM partitions the latent embeddings into paired groups and applies Givens rotations to each pair, with rotation angles adaptively conditioned on pairwise energy statistics. Consequently, the core component, SICM, guarantees strict orthogonality and structural energy preservation while reducing computational complexity to $\mathcal{O}(N \log N)$, thereby capturing more robust inter-channel dependencies and accommodating signed channel interactions. Code is available at `https://anonymous.4open.science/r/SICDNet`. We provide the following key contributions:

(1) **State-conditioned structured channel mixing.** We introduce SICM, a multi-scale cascade of dyadic, group-wise Givens rotations for cross-channel information exchange. Its rotation angles are conditioned from pair-group energies that are invariant under the corresponding rotations, enabling exact cache-free inversion of the structured mixer in exact arithmetic without storing intermediate states or dense $N \times N$ interaction maps.

(2) **Latent-channel adaptive spectral modulation.** We introduce SGM, which applies a DCT-based transform along the latent channel dimension of each inverted variable token. A learnable static spectral response and an input-dependent gate modulate the spectral coefficients, while multiplicative fusion with a feature-domain projection provides nonlinear within-token refinement.

(3) **Backbone-agnostic generality.** We empirically evaluate SICM as a plug-in cross-series mixer across multiple forecasting backbones, datasets, and prediction horizons. The results show that SICM can be integrated with different architectures without changing their basic forecasting pipelines, while improving predictive performance in most forecasting scenarios.

2 Related Work

We defer the background and related work of Givens rotations to Appendix A.3 and focus this section exclusively on multivariate time series analysis.

Channel-Independent and Inverted Transformer Architectures Early Transformer-based forecasting models (e.g., Autoformer (Wu et al., 2021), Fedformer (Zhou et al., 2022), PatchTST (Nie et al., 2023)) primarily tokenized time points or sub-series patches along the temporal axis. iTransformer instead embeds each complete series history as a token and applies attention across inverted variable tokens N (Liu et al., 2024). Building upon this foundation, subsequent methods such as Vcformer (Yang et al., 2024), DUET (Qiu et al., 2025), Soatten (Wu, 2025), EiT (Li & Yang, 2025), RoadDiff (Li et al., 2025a) and FAiT (He et al., 2026) introduced further improvements. While iTransformer established strong results on multiple benchmarks, its standard full-attention mechanism possesses an $\mathcal{O}(N^2)$ complexity and is unconstrained, often introducing noise when cross-channel dependencies are sparse or non-stationary (Ma et al., 2025b; Yu et al., 2026).

Adaptive Graph Learning and Spatio-Temporal Dependency Modeling ST-GNNs, such as Graph WaveNet (Wu et al., 2019), MTGNN (Wu et al., 2020), CrossGNN (Huang et al., 2023), STAGANN (Li et al., 2025b) and CGTFra (Yu et al., 2026), model physical dependencies by estimating adaptive adjacency matrices $\mathbf{A} \in \mathbb{R}^{N\times N}$. Norm-bounded graph attention has also been studied with generalization guarantees for multivariate forecasting (Zhang et al., 2024). Recently, MAGE (Ma et al., 2025b) demonstrated that excessive graph parameterization and complex non-linear message passing can impair generalizability due to over-smoothing and noise amplification, proposing linear adaptive graph convolutions to stabilize training. More recently, SEED (Xiong et al., 2026) introduced a spectral entropy-guided perspective, demonstrating that the structural complexity and entropy of spatiotemporal graphs directly dictate forecasting generalizability.

Although these methods enhance inter-channel dependency modeling from different aspects, they still rely on learnable dense attention maps or adjacency matrices, utilizing Softmax to compress interactions across all variables and aggregating multiple channels into a weighted sum. **However, when certain variables are inherently independent of the rest, such forced all-to-all aggregation tends to overfit spurious correlations and causes destructive inter-channel signal cancellation. Furthermore, unconstrained linear mixtures lack orthogonal and energy-preserving constraints, leading to representation distortion (Huang et al., 2022; Zhang & Martens, 2026) while imposing an $\mathcal{O}(N^2)$ computational overhead.** To address these limitations, we propose `SICM`, which establishes a multi-scale sparse propagation path via multi-stage, iterative grouped Givens rotations to progressively model pairwise variable interactions, with the rotation angles dynamically determined by the pairwise interaction energy. Theoretical analysis demonstrates that `SICM` achieves favorable computational complexity while guaranteeing strict orthogonality and structural energy preservation.

3 METHODOLOGY

3.1 PROBLEM FORMULATION AND INVERTED TOKENIZATION

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]^\top \in \mathbb{R}^{L\times N}$ denote a multivariate historical time series over a lookback window of length L across N variates. The objective of MTSF is to forecast the future values $\mathbf{Y} = [\mathbf{x}_{L+1}, \ldots, \mathbf{x}_{L+T}]^\top \in \mathbb{R}^{T\times N}$ over a forecast horizon T.

Following the reversible instance normalization (RevIN) paradigm (Kim et al., 2022), we compute the channel-wise temporal mean $\boldsymbol{\mu} \in \mathbb{R}^{1\times N}$ and standard deviation $\boldsymbol{\sigma} \in \mathbb{R}^{1\times N}$ to normalize $\mathbf{X}$: $\widetilde{\mathbf{X}} = (\mathbf{X} - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma} \in \mathbb{R}^{L\times N}$, where $\oslash$ denotes element-wise division. In accordance with the inverted tokenization strategy, the normalized time series of the i-th variable across all L timestamps, $\widetilde{\mathbf{x}}_{:,i} \in \mathbb{R}^L$, is projected into a D-dimensional latent space:

$$\mathbf{h}_i^{(0)} = \text{Dropout}(\text{LayerNorm}(\mathbf{W}_{\text{emb}}\widetilde{\mathbf{x}}_{:,i} + \mathbf{b}_{\text{emb}})) \in \mathbb{R}^D, \quad \forall i \in \{1, \ldots, N\}, \tag{1}$$

forming the initial token matrix $\mathbf{H}^{(0)} = [\mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_N^{(0)}]^\top \in \mathbb{R}^{N\times D}$. The representation $\mathbf{H}^{(0)}$ is subsequently processed by a stack of E SICD blocks. For the ℓ-th block, the input $\mathbf{H}^{(\ell-1)}$ first undergoes structured inter-channel mixing `SICM`, producing $\mathbf{H}^{(\ell)}_{\text{mixed}}$. The mixed variable tokens are then refined along the latent dimension by the `SGM`:

$$\mathbf{H}^{(\ell)} = \mathbf{H}^{(\ell)}_{\text{mixed}} + \text{SGM}(\text{LayerNorm}(\mathbf{H}^{(\ell)}_{\text{mixed}})). \tag{2}$$

Algorithm 1 summarizes the complete forward pass of one SICD block. The structured mixer is detailed in Section 3.2. And the overall `SICDNet` framework is shown in Figure 2.

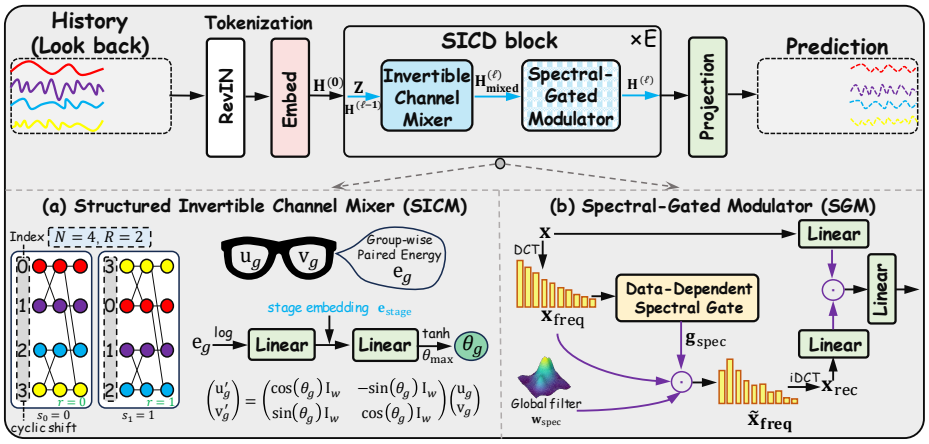


Figure 2: Illustration of SICDNet. (a) Structured Invertible Channel Mixer (SICM) executes multi-round Givens rotations conditioned on rotation-invariant pair energy. (b) Intra-series Spectral-Gated Modulator (SGM) executes dynamic spectral filtering and spatial-frequency gated fusion.

3.2 Inter-Channel Structured Invertible Mixer (SICM)

Instead of dense attention $\mathrm{Softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{D})\mathbf{V}$ or explicit graph aggregation, the SICM establishes cross-channel communication through a structured sequence of dyadic, state-conditioned Givens rotations. Let $\mathbf{Z} \in \mathbb{R}^{N\times D}$ denote the current token matrix within the mixer. Each row of $\mathbf{Z}$ represents one channel (i.e., variable), and the mixer updates selected pairs of rows while preserving the overall tensor shape. An overview of the algorithmic details of SICM is provided in Algorithm 1.

Dyadic Multi-Scale Pairing Schedule. For N variables ($N \geq 2$), each round $r \in \{0, \ldots, R-1\}$ consists of $S_{\text{stage}} = \lceil \log_2 N \rceil$ hierarchical stages. Before the stages of round r, a cyclic permutation shift $s_r = r \pmod N$ is applied to $\mathbf{Z}$ along the variable dimension. The dyadic pairing schedule is subsequently applied to this permuted variable order. In our implementation, the pairing schedule below uses zero-based variable indices $j \in \{0, \ldots, N-1\}$. At level (stage) $k \in \{0, \ldots, S_{\text{stage}} - 1\}$, with stride $\Delta = 2^k$, the paired index sets $(\mathcal{I}^{(k)}_{\text{left}}, \mathcal{I}^{(k)}_{\text{right}})$ are defined as

$$\mathcal{I}^{(k)}_{\text{left}} = \{j = 2m\Delta + p \mid 0 \leq j < N - \Delta,\ m \in \mathbb{N}_0,\ 0 \leq p < \Delta\}, \tag{3}$$

$$\mathcal{I}^{(k)}_{\text{right}} = \left\{ j + \Delta \;\middle|\; j \in \mathcal{I}^{(k)}_{\text{left}} \right\}. \tag{4}$$

Each index $j \in \mathcal{I}^{(k)}_{\text{left}}$ is paired with the index $j + \Delta \in \mathcal{I}^{(k)}_{\text{right}}$. At the entrance to stage k, the associated variable representations are gathered from the current token matrix as $\mathbf{u} = \mathbf{Z}[j, :], \mathbf{v} = \mathbf{Z}[j + \Delta, :], \mathbf{u}, \mathbf{v} \in \mathbb{R}^D$. Thus, Eqs. (3)–(4) determine which rows of $\mathbf{Z}$ communicate at the current scale. All valid pairs within the same stage are disjoint and can therefore be processed in parallel.

Channel Grouping and Group-Wise Givens Rotation. For each paired representation $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ extracted above, the channel dimension D is partitioned into G disjoint groups, each of width $W = D/G$. The paired vectors are reshaped into $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{G\times W}$, where $\mathbf{u}_g, \mathbf{v}_g \in \mathbb{R}^W$ denote the g-th channel groups for $g \in \{1, \ldots, G\}$.

For the pair anchored at $j \in \mathcal{I}^{(k)}_{\text{left}}$, let $\theta^{(r,k)}_{j,g}$ denote the rotation angle for channel group g, where the paired variables are j and $j + \Delta$. In the following, we abbreviate $\theta^{(r,k)}_{j,g}$ to θ_g for notational clarity. The angle may vary across valid anchor indices j, channel groups g, rounds r, and stages k, although the angle-generation network is shared across pairs. Once generated, the scalar angle is shared across the W coordinates within the corresponding channel group, as indicated by $\mathbf{I}_W$. For each group g, the state-conditioned rotation angle θ_g is applied as

$$\begin{bmatrix} \mathbf{u}'_g \\ \mathbf{v}'_g \end{bmatrix} = \begin{bmatrix} \cos(\theta_g)\mathbf{I}_W & -\sin(\theta_g)\mathbf{I}_W \\ \sin(\theta_g)\mathbf{I}_W & \cos(\theta_g)\mathbf{I}_W \end{bmatrix} \begin{bmatrix} \mathbf{u}_g \\ \mathbf{v}_g \end{bmatrix}. \tag{5}$$

Here, $\mathbf{I}_W \in \mathbb{R}^{W\times W}$ denotes the W-dimensional identity matrix. Thus, the block matrix in Eq. (5) is a $2W \times 2W$ orthogonal rotation, which applies the same scalar angle θ_g to the corresponding coordinates of $\mathbf{u}_g$ and $\mathbf{v}_g$. The rotated groups are concatenated to reconstruct the updated D-dimensional representations: $\mathbf{u}' = \left[(\mathbf{u}'_1)^\top, \ldots, (\mathbf{u}'_G)^\top\right]^\top, \mathbf{v}' = \left[(\mathbf{v}'_1)^\top, \ldots, (\mathbf{v}'_G)^\top\right]^\top$. They are then written back to the same paired positions: $\mathbf{Z}[j,:] \leftarrow \mathbf{u}', \mathbf{Z}[j+\Delta,:] \leftarrow \mathbf{v}', j \in \mathcal{I}^{(k)}_{\text{left}}$. Variables whose indices do not belong to $\mathcal{I}^{(k)}_{\text{left}} \cup \mathcal{I}^{(k)}_{\text{right}}$ remain unchanged at the current stage. After all valid pairs have been updated, the resulting $\mathbf{Z}$ becomes the input to stage $k+1$. Once all S_{stage} stages in round r have been completed, the cyclic shift is reversed. The restored representation is then passed to the next round. A concrete example is in Appendix A.7 to illustrate the modeling strategy in SICM.

Invariant Angle Conditioner and Exact Invertibility. Because the rotation angles are conditioned on the current paired representations, exact inversion would ordinarily require caching the angles generated during the forward pass. To avoid this requirement, the angle conditioner is constructed solely from the group-wise paired energy. **Property 1** shows that this conditioning statistic is exactly preserved by Eq. (5), allowing the same angles to be recomputed from the rotated outputs. Further characteristics of SICM are detailed in Appendix A.4.

Property 1 (**Energy Invariance under Group-Wise Givens Rotation**). *Let $\mathbf{u}_g, \mathbf{v}_g \in \mathbb{R}^W$ be transformed into $\mathbf{u}'_g, \mathbf{v}'_g$ via Eq. (5). The group-wise normalized paired energy e_g satisfies:*

$$e'_g = \frac{1}{W}\left(\|\mathbf{u}'_g\|_2^2 + \|\mathbf{v}'_g\|_2^2\right) = \frac{1}{W}\left(\|\mathbf{u}_g\|_2^2 + \|\mathbf{v}_g\|_2^2\right) = e_g. \tag{6}$$

For a fixed input instance and SICM layer, consider the variable pair anchored at $j \in \mathcal{I}^{(k)}_{\text{left}}$ at stage k of round r. Let $\mathbf{e}^{(r,k)}_j = [e^{(r,k)}_{j,1}, \ldots, e^{(r,k)}_{j,G}]^\top \in \mathbb{R}^G$ collect its current group-wise paired energies, as defined in Eq. (6). By **Property 1**, this vector is invariant under the corresponding group-wise rotations. We combine these pair-specific energy features with a learnable stage-specific embedding $\mathbf{e}^{(r,k)}_{\text{stage}} \in \mathbb{R}^{d_h}$ to generate the angle vector $\boldsymbol{\theta}^{(r,k)}_j = [\theta^{(r,k)}_{j,1}, \ldots, \theta^{(r,k)}_{j,G}]^\top \in \mathbb{R}^G$:

$$\mathbf{z}^{(r,k)}_{e,j} = \text{LayerNorm}(\log(1 + \mathbf{e}^{(r,k)}_j)), \tag{7}$$

$$\boldsymbol{\theta}^{(r,k)}_j = \theta_{\max} \cdot \tanh(\mathbf{W}_\theta \,\text{GELU}(\mathbf{W}_e \mathbf{z}^{(r,k)}_{e,j} + \mathbf{b}_e + \mathbf{e}^{(r,k)}_{\text{stage}}) + \mathbf{b}_\theta). \tag{8}$$

Here, $\mathbf{W}_e \in \mathbb{R}^{d_h \times G}$, $\mathbf{b}_e \in \mathbb{R}^{d_h}$, $\mathbf{W}_\theta \in \mathbb{R}^{G\times d_h}$, and $\mathbf{b}_\theta \in \mathbb{R}^G$ are shared across input instances, variable pairs, rounds, and stages within each SICM layer. The embedding $\mathbf{e}^{(r,k)}_{\text{stage}}$ is shared across all pairs and input instances at the same round–stage combination. The scalar $\theta_{\max} \in (0, \pi/2]$ bounds the rotation angles. Sample and layer indices are omitted throughout this formulation for readability.

3.3 Intra-Series Spectral-Gated Modulator (SGM)

The input is first processed by the SICM to explicitly capture inter-series dependencies, yielding an intermediate representation $\mathbf{H}_{\text{mixed}} \in \mathbb{R}^{N\times D}$. Subsequently, we design the Spectral-Gated Modulator (SGM) to further extract intra-series pattern fluctuations in the frequency domain. Specifically, applying the orthonormal Type-II Discrete Cosine Transform (DCT-II) along the temporal dimension D effectively maps the continuous temporal basis into orthogonal harmonic orders. Owing to the energy compaction property of DCT-II, dominant seasonalities and low-frequency trends are concentrated into compact spectral coefficients. Since the DCT operates on each variable independently, for notational simplicity, dimension N is omitted in subsequent SGM formulation.

For an input token $\mathbf{x} \in \mathbb{R}^D$, the Type-II DCT $\mathbf{x}_{\text{freq}} = \mathcal{DCT}(\mathbf{x}) \in \mathbb{R}^D$ is defined as

$$[\mathbf{x}_{\text{freq}}]_\nu = \alpha_\nu \sum_{t=0}^{D-1} [\mathbf{x}]_t \cos\left(\frac{\pi(2t+1)\nu}{2D}\right), \qquad \nu \in \{0, \ldots, D-1\}, \tag{9}$$

where ν denotes the frequency-bin index, $\alpha_0 = 1/\sqrt{D}$, and $\alpha_\nu = \sqrt{2/D}$ for $\nu \geq 1$.

Adaptive Spectral Modulation. The frequency spectrum $\mathbf{x}_{\text{freq}}$ is modulated via two complementary factors: (1) Learnable Static Spectral Prior: A trainable weight vector $\mathbf{w}_{\text{spec}} \in \mathbb{R}^D$ initialized around

1.0 and broadcast over the batch and variable dimensions, acting as a learnable global frequency filter. (2) Dynamic Data-Dependent Spectral Gate: A lightweight bottleneck MLP mixes information across latent spectral bins and computes instance-specific spectral gating weights to dynamically emphasize informative spectral components and suppress irrelevant frequency variations:

$$\mathbf{g}_{\text{spec}} = 2 \cdot \sigma(\mathbf{W}_{g2}\text{GELU}(\mathbf{W}_{g1}\mathbf{x}_{\text{freq}})) \in \mathbb{R}^D, \tag{10}$$

where $\mathbf{W}_{g1} \in \mathbb{R}^{d_{\text{gate}} \times D}$, $\mathbf{W}_{g2} \in \mathbb{R}^{D \times d_{\text{gate}}}$, with bottleneck dimension d_{gate}. The modulated spectrum is activated non-linearly, followed by the inverse DCT to reconstruct the temporal representations:

$$\widetilde{\mathbf{x}}_{\text{freq}} = \text{GELU}(\mathbf{x}_{\text{freq}} \odot \mathbf{w}_{\text{spec}} \odot \mathbf{g}_{\text{spec}}) \in \mathbb{R}^D, \tag{11}$$

$$[\mathbf{x}_{\text{rec}}]_t = \sum_{\nu=0}^{D-1} \alpha_\nu [\widetilde{\mathbf{x}}_{\text{freq}}]_\nu \cos\left(\frac{\pi(2t+1)\nu}{2D}\right), \quad t \in \{0, \ldots, D-1\}. \tag{12}$$

Dual-Domain Gated Fusion. We propose a dual-domain gated fusion mechanism that collaboratively combines the globally filtered spectral features with the hidden representations. Specifically, the reconstructed spectral signal $\mathbf{x}_{\text{rec}}$ and the original temporal input $\mathbf{x}$ are projected into a shared space:

$$\mathbf{h}_{\text{freq}} = \mathbf{W}_{\text{freq}}\mathbf{x}_{\text{rec}} + \mathbf{b}_{\text{freq}} \in \mathbb{R}^{d_{\text{ff}}}, \tag{13}$$

$$\mathbf{h}_{\text{time}} = \text{GELU}(\mathbf{W}_{\text{time}}\mathbf{x} + \mathbf{b}_{\text{time}}) \in \mathbb{R}^{d_{\text{ff}}}. \tag{14}$$

Subsequently, the two representations are fused through element-wise multiplicative modulation and the fused representation is projected back to the original dimension:

$$\text{SGM}(\mathbf{x}) = \text{Dropout}(\mathbf{W}_{\text{out}}(\mathbf{h}_{\text{freq}} \odot \mathbf{h}_{\text{time}}) + \mathbf{b}_{\text{out}}) \in \mathbb{R}^D. \tag{15}$$

3.4 Forecasting and Loss Function

Forecasting Projection. Given the final output $\mathbf{H}^{(E)} \in \mathbb{R}^{N \times D}$ from the E-layer SICD encoder, a shared linear projection head maps representations to predictions: $\widehat{\mathbf{Y}}_{\text{norm}} = \text{LayerNorm}(\mathbf{H}^{(E)})\mathbf{W}_{\text{proj}} + \mathbf{b}_{\text{proj}} \in \mathbb{R}^{N \times T}$. Transposing to $\mathbb{R}^{T \times N}$ and inverting the instance normalization yields the final forecast: $\widehat{\mathbf{Y}} = \widehat{\mathbf{Y}}_{\text{norm}}^\top \odot \boldsymbol{\sigma} + \boldsymbol{\mu} \in \mathbb{R}^{T \times N}$.

Joint Objective Optimization. For the regularizer, let $s = rS_{\text{stage}} + k + 1 \in \{1, \ldots, RS_{\text{stage}}\}$ denote the one-based global stage index corresponding to round r and level k. We denote the set of disjoint variable pairs processed at stage s by $\mathcal{P}_s = \left\{(u, v) \;\middle|\; u \in \mathcal{I}_{\text{left}}^{(k)},\ v = u + 2^k\right\}$. The indices in $\mathcal{P}_s$ refer to positions in the round-r cyclically shifted variable order. Hence, $|\mathcal{P}_s|$ is the number of valid variable pairs at stage s. The angle regularization penalty $\mathcal{L}_{\text{angle}}$ averages the squared rotation angles across layers, channel groups, stages, active variable pairs, and samples:

$$\mathcal{L}_{\text{angle}} = \frac{1}{E \cdot R \cdot S_{\text{stage}}} \sum_{\ell=1}^{E} \sum_{s=1}^{R \cdot S_{\text{stage}}} \left(\frac{1}{B \cdot |\mathcal{P}_s| \cdot G} \sum_{b=1}^{B} \sum_{(u,v) \in \mathcal{P}_s} \sum_{g=1}^{G} \left(\theta_{b,u,v,g,s}^{(\ell)}\right)^2 \right). \tag{16}$$

The entire framework is optimized end-to-end via:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{forecast}} + \lambda \mathcal{L}_{\text{angle}}. \tag{17}$$

where $\mathcal{L}_{\text{forecast}}$ is the loss between $\mathbf{Y}$ and $\widehat{\mathbf{Y}}$, and $\lambda > 0$ is a hyperparameter balancing forecasting precision and structural sparsity.

4 Experiments

4.1 Experimental Setup

Datasets. To comprehensively evaluate the effectiveness of `SICDNet`, we conduct extensive experiments on 13 widely-used benchmark datasets in time series forecasting. These datasets encompass both long-term forecasting benchmarks (i.e., ETTm1, ETTm2, ETTh1, ETTh2, Exchange, Weather, Electricity, Solar-Energy, Traffic) and short-term forecasting benchmarks (i.e., PEMS03, PEMS04, PEMS07, and PEMS08). Detailed descriptions of all datasets are provided in Appendix A.5.

Baselines and Experimental Settings. We compare our SICDNet against nine representative baselines, which are broadly divided into two categories: (1) methods emphasizing inter-series dependency modeling: DUET (Qiu et al., 2025), Soatten (Wu, 2025), Leddam (Yu et al., 2024), iTransformer (Liu et al., 2024), MSGNet (Cai et al., 2024), and (2) methods focusing on intra-series pattern extraction: SSformer (Liu et al., 2026), TimePro (Ma et al., 2025c), PatchTST (Nie et al., 2023), TimesNet (Wu et al., 2023). We provide experimental settings in Appendix A.6.

4.2 Main Results

Table 1 summarizes the averaged performance comparison for long-term forecasting. The empirical results demonstrate that SICDNet achieves consistently superior predictive performance across diverse benchmarks, ranging from datasets characterized by challenging temporal patterns (e.g., ETTh1 and Solar) to high-dimensional datasets with complex inter-variable dependencies (e.g., Weather and Traffic). Notably, compared to DUET, which jointly models both intra- and inter-channel dependencies, SICDNet reduces the forecasting MSE by 3.6% and 4.7% on ETTh1 and Traffic, respectively. The short-term forecasting results are provided in Appendix A.8.

Table 1: Averaged long-term forecasting results with forecasting horizons $T \in \{96, 192, 336, 720\}$ and fixed look-back length L=96. **Bold**/second: Best/second best one. "-" indicates that the original method was not evaluated in the corresponding scenario. The standard deviations are within 0.005 for MSE and within 0.003 for MAE.

Models	**SICDNet** (ours)		SSformer (2026)		DUET (2025)		TimePro (2025)		Soatten (2025)		iTransformer (2024)		MSGNet (2024)		Leddam (2024)		PatchTST (2023)		TimesNet (2023)	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.380**	**0.381**	**0.380**	0.394	0.390	0.393	0.391	0.400	0.394	0.402	0.407	0.410	0.398	0.411	0.386	0.397	0.387	0.400	0.400	0.406
ETTm2	**0.275**	**0.317**	0.279	0.322	0.280	0.324	0.281	0.326	0.287	0.331	0.288	0.332	0.288	0.330	0.281	0.325	0.281	0.326	0.291	0.333
ETTh1	**0.427**	**0.426**	0.452	0.443	0.443	0.436	0.438	0.438	0.447	0.440	0.454	0.447	0.452	0.452	0.431	0.429	0.469	0.454	0.458	0.450
ETTh2	0.364	0.392	**0.359**	**0.389**	0.372	0.397	0.377	0.403	0.379	0.405	0.383	0.407	0.396	0.417	0.373	0.399	0.387	0.407	0.414	0.427
Exchange	0.322	0.386	0.362	0.410	**0.318**	**0.384**	0.352	0.399	0.359	0.404	0.360	0.403	0.399	0.430	-	-	0.367	0.404	0.416	0.443
Weather	**0.237**	**0.260**	**0.237**	0.268	0.251	0.273	0.251	0.276	0.245	0.273	0.258	0.278	0.249	0.278	0.242	0.272	0.259	0.281	0.259	0.287
Electricity	**0.165**	**0.256**	0.170	0.265	0.172	0.258	0.169	0.262	0.166	0.259	0.178	0.270	0.194	0.300	0.169	0.263	0.205	0.290	0.192	0.295
Solar	**0.224**	**0.226**	0.228	0.266	0.237	0.233	0.232	0.266	0.229	0.261	0.233	0.262	-	-	0.230	0.264	0.270	0.307	0.301	0.319
Traffic	0.430	**0.262**	0.452	0.299	0.451	0.269	-	-	0.437	0.286	**0.428**	0.282	-	-	0.467	0.294	0.555	0.362	0.620	0.336
1^{st} Count	**6**	**7**	3	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0

4.3 Framework Generality

To verify the generalizability of SICM's inter-channel dependency modeling capability, we substitute the inter-series dependency modules in four representative architectures—DUET, iTransformer, SOFTS (Han et al., 2024) and VCformer—with SICM, and conduct comparative experiments using their publicly released hyperparameters under identical hardware environments. Table 2 summarizes the averaged results across four forecasting scenarios. Detailed introductions to these baselines and the complete experimental results are provided in Appendix A.9. The incorporation of SICM brings performance gains across the majority of diverse datasets, achieving up to a 9.14% reduction in MSE (VCformer on the ECL). Nonetheless, we also observe that the improvements on Solar and Traffic are somewhat unstable; for instance, the full results in Table 7 indicate that SICM enhances DUET's forecasting performance on Solar for prediction horizons $T \in \{96, 192\}$. We acknowledge that consistently boosting performance across such massive-scale datasets under strictly identical hyperparameter configurations remains non-trivial. We leave an in-depth investigation of the underlying mechanisms and tailored architectural improvements for future work.

4.4 Inference-based analysis of SICM

We further examine whether the predictive improvement of SICM is attributable to its learned cross-variable rotations. To isolate the contribution of SICM, we construct an identity-control model (from the same trained checkpoint), where all learned Givens angles are set to zero during inference, i.e., $\theta^{(i,j)}_{\ell,r,k,g} \leftarrow 0$, while keeping all other components frozen. Consequently, the identity control removes the cross-variable rotations without changing the rest of the predictor. Figure 3 summarizes the resulting evidence on the entire test set of the ETTm1 dataset. Further analyses of SICM are provided in Appendix A.10.

Table 2: Verification of Framework Generality. "-" denotes that the original method was not evaluated on certain datasets, or that we encountered out-of-memory issues. To ensure a fair comparison, the results for DUET and iTransformer in Table 1 are directly adopted from their officially published metrics, whereas the results in Table 2 are rigorously reproduced using their publicly released codebases under our unified environment.

Datasets		ETTm1		ETTm2		ETTh1		ETTh2		Exchange		Weather		ECL		Solar		Traffic	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DUET	original	0.391	0.394	0.279	0.322	0.449	0.440	**0.372**	0.398	0.309	0.380	0.247	0.270	**0.172**	**0.258**	0.241	0.246	**0.451**	**0.269**
	+ **SICM**	**0.382**	**0.389**	**0.276**	**0.319**	**0.440**	**0.433**	**0.372**	**0.397**	**0.302**	**0.375**	**0.234**	**0.259**	0.186	0.289	**0.226**	**0.231**	0.463	0.287
iTransformer	original	0.408	0.412	0.293	0.337	0.457	0.449	0.384	0.407	0.369	0.409	0.262	0.283	0.176	0.268	0.235	**0.261**	**0.422**	**0.282**
	+ **SICM**	**0.397**	**0.403**	**0.287**	**0.331**	**0.445**	**0.438**	**0.379**	**0.405**	**0.358**	**0.402**	**0.250**	**0.277**	**0.172**	**0.265**	**0.234**	0.266	0.431	0.288
SOFTS	original	**0.393**	0.404	0.293	0.331	0.452	0.445	0.381	0.404	-	-	0.260	0.281	0.175	0.266	0.231	**0.257**	**0.414**	**0.271**
	+ **SICM**	**0.393**	**0.401**	**0.283**	**0.326**	**0.444**	**0.440**	**0.373**	**0.402**	-	-	**0.251**	**0.278**	**0.169**	**0.264**	**0.230**	0.265	0.429	0.281
VCformer	original	0.404	0.406	0.292	0.334	0.488	0.460	0.384	0.405	0.358	0.403	0.269	0.286	0.186	0.278	-	-	-	-
	+ **SICM**	**0.395**	**0.400**	**0.285**	**0.328**	**0.456**	**0.438**	**0.381**	**0.402**	**0.353**	**0.400**	**0.260**	**0.284**	**0.169**	**0.262**	-	-	-	-

Global Impact and Local Temporal Gains. As shown in Figure 3(a), the original SICM obtains an MAE of 0.3397, whereas the identity control obtains an MAE of 0.3486, corresponding to a relative increase of approximately 2.60%. The strictly positive 95% moving-block bootstrap interval confirms that the model functionally relies on the cross-variable mixing mechanism to minimize prediction errors under the same trained checkpoint. The horizon-resolved comparison in Figure 3(b) provides a more detailed view of this effect. The plotted quantity is $\Delta_h(i) = \mathrm{MAE}_{\mathrm{identity}}(i,h) - \mathrm{MAE}_{\mathrm{SICM}}(i,h)$, where i denotes the target variable and h denotes the forecasting horizon. The predominantly positive regions across target variables and forecast horizons show that the magnitude of the gain varies across variables and horizons.

Verifying Cross-Channel Information Routing. To examine the structure of the learned communication, we replace one source variable at a time, keep all other input variables unchanged, and rerun the same frozen predictor to measure the resulting change in the MAE of each target variable. For each source variable c, its historical input is replaced by a training-set donor history by matching its historical mean, standard deviation, and time of day. The entry at row j and column c is $\Delta_{j\leftarrow c} = \mathrm{MAE}_j^{\mathrm{replaced}(c)} - \mathrm{MAE}_j^{\mathrm{SICM}}$. Positive values indicate increased target error after replacing source c, thereby measuring predictive reliance on that source history. The off-diagonal heatmaps in Figure 3(c, d) therefore summarize the (averaged) target-specific sensitivity to each replaced source variable, with the structured responses of the original SICM model contrasting with the weaker responses of the identity control. This demonstrates the presence of non-uniform, directional predictive dependencies within the original SICM, as opposed to homogeneous interactions across all variables.

Figure 4 visualizes the θ values in the hidden layers during test-set inference. The obtained θ by SICM exhibit a heterogeneous allocation across encoder layers, dyadic stages, variable pairs and channel groups. This pattern indicates that SICM implements a structured, layer-dependent and group-specific communication operator rather than a shared static mixing matrix. And the diverse angular distributions (including both positive and negative rotations across various layers and groups) provide corroborating evidence that SICM is immune to destructive signal cancellation and over-smoothing.

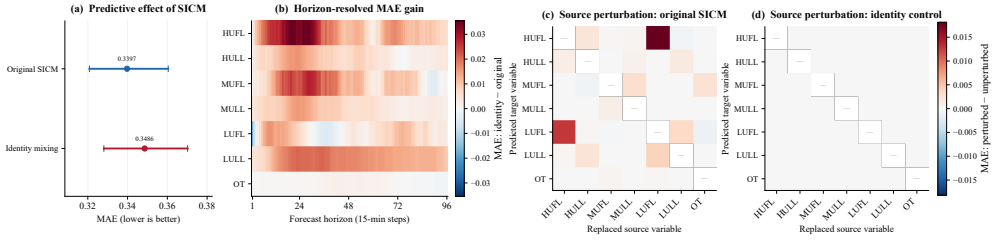


Figure 3: Functional contribution and source dependence of SICM. All panels use the same trained checkpoint evaluated on **the whole 11,425 test windows** with a 96-to-96 forecasting setting. (a) Mean MAE and 95% circular moving-block bootstrap intervals for the original SICM and the identity-mixing intervention, which sets all Givens angles to zero. (b) Horizon- and target-specific MAE difference $\mathrm{MAE}^{\mathrm{identity}} - \mathrm{MAE}^{\mathrm{SICM}}$. (c) Change in target MAE after replacing one source history under the original SICM. (d) The same intervention under identity mixing.

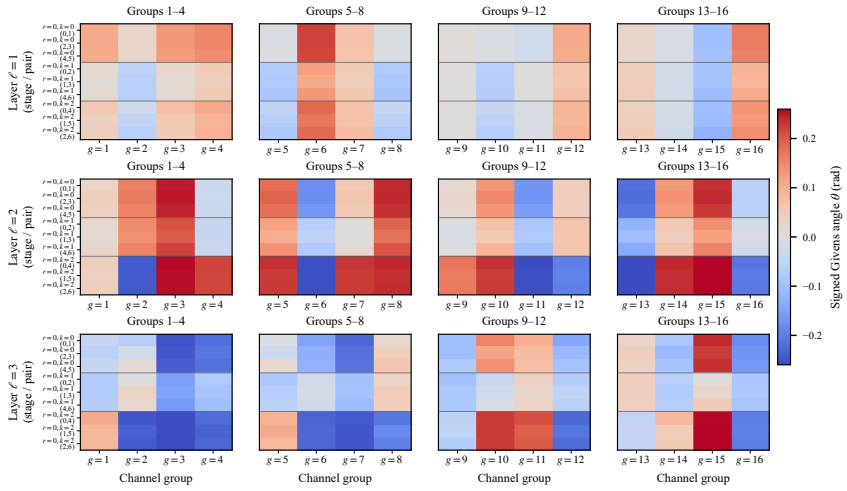


Figure 4: Layer- and group-wise Givens-angle allocation (values averaged across all test samples).

4.5 Ablation Study

We conducted ablation studies in a coarse-to-fine manner. Initially, we performed module-level ablations by removing the SICM and/or SGM to validate the effectiveness of inter-series and intra-series dependency modeling. Furthermore, to investigate the internal mechanisms of the SICM, we designed the following ablations: (1) **w/o pair-energy cond.**: We remove $\mathbf{W}_e\mathbf{z}_{e,j} + \mathbf{b}_e$ in Eq. (8) and only retain $\mathbf{e}_{\text{stage}}$; (2) **w/o stage embedding**: Remove $\mathbf{e}_{\text{stage}}$ in Eq. (8); (3) **w/o multiple rounds**: We remove the multi-round strategy while retaining the multi-stage variate-pair scheduling; (4) **fixed dense pairing schedule**: We replace the multi-stage, multi-round strategy with a dense self-attention layer, where every channel pair interacts with each other. The average experimental results across four diverse datasets are summarized in Table 3. And the complete ablation results and detailed analyses are provided in Appendix A.8. Overall ablations confirm the indispensability of both intra- and inter-series modeling. In-depth analysis of SICM shows that pairwise energy conditioning, stage embeddings, and multi-round rotations are all crucial for performance enhancement. The significant performance drop caused by fixed dense pairing further proves that input-adaptive structured interactions outperform dense channel connections. Meanwhile, ablating SGM reveals that omitting spectral gating and global filtering causes only marginal performance degradation.

Table 3: Ablation study of the complete SICDNet architecture. All results are validation-set averages over $T \in \{96, 192, 336, 720\}$. Each entry reports MSE/MAE. Full results are in Appendix A.8.

	SICDNet Variant	ETTh1	Weather	ECL	Traffic
	Full SICDNet	**.427 / .426**	**.237 / .260**	**.165 / .256**	**.430 / .262**
Core	w/o SICM	.428 / .427	.265 / .275	.189 / .271	.458 / .277
	w/o SGM	.445 / .431	.252 / .275	.208 / .289	.511 / .313
	w/o SICM & SGM	.446 / .431	.278 / .288	.218 / .294	.599 / .359
SICM	w/o pair-energy cond.	.428 / .427	.241 / .263	.174 / .264	.445 / .270
	w/o stage embedding	.428 / .427	.239 / .262	.167 / .258	.437 / .266
	w/o multiple rounds	.432 / .430	.245 / .266	.168 / .260	.444 / .271
	fixed dense pairing	.436 / .431	.252 / .270	.184 / .268	.453 / .274
SGM	w/o Spectral Gate	.429 / .427	.239 / .263	.168 / .258	.434 / .264
	w/o Global Filter	.428 / .427	.240 / .263	.167 / .258	.434 / .264
	w/o both	.429 / .428	.244 / .266	.168 / .260	.437 / .266

5 Conclusion

In this work, we presented SICDNet, a multivariate time series forecasting architecture that fundamentally redesigns inter-series dependency mixing and intra-series feature extraction. By exploiting the energy invariance property of orthogonal transformations, the SICM achieves invertible, multi-scale cross-variable interactions in $\mathcal{O}(N \log N)$ complexity without storing explicit graph or attention matrices. Complementarily, the SGM use dynamic, real-valued DCT spectral gating and dual-domain fusion to capture complex temporal pattern in frequency domain. Another notable property is that we validate SICM as a universal paradigm for modeling inter-series correlation, analogous to the self-attention mechanism.

AI USE STATEMENT

During the preparation of this manuscript, AI-assisted tools were employed solely for language polishing and proofreading, including grammar checking, sentence restructuring, and standardizing mathematical expressions. The core methodology proposed in this work was developed entirely without the use of GPT or other generative models. Furthermore, no figures, tables, or experimental results were generated by GPT.

ETHICS STATEMENT

This work strictly adheres to the Code of Ethics. Our research does not involve human subjects, personal data, or animal testing. All experiments were conducted using publicly available benchmark datasets in compliance with their respective licenses and terms of use. The proposed methodology focuses on foundational algorithms, and we foresee no direct negative societal impacts, privacy violations, or ethical concerns associated with this work.

REPRODUCIBILITY STATEMENT

The proposed model, along with the complete pipelines for training, validation, and testing (including all involved hyperparameters), has been made publicly available via an anonymous repository (`https://anonymous.4open.science/r/SICDNet`). A step-by-step reproduction guide is provided in the README file. Notably, we also release the full source code (including all hyperparameters) for the generalizability evaluation (i.e., the experiments reported in Table 2 and Table 7), allowing readers to directly reproduce the results and verify the effectiveness and generalizability of the proposed `SICM` in modeling inter-channel dependencies.

REFERENCES

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11141–11149, 2024.

Wallace Givens. Computation of plain unitary rotations transforming a general matrix to triangular form. *Journal of the Society for Industrial and Applied Mathematics*, 6(1):26–50, 1958.

Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2): 205–224, 1965.

Kai Guo, Kaixiong Zhou, Xia Hu, Yu Li, Yi Chang, and Xin Wang. Orthogonal graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3996–4004, 2022.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 64145–64175. Curran Associates, Inc., 2024. doi: 10.52202/079017-2046. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/754612bde73a8b65ad8743f1f6d8ddf6-Paper-Conference.pdf`.

Peng He, Yao Liu, Yanglei Gan, Run Lin, Yuxiang Cai, and Qiao Liu. Fait: Frequency-aware inverted transformer for multivariate time series forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, pp. 1614–1625, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3818005. URL `https://doi.org/10.1145/3770855.3818005`.

Huaibo Huang, Xiaoqiang Zhou, and Ran He. Orthogonal transformer: An efficient vision transformer backbone with token orthogonalization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 14596–14607. Curran Associates, Inc., 2022. doi: 10.52202/068431-1061. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/5d8c01de2dc698c54201c1c7d0b86974-Paper-Conference.pdf`.

Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46885–46902. Curran Associates, Inc., 2023. doi: 10.52202/075280-2031. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9278abf072b58caf21d48dd670b4c721-Paper-Conference.pdf.

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

Shuhao Li, Weidong Yang, Yue Cui, Xiaoxing Liu, Lingkai Meng, Lipeng Ma, and Fan Zhang. Fine-grained traffic inference from road to lane via spatio-temporal graph node generation. KDD '25, pp. 1529–1540, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3736961. URL https://doi.org/10.1145/3711896.3736961.

Xin-Yi Li and Yu-Bin Yang. Enhanced inverted transformer: advancing variate token encoding and blending for time series forecasting. *Applied Intelligence*, 55(15):974, 2025.

Yujie Li, Shao Zezhi, Chengqing Yu, Tangwen Qian, Zhao Zhang, Yifan Du, Shaoming He, Fei Wang, and Yongjun Xu. Sta-gann: A valid and generalizable spatio-temporal kriging approach. CIKM '25, pp. 1726–1736, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400720406. doi: 10.1145/3746252.3761045. URL https://doi.org/10.1145/3746252.3761045.

Ying Liu, Bo Liu, Sheng Huang, Gang Luo, Wenbo Hu, Meng Wang, and Richang Hong. Sparse-scale transformer with bidirectional awareness for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23899–23907, 2026.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. Mofo: Empowering long-term time series forecasting with periodic pattern modeling. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 84871–84902. Curran Associates, Inc., 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/7a99ad21706dec5b28f9ad715e12197f-Paper-Conference.pdf.

Jiaming Ma, Binwu Wang, Guanjun Wang, Kuo Yang, Zhengyang Zhou, Pengkun Wang, Xu Wang, and Yang Wang. Less but more: Linear adaptive graph learning empowering spatiotemporal forecasting. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 58382–58418. Curran Associates, Inc., 2025b. doi: 10.52202/085713-1958. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/54c9bfb0885ae07f23607f617ab64c2b-Paper-Conference.pdf.

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*, 2025c. URL https://openreview.net/forum?id=s69Ei2VrIW.

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

Christopher C Paige and Michael A Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):43–71, 1982.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1185–1196, 2025.

Zongjiang Shang, Ling Chen, Binqing Wu, and Dongliang Cui. Ada-mshyper: Adaptive multi-scale hypergraph transformer for time series forecasting. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 33310–33337. Curran Associates, Inc., 2024. doi: 10.52202/079017-1050. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3a6935d11910d6f9142b0a1e36fc6753-Paper-Conference.pdf.

Zezhi Shao, Yujie Li, Fei Wang, Chengqing Yu, Yisong Fu, Tangwen Qian, Bin Xu, Boyu Diao, Yongjun Xu, and Xueqi Cheng. Blast: Balanced sampling time series corpus for universal forecasting models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 2502–2513, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3736860. URL https://doi.org/10.1145/3711896.3736860.

Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4): 551–566, 1993.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.127063. URL https://www.sciencedirect.com/science/article/pii/S0925231223011864.

Daoyu Wang, Mingyue Cheng, Zhiding Liu, and Qi Liu. TimeDART: A diffusion autoregressive transformer for self-supervised time series representation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=v2G9HML7ep.

Binqing Wu, Jianlong Huang, Zongjiang Shang, and Ling Chen. St-hyper: Learning high-order dependencies across multiple spatial-temporal scales for multivariate time series forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, pp. 3333–3343, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720406. doi: 10.1145/3746252.3761281. URL https://doi.org/10.1145/3746252.3761281.

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

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. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 1907–1913. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/264. URL https://doi.org/10.24963/ijcai.2019/264.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 753–763, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403118. URL https://doi.org/10.1145/3394486.3403118.

Feng Xiong, Zongxia Xie, Yanru Sun, Haoyu Wang, and Jianhong Lin. SEED: Spectral entropy-guided evaluation of spatial-temporal dependencies for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 27153–27161, 2026.

Yingnan Yang, Qingling Zhu, and Jianyong Chen. Vcformer: variable correlation transformer with inherent lagged correlation for multivariate time series forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/590. URL https://doi.org/10.24963/ijcai.2024/590.

Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57818–57841. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yu24s.html.

Wuqing Yu, Weichen Guo, Jian Zhou, Shuyu Luo, and Jiacai Zhang. Robust inter-series dependency modeling for time series forecasting via information-theoretic alignment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YQPQuHVLLQ.

Leo Zhang and James Martens. Orthogonal self-attention, 2026. URL https://arxiv.org/abs/2602.05996.

Zhi Zhang, Weijian Li, and Han Liu. Multivariate time series forecasting by graph attention networks with theoretical guarantees. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2845–2853. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/zhang24g.html.

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 APPENDIX

A.1 NOTATION AND DEFINITIONS

To enhance readability and clarity, we summarize the primary mathematical notations and symbols used throughout this paper in Table 4. Tensor dimensions are specified for a single input instance unless stated otherwise.

Table 4: Principal notation and definitions for `SICDNet`.

Notation	Definition
Forecasting task and representations	
$B,\ N$	Batch size and number of variables, respectively.
$L,\ T$	Historical look-back length and forecasting horizon.
$D,\ E$	Latent token dimension and number of stacked SICD blocks.
$\mathbf{X},\ \widetilde{\mathbf{X}}$	Historical observations and instance-normalized observations; both in $\mathbb{R}^{L\times N}$.
$\mathbf{Y},\ \widehat{\mathbf{Y}}$	Ground-truth future observations and final predictions; both in $\mathbb{R}^{T\times N}$.
$\boldsymbol{\mu},\ \boldsymbol{\sigma}$	Per-variable temporal mean and standard deviation of the historical window; both in $\mathbb{R}^{1\times N}$.
$\mathbf{H}^{(0)},\ \mathbf{H}^{(\ell)}$	Initial variable-token matrix and output of block ℓ; both in $\mathbb{R}^{N\times D}$, with $\ell = 1,\dots,E$.
$\mathbf{Z},\ \mathbf{H}^{(\ell)}_{\text{mixed}}$	Current mixer state and final `SICM` output in block ℓ; both in $\mathbb{R}^{N\times D}$.
$\widehat{\mathbf{Y}}_{\text{norm}}$	Normalized predictions in variable-major order, $\mathbb{R}^{N\times T}$.
Structured invertible channel mixer (`SICM`)	
$R,\ S_{\text{stage}}$	Number of mixing rounds and stages per round, where $S_{\text{stage}} = \lceil \log_2 N \rceil$.
$r,\ k$	Zero-based round and dyadic-level indices: $0 \le r < R$ and $0 \le k < S_{\text{stage}}$.
$G,\ W$	Number of latent feature groups and width of each group, with $W = D/G$.
$s_r,\ \Delta$	Round-wise cyclic shift $s_r = r \bmod N$ and dyadic pairing stride $\Delta = 2^k$.
$\tau,\ s$	Zero-based stage-embedding index $\tau = rS_{\text{stage}} + k$ and one-based regularization index $s = \tau + 1$.
$\mathcal{I}^{(k)}_{\text{left}},\ \mathcal{I}^{(k)}_{\text{right}}$	Paired index sets in the cyclically shifted variable order; each valid anchor j is paired with $j + 2^k$.
$\mathcal{P}_s$	Set of disjoint variable pairs processed at global stage s; $\lvert\mathcal{P}_s\rvert$ is its cardinality.
$\mathbf{u}, \mathbf{v};\ \mathbf{u}_g, \mathbf{v}_g$	Paired tokens in $\mathbb{R}^D$ and their g-th feature groups in $\mathbb{R}^W$; primes denote rotated outputs.
$\mathbf{e}^{(r,k)}_j,\ \mathbf{z}^{(r,k)}_{e,j}$	Pair-energy vector and its log-transformed, layer-normalized features; both in $\mathbb{R}^G$. Each component is $e_g = (\lVert\mathbf{u}_g\rVert_2^2 + \lVert\mathbf{v}_g\rVert_2^2)/W$.
$\mathbf{e}^{(r,k)}_{\text{stage}},\ d_h$	Learned round–stage embedding in $\mathbb{R}^{d_h}$ and hidden dimension of the angle conditioner.
$\boldsymbol{\theta}^{(r,k)}_j,\ \theta_{\max}$	Group-wise angle vector in $\mathbb{R}^G$ and its bound: $\lvert\theta^{(r,k)}_{j,g}\rvert \le \theta_{\max}$, with $0 < \theta_{\max} \le \pi/2$.
$P_r,\ F_{r,k},\ \mathcal{M}$	Round-wise cyclic permutation, state-conditioned stage transformation, and complete multi-round `SICM`, respectively.
Spectral-gated modulator (`SGM`)	
$\mathbf{x},\ \mathbf{x}_{\text{freq}}$	A normalized input token and its orthonormal DCT-II coefficients; both in $\mathbb{R}^D$. The transform acts along the latent feature axis.
$\nu,\ \alpha_\nu$	Spectral-bin index $0 \le \nu < D$ and DCT normalization.
$\mathbf{w}_{\text{spec}},\ \mathbf{g}_{\text{spec}}$	Learned static spectral response in $\mathbb{R}^D$ and input-dependent spectral gate in $(0,2)^D$.
$\widetilde{\mathbf{x}}_{\text{freq}},\ \mathbf{x}_{\text{rec}}$	Modulated spectral coefficients and their inverse-DCT reconstruction; both in $\mathbb{R}^D$.
$d_{\text{gate}},\ d_{\text{ff}}$	Spectral-gate bottleneck width, hidden width.
$\mathbf{h}_{\text{freq}},\ \mathbf{h}_{\text{time}}$	Reconstructed signal and temporal features in $\mathbb{R}^{d_{\text{ff}}}$, where d_{ff} is the expanded fusion dimension.

A.2 Example: Visualizing Signal Cancellation and Over-Smoothing

Standard message-passing and self-attention aggregate neighboring representations through convex or approximately convex weighted sums. Weighted-sum-based message passing is susceptible to **destructive interference** when input variables exhibit out-of-phase, opposite-sign, or heterogeneous local patterns; furthermore, the iterative application of similar aggregation weights across nodes progressively diminishes representational distinctions, inevitably culminating in **over-smoothing**.

To provide an intuitive understanding of the flaws inherent in standard message passing mechanisms (e.g., GNNs and Self-Attention) and the superiority of our `SICM`, we construct a synthetic example. As illustrated in Figure 5(a), we simulate two input latent temporal features, V_1 and V_2, belonging to two distinct variables ($V_1(t) = \sin(t)\, e^{-\frac{(t-5)^2}{4}}, V_2(t) = 0.8\sin(t+2.5)\, e^{-\frac{(t-5)^2}{4}}, t \in [0, 10]$.). These features exhibit out-of-phase temporal anomalies, representing heterogeneous local patterns. The y-axis, *Feature Amplitude*, denotes the activation magnitude of the latent representations, where the squared amplitude reflects the ℓ_2-energy of the signal.

The Failure of Standard GNN/Attention. Figure 5(b) visualizes the consequence of applying a standard attention mechanism, which updates node representations via convex combinations (i.e., weighted sums, $\sum A_{i,j}V_j$). When fusing the out-of-phase signals V_1 and V_2, the updated representations $Y_1^{(\text{Attn})}$ and $Y_2^{(\text{Attn})}$ risk a dual failure: ❶ **Destructive signal cancellation:** The amplitudes of both outputs are attenuated compared to the inputs, implying an irreversible loss of feature energy and critical time-series variations. ❷ **Over-smoothing:** The distinct characteristics of the original variables are washed out, rendering the two output trajectories nearly indistinguishable.

Energy-Preserving Mixing via `SICM`. In stark contrast, Figure 5(c) demonstrates the outputs processed by the orthogonal Givens rotations in `SICM`. By avoiding weighted additions, mathematically, the Givens rotation generates the output via $Y_1 = V_1 \cos\theta - V_2 \sin\theta$. Given that V_1 is negative and V_2 is positive (i.e., out-of-phase), the subtraction in the orthogonal basis *constructively* adds their magnitudes together (e.g., a negative value minus a positive value yields a larger negative value). This explicitly explains the deep negative trough of Y_1 observed in Figure 5(c).

We highlight two advantages of this behavior: **(1) Constructive amplitude concentration:** Unlike standard attention (Figure 5(b))), which performs scalar addition and destructively cancels out-of-phase signals towards 0, `SICM` reorganizes the coordinate axes. It successfully concentrates the dispersed temporal anomalies from both V_1 and V_2 into a single high-magnitude latent component (Y_1). **(2) Information reversibility:** In latent spaces, neural networks are largely sign-agnostic; a large negative amplitude carries the exact same amount of predictive information as a large positive one. Because the transformation is strictly orthogonal ($\|Y_1\|^2 + \|Y_2\|^2 = \|V_1\|^2 + \|V_2\|^2$), no information is permanently lost. The features can be inverted if necessary (For instance, the subsequent linear layer can readily invert the sign of the representations.), overcoming the irreversible information bottleneck inherent in standard graph message passing.

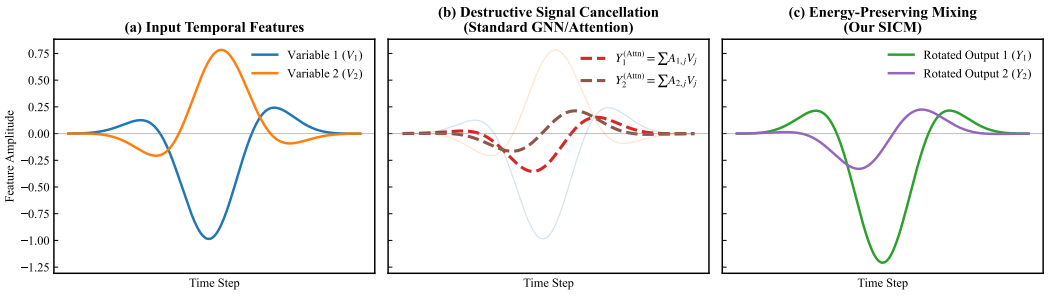


Figure 5: A example comparing weighted sum aggregation versus orthogonal mixing. **(a)** Two input temporal features with out-of-phase anomalies. **(b)** The convex combination in standard GNN/Attention results in destructive signal cancellation (amplitude decay) and over-smoothing (collapsing trajectories). **(c)** The Givens rotations in `SICM` enable lossless, energy-preserving feature mixing without amplitude degradation. *(For illustration, the standard attention assumes a normalized weight matrix* $\mathbf{A} = [0.6, 0.4; 0.4, 0.6]$*, while* `SICM` *applies a* $45°$ *Givens rotation)*

A.3 GIVENS ROTATIONS AND ROTARY POSITION EMBEDDINGS

Background on Givens Rotations. A Givens rotation is a sparse orthogonal transformation that acts on only two coordinates of a vector (Givens, 1958). For two coordinates i and j, the corresponding two-dimensional update is

$$\begin{bmatrix} y_i \\ y_j \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix}, \qquad y_q = x_q, \quad q \notin \{i, j\}. \tag{18}$$

Formally, this element-wise update is equivalent to a linear transformation $\mathbf{y} = \mathbf{G}_{ij}(\theta)\mathbf{x}$. The embedded transformation is orthogonal:

$$\mathbf{G}_{ij}(\theta)^\top \mathbf{G}_{ij}(\theta) = \mathbf{I}, \qquad \mathbf{G}_{ij}(\theta)^{-1} = \mathbf{G}_{ij}(-\theta) = \mathbf{G}_{ij}(\theta)^\top. \tag{19}$$

Consequently, for any vector $\mathbf{x}$, $\|\mathbf{G}_{ij}(\theta)\mathbf{x}\|_2 = \|\mathbf{x}\|_2$.

The numerical linear algebra literature introduced Givens rotations as a constructive way to transform a matrix into triangular form. Givens described this procedure in a formal journal article on unitary rotations and matrix triangularization (Givens, 1958). The key operation is to choose c and s so that one selected matrix entry is annihilated while the orthogonality of the transformed matrix is preserved. Applying a sequence of such sparse plane rotations yields QR-type factorizations without introducing a dense orthogonal matrix explicitly.

Givens rotations and closely related orthogonal transformations subsequently became standard primitives in singular-value and least-squares algorithms. Golub and Kahan developed an orthogonal-transformation-based procedure for computing singular values and the pseudoinverse of a matrix (Golub & Kahan, 1965). The LSQR algorithm uses Golub–Kahan bidiagonalization together with orthogonal plane-rotation updates to solve sparse linear systems and sparse least-squares problems (Paige & Saunders, 1982). The historical development of these orthogonal matrix factorizations and their relationship to the singular value decomposition is reviewed by Stewart (Stewart, 1993).

The use of Givens rotations in `SICDNet` follows this established numerical primitive but serves a different purpose. `SICDNet` does not use the rotations to compute a QR factorization, or an Singular Value Decomposition (SVD). Instead, the rotations in Eq. (5) are applied to paired variable tokens according to the fixed dyadic schedule $\mathcal{I}^{(k)}_{\text{left}}$ and $\mathcal{I}^{(k)}_{\text{right}}$. The hidden dimension D of each variable token is partitioned into G groups of width $W = D/G$. For each input instance, at stage k of round r, every valid pair $(j, j + 2^k)$ in the cyclically shifted variable order receives a group-wise angle vector $\boldsymbol{\theta}^{(r,k)}_j \in \mathbb{R}^G$. Its g-th component, $\theta^{(r,k)}_{j,g}$, is shared across the W corresponding coordinates of that pair's g-th group, but may vary across input instances, variable pairs, groups, rounds, and stages.

Within each `SICM` layer, a shared angle-generation network predicts this vector from the pair's current G-dimensional group-energy vector and a stage embedding indexed by $rS_{\text{stage}} + k$. For consistency with the main text, the sample and layer indices are omitted, and θ_g denotes the angle for the particular pair, round, and stage under consideration. Thus, each local $2W \times 2W$ rotation block remains orthogonal despite its state-dependent angle.

Distinction from Rotary Position Embeddings. During our literature review, we noted that rotary position embeddings (RoPE) (Su et al., 2024) also employs a similar rotational mechanism. Although our motivation for proposing `SICM` does not stem from RoPE, we provide a detailed clarification here to offer readers a clearer distinction.

SICM shares the rotation primitive with RoPE, but differs in its transformation axis, conditioning mechanism, and computational role. Standard RoPE rotates feature pairs within each query and key vector using position-dependent angles, thereby introducing relative-position information into attention scores. The positional rotation itself does not exchange representations between tokens. In contrast, `SICM` jointly rotates corresponding feature groups from two distinct variable tokens, directly performing cross-variable information exchange without query–key attention. Its angles are conditioned on the current pair-group energies and the mixing stage, rather than on token positions. Because these energies are invariant under the corresponding rotations, the angles can be recomputed during inversion. Thus, our contribution is not the rotation primitive itself, but its use in a multi-scale, state-conditioned variable mixer whose invertibility follows from invariant conditioning.

For standard RoPE, let $\mathbf{q}_m, \mathbf{k}_n \in \mathbb{R}^d$ denote Query and Key vectors at positions m and n. The position-dependent block-diagonal rotation matrix $\mathbf{R}(m)$ is explicitly formulated as:

$$\mathbf{R}(m) = \text{diag}\left(\mathbf{R}_1(m), \mathbf{R}_2(m), \ldots, \mathbf{R}_{d/2}(m)\right), \tag{20}$$

where each 2×2 sub-matrix $\mathbf{R}_i(m)$ is given by:

$$\mathbf{R}_i(m) = \begin{pmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{pmatrix}. \tag{21}$$

Here, θ_i is the pre-defined rotation frequency for the i-th dimension pair, typically defined as an exponential decay function: $\theta_i = \Theta^{-2(i-1)/d}$ for $i \in \{1, 2, \ldots, d/2\}$. Θ is the base commonly set to $10,000$ (or larger values for extended context lengths). Position-dependent block-diagonal rotations give $\widetilde{\mathbf{q}}_m = \mathbf{R}(m)\mathbf{q}_m, \widetilde{\mathbf{k}}_n = \mathbf{R}(n)\mathbf{k}_n$, and hence

$$\widetilde{\mathbf{q}}_m^\top \widetilde{\mathbf{k}}_n = \mathbf{q}_m^\top \mathbf{R}(n-m)\mathbf{k}_n. \tag{22}$$

SICM instead acts jointly on two variable tokens. At a given mixing stage, let $\mathbf{h}_{i,g}, \mathbf{h}_{j,g} \in \mathbb{R}^W$ be the corresponding feature groups of paired variables i and j. Their update is

$$\begin{bmatrix} \mathbf{h}'_{i,g} \\ \mathbf{h}'_{j,g} \end{bmatrix} = \begin{bmatrix} \cos\theta_{ij,g}\,\mathbf{I}_W & -\sin\theta_{ij,g}\,\mathbf{I}_W \\ \sin\theta_{ij,g}\,\mathbf{I}_W & \cos\theta_{ij,g}\,\mathbf{I}_W \end{bmatrix} \begin{bmatrix} \mathbf{h}_{i,g} \\ \mathbf{h}_{j,g} \end{bmatrix}. \tag{23}$$

Unlike a token-wise positional rotation, this operation directly couples the representations of two variables. The angle vector is generated as

$$\boldsymbol{\theta}_{ij} = f_\phi(\mathbf{e}_{ij}, s), \qquad e_{ij,g} = \frac{\|\mathbf{h}_{i,g}\|_2^2 + \|\mathbf{h}_{j,g}\|_2^2}{W}, \tag{24}$$

where $\mathbf{e}_{ij}$ collects the energies of all feature groups and s identifies the mixing stage, not a temporal position. Since $\mathbf{e}'_{ij} = \mathbf{e}_{ij}$, the same angles can be recomputed from the stage outputs. Reversing the stage order and applying the negative angles therefore inverts the mixer in exact arithmetic.

In summary, the fundamental distinctions between RoPE and our proposed SICM regarding the utilization of rotations are twofold: ❶ **Determination of θ:** RoPE relies on a static, predefined scheduling strategy for rotation angles, whereas SICM dynamically computes θ conditioned on the pairwise energy of the interacting variables. ❷ **Modeling paradigm:** While RoPE rotates the Query and Key vectors to inject relative positional embeddings within self-attention mechanisms, SICM leverages Givens rotations to explicitly model pairwise inter-variable dependencies entirely independent of self-attention. Specifically, SICM incorporates a *Dyadic Multi-Scale Pairing Schedule*, utilizing multi-stage and multi-round strategies to maximize the coverage of pairwise information exchange across a broader range of variables.

A.4 Algorithm and Properties of SICM

In this subsection, we provide a detailed analysis of the algorithmic specifics, theoretical properties, and key characteristics of SICM. First, the procedure of a single-layer SICM is summarized in Algorithm 1. Notably, SICM acts as a plug-and-play module that can be seamlessly embedded into any existing architecture for inter-series dependency modeling. Furthermore, we present the formal proof of its Energy Invariance property below.

Property 1 (Energy Invariance under Group-Wise Givens Rotation). Let $\mathbf{u}_g, \mathbf{v}_g \in \mathbb{R}^W$ be transformed into $\mathbf{u}'_g, \mathbf{v}'_g$ via Eq. (5). The group-wise normalized paired energy e_g satisfies:

$$e'_g = \frac{1}{W}\left(\|\mathbf{u}'_g\|_2^2 + \|\mathbf{v}'_g\|_2^2\right) = \frac{1}{W}\left(\|\mathbf{u}_g\|_2^2 + \|\mathbf{v}_g\|_2^2\right) = e_g. \tag{25}$$

Proof. Expanding the norm of the rotated vectors:

$$\begin{aligned} \|\mathbf{u}'_g\|_2^2 + \|\mathbf{v}'_g\|_2^2 &= \|\cos(\theta_g)\mathbf{u}_g - \sin(\theta_g)\mathbf{v}_g\|_2^2 + \|\sin(\theta_g)\mathbf{u}_g + \cos(\theta_g)\mathbf{v}_g\|_2^2 \\ &= \cos^2(\theta_g)\|\mathbf{u}_g\|_2^2 + \sin^2(\theta_g)\|\mathbf{v}_g\|_2^2 - 2\sin(\theta_g)\cos(\theta_g)\langle\mathbf{u}_g, \mathbf{v}_g\rangle \\ &\quad + \sin^2(\theta_g)\|\mathbf{u}_g\|_2^2 + \cos^2(\theta_g)\|\mathbf{v}_g\|_2^2 + 2\sin(\theta_g)\cos(\theta_g)\langle\mathbf{u}_g, \mathbf{v}_g\rangle \\ &= (\cos^2\theta_g + \sin^2\theta_g)\|\mathbf{u}_g\|_2^2 + (\sin^2\theta_g + \cos^2\theta_g)\|\mathbf{v}_g\|_2^2 \\ &= \|\mathbf{u}_g\|_2^2 + \|\mathbf{v}_g\|_2^2. \end{aligned} \tag{26}$$

Dividing by W completes the proof. □

Algorithm 1 Forward Pass of Single SICM Block

Require: Token representation $\mathbf{H}^{(\ell-1)} \in \mathbb{R}^{N\times D}$, rounds R, stages $S_{\text{stage}} = \lceil \log_2 N \rceil$.
Ensure: Updated representation $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N\times D}$, auxiliary loss $\mathcal{L}_{\text{angle}}$.

$\mathbf{Z} \leftarrow \mathbf{H}^{(\ell-1)}$
Initialize penalty list $\mathcal{A} \leftarrow \emptyset$
for $r = 0$ **to** $R-1$ **do**
  $s_r \leftarrow r \bmod N$
  if $s_r > 0$ **then**
    $\mathbf{Z} \leftarrow \mathrm{Roll}(\mathbf{Z}, \text{shift} = s_r, \dim = \text{variates})$
  end if
  for $k = 0$ **to** $S_{\text{stage}} - 1$ **do**
    $\tau \leftarrow rS_{\text{stage}} + k$
    $s \leftarrow \tau + 1$
    $(\mathcal{I}_{\text{left}}, \mathcal{I}_{\text{right}}) \leftarrow \mathrm{DyadicPairs}(N, \text{stride} = 2^k)$
    $\mathcal{P}_s \leftarrow \{(u, v) \mid u \in \mathcal{I}_{\text{left}},\ v = u + 2^k\}$
    $(\mathbf{Z}[\mathcal{I}_{\text{left}}], \mathbf{Z}[\mathcal{I}_{\text{right}}], \boldsymbol{\theta}) \leftarrow \mathrm{GivensRotate}(\mathbf{Z}[\mathcal{I}_{\text{left}}], \mathbf{Z}[\mathcal{I}_{\text{right}}]; \tau)$
    Append $\mathrm{Mean}(\boldsymbol{\theta}^2)$ to $\mathcal{A}$
  end for
  if $s_r > 0$ **then**
    $\mathbf{Z} \leftarrow \mathrm{Roll}(\mathbf{Z}, \text{shift} = -s_r, \dim = \text{variates})$
  end if
end for
$\mathbf{H}^{(\ell)}_{\text{mixed}} \leftarrow \mathbf{Z}$
$\mathbf{H}^{(\ell)} \leftarrow \mathbf{H}^{(\ell)}_{\text{mixed}} + \mathrm{SGM}(\mathrm{LayerNorm}(\mathbf{H}^{(\ell)}_{\text{mixed}}))$
$\mathcal{L}_{\text{angle}} \leftarrow \mathrm{Mean}(\mathcal{A})$
return $\mathbf{H}^{(\ell)}, \mathcal{L}_{\text{angle}}$

SICM intrinsically possesses the following key characteristics:

❶ **Mathematical Inversion of the SICM.** Because the group-wise energy satisfies $e'_g \equiv e_g$ at every stage (**Property 1**), the exact same rotation angles $\boldsymbol{\theta}$ can be deterministically recomputed from the output states $(\mathbf{u}'_g, \mathbf{v}'_g)$. The local inverse operator $F^{-1}_{r,k}$ for each pair at stage k in round r is given by the transposed orthogonal matrix:

$$\begin{bmatrix}\mathbf{u}_g \\ \mathbf{v}_g\end{bmatrix} = \begin{bmatrix}\cos\theta_g \mathbf{I}_W & \sin\theta_g \mathbf{I}_W \\ -\sin\theta_g \mathbf{I}_W & \cos\theta_g \mathbf{I}_W\end{bmatrix} \begin{bmatrix}\mathbf{u}'_g \\ \mathbf{v}'_g\end{bmatrix}. \tag{27}$$

To formalize the invertibility of the entire multi-scale cascade, let P_r denote the cyclic permutation operator (with inverse P_r^{-1}) for round r, and let $F_{r,k}$ denote the orthogonal Givens transformation at stage $k \in \{0, \ldots, S_{\text{stage}} - 1\}$. The forward mixing transformation for round r is a sequential composition:

$$\mathcal{M}_r = P_r^{-1} \circ F_{r,S_{\text{stage}}-1} \circ \cdots \circ F_{r,0} \circ P_r. \tag{28}$$

By the anti-homomorphism of function composition under inversion, the exact inverse of the r-th round is given by:

$$\mathcal{M}_r^{-1} = P_r^{-1} \circ F_{r,0}^{-1} \circ \cdots \circ F_{r,S_{\text{stage}}-1}^{-1} \circ P_r, \tag{29}$$

where each $F^{-1}_{r,k}$ evaluates Eq. (27) on paired variables. The full SICM $\mathcal{M} = \mathcal{M}_{R-1} \circ \cdots \circ \mathcal{M}_0$ is thus inverted via $\mathcal{M}^{-1} = \mathcal{M}_0^{-1} \circ \cdots \circ \mathcal{M}_{R-1}^{-1}$ in reverse sequential execution order. Thus, for fixed model parameters, the inter-series mixer is information-preserving and invertible in exact arithmetic, without requiring cached per-stage angles or intermediate mixer states.

❷ **Multi-Scale Mixing and Progressive Reachability.** Dyadic pairing establishes hierarchical multi-scale communication. For arbitrary N, while boundary variables may remain unpaired at specific stages in a single round, the multi-round cyclic permutation ($s_r = r \pmod N$) continuously reorganizes spatial pairings, enabling progressive all-to-all dependency propagation across $R \geq 2$ rounds.

❸ Sparsity Prior via Angle Regularization. When $\boldsymbol{\theta} \to \mathbf{0}$, the Givens rotation matrix continuously converges to the identity matrix $\mathbf{I}_{2W}$. Thus, the auxiliary regularization $\mathcal{L}_{\text{angle}}$ imposes a structural inductive bias: when inter-channel correlations are negligible, `SICDNet` gracefully degenerates into an optimal Channel-Independent (CI) baseline, avoiding the spurious cross-series overfitting common in dense attention.

❹ Signed Cross-Variable Interactions. Orthogonality also determines the form of the cross-variable coefficients. The self-variable pathways are scaled by $\cos(\theta_g)$, whereas the two cross-variable pathways are scaled by $+\sin(\theta_g)$ and $-\sin(\theta_g)$:

$$\begin{aligned} \mathbf{u}'_g &= \underbrace{\cos(\theta_g)\mathbf{u}_g}_{\text{self pathway}} - \underbrace{\sin(\theta_g)\mathbf{v}_g}_{\text{cross-variable pathway}}, \\ \mathbf{v}'_g &= \underbrace{\sin(\theta_g)\mathbf{u}_g}_{\text{cross-variable pathway}} + \underbrace{\cos(\theta_g)\mathbf{v}_g}_{\text{self pathway}}. \end{aligned} \tag{30}$$

Consequently, `SICM` is not restricted to non-negative convex combinations. The signs of the cross-variable terms allow the mixer to represent both reinforcing and cancelling interactions between variables. This is structurally different from row-normalized softmax weights, which satisfy $A_{ij} \geq 0$ and $\sum_j A_{ij} = 1$.

❺ Computational Complexity. Table 5 summarizes the theoretical properties of `SICDNet`. While standard cross-attention requires $\mathcal{O}(N^2D)$ operations and ST-GNNs evaluate dense $\mathcal{O}(N^2)$ graphs, the `SICM` executes $R\log_2 N$ dyadic Givens stages, achieving a quasi-linear $\mathcal{O}(R \cdot N \log N \cdot D)$ computational complexity. We set R to a small value (e.g., 4), generally smaller than the number of variables N and the feature dimension D.

Table 5: Comparison of inter-series dependency modeling. N: number of variables; D_{node}, D: node, embedding dimension; L: look-back length; R: number of `SICM` rounds; K: graph sparsity.

Model	Dependency Operator	Inter-Series Complexity	Explicit Dense $N \times N$ Map	Exact Invertibility	Norm Conservation
iTransformer	Inverted self-attention	$\mathcal{O}(N^2D)$	Yes	✗	✗
DUET	Inverted self-attention with dual clustering	$\mathcal{O}(N^2D)$	Yes	✗	✗
VCformer	FFT-based correlation modeling	$\mathcal{O}(N^2L\log L + N^2D)$	Yes	✗	✗
MTGNN	Adaptive graph convolution	$\mathcal{O}(N^2D + KND)$	Yes	✗	✗
MAGE	Factorized sparse expert graph propagation	$\mathcal{O}(KNDD_{node})$	No†	✗	✗
SICDNet (Ours)	**Multi-scale dyadic Givens mixer**	$\boldsymbol{\mathcal{O}(RND\log N)}$	**No**	**✓ Exact**	**✓ Orthogonal**

† MAGE still defines an implicit $N \times N$ graph operator, but evaluates it through low-rank factorized multiplication and sparse expert routing without materializing a dense adjacency matrix. Given that the datasets and forecasting scenarios evaluated in MTGNN differ from ours, and MAGE has not publicly released the hyperparameters for all datasets, a fair comparison with these methods cannot be conducted in this study.

A.5 Dataset Details

Table 6 summarizes the detailed statistics of the datasets employed for both long-term and short-term forecasting tasks. According to the predictability metrics, the ETT series and the Solar benchmark pose substantial forecasting challenges. Furthermore, we observe that existing baselines also struggle significantly on high-dimensional datasets containing a massive number of variables (e.g., Traffic).

Table 6: Details of different datasets. Forecastability is one minus the entropy of Fourier domain.

Tasks	Datasets	# Variables	# Samples*	Frequency	Forecastability	Information
Long-term forecasting	ETTm1	7	(34465, 11521, 11521)	15min	0.46	Temperature
	ETTm2	7	(34465, 11521, 11521)	15min	0.55	Temperature
	ETTh1	7	(8545, 2881, 2881)	1hour	0.38	Temperature
	ETTh2	7	(8545, 2881, 2881)	1hour	0.45	Temperature
	Exchange	8	(5120, 665, 1422)	Daily	–	Finance
	Weather	21	(36792, 5271, 10540)	10min	0.75	Weather
	Solar-Energy	137	(36601, 5161, 10417)	10min	0.33	Electricity
	Electricity	321	(18317, 2633, 5261)	1hour	0.77	Electricity
	Traffic	862	(12185, 1757, 3509)	1hour	0.68	Transportation
Short-term forecasting	PEMS03	358	(15701, 5216, 434)	5min	0.65	Transportation
	PEMS04	307	(10172, 3375, 281)	5min	0.45	Transportation
	PEMS07	883	(16911, 5622, 468)	5min	0.58	Transportation
	PEMS08	170	(10690, 3548, 265)	5min	0.52	Transportation

* The number of samples indicates the split ratio (Train, Val, Test).

A.6 Implementation Details

We develop our framework using Python 3.8 and PyTorch 2.0.1, with all computational runs conducted on a computer equipped with two NVIDIA GeForce RTX 3090 GPUs. The entire architecture is trained in an end-to-end fashion using the Adam optimizer. To jointly guarantee forecasting precision and encourage structural sparsity, we minimize the unified loss function: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{forecast}} + \lambda \mathcal{L}_{\text{angle}}$, where $\mathcal{L}_{\text{forecast}}$ is selected for each dataset as either the Mean Absolute Error (MAE) loss or the Huber loss. For the Huber loss, δ denotes the specified Huber threshold, and $\lambda > 0$ controls the auxiliary structural penalty. For performance evaluation, we report two widely recognized metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE).

For `SICDNet`, the core hyperparameters include the number of stacked layers E, the number of groups G, the number of rounds R, the rotation angle bound ($\theta_{\max}$) and the balancing parameter λ. By default, E is selected from $\{1, 2, 3, 4\}$, both the number of groups and rounds are selected within the range of $[1, 32]$, and λ is bounded within $[0, 1]$. While a few hyperparameters may vary slightly across different forecasting scenarios, a comprehensive sensitivity analysis of these hyperparameters is provided in Appendix A.11.

For the framework generality experiments of `SICM`, we re-run the publicly available codebases of all baselines within our unified hardware and software environment, and subsequently execute the modified models integrated with `SICM` under the exact same hyperparameters to guarantee a strictly fair comparison. Detailed implementation specifications are provided in Appendix A.9.

A.7 A Concrete Example of Dyadic Variable Mixing

The single-pair update in Eq. (5) extends to all disjoint pairs at a given stage by forming a block-diagonal operator. For the following example, we fix one sample and one channel group g, and suppress the corresponding indices. The angles associated with pairs $(0, 1)$ and $(2, 3)$ are denoted by θ_{01} and θ_{23}, respectively; they are generally different because the angle network produces one angle for each variable pair and channel group.

This subsection illustrates how the dyadic schedule creates cross-variable dependencies. Consider one round with$N = 4, R = 1$. The number of stage is $S_{\text{stage}} = \lceil \log_2 N \rceil = 2$. Because $s_0 = 0$, the cyclic permutation is the identity in this example. The two stages use the following paired indices:

$$\Delta = 1 : \quad (0, 1),\ (2, 3), \tag{31}$$

$$\Delta = 2 : \quad (0, 2),\ (1, 3). \tag{32}$$

The indices are zero-based, as in Eqs. (3)–(4).

For clarity, fix one sample and one channel group g. Let $\mathbf{h}_{i,g}^{(0)} \in \mathbb{R}^W$ denote the g-th group slice of the D-dimensional token $\mathbf{h}_i^{(0)}$, and define the local matrix

$$\mathbf{H}_0 = \begin{bmatrix} (\mathbf{h}_{0,g}^{(0)})^\top \\ (\mathbf{h}_{1,g}^{(0)})^\top \\ (\mathbf{h}_{2,g}^{(0)})^\top \\ (\mathbf{h}_{3,g}^{(0)})^\top \end{bmatrix} \in \mathbb{R}^{4\times W}. \tag{33}$$

The sample, group, round, and stage indices are suppressed below. Define

$$c_{ij} = \cos(\theta_{ij}), \qquad s_{ij} = \sin(\theta_{ij}). \tag{34}$$

At the first stage, the pairs $(0,1)$ and $(2,3)$ are processed in parallel. The updated matrix is

$$\mathbf{H}_1 = \mathbf{A}_1(\mathbf{H}_0)\mathbf{H}_0, \tag{35}$$

where

$$\mathbf{A}_1(\mathbf{H}_0) = \begin{bmatrix} c_{01} & -s_{01} & 0 & 0 \\ s_{01} & c_{01} & 0 & 0 \\ 0 & 0 & c_{23} & -s_{23} \\ 0 & 0 & s_{23} & c_{23} \end{bmatrix}. \tag{36}$$

Equivalently,

$$(\mathbf{H}_1)_{0,:} = c_{01}\mathbf{h}_{0,g}^{(0)} - s_{01}\mathbf{h}_{1,g}^{(0)}, \tag{37}$$

$$(\mathbf{H}_1)_{1,:} = s_{01}\mathbf{h}_{0,g}^{(0)} + c_{01}\mathbf{h}_{1,g}^{(0)}, \tag{38}$$

$$(\mathbf{H}_1)_{2,:} = c_{23}\mathbf{h}_{2,g}^{(0)} - s_{23}\mathbf{h}_{3,g}^{(0)}, \tag{39}$$

$$(\mathbf{H}_1)_{3,:} = s_{23}\mathbf{h}_{2,g}^{(0)} + c_{23}\mathbf{h}_{3,g}^{(0)}. \tag{40}$$

At the second stage, the implementation reads the already updated matrix $\mathbf{H}_1$. It then applies the pairs $(0,2)$ and $(1,3)$:

$$\mathbf{H}_2 = \mathbf{A}_2(\mathbf{H}_1)\mathbf{H}_1, \tag{41}$$

with

$$\mathbf{A}_2(\mathbf{H}_1) = \begin{bmatrix} c_{02} & 0 & -s_{02} & 0 \\ 0 & c_{13} & 0 & -s_{13} \\ s_{02} & 0 & c_{02} & 0 \\ 0 & s_{13} & 0 & c_{13} \end{bmatrix}. \tag{42}$$

The notation $\mathbf{A}_2(\mathbf{H}_1)$ emphasizes that the second-stage angles are generated from the pair energies computed from the current stage input $\mathbf{H}_1$. Therefore, the complete two-stage operation is more precisely written as

$$\mathbf{H}_2 = \mathbf{A}_2(\mathbf{H}_1)\mathbf{A}_1(\mathbf{H}_0)\mathbf{H}_0, \qquad \mathbf{H}_1 = \mathbf{A}_1(\mathbf{H}_0)\mathbf{H}_0. \tag{43}$$

Writing $\mathbf{H}_2 = \mathbf{A}_2\mathbf{A}_1\mathbf{H}_0$ is only a conditional shorthand for one forward evaluation after the angles have been computed; $\mathbf{A}_1$ and $\mathbf{A}_2$ are not globally fixed matrices independent of the input state.

For example, the first output variable after the second stage is

$$\begin{aligned}(\mathbf{H}_2)_{0,:} &= c_{02}\left(c_{01}\mathbf{h}_{0,g}^{(0)} - s_{01}\mathbf{h}_{1,g}^{(0)}\right) - s_{02}\left(c_{23}\mathbf{h}_{2,g}^{(0)} - s_{23}\mathbf{h}_{3,g}^{(0)}\right) \\ &= c_{02}c_{01}\mathbf{h}_{0,g}^{(0)} - c_{02}s_{01}\mathbf{h}_{1,g}^{(0)} - s_{02}c_{23}\mathbf{h}_{2,g}^{(0)} + s_{02}s_{23}\mathbf{h}_{3,g}^{(0)}.\end{aligned} \tag{44}$$

Thus, although variable 0 is paired only with variable 1 at the first stage, its second-stage representation contains information originating from all four variables. The first stage performs local mixing, while the subsequent dyadic stage propagates information between the previously mixed groups.

In the full implementation, Eq. (36) and Eq. (42) are applied independently to every channel group, with group-specific angles. Equivalently, each scalar coefficient is lifted to a $W \times W$ block $c_{ij,g}\mathbf{I}_W$ or $s_{ij,g}\mathbf{I}_W$, and the complete D-dimensional operation is the direct sum of the group-wise rotations.

The rotations are applied to the current $\mathbf{Z}$, written back to the same variable positions, and the updated $\mathbf{Z}$ is passed to the next stage.

For an odd number of variables, an index that is not contained in $\mathcal{I}_{\text{left}}^{(k)} \cup \mathcal{I}_{\text{right}}^{(k)}$ is unchanged at that stage. For example, when $N = 7$, the implementation generates

$$\Delta = 1: \quad (0,1), (2,3), (4,5), \tag{45}$$

$$\Delta = 2: \quad (0,2), (1,3), (4,6), \tag{46}$$

$$\Delta = 4: \quad (0,4), (1,5), (2,6). \tag{47}$$

The unmatched variables are carried forward unchanged until they participate in a later valid pair. After all dyadic stages of the mixer have finished, the resulting matrix is $\mathbf{H}_{\text{mixed}}^{(\ell)}$. Only then is the latent-dimension `SGM` applied: $\mathbf{H}^{(\ell)} = \mathbf{H}_{\text{mixed}}^{(\ell)} + \text{SGM}(\text{LayerNorm}(\mathbf{H}_{\text{mixed}}^{(\ell)}))$. The `SGM` operates along the last dimension D for each variable token; it does not introduce an additional $N \times N$ variable-pairing operation.

A.8 Additional Results

Full Comparison Results. Table 8 provides a comprehensive comparison across all prediction horizons for long-term forecasting tasks. Compared to representative baselines that explicitly model inter-series dependencies—including DUET, Soatten, iTransformer, MSGNet, and Leddam—`SICDNet` achieves consistently superior averaged performance on both low-dimensional (e.g., ETT series with 7 variables) and high-dimensional (e.g., Traffic with 862 variables) datasets. In particular, `SICDNet` achieves a 4.7% reduction in MSE over DUET on the challenging Traffic benchmark. Furthermore, since several baselines compared in the long-term setting do not report evaluations on short-term benchmarks (i.e., the PEMS series), the comparative baseline results in Table 9 are directly adopted from the original iTransformer paper. In these short-term forecasting scenarios, `SICDNet` likewise demonstrates decent predictive capability.

Full Ablation Results. We select four diverse datasets to conduct comprehensive ablation studies. Table 10 summarizes the full ablation results across four forecasting horizons, from which we observe three key underlying insights: (1) Removing either `SICM` or `SGM` leads to varying degrees of performance degradation, while removing both modules results in the most pronounced deterioration. This finding demonstrates the significant complementarity between `SICM` and `SGM`. (2) `SICM` and `SGM` exert different effects across datasets, revealing heterogeneous and complex patterns. `SICM` has a greater impact than `SGM` on the Weather dataset, whereas `SGM` appears to make a more substantial contribution on ETTh1. On the ECL dataset, `SICM` contributes more prominently when the forecasting horizon is 96, while `SGM` has a greater impact under the other forecasting settings. (3) Overall, stage embeddings contribute positively to the model performance, although their effect is generally less pronounced than that of pair-energy conditioning. (4) The fine-grained ablation results of `SGM` indicate that the spectral gate and global filter have relatively smaller effects than the key mechanisms incorporated in `SICM`.

A.9 Verification of Framework Generality

To evaluate the effectiveness and framework generality of our proposed inter-series dependency modeling approach, we seamlessly integrate the `SICM` into four representative inter-variable modeling architectures: DUET (Qiu et al., 2025), iTransformer (Liu et al., 2024), SOFTS (Han et al., 2024) and VCformer (Yang et al., 2024). To ensure a fair and rigorous comparison, **we strictly adopt their officially released hyperparameters and conduct all benchmarks under identical software and hardware environments**. The comparative results are summarized in Table 7. All source code, execution scripts, and configuration files to fully reproduce these results are publicly available in our anonymous repository. As illustrated in Table 7, integrating `SICM` delivers favorable performance gains across the vast majority of forecasting scenarios.

- DUET (Qiu et al., 2025): DUET highlights that inter-channel interactions in multivariate time series are intrinsically complex, entangled, and noisy, arguing that indiscriminate all-to-all connectivity inevitably introduces spurious correlations. To address this, DUET first utilizes the Discrete Fourier Transform (DFT) to map each channel's series into the frequency

domain, extracting amplitude and phase spectral representations. It then employs metric learning to measure structural similarity distances between channels in the spectral domain, softly assigning highly correlated channels to shared cluster centers. Finally, it executes inter-channel communication conditioned on this clustering structure and incorporates a sparsification gating mechanism to prune low-confidence, noisy connections, thereby retaining only salient correlations among dominant channels. https://github.com/decisionintelligence/DUET

- iTransformer (Liu et al., 2024): Departing from the conventional time-centric paradigm that tokenizes all variables across the same timestamp into a unified token, iTransformer introduces an Inverted Transformer architecture. Specifically, iTransformer independently embeds the entire 1D temporal sequence of each variable over the lookback horizon T into a high-dimensional "variate token". The embeddings of all N variables collectively constitute a representation matrix, upon which standard Multi-Head Self-Attention (MHSA) is directly applied along the variable dimension (i.e., channel dimension) to capture dense pairwise cross-variable interactions. https://github.com/thuml/iTransformer
- SOFTS (Han et al., 2024): SOFTS facilitates inter-channel interactions via a two-stage mechanism: (1) Aggregation: First, the model extracts and aggregates information from all individual time series (channels) to generate a unified global core representation. (2) Dispatch and Fusion: Subsequently, this global core representation, encapsulating the global contextual information, is redistributed and fused with the distinct representation of each individual series. https://github.com/Secilia-Cxy/SOFTS
- VCformer (Yang et al., 2024): Grounded in stochastic process theory, VCformer explicitly calculates the cross-correlation sequences between variable pairs across varying time-lag steps. It subsequently retrieves and aggregates cross-correlation scores under optimal time lags between Queries and Keys, constructing a lagged variable correlation matrix that explicitly accommodates phase shifts and temporal delays. https://github.com/CSyyn/VCformer

Table 7: Verification of framework generality. Full forecasting results across four prediction lengths with a fixed lookback length $L = 96$. "iTrans" denotes iTransformer. "-" indicates that the baseline was not evaluated in the original work or encountered Out-of-Memory (OOM) errors.

Datasets			ETTm1		ETTm2		ETTh1		ETTh2		Exchange		Weather		ECL		Solar		Traffic	
Metrics			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DUET	original	96	0.322	0.354	0.174	0.254	0.389	0.400	0.295	0.345	0.084	0.203	0.162	0.201	**0.146**	**0.233**	0.249	0.269	**0.407**	**0.252**
		192	0.370	0.380	0.239	0.299	0.431	0.426	0.370	0.391	0.179	0.300	0.217	0.251	**0.163**	**0.249**	0.221	0.230	**0.431**	**0.261**
		336	0.407	0.403	0.301	0.339	0.473	0.450	0.408	**0.419**	**0.285**	**0.390**	0.268	0.290	**0.174**	**0.261**	**0.245**	**0.242**	**0.458**	**0.271**
		720	0.464	0.439	0.400	0.396	0.502	0.484	**0.415**	**0.435**	0.686	0.625	0.342	0.339	**0.204**	**0.288**	**0.247**	**0.244**	**0.504**	**0.291**
DUET	+ SICM	96	**0.316**	**0.351**	**0.172**	**0.253**	**0.379**	**0.395**	**0.293**	**0.341**	**0.083**	**0.200**	**0.151**	**0.190**	0.163	0.270	**0.189**	**0.213**	0.425	0.274
		192	**0.367**	**0.377**	**0.236**	**0.295**	**0.422**	**0.418**	**0.367**	**0.387**	**0.175**	**0.297**	**0.202**	**0.240**	0.179	0.282	**0.213**	**0.226**	0.441	0.280
		336	**0.392**	**0.395**	**0.300**	**0.337**	**0.461**	**0.439**	**0.406**	0.420	0.300	0.396	**0.257**	**0.281**	0.193	0.296	0.250	**0.242**	0.476	0.289
		720	**0.455**	**0.432**	**0.395**	**0.393**	**0.499**	**0.479**	0.421	0.439	**0.649**	**0.606**	**0.325**	**0.327**	0.209	0.310	0.252	0.245	0.511	0.307
iTrans	original	96	0.342	0.377	0.186	0.272	0.387	0.405	0.301	0.350	0.086	0.206	0.181	0.221	0.148	0.239	**0.201**	**0.234**	**0.392**	**0.268**
		192	0.383	0.396	0.254	0.314	0.441	0.436	0.381	0.399	0.181	0.303	0.226	0.259	0.167	0.258	0.239	**0.263**	**0.413**	**0.277**
		336	0.418	0.418	0.317	0.353	0.491	0.462	0.423	0.432	0.338	0.422	0.283	0.300	0.181	0.275	**0.248**	**0.272**	**0.425**	**0.283**
		720	0.487	0.456	0.416	0.408	0.509	0.494	0.430	0.446	0.869	0.704	0.359	0.351	**0.209**	**0.299**	**0.250**	**0.275**	**0.459**	**0.300**
iTrans	+ SICM	96	**0.336**	**0.372**	**0.183**	**0.267**	**0.384**	**0.399**	**0.297**	**0.348**	**0.084**	**0.204**	**0.168**	**0.215**	**0.143**	**0.238**	0.203	0.242	0.400	0.277
		192	**0.376**	**0.388**	**0.249**	**0.309**	**0.436**	**0.429**	**0.377**	**0.397**	**0.177**	**0.299**	**0.214**	**0.255**	**0.160**	**0.252**	**0.233**	0.264	0.422	0.283
		336	**0.408**	**0.408**	**0.311**	**0.348**	**0.475**	**0.448**	**0.419**	**0.431**	**0.331**	**0.417**	**0.271**	**0.295**	**0.176**	**0.269**	0.250	0.279	0.438	0.289
		720	**0.470**	**0.443**	**0.407**	**0.402**	**0.484**	**0.476**	**0.425**	**0.444**	**0.840**	**0.690**	**0.347**	**0.344**	**0.209**	0.300	0.251	0.279	0.464	0.305
SOFTS	original	96	**0.318**	0.359	0.180	0.262	0.388	0.404	0.294	**0.344**	-	-	0.178	0.219	0.145	0.236	0.204	**0.231**	**0.390**	**0.265**
		192	0.374	0.390	0.246	0.305	0.441	0.436	0.371	0.394	-	-	0.222	0.257	0.163	0.253	0.230	**0.255**	**0.398**	**0.262**
		336	0.408	0.412	0.327	0.357	0.479	0.452	0.430	0.434	-	-	0.281	0.298	0.179	0.271	**0.243**	**0.269**	**0.416**	**0.269**
		720	**0.473**	0.454	0.421	0.408	0.499	0.488	0.427	0.445	-	-	0.357	**0.350**	0.213	0.302	**0.246**	**0.272**	**0.450**	**0.289**
SOFTS	+ SICM	96	0.320	**0.356**	**0.179**	**0.260**	**0.376**	**0.398**	**0.293**	**0.344**	-	-	**0.165**	**0.213**	**0.139**	**0.235**	**0.200**	0.242	0.395	0.262
		192	**0.372**	**0.387**	**0.244**	**0.302**	**0.435**	**0.429**	**0.370**	**0.393**	-	-	**0.211**	**0.252**	**0.156**	**0.250**	**0.229**	0.263	0.413	0.273
		336	**0.403**	**0.408**	**0.305**	**0.342**	**0.476**	**0.451**	**0.415**	**0.430**	-	-	**0.271**	**0.295**	**0.177**	**0.270**	0.246	0.277	0.443	0.291
		720	0.476	0.452	**0.404**	**0.401**	**0.487**	**0.480**	**0.412**	**0.441**	-	-	**0.355**	**0.350**	**0.205**	**0.300**	0.247	0.278	0.466	0.299
VCformer	original	96	0.331	0.364	0.184	0.266	0.405	0.410	0.302	0.349	**0.085**	0.206	0.186	0.224	0.152	0.246	-	-	-	-
		192	0.379	0.389	0.250	0.309	0.455	0.439	0.383	0.396	**0.175**	0.300	0.238	0.266	0.170	0.261	-	-	-	-
		336	0.419	0.416	0.318	0.352	0.530	0.476	**0.421**	**0.430**	0.327	0.415	0.288	0.303	0.186	0.277	-	-	-	-
		720	0.487	0.453	0.414	0.407	0.561	0.515	0.429	0.446	0.844	0.691	0.365	0.352	0.235	0.328	-	-	-	-
VCformer	+ SICM	96	**0.330**	**0.364**	**0.180**	**0.261**	**0.396**	**0.400**	**0.293**	**0.341**	**0.085**	**0.205**	**0.177**	**0.224**	**0.139**	**0.235**	-	-	-	-
		192	**0.374**	**0.385**	**0.245**	**0.305**	**0.445**	**0.427**	**0.379**	**0.393**	**0.175**	**0.298**	**0.227**	**0.263**	**0.159**	**0.250**	-	-	-	-
		336	**0.406**	**0.407**	**0.308**	**0.345**	**0.489**	**0.449**	0.425	0.431	**0.320**	**0.410**	**0.281**	**0.301**	**0.174**	**0.266**	-	-	-	-
		720	**0.469**	**0.442**	**0.409**	**0.401**	**0.495**	**0.475**	**0.426**	**0.443**	**0.834**	**0.686**	**0.354**	**0.349**	**0.205**	**0.299**	-	-	-	-

Table 8: Long-term forecasting results with forecasting horizons $T \in \{96, 192, 336, 720\}$ and fixed look-back length L=96. **Bold**/underline: Best/second best one. "-" indicates that the original method was not evaluated in the corresponding scenario. The standard deviations are within 0.005 for MSE and within 0.003 for MAE.

Models		SICDNet (ours)		SSformer (2026)		DUET (2025)		TimePro (2025)		Soatten (2025)		iTransformer (2024)		MSGNet (2024)		Leddam (2024)		PatchTST (2023)		TimesNet (2023)	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.310**	**0.340**	0.318	0.355	0.324	0.354	0.326	0.364	0.329	0.365	0.334	0.368	0.319	0.366	0.319	0.359	0.329	0.367	0.338	0.375
	192	**0.358**	**0.366**	0.360	0.379	0.369	0.379	0.367	0.383	0.370	0.387	0.377	0.391	0.376	0.397	0.369	0.383	0.367	0.385	0.374	0.387
	336	0.391	**0.390**	0.393	0.402	0.404	0.402	0.402	0.409	0.401	0.407	0.426	0.420	0.417	0.422	0.394	0.402	0.399	0.410	0.410	0.411
	720	0.462	**0.429**	**0.449**	0.440	0.463	0.437	0.469	0.446	0.474	0.447	0.491	0.459	0.481	0.458	0.460	0.442	0.454	0.439	0.478	0.450
	AVG	**0.380**	**0.381**	**0.380**	0.394	0.390	0.393	0.391	0.400	0.394	0.402	0.407	0.410	0.398	0.411	0.386	0.397	0.387	0.400	0.400	0.406
ETTm2	96	0.172	**0.251**	**0.171**	0.254	0.174	0.255	0.178	0.260	0.180	0.264	0.180	0.264	0.177	0.262	0.176	0.257	0.175	0.259	0.187	0.267
	192	0.238	**0.294**	**0.235**	0.302	0.243	0.302	0.242	0.303	0.245	0.306	0.250	0.309	0.245	0.306	0.243	0.303	0.241	0.302	0.249	0.309
	336	**0.296**	**0.332**	0.307	0.338	0.304	0.341	0.303	0.342	0.312	0.349	0.311	0.348	0.312	0.346	0.303	0.341	0.305	0.343	0.321	0.351
	720	**0.396**	**0.392**	0.403	0.395	0.399	0.397	0.400	0.399	0.411	0.406	0.412	0.407	0.414	0.403	0.400	0.398	0.402	0.400	0.408	0.403
	AVG	**0.275**	**0.317**	0.279	0.322	0.280	0.324	0.281	0.326	0.287	0.331	0.288	0.332	0.288	0.330	0.281	0.325	0.281	0.326	0.291	0.333
ETTh1	96	**0.373**	**0.388**	**0.373**	0.393	0.377	0.393	0.375	0.398	0.383	0.400	0.386	0.405	0.390	0.411	0.377	0.394	0.414	0.419	0.384	0.402
	192	0.426	**0.422**	0.432	0.431	0.429	0.425	0.427	0.429	0.440	0.433	0.441	0.436	0.442	0.442	**0.424**	**0.422**	0.460	0.445	0.436	0.429
	336	0.460	**0.440**	0.483	0.454	0.471	0.446	0.472	0.450	0.475	0.449	0.487	0.458	0.480	0.468	**0.459**	0.442	0.501	0.466	0.491	0.469
	720	**0.451**	**0.456**	0.519	0.495	0.496	0.480	0.476	0.474	0.491	0.477	0.503	0.491	0.494	0.488	0.463	0.459	0.500	0.488	0.521	0.500
	AVG	**0.427**	**0.426**	0.452	0.443	0.443	0.436	0.438	0.438	0.447	0.440	0.454	0.447	0.452	0.452	0.431	0.429	0.469	0.454	0.458	0.450
ETTh2	96	0.283	0.334	**0.281**	**0.332**	0.296	0.345	0.293	0.345	0.295	0.348	0.297	0.349	0.328	0.371	0.292	0.343	0.302	0.348	0.340	0.374
	192	**0.362**	**0.382**	0.367	0.385	0.368	0.389	0.367	0.394	0.380	0.398	0.380	0.400	0.402	0.414	0.367	0.389	0.388	0.400	0.402	0.414
	336	0.403	**0.418**	**0.394**	**0.418**	0.411	0.422	0.419	0.431	0.420	0.431	0.428	0.432	0.435	0.443	0.412	0.424	0.426	0.433	0.452	0.452
	720	0.410	0.433	**0.395**	**0.420**	0.412	0.434	0.427	0.445	0.419	0.441	0.427	0.445	0.417	0.441	0.419	0.438	0.431	0.446	0.462	0.468
	AVG	0.364	0.392	**0.359**	**0.389**	0.372	0.397	0.377	0.403	0.379	0.405	0.383	0.407	0.396	0.417	0.373	0.399	0.387	0.407	0.414	0.427
Exchange	96	**0.083**	0.201	0.085	**0.200**	0.086	0.205	0.085	0.204	0.085	0.204	0.086	0.206	0.102	0.230	-	-	0.088	0.205	0.107	0.234
	192	0.173	0.299	**0.171**	**0.296**	0.182	0.305	0.178	0.299	0.175	0.299	0.177	0.299	0.195	0.317	-	-	0.176	0.299	0.226	0.344
	336	0.324	0.414	0.348	0.429	0.310	0.403	0.328	0.414	0.330	0.417	0.331	0.417	0.359	0.436	-	-	**0.301**	**0.397**	0.367	0.448
	720	0.707	0.631	0.843	0.715	**0.693**	**0.624**	0.817	0.679	0.844	0.695	0.847	0.691	0.940	0.738	-	-	0.901	0.714	0.964	0.746
	AVG	0.322	0.386	0.362	0.410	**0.318**	**0.384**	0.352	0.399	0.359	0.404	0.360	0.403	0.399	0.430	-	-	0.367	0.404	0.416	0.443
Weather	96	**0.153**	**0.191**	0.156	0.204	0.163	0.202	0.166	0.207	0.161	0.206	0.174	0.214	0.163	0.212	0.156	0.202	0.177	0.218	0.172	0.220
	192	**0.203**	**0.239**	**0.203**	0.245	0.218	0.252	0.216	0.254	0.208	0.250	0.221	0.254	0.212	0.254	0.207	0.250	0.225	0.259	0.219	0.261
	336	0.253	**0.276**	**0.250**	0.284	0.274	0.294	0.273	0.296	0.264	0.291	0.278	0.296	0.272	0.299	0.262	0.291	0.278	0.297	0.280	0.306
	720	0.339	**0.335**	**0.338**	0.337	0.349	0.343	0.351	0.346	0.347	0.346	0.358	0.347	0.350	0.348	0.343	0.343	0.354	0.348	0.365	0.359
	AVG	**0.237**	**0.260**	**0.237**	0.268	0.251	0.273	0.251	0.276	0.245	0.273	0.258	0.278	0.249	0.278	0.242	0.272	0.259	0.281	0.259	0.287
Electricity	96	0.139	**0.230**	0.141	0.238	0.145	0.233	0.139	0.234	**0.137**	0.232	0.148	0.240	0.165	0.274	0.141	0.235	0.181	0.270	0.168	0.272
	192	0.156	**0.247**	0.157	0.251	0.163	0.248	0.156	0.249	**0.155**	**0.247**	0.162	0.253	0.184	0.292	0.159	0.252	0.188	0.274	0.184	0.289
	336	**0.171**	0.263	0.174	0.268	0.175	**0.262**	0.172	0.267	**0.171**	0.265	0.178	0.269	0.195	0.302	0.173	0.268	0.204	0.293	0.198	0.300
	720	**0.194**	**0.286**	0.209	0.303	0.204	0.291	0.209	0.299	0.200	0.290	0.225	0.317	0.231	0.332	0.201	0.295	0.246	0.324	0.220	0.320
	AVG	**0.165**	**0.256**	0.170	0.265	0.172	0.258	0.169	0.262	0.166	0.259	0.178	0.270	0.194	0.300	0.169	0.263	0.205	0.290	0.192	0.295
Solar	96	**0.190**	**0.200**	0.197	0.236	0.200	0.207	0.196	0.237	0.198	0.239	0.203	0.237	-	-	0.197	0.241	0.234	0.286	0.250	0.292
	192	**0.220**	**0.222**	0.222	0.268	0.228	0.233	0.231	0.263	0.228	0.259	0.233	0.261	-	-	0.231	0.264	0.267	0.310	0.296	0.318
	336	0.242	**0.243**	0.245	0.271	0.262	0.244	0.250	0.281	0.244	0.272	0.248	0.273	-	-	0.241	0.268	0.290	0.315	0.319	0.330
	720	**0.246**	**0.240**	0.247	0.287	0.258	0.249	0.253	0.285	**0.246**	0.275	0.249	0.275	-	-	0.250	0.281	0.289	0.317	0.338	0.337
	AVG	**0.224**	**0.226**	0.228	0.266	0.237	0.233	0.232	0.266	0.229	0.261	0.233	0.262	-	-	0.230	0.264	0.270	0.307	0.301	0.319
Traffic	96	**0.392**	**0.248**	0.420	0.288	0.407	0.252	-	-	0.401	0.270	0.395	0.268	-	-	0.426	0.276	0.544	0.359	0.593	0.321
	192	0.419	**0.253**	0.434	0.298	0.431	0.262	-	-	0.424	0.281	**0.417**	0.276	-	-	0.458	0.289	0.540	0.354	0.617	0.336
	336	0.436	**0.264**	0.468	0.301	0.456	0.269	-	-	0.445	0.288	**0.433**	0.283	-	-	0.486	0.297	0.551	0.358	0.629	0.336
	720	0.472	**0.282**	0.487	0.309	0.509	0.292	-	-	0.479	0.306	**0.467**	0.302	-	-	0.498	0.313	0.586	0.375	0.640	0.350
	AVG	0.430	**0.262**	0.452	0.299	0.451	0.269	-	-	0.437	0.286	**0.428**	0.282	-	-	0.467	0.294	0.555	0.362	0.620	0.336
1^{st} Count		**16**	**29**	11	5	1	2	0	0	4	1	3	0	0	0	2	1	1	1	0	0

A.10 Empirical Analysis of State-Conditioned SICM

In this subsection, all visualizations are generated by performing a forward pass on the test set of the Weather dataset (either averaged across all test windows or on an individual window), using the best model weights selected on the validation set.

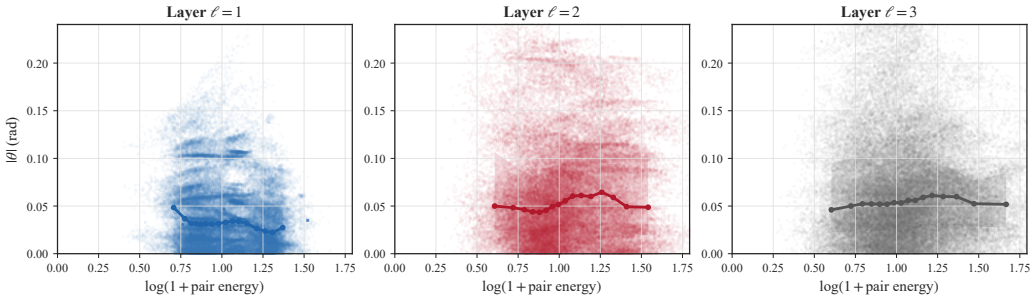


Figure 6: Angle–energy response visualization on Weather dataset.

Table 9: Short-term forecasting results with forecasting horizons $T \in \{12, 24, 48, 96\}$ and fixed look-back length L=96.

Models		SICDNet (*Ours*)		iTransformer (2024)		PatchTST (2023)		TimesNet (2023)	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	12	**0.057**	**0.160**	0.071	0.174	0.099	0.216	0.085	0.192
	24	**0.075**	**0.183**	0.093	0.201	0.142	0.259	0.118	0.223
	48	**0.118**	**0.230**	0.125	0.236	0.211	0.319	0.155	0.260
	96	0.165	**0.274**	**0.164**	0.275	0.269	0.370	0.228	0.317
PEMS04	12	**0.068**	**0.168**	0.078	0.183	0.105	0.224	0.087	0.195
	24	**0.086**	**0.190**	0.095	0.205	0.153	0.275	0.103	0.215
	48	**0.110**	**0.221**	0.120	0.233	0.229	0.339	0.136	0.250
	96	**0.148**	**0.259**	0.150	0.262	0.291	0.389	0.190	0.303
PEMS07	12	**0.054**	**0.149**	0.067	0.165	0.095	0.207	0.082	0.181
	24	**0.075**	**0.169**	0.088	0.190	0.150	0.262	0.101	0.204
	48	**0.099**	**0.198**	0.110	0.215	0.253	0.340	0.134	0.238
	96	0.140	**0.244**	**0.139**	0.245	0.346	0.404	0.181	0.279
PEMS08	12	**0.075**	**0.170**	0.079	0.182	0.168	0.232	0.112	0.212
	24	**0.099**	**0.194**	0.115	0.219	0.224	0.281	0.141	0.238
	48	**0.155**	0.247	0.186	**0.235**	0.321	0.354	0.198	0.283
	96	0.237	0.280	**0.221**	**0.267**	0.408	0.417	0.320	0.351
1^{st} Count		**13**	**14**	3	2	0	0	0	0

Table 10: Complete ablation results of the `SICDNet` architecture across four benchmark datasets and four forecasting horizons ($T \in \{96, 192, 336, 720\}$). Each entry reports test-set MSE/MAE.

(a) $T = 96$

	SICDNet Variant	ETTh1	Weather	ECL	Traffic
	Full `SICDNet`	**.373 / .388**	**.153 / .191**	**.139 / .230**	**.392 / .248**
Core	w/o `SICM`	.374 / .390	.179 / .209	.170 / .249	.429 / .260
	w/o `SGM`	.379 / **.388**	.171 / .211	.164 / .253	.552 / .345
	w/o `SICM` & `SGM`	.380 / **.388**	.201 / .230	.192 / .266	.624 / .363
SICM	w/o pair-energy cond.	.374 / .389	.155 / **.191**	.144 / .236	.406 / .253
	w/o stage embedding	.374 / .389	.156 / .193	.142 / .233	.398 / .250
	w/o multiple rounds	.375 / .390	.157 / .195	.146 / .236	.400 / .251
	fixed dense pairing	.379 / .392	.158 / .195	.153 / .240	.417 / .257
SGM	w/o Spectral Gate	**.373** / .389	**.153** / .193	.139 / .231	.396 / .250
	w/o Global Filter	.374 / .389	.155 / .194	.139 / .231	.393 / .249
	w/o both	.374 / .389	.154 / .194	.142 / .236	.398 / .252

(b) $T = 192$

	SICDNet Variant	ETTh1	Weather	ECL	Traffic
	Full `SICDNet`	**.426 / .422**	**.203 / .239**	**.156 / .247**	**.419 / .253**
Core	w/o `SICM`	.425 / **.422**	.228 / .252	.173 / .257	.442 / .267
	w/o `SGM`	.436 / **.422**	.216 / .251	.185 / .268	.471 / .293
	w/o `SICM` & `SGM`	.437 / .423	.241 / .264	.195 / .272	.555 / .349
SICM	w/o pair-energy cond.	**.426 / .422**	.205 / .242	.162 / .253	.439 / .266
	w/o stage embedding	**.426 / .422**	.206 / .243	.157 / .248	.426 / .258
	w/o multiple rounds	.427 / .423	.211 / .246	.160 / .250	.435 / .265
	fixed dense pairing	.430 / .424	.215 / .248	.169 / .255	.442 / .268
SGM	w/o Spectral Gate	**.426 / .422**	.204 / .240	.157 / **.247**	.422 / .255
	w/o Global Filter	**.426 / .422**	.204 / .240	.157 / .248	.424 / .255
	w/o both	**.426** / .423	.206 / .241	.157 / **.247**	.425 / .257

(c) $T = 336$

	SICDNet Variant	ETTh1	Weather	ECL	Traffic
	Full `SICDNet`	**.460 / .440**	**.253 / .276**	**.171 / .263**	**.436 / .264**
Core	w/o `SICM`	.461 / **.440**	.285 / .293	.187 / .272	.459 / .276
	w/o `SGM`	.479 / .445	.277 / .296	.214 / .296	.485 / .294
	w/o `SICM` & `SGM`	.481 / .445	.298 / .304	.218 / .299	.570 / .355
SICM	w/o pair-energy cond.	**.460 / .440**	.258 / .279	.179 / .270	.445 / .268
	w/o stage embedding	.461 / .441	.255 / .277	.173 / .265	.442 / .267
	w/o multiple rounds	.467 / .445	.268 / .286	.173 / .266	.452 / .273
	fixed dense pairing	.472 / .447	.277 / .295	.189 / .273	.457 / .274
SGM	w/o Spectral Gate	.461 / **.440**	.259 / .282	.174 / .265	.440 / .266
	w/o Global Filter	.461 / **.440**	.260 / .282	.174 / .265	.442 / .267
	w/o both	.461 / .441	.261 / .284	.173 / .264	.445 / .267

(d) $T = 720$

	SICDNet Variant	ETTh1	Weather	ECL	Traffic
	Full `SICDNet`	**.451 / .456**	**.339 / .335**	**.194 / .286**	**.472 / .282**
Core	w/o `SICM`	.452 / **.456**	.366 / .344	.226 / .306	.501 / .305
	w/o `SGM`	.486 / .467	.345 / .340	.267 / .338	.537 / .321
	w/o `SICM` & `SGM`	.486 / .467	.372 / .352	.267 / .338	.648 / .367
SICM	w/o pair-energy cond.	.452 / **.456**	.346 / .340	.212 / .298	.488 / .292
	w/o stage embedding	.452 / **.456**	.341 / .336	.198 / .287	.483 / .290
	w/o multiple rounds	.458 / .460	.344 / .337	.195 / **.286**	.491 / .293
	fixed dense pairing	.464 / .462	.359 / .340	.223 / .305	.497 / .295
SGM	w/o Spectral Gate	.455 / .457	.340 / .336	.201 / .291	.479 / .285
	w/o Global Filter	.452 / **.456**	.342 / .337	.197 / .288	.475 / .285
	w/o both	.455 / .458	.356 / .344	.201 / .291	.481 / .286

Notes. We conduct ablation studies in a coarse-to-fine manner. For a fair comparison, all model variants adopt identical hyperparameters by default. The best performance in each column is highlighted in **bold**.

State-Conditioned Angle Response. Figure 6 visualizes the Givens angle changes with the rotation-invariant pair energy. For a scheduled pair (u, v) and channel group g, the energy used by the angle network is $e_{\ell,r,k,p,g} = \frac{1}{W}(\|u_{\ell,r,k,p,g}\|_2^2 + \|v_{\ell,r,k,p,g}\|_2^2)$, where ℓ denotes the SICD layer, r the `SICM`

round, k the dyadic level, p the scheduled pair index, and W the channel-group width. The horizontal coordinate in the figure is $\log(1 + e_{\ell,r,k,p,g})$, whereas the vertical coordinate is $|\theta_{\ell,r,k,p,g}|$.

The scatter points represent the realized angle–energy observations. For visualization, a fixed number of observations is sampled per layer, and the solid curve with the shaded region denotes the bin-wise median and interquartile range, respectively. The non-degenerate angle distributions and their layer-dependent response profiles indicate that the mixer does not use a single fixed rotation magnitude for all inputs. Instead, the realized mixing strength varies with the input state and the encoder depth ℓ. The response is not strictly monotonic over the entire energy range, which is consistent with the fact that the angle network also receives a learned stage embedding. The local rotation is determined from an invariant statistic of the paired representations, allowing the mixer to adapt its coupling strength to the current representation.

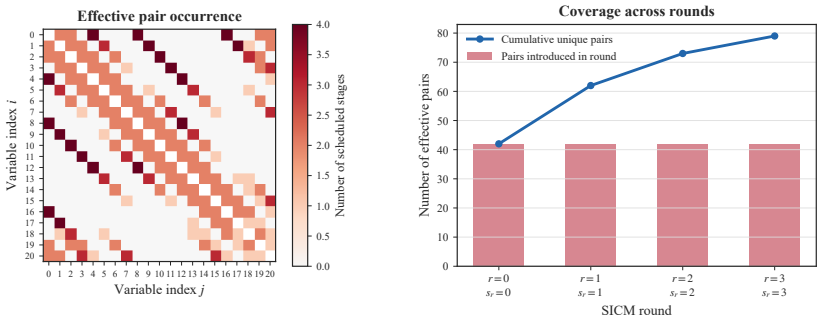


Figure 7: Analysis of the dyadic pairing schedule and cross-variable coverage in SICM. Left: Heatmap of effective pair occurrence across 21 variables of Weather dataset. Right: Number of newly introduced pairs per round and the cumulative unique pairs across four SICM rounds ($r = 0, \ldots, 3$ with cyclic shifts $s_r = 0, \ldots, 3$).

Effective Dyadic Pair Coverage. The Figure 7 visualizes the structural coverage induced by the dyadic pairing schedule and the round-wise cyclic permutation. We represent each scheduled interaction as an unordered variable pair. Let $\mathcal{P}_{r,k}^{\mathrm{roll}}$ denote the set of unordered pairs generated at dyadic level k in round r before undoing the cyclic shift. The effective pair set in the original variable coordinates is defined as

$$\mathcal{P}_{r,k}^{\mathrm{eff}} = \left\{ \{(i - s_r) \bmod N,\ (j - s_r) \bmod N\} : \{i, j\} \in \mathcal{P}_{r,k}^{\mathrm{roll}} \right\}, \qquad s_r = r \bmod N. \tag{48}$$

The left panel reports the number of scheduled stage occurrences for each effective unordered pair:

$$C_{ij} = \sum_{r=0}^{R-1} \sum_{k=0}^{S_{\mathrm{stage}}-1} \mathbb{I}\left[\{i, j\} \in \mathcal{P}_{r,k}^{\mathrm{eff}}\right], \qquad i \neq j. \tag{49}$$

The diagonal entries are masked because the dyadic schedule contains no explicit self-pairs (i, i); this diagnostic therefore counts only cross-variable pair occurrences.

For the Weather configuration, $N = 21$ and $K = \lceil \log_2 N \rceil = 5$. The five dyadic levels contain 10, 10, 9, 8, and 5 scheduled pairs, respectively. Hence, each round contains $10 + 10 + 9 + 8 + 5 = 42$ scheduled pair occurrences. The bars in the right panel therefore have height 42 for every round and should not be interpreted as 42 newly introduced unique pairs.

Because pair sets from different rounds overlap, the cumulative number of unique effective unordered pairs increases from 42 after round $r = 0$ to 62, 73, and 79 after rounds $r = 1$, $r = 2$, and $r = 3$, respectively. Thus, the four-round architecture schedules 79 of the $\binom{21}{2} = 210$ possible unordered variable pairs in the original variable coordinates, corresponding to approximately 37.6% structural coverage.

The increasing cumulative coverage indicates that the multi-round schedule, together with the round-wise cyclic shifts, changes the locations of dyadic interactions in the original variable coordinates and expands the structural interaction range.

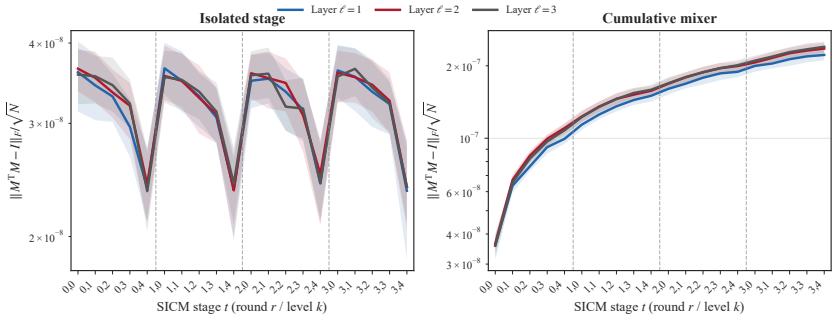


Figure 8: Orthogonality residuals of stage-wise and cumulative SICM operators.

Orthogonality of the Realized SICM. The Figure 8 evaluates the numerical orthogonality of both an isolated SICM stage and the cumulative variable mixer. For a realized stage or cumulative mixing matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$ corresponding to one sample and one channel group, we define

$$\varepsilon(\mathbf{M}) = \frac{\left\|\mathbf{M}^{\mathsf{T}}\mathbf{M} - \mathbf{I}_N\right\|_F}{\sqrt{N}}, \tag{50}$$

where $\mathbf{I}_N$ is the N-dimensional identity matrix.

The isolated-stage and cumulative matrices are reconstructed from the stored Givens angles and the effective pair indices after undoing the round-wise cyclic shift. The curves summarize an evenly spaced diagnostic subset of at most 256 processed test windows; the solid curve denotes the median residual and the shaded region denotes the interquartile range.

The isolated-stage residuals remain at approximately the 10^{-8} level, whereas the cumulative residuals remain at approximately the 10^{-7} level. These values are consistent with floating-point round-off and the accumulation of numerical errors during matrix composition. Analytically, each Givens rotation is orthogonal, and the product of orthogonal matrices is also orthogonal. The observed residuals therefore support the numerical implementation of the structured SICM and its variable-space norm-preserving property.

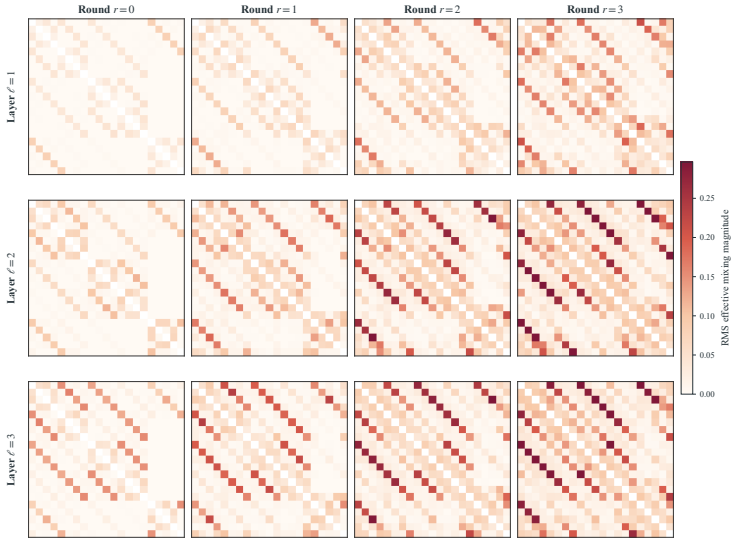


Figure 9: Cumulative inter-variable mixing patterns across SICM layers and rounds.

Round-Level Cumulative Mixing. The Figure 9 summarizes the effective cumulative mixer at the end of each SICM round. The magnitude generally increases as the rounds are composed, indicating

that successive dyadic rotations progressively accumulate cross-channel mixing. Differences between encoder layers (SICD block) show that the strength of the realized mixer is depth-dependent.

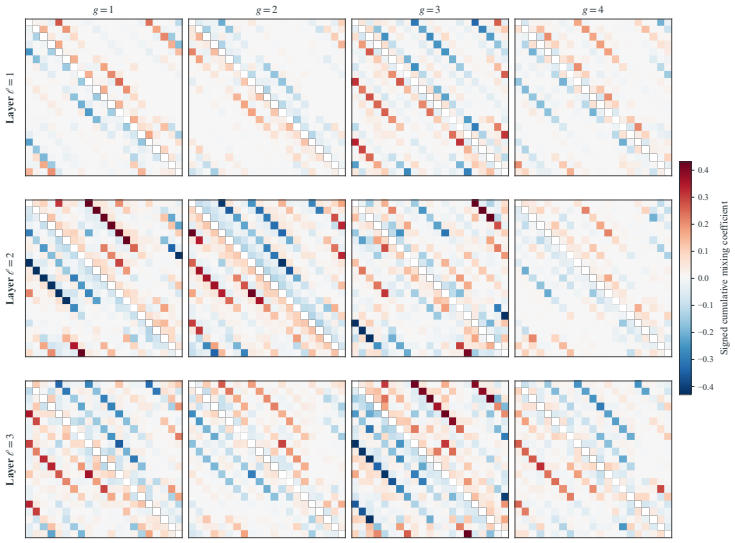


Figure 10: Group-specific signed cumulative inter-series mixing in SICM.

Group-Specific Signed Mixing. The Figure 10 shows the signed cumulative coefficients at the final stage separately for every channel group, for a selected test sample. Differences between groups indicate that the channel branches may realize different state-conditioned transformations.

A.11 Hyperparameter Sensitivity Analysis

Key hyperparameters governing SICDNet include the number of groups (G), the number of mixing rounds (R), the rotation angle bound ($\theta_{\max}$), and the number of SICD layer E. Figure 11, Figure 12, Figure 13, and Figure 14 illustrate the impact of these four hyperparameters on the forecasting performance of SICDNet across the ETTh1 and Weather benchmarks, from which we observe:

(1) The scheduling parameters, G and R, dictate the breadth and density of the pairwise interaction graph. Empirical trends suggest a trade-off: moderate values (e.g., $R = 4$ and $G = 4$) generally deliver a good result. Extremely small values fail to guarantee sufficient coverage of cross-variable paths (under-mixing), whereas excessively large values may introduce redundant interactions or propagate spurious noise (over-mixing).

(2) The parameter $\theta_{\max}$ acts as a bottleneck for the intensity of cross-channel information exchange. The parameter $\theta_{\max}$ exerts a relatively marginal impact on the ETTh1 dataset, whereas tightly constraining the rotation angle (e.g., $\theta_{\max} = \pi/16$ or $\pi/32$) leads to a degradation in forecasting performance on the Weather dataset.

(3) The optimal number of layers E exhibits a strong correlation with the intrinsic complexity and dimensionality of the dataset. For the relatively low-dimensional ETTh1 dataset ($N = 7$), SICDNet achieves peak performance with a shallower architecture ($E \in \{1, 2\}$), indicating that excessive depth may lead to overfitting on simpler topologies. Conversely, on the high-dimensional Weather dataset ($N = 21$) with complex meteorological dynamics, a deeper architecture ($E = 3$) yields the lowest MSE and MAE. This demonstrates that SICDNet scales effectively to accommodate higher-order cross-channel interactions when data complexity demands it.

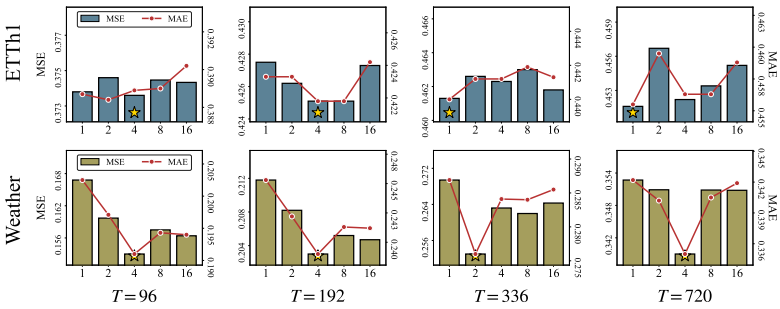


Figure 11: Sensitivity experiment on the number of groups G.

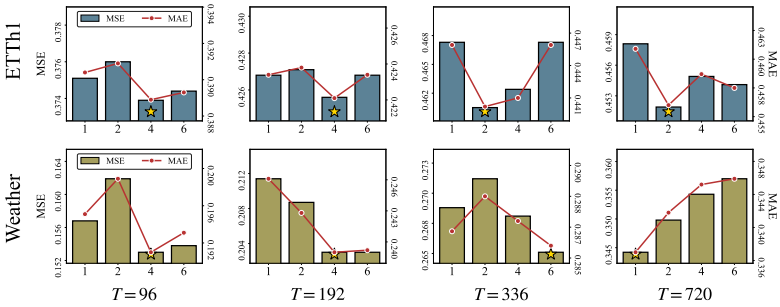


Figure 12: Sensitivity experiment on the number of rounds R.

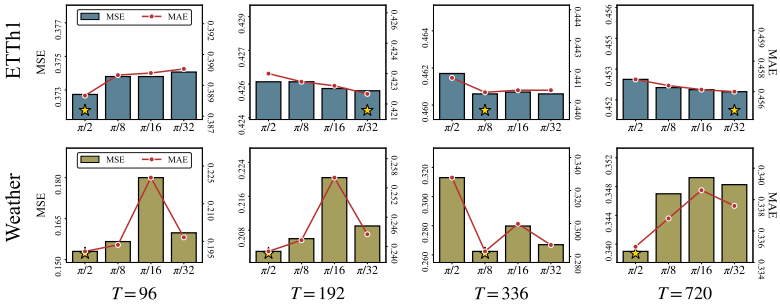


Figure 13: Sensitivity experiment on the rotation angle bound $\theta_{\max}$.

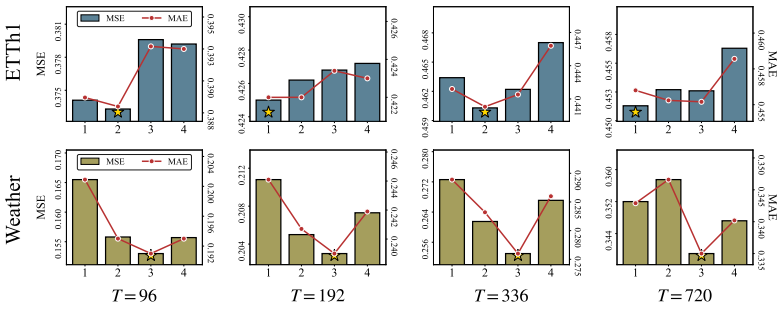
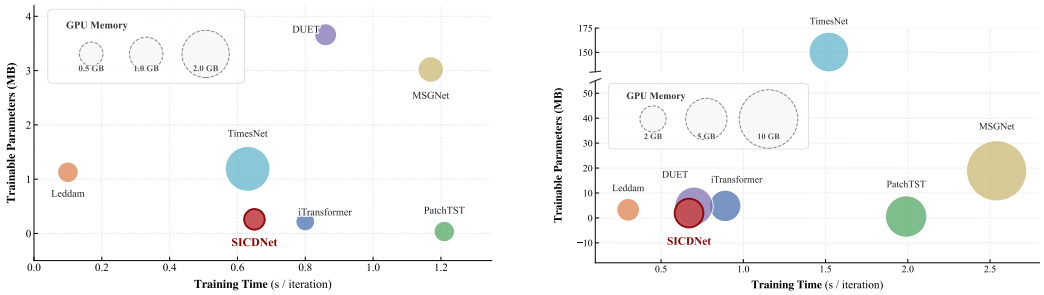


Figure 14: Sensitivity experiment on the number of SICD layer E.

A.12 COMPUTATIONAL EFFICIENCY ANALYSIS

While SICDNet demonstrates superior predictive capabilities across the majority of forecasting scenarios, as evidenced in Table 1 (or Table 8), it is equally crucial to ascertain whether these performance gains are accompanied by high computational efficiency. Therefore, as illustrated in Figure 15, we benchmark SICDNet against six representative baselines across three key dimensions: training speed (seconds/iteration), GPU memory footprint (GB), and parameter count (MB). Although SICDNet trails Leddam in terms of training speed, it generally preserves a relatively low parameter footprint and high computational efficiency, simultaneously delivering enhanced predictive performance.



(a) ETTh2 (7 Variables)

(b) ECL (321 Variables)

Figure 15: Computation effectiveness analysis on ETTh2 and ECL (Input 96-Predict 96). Due to the unavailability of public codebases or execution scripts for certain baselines compared in Table 1, their computational efficiency plot are omitted. To ensure a fair comparison, the batch size is uniformly set to 32 across all evaluated methods. Additionally, the size of the circles denotes the peak GPU memory footprint.