# REPROGAMMING CROSS-CHANNEL INTERACTION FOR NOISE-RESILIENT TIME SERIES FORECASTING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Transformer-based models have achieved remarkable performance in multivariate time series forecasting, benefiting from the attention mechanism. However, current Transformer-based forecasting paradigms still suffer from channel noise to the point that even perturbations from a single channel can significantly degrade overall performance. This vulnerability arises because: noise propagation caused by token-to-token attention (*mechanism level*), loss of global information (*representation level*), and the lack of cross-channel interaction regularization (*learning level*). To address these challenges, we propose three innovations: *(i)* reprogramming attention to an MLP-based 'Contextual-weighted Interaction Block' (CoIn), *(ii)* introducing a learnable 'Global Token' to preserve global information, and *(iii)* employing 'Channel Decoupling Loss' to regularize cross-channel interaction. Particularly, we renovate the traditional attention, which inevitably causes noise propagation between channels, into CoIn, which suppresses channel noise by reprogramming information flow in attention. A learnable 'Global Token' is employed to aggregate and redistribute global information. We also introduce 'Channel Decoupling Loss', which regularizes cross-channel interactions during the learning process. Experiments on benchmark datasets show that these tools improve robustness to channel noise and outperform Transformer-based models in forecasting accuracy and efficiency. Code is available at this **Anonymous Repo**: https://anonymous.4open.science/r/ResNo-laoda24/.

## 1 INTRODUCTION

Time series forecasting has been widely applied across various fields (Torres et al., 2021), including economics (Lin et al., 2024b), energy (Zhou et al., 2022), weather (Du et al., 2023; Challu et al., 2023), and traffic (Shao et al., 2022) in recent years. With the rapid development of deep learning technologies in the past decade, focus on time series forecasting has shifted from traditional statistical methods like ARIMA (Geurts et al., 1977) to deep learning models such as Convolution Neural Networks (CNNs) (Liu et al., 2022; Li et al., 2025), Recurrent Neural Networks (RNNs) (Lin et al., 2023; Jia et al., 2024), Multi-Layer Perceptron (MLPs) (Wang et al., 2024a; Das et al., 2023), and Transformers (Woo et al., 2022; Cirstea et al., 2022). In particular, Transformer-based models have demonstrated remarkable capability with self-attention design, especially in capturing long-term dependencies for multivariate time series forecasting (Nie et al., 2023; Dai et al., 2024).

Despite its outstanding performance, Transformer models with self-attention mechanisms still struggle to manage the ***Channel noise***, a common issue in real-world multivariate time series data (Touvron et al., 2021b; Ilyas et al., 2019; Jain & Wallace, 2019). Channel noise refers to disturbances affecting individual variables or channels, typically caused by sensor inaccuracies, environmental variations, or system-level disruptions (Zerveas et al., 2021; Yue et al., 2022). If not properly suppressed, such noise can propagate across channels and significantly degrade model robustness. For instance, the widely used iTransformer Liu et al. (2024b) suffers from noticeable performance degradation and rank collapse (reduction of rank in attention maps) when a single channel is corrupted with noise, as illustrated in Figure 1 **(a)**. To understand this vulnerability, we visualize the final-layer attention maps in Figure 1 **(b)**, revealing a phenomenon known as *attention collapse* (Ilbert et al., 2024; Zhai et al., 2023; Anagnostidis et al., 2022), where attention scores concentrate sharply, reducing rank and diversity in representation. This collapse limits the model's expressiveness and contributes to forecasting failure under noisy conditions.



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.

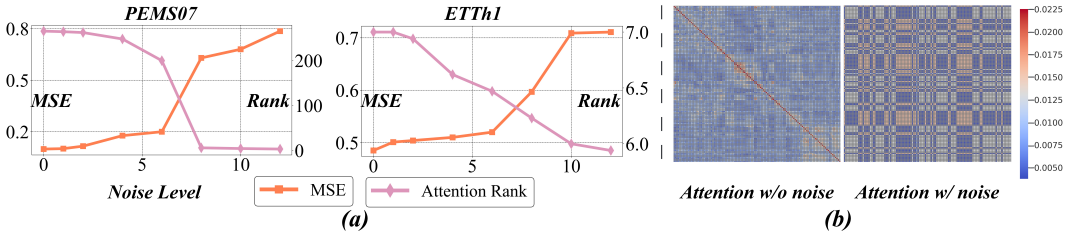


Figure 1: **(a)**: Visualization of **iTransformer**'s MSE and average Rank of the attention map on PEMS07 (883 channels) and ETTh1 (7 channels) dataset. We consistently increase the intensity of Gaussian random noise on the last channel. **(b)**: Example of attention collapse, where the attention map shows a lower rank and entropy.

These observations reveal that channel noise undermines Transformer-based time series models across three interconnected levels: ***(i) Mechanism level – Noise Propagation:*** Due to joint attention computation across channels, noise in one channel can propagate and contaminate others (Touvron et al., 2021b; Ilyas et al., 2019; Jain & Wallace, 2019), leading to information entanglement and reduced robustness. ***(ii) Representation level – Global Context Distortion:*** Channel noise disrupts global token interactions, degrading the model's ability to capture long-range dependencies (Khan et al., 2022; Touvron et al., 2021c; Han et al., 2021), a crucial aspect often overlooked in time series modeling. ***(iii) Learning level – Insufficient Regularization:*** Without explicit regularization, models tend to overfit spurious cross-channel correlations. While regularization strategies like token dropping have shown promise in vision (Yin et al., 2022; Gupta et al., 2023; Touvron et al., 2021a), they remain underexplored in time series, where each series may carry distinct semantic signals. Therefore, it is necessary to design a principled framework that explicitly suppresses noise propagation, preserves global representations, and regularizes cross-channel interactions to enhance robustness in multivariate time series forecasting.

To address the multi-level challenges posed by channel noise, we propose **ResNo**, a noise-resilient and efficient framework for multivariate time series forecasting. At the *mechanism level*, ResNo addresses *noise propagation* by replacing conventional attention with a **Contextual-weighted Interaction Block (CoIn)**, which reprograms attention flow to isolate and suppress the influence of noisy channels. At the *representation level*, to preserve *global contextual information*, a largely overlooked but crucial aspect in time series modeling (Han et al., 2024; Zhou et al., 2024; Wang et al., 2023; Yang et al., 2024), we incorporate a **learnable Global Token (GloT)**, inspired by Vision Transformers (Dosovitskiy et al., 2021). GloT aggregates holistic features during encoding and redistributes them across channels, enhancing global coherence. At the *learning level*, to counteract the model's tendency to overfit spurious cross-channel dependencies, we introduce a novel **Channel Decoupling Loss (CDL)**. Unlike token-level dropout used in vision (Yin et al., 2022; Gupta et al., 2023; Touvron et al., 2021a), CDL weakens inter-channel interactions by design, promoting channel-wise independence and preventing early-stage noise entanglement. Together, they make ResNo a principled and robust solution to the pervasive challenge of channel noise in multivariate time series forecasting. Furthermore, we conduct extensive experiments across nine benchmarks, demonstrating that **ResNo** achieves state-of-the-art performance and strong noise resilience. Besides, ResNo is far more efficient than Transformer-based methods, introducing only **Linear complexity**.

## 2 Related Work

**Advancement in Deep Learning-based Time Series Forecasting** Recent years have witnessed significant advances in deep learning for time series forecasting, driven by the adaptation of various model architectures. Recurrent Neural Networks (RNNs) and their variants were among the earliest deep learning models applied to sequential data (Lin et al., 2023). Convolutional Neural Networks (CNNs) have also been utilized to capture local temporal patterns (Deng et al., 2024). The emergence of Transformer-based models marked a turning point, enabling more effective modeling of long-range dependencies (Chen et al., 2024) and inter-series dependencies (Wang et al., 2024b; Yu et al., 2024). Other recent architectures such as Mamba (Wang et al., 2024d), MLP-based models (Lin et al., 2024a), and Graph Neural Networks (GNNs) (Yi et al., 2023a) have further diversified the design space. More recently, Large Language Models (LLMs) have been adapted for time series tasks, demonstrating strong generalization capabilities (Jin et al., 2024; Gruver et al., 2024).

**Channel Noise in Multivariate Time Series** Channel noise, *i.e.*, perturbations affecting individual time series variables due to sensor faults, environmental shifts, or system-level disturbances, is prevalent in real-world scenarios (Zerveas et al., 2021; Yue et al., 2022). This noise can lead to reduced forecasting accuracy (Liu et al., 2024a), unstable learned representations (Liu et al., 2023), and decreased model robustness (Niu et al., 2024). While robustness to noise has been studied in domains like computer vision (Yin et al., 2022; Gupta et al., 2023; Touvron et al., 2021a), relatively few works have investigated its implications for multivariate time series forecasting. In particular, the risk of noise propagation across variables, where a corrupted channel affects others through shared modeling mechanisms, is rarely addressed in the literature.

**Preserving Global Context in Forecasting Models** Global context is essential in multivariate forecasting for capturing overarching temporal trends and relationships that span across variables (Han et al., 2024; Zhou et al., 2024; Wang et al., 2023; Yang et al., 2024). However, most time series models focus on local or pairwise dependencies and lack explicit mechanisms to aggregate and preserve global information (Liu et al., 2024b; Wang et al., 2024b). In the general vision domain, Vision Transformers have introduced learnable global tokens to enhance context modeling (Dosovitskiy et al., 2021). Despite their success, similar strategies remain largely unexplored in time series forecasting. This gap is particularly significant in noisy environments, where global features can serve as a stabilizing signal amid localized perturbations (Khan et al., 2022; Han et al., 2021).

**Regularization for Causal and Disentangled Representations** In the vision domain (Yin et al., 2022; Gupta et al., 2023; Touvron et al., 2021a), regularization techniques such as token dropping have shown potential in improving robustness. But in multivariate time series analysis, each series possesses unique semantic information that cannot be inferred from other variates (Granger, 1969). Relying solely on information from other channels to reconstruct the representation of the current channel is almost infeasible and unreliable, as the other series are merely external variables (Hyndman & Athanasopoulos, 2018; Lütkepohl, 2005). Thus, a more suitable regularization technic is needed.

*In light of these observations, there is a pressing need for models that are resilient to channel noise, capable of robust cross-channel interaction modeling, and equipped to preserve global context.*

## 3 METHODOLOGY

Given a multivariate time series input $X \in \mathbb{R}^{N\times T}$, multivariate time series forecasting tasks are designed to predict its future $F$ time steps $Y \in \mathbb{R}^{N\times F}$ using past $T$ steps, where $N$ denotes the number of variates or channels.

### 3.1 MOTIVATING ANALYSIS: WHY ATTENTION PROPAGATES NOISE

Transformers have demonstrated strong performance in multivariate time series forecasting due to their ability to capture complex inter-channel dependencies. However, in noisy environments, the vanilla attention mechanism becomes a double-edged sword: while it enables global interaction, it also indiscriminately propagates noise across channels.

In a standard attention block, each channel's updated representation is computed by aggregating information (*e.g.* Query, Key, Value) from *all* channels, weighted by attention scores, as shown in Figure 2 (**Left**). Formally, for an input embedding $E \in \mathbb{R}^{N\times D}$ (where $N$ is the number of channels and $D$ the embedding dimension), attention operates via:

$$Q = EW_Q + b, \quad K = EW_K + b, \quad V = EW_V + b, \quad Q, K, V \in \mathbb{R}^{N\times D},$$

$$A_{i,:} = \sum_{j=1}^{N} \alpha_{ij} V_{j,:}, \quad \alpha_{ij} = \frac{\exp(Q_{i,:} \cdot K_{j,:}/\sqrt{D})}{\sum_{k=1}^{N} \exp(Q_{i,:} \cdot K_{k,:}/\sqrt{D})}. \tag{1}$$

If even a single channel (*e.g.*, channel $N$) is corrupted by noise, its key and value vectors ($K_{N,:}$, $V_{N,:}$) will influence all other channels via their attention weights $\alpha_{iN}$, for all $i \in \{1, \ldots, N\}$. This leads to *attention collapse*, where noisy or unreliable information is broadcast across the entire model, degrading both representation quality and forecasting accuracy.

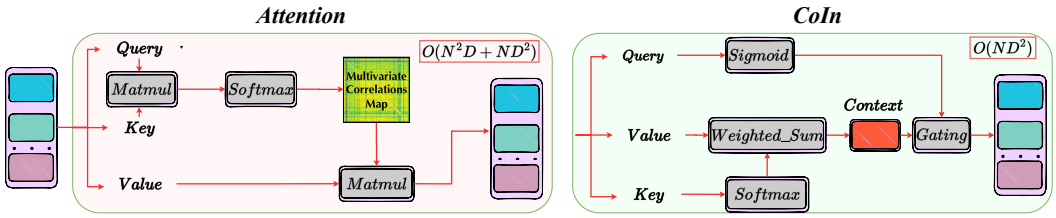


Figure 2: Illustration of attention and CoIn. While attention updates its feature using attention maps based on matrix multiplications between $Query$ and $Key$, CoIn updates its representation by interacting individually with a Channel-Dependent token $Context$ using Channel-Independent $Gate$.

### 3.2 Contextual-weighted Interaction Block (CoIn)

To address the attention collapse, we ask: *can we maintain the benefits of attention (flexible, dynamic weighting) while preventing one noisy channel from contaminating the rest?* Motivated by this, we propose the **Contextual-weighted Interaction Block (CoIn)**, a plug-in module that decouples attention into two interpretable and noise-tolerant stages, as shown in Figure 2 (**Right**):

- **Contextual aggregation**: Channel-wise features are softly aggregated by the weighted-sum of $Value$ into a single shared token $Context$, using channel-dependent softmax weights calculated from $Key$ (analogous to attention over value).
- **Independent gating**: Each channel then independently determines how much of the shared token $Conetxt$ to absorb, via a channel-specific gating mechanism (analogous to query information for each channel).

Specifically, given the same embedding $E \in \mathbb{R}^{N\times D}$ and obtained Q, K, V, CoIn operates via:

$$G_{i,d} = \text{Sigmoid}(Q_{i,d}), \quad w_{j,d} = \frac{\exp(K_{j,d})}{\sum_{k=1}^{N}\exp(K_{k,d})}, \quad H_d = \sum_{j=1}^{N} w_{j,d}V_{j,d}, \quad A_{i,:} = G_{i,:} \cdot H. \tag{2}$$

Here, $H$ is the shared contextual token $Context$, $G_{i,:}$ is the channel-independent gate, and $A_{i,:}$ is the updated representation for channel $i$. Crucially, a noisy channel may affect $H$, but the final influence on any other channel is mitigated by its own gate. This significantly *localizes noise impact*.

**Computational efficiency of CoIn** CoIn introduces $O(ND^2)$ complexity compared to $O(N^2D + ND^2)$ in standard attention. This improvement stems from eliminating pairwise dot-product computations between Q and K. As a result, CoIn is not only more robust under noise, but also computationally lighter, making it suitable for high-dimensional multivariate forecasting.

### 3.3 A Noise Resilient Multivariate Time Series Forecasting Framework

Building on CoIn, we present **ResNo**, a framework designed to improve the robustness of multivariate time series forecasting under channel noise. As illustrated in Figure 3, ResNo comprises three key components: *(i)* CoIn for learning stable inter-channel dependencies, *(ii)* CDL to explicitly regularize cross-channel noise propagation, and *(iii)* GloT to enhance global contextual modeling.

Given a multivariate time series input $X \in \mathbb{R}^{N\times T}$, we use a Linear projection to embed $X$ into the Variate Embedding (Liu et al., 2024b). Then a Global token, which is learnable and initialized using *Normal Distribution*, is concatenated in the **channel dimension**, as in VisionTransformer (Dosovitskiy et al., 2021), to result in $E \in \mathbb{R}^{(N+1)\times D}$. Then, $E$ goes through standard **Transformer Encoder** by $S$ times with attention replaced using CoIn, generating representation $O_S \in \mathbb{R}^{(N+1)\times D}$ ($O_0 = E$). The Global token is repeated and concatenated to the **temporal dimension** of each token to distribute global information, following the STAR operation (Han et al., 2024), which results in $\hat{O}_S \in \mathbb{R}^{N\times D}$. This is followed by an MLP to fuse global information and channel-wise local information. Then, a Linear projection is used to obtain the final forecasting $\hat{Y}$. To stabilize learning and improve adaptability across scales, we apply Reversible Instance Normalization (RevIN) (Kim et al., 2022), which normalizes each channel to zero mean and unit variance. This normalization is then reversed to restore the original scale in the obtained $\hat{Y}$.

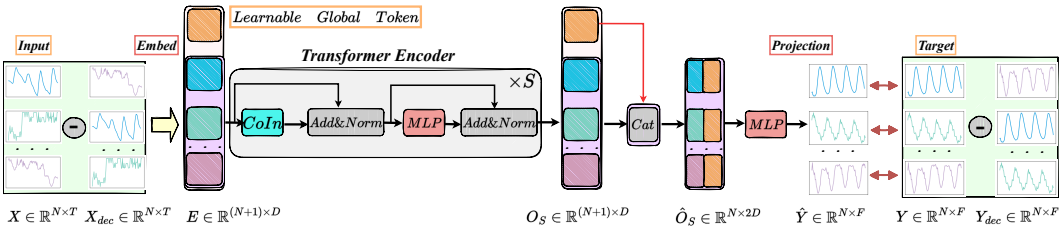


Figure 3: Overall architecture of the proposed **ResNo**, a noise-resilient framework for multivariate time series forecasting. We employ 'Contextual-weighted Interaction Block' (CoIn) to efficiently capture inter-series dependencies, utilize 'Channel Decoupling Loss' (CDL) to constrain cross-channel interactions, and leverage a learnable 'Global Token' (GloT) to preserve and redistribute global information.

**Channel Decoupling Loss** To explicitly prevent the influence of noisy channels on clean ones, we introduce *Channel Decoupling Loss*, which weakens cross-channel coupling during training. Given a sample pair $X \in \mathbb{R}^{N\times T}$ and target $Y \in \mathbb{R}^{N\times F}$, we construct a decoupled version by subtracting randomly selected channel-wise signals:

$$\tilde{X}_{i,:} = X_{i,:} - X_{j,:}, \quad \tilde{Y}_{i,:} = Y_{i,:} - Y_{j,:}, \quad i \neq j. \tag{3}$$

The index $j$ is randomly sampled per channel based on the largest element in a noise-injected matrix $\omega_{j,:} \in \mathbb{R}^N$, whose elements follow a normal distribution. These yields decoupled samples $X_{\text{dec}}$ and $Y_{\text{dec}}$, forming:

$$\tilde{X} = X - X_{\text{dec}}, \quad \tilde{Y} = Y - Y_{\text{dec}}. \tag{4}$$

Let $\hat{Y}$ and $\hat{\tilde{Y}}$ denote the model predictions for $X$ and $\tilde{X}$, respectively. The overall training loss combines both the original and decoupled objectives:

$$\mathcal{L} = \frac{1}{NF}\sum_{i=1}^{N}\sum_{t=1}^{F}|\hat{Y}_{i,t} - Y_{i,t}|^2 + \frac{1}{NF}\sum_{i=1}^{N}\sum_{t=1}^{F}|\hat{\tilde{Y}}_{i,t} - \tilde{Y}_{i,t}|^2. \tag{5}$$

Because the same channel-wise transformation is applied consistently to both the historical input and its corresponding forecasting target, the auxiliary task remains aligned with the original forecasting objective rather than introducing an artificial prediction target. Joint optimization on the original and decoupled views encourages the model to preserve channel-specific predictive information while reducing its reliance on brittle cross-channel shortcuts. Moreover, the random channel pairing exposes the model to diverse channel-decoupled views throughout training, acting as a structured regularization that improves robustness to heterogeneous channels. Importantly, CDL is used only during training and introduces no computational overhead at inference; ResNo directly forecasts from the original input through the CoIn-based encoder and global-token pathway.

### 3.4 Complexity Analysis

We analyze the complexity of each component of ResNo step by step concerning window length $T$, number of channels $N$, model dimension $D$, and forecasting horizon $F$. The complexity of the RevIN and Variate Embedding is $O(NT)$ and $O(NTD)$ respectively. In ResNo, CoIn has complexity $O(ND^2)$, the MLP is a $\mathbb{R}^D \mapsto \mathbb{R}^D$ mapping with complexity $O(ND^2)$. The complexity of the forecasting projection is $O(NDF)$. In all, the complexity of ResNo is $O(NTD + ND^2 + NDF)$, which is linear to $N$, $T$, and $F$. Ignoring the model dimension $D$, which is a constant in the algorithm and irrelevant to the problem, we compare the complexity of several popular forecasters in Table 1.

Table 1: Complexity comparison between popular time series forecasters concerning window length $T$, number of channels $N$ and forecasting horizon $F$. Our method achieves only linear complexity.

| | **ResNo (ours)** | iTransformer | PatchTST | Transformer |
|---|---|---|---|---|
| Complexity | $\boldsymbol{O(NT + NF)}$ | $O(N^2 + NT + NF)$ | $O(NT^2 + NF)$ | $O(NT + T^2 + TF + NF)$ |

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL SETUP

**Datasets and preprocessing** We conduct empirical analyses on **NINE** datasets spanning multiple domains, including traffic, energy, and weather. Specifically, ETT (4 subsets) (Zhou et al., 2022), Traffic (Wu et al., 2021), Electricity (Wu et al., 2021), Weather (Wu et al., 2021), Solar_Energy (Lai et al., 2018), and PEMS (PEMS07 that has the largest 883 channels) (Liu et al., 2022) are used. We preprocess all datasets according to the methods outlined in (Liu et al., 2024b) and normalize them with the standard normalization method. For a fair comparison, we follow the same standard protocol (Liu et al., 2024b) and split all forecasting datasets into training, validation, and test sets by the ratio of 6:2:2 for the ETT and PEMS datasets, and 7:1:2 for the other datasets. The characteristics of these datasets can be found in Appendix A.1.

**Evaluation protocol** Following iTransformer (Liu et al., 2024b), we use Mean Squared Error (MSE) and Mean Absolute Error (MAE) for the evaluation. We follow the same evaluation protocol, where the input length is set as $T = 96$, the forecasting lengths $F \in \{12, 24, 48, 96\}$ for PEMS, and $F \in \{96, 192, 336, 720\}$ for others. To foster reproducibility, we make our code and training scripts available in this **Anonymous Repo**[1]. Full implementation details are in Appendix A.2.

**Benchmark models** We carefully choose **ELEVEN** well-acknowledged forecasting models as our baselines, including 1) Transformer-based methods: iTransformer (Liu et al., 2024b), Crossformer (Zhang & Yan, 2023), PatchTST (Nie et al., 2023); 2) Linear-based or MLP-based methods: DLinear (Zeng et al., 2023), FITS (Xu et al., 2024), TSMixer (Chen et al., 2023), FreTS (Yi et al., 2023b); 3) CNN-based methods: TimesNet (Wu et al., 2023), ModernTCN (Donghao & Xue, 2024); 4) Filter-based methods: FilterNet (Yi et al., 2024); 5) GNN-based methods: CrossGNN (Huang et al., 2023). These models represent the latest advancements in multivariate time series forecasting and encompass all mainstream prediction model types.

## 4.2 MAIN RESULTS

Table 2: Multivariate forecasting results with fixed lookback length $T = 96$. Results are averaged from all prediction lengths (prediction lengths $F \in \{12, 24, 48, 96\}$ for PEMS and $F \in \{96, 192, 336, 720\}$ for others). ***Avg*** means further averaged by subsets. The best is **Red**, and the second is Blue.

| Models | **ResNo (Ours)** | | FilterNet (2024) | | iTransformer (2024b) | | ModernTCN (2024) | | FITS (2024) | | PatchTST (2023) | | Crossformer (2023) | | TimesNet (2023) | | TSMixer (2023) | | DLinear (2023) | | FreTS (2023b) | | CrossGNN (2023) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETT(Avg) | **0.371** | **0.389** | 0.375 | 0.394 | 0.383 | 0.399 | 0.374 | 0.390 | 0.376 | 0.395 | 0.384 | 0.396 | 0.381 | 0.397 | 0.685 | 0.578 | 0.391 | 0.404 | 0.388 | 0.402 | 0.442 | 0.444 | 0.419 | 0.428 |
| ECL | **0.171** | **0.264** | 0.173 | 0.268 | 0.178 | 0.270 | 0.197 | 0.282 | 0.217 | 0.295 | 0.205 | 0.290 | 0.244 | 0.334 | 0.192 | 0.295 | 0.186 | 0.287 | 0.212 | 0.300 | 0.189 | 0.278 | 0.201 | 0.300 |
| Traffic | **0.409** | **0.265** | 0.463 | 0.310 | 0.428 | 0.282 | 0.546 | 0.348 | 0.627 | 0.376 | 0.481 | 0.304 | 0.550 | 0.304 | 0.620 | 0.336 | 0.522 | 0.357 | 0.625 | 0.383 | 0.618 | 0.390 | 0.583 | 0.323 |
| Weather | 0.253 | 0.276 | **0.245** | **0.272** | 0.258 | 0.279 | 0.247 | 0.272 | 0.249 | 0.276 | 0.259 | 0.281 | 0.259 | 0.315 | 0.259 | 0.287 | 0.256 | 0.279 | 0.265 | 0.317 | 0.250 | 0.270 | 0.247 | 0.289 |
| Solar_Energy | **0.230** | **0.257** | 0.243 | 0.283 | 0.233 | 0.262 | 0.244 | 0.286 | 0.395 | 0.407 | 0.270 | 0.307 | 0.641 | 0.639 | 0.301 | 0.319 | 0.260 | 0.297 | 0.330 | 0.401 | 0.248 | 0.296 | 0.249 | 0.313 |
| PEMS | **0.086** | **0.182** | 0.149 | 0.258 | 0.101 | 0.204 | 0.160 | 0.257 | 0.390 | 0.416 | 0.211 | 0.303 | 0.235 | 0.315 | 0.124 | 0.225 | 0.112 | 0.217 | 0.329 | 0.395 | 0.185 | 0.282 | 0.170 | 0.281 |

**ResNo** consistently achieves SOTA performance across multiple datasets, as shown in Table 2. Notably, it excels in forecasting high-dimensional datasets, such as PEMS (883 variates) and Traffic (862 variates), where capturing **inter-series dependencies** is crucial. Interestingly, on these datasets, the second-best model remains **iTransformer**, significantly outperforming other baselines. This observation suggests that native Transformer components are relatively powerful at capturing multivariate correlations. However, ResNo further surpasses iTransformer by a significant margin, illustrating its enhanced capacity for robust dependency modeling. For instance, it achieves a notable MSE reduction on PEMS (0.101→**0.086**), a 15% improvement, and Traffic (0.428→**0.409**), a 4% improvement. Moreover, on datasets (ETT) with fewer variates (*e.g.* 7 variates), where ***temporal variation*** is the dominant factor, models with Channel-Independent designs, such as PatchTST and ModernTCN, typically perform well. Nonetheless, our method **ResNo** still achieves the best performance. These results confirm that ResNo is not only capable of modeling complex inter-series dependencies in high-dimensional data but also excels in capturing temporal dynamics in lower-dimensional settings.

[1] https://anonymous.4open.science/r/ResNo-laoda24/

Table 3: Ablation of the 'Channel Decoupling Loss (**CDL**)' and the 'Global Token (**GloT**)'. Results are averaged from all prediction lengths (prediction lengths $F \in \{12, 24, 48, 96\}$ for PEMS and $F \in \{96, 192, 336, 720\}$ for others).

| CDL | GloT | ETTm1 | | ETTm2 | | ETTh1 | | ETTh2 | | ECL | | Traffic | | Weather | | Solar_Energy | | PEMS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| - | - | 0.398 | 0.404 | 0.287 | 0.332 | 0.448 | 0.442 | 0.386 | 0.409 | 0.177 | 0.270 | 0.417 | 0.270 | 0.258 | 0.280 | 0.236 | 0.267 | 0.093 | 0.196 |
| - | ✓ | 0.395 | 0.402 | 0.285 | 0.330 | 0.446 | 0.441 | 0.378 | 0.403 | 0.175 | 0.268 | 0.417 | 0.270 | 0.257 | 0.279 | 0.234 | 0.263 | 0.090 | 0.188 |
| ✓ | - | 0.396 | 0.401 | 0.283 | 0.327 | 0.444 | 0.440 | 0.386 | 0.409 | 0.172 | 0.264 | 0.411 | 0.265 | 0.254 | 0.276 | 0.232 | 0.262 | 0.087 | 0.187 |
| ✓ | ✓ | **0.392** | **0.399** | **0.278** | **0.322** | **0.438** | **0.435** | **0.377** | **0.402** | **0.171** | **0.264** | **0.409** | **0.265** | **0.253** | **0.276** | **0.230** | **0.257** | **0.086** | **0.182** |

Table 4: Further ablation study of the proposed 'Channel Decoupling Loss (**CDL**)' and the 'Global Token (**GloT**)'. We test their generalizability on **iTransformer** (Liu et al., 2024b) with prediction length $F = 720$ and fixed look-back length $T = 96$.

| | CDL | GloT | ETTm1 | | ETTm2 | | ETTh1 | | ETTh2 | | ECL | | Traffic | | Weather | | Solar_Energy | | PEMS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| iTransformer | - | - | 0.486 | 0.455 | 0.411 | 0.405 | 0.493 | 0.483 | 0.429 | 0.447 | 0.216 | 0.304 | 0.467 | 0.307 | 0.358 | 0.350 | 0.254 | 0.281 | 0.159 | 0.263 |
| | - | ✓ | 0.488 | 0.455 | 0.409 | 0.404 | 0.481 | 0.479 | 0.421 | 0.441 | 0.209 | 0.299 | 0.458 | 0.297 | 0.356 | 0.349 | 0.252 | 0.278 | 0.155 | 0.261 |
| | ✓ | - | 0.484 | 0.453 | 0.404 | 0.400 | **0.481** | **0.476** | 0.427 | 0.446 | 0.213 | 0.303 | 0.454 | 0.296 | 0.358 | 0.348 | 0.252 | 0.280 | 0.134 | 0.236 |
| | ✓ | ✓ | **0.475** | **0.446** | **0.400** | **0.397** | 0.488 | 0.482 | **0.421** | **0.441** | **0.207** | **0.297** | **0.449** | **0.289** | **0.355** | **0.348** | **0.247** | **0.275** | **0.130** | **0.233** |

Table 5: The performance of **CoIn** in different models. The attention replaced by **CoIn** here are the temporal attention in PatchTST (Nie et al., 2023), the channel attention in iTransformer (Liu et al., 2024b). The results demonstrate that replacing attention with CoIn, which requires fewer computational resources, could significantly improve the models' performance. We set the prediction length $F = 720$ and look-back length $T = 96$.

| Model | Design | ETTm1 | | ETTm2 | | ETTh1 | | ETTh2 | | ECL | | Traffic | | Weather | | Solar_Energy | | PEMS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| PatchTST | Attention | 0.466 | 0.443 | 0.403 | 0.401 | 0.490 | 0.476 | 0.437 | 0.451 | 0.235 | 0.316 | 0.488 | 0.301 | **0.353** | **0.347** | 0.259 | 0.282 | 0.375 | 0.412 |
| | CoIn | **0.462** | **0.439** | **0.401** | **0.400** | **0.488** | **0.474** | **0.421** | **0.442** | **0.232** | **0.312** | **0.481** | **0.299** | 0.355 | 0.348 | **0.252** | **0.278** | **0.363** | **0.403** |
| iTransformer | Attention | 0.486 | 0.455 | 0.411 | 0.405 | 0.493 | 0.483 | **0.429** | **0.447** | 0.215 | 0.304 | 0.467 | 0.307 | 0.358 | 0.350 | 0.254 | 0.281 | 0.159 | 0.263 |
| | CoIn | **0.479** | **0.449** | **0.409** | **0.404** | **0.476** | **0.476** | 0.435 | 0.451 | **0.213** | **0.303** | **0.459** | **0.299** | **0.355** | **0.349** | **0.251** | **0.279** | **0.140** | **0.246** |

### 4.3 Ablation Study

**Ablation on CDL and GloT** To verify the functionality of '**C**hannel **D**ecouling **L**oss' (CDL) and '**Glo**bal **T**oken' (GloT), we provide detailed ablations in Table 3. Both CDL and GloT significantly enhance the performance of the raw model, and their combination leads to even further improvements. It is highlighted on high-dimensional datasets, such as ECL (321 variates), Traffic (862 variates), and PEMS (883 variates). This suggests that preserving and redistributing global information, as done in Vision Transformer, can enhance the model's ability. Additionally, CDL further strengthens this effect by explicitly constraining the model to learn channel correlations. The generalization capability of both methods is also demonstrated in Table 4, further highlighting their adaptability across different models.

**Universality of CoIn** The 'Contextual-weighted Interaction Block' (CoIn) is a versatile module that can replace the attention mechanism in any Transformer-based model. We evaluate the effectiveness of CoIn across renowned Transformer-based forecasters, PatchTST and iTransformer. The former represents the most advanced Channel-Independent approach, while the latter is an outstanding Channel-Dependent multivariate time series forecaster. The results in Table 5, demonstrate that replacing the attention with CoIn, which requires fewer computational resources, can enhance model performance consistently.

### 4.4 Model Analysis

**Impact of CoIn and CDL on series embedding** To intuitively illustrate the roles of CoIn and CDL, we visualize the series embeddings for all variates on ECL dataset before and after their

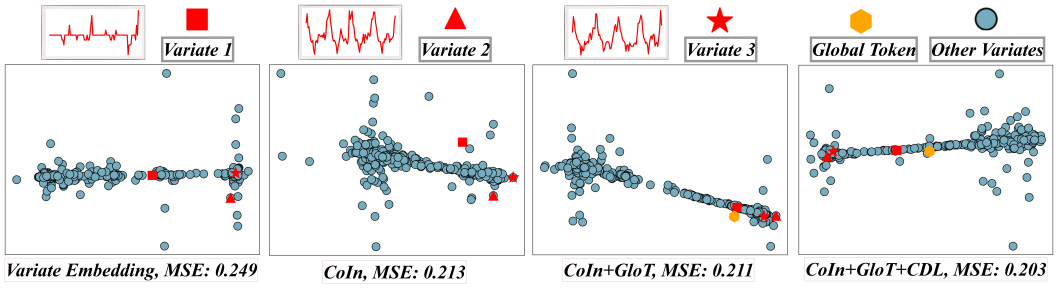


Figure 4: T-SNE visualization of the impact of CoIn and CDL on series embeddings. We chose the $input-96-forecast-720$ task on ECL dataset. Three example variates are highlighted: variates 2&3 shared a channel correlation, while variate 1 did not. We also plotted the ***Global Token***.

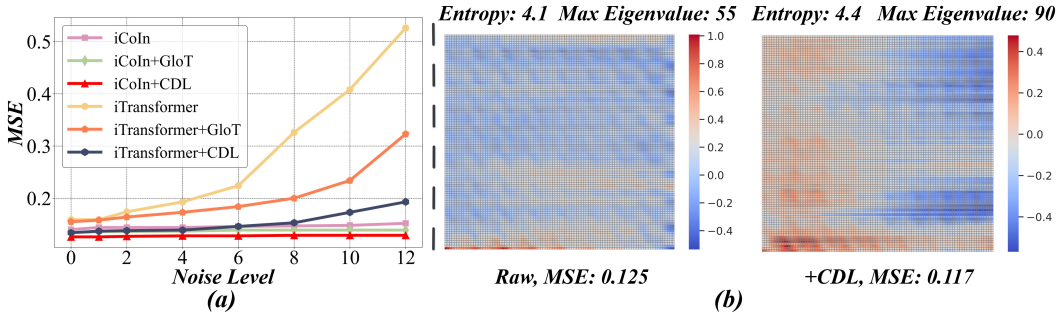


Figure 5: Impact of CDL and GloT on different methods and visualization of model weights. We choose $input-96-forecast-96$ tasks on PEMS dataset. **(a)**: Impact of CDL and GloT on iTransformer and iCoIn (Replace self-attention with CoIn Block). We consistently increase the intensity (the standard deviation) of Gaussian random noise from 0.0 to 12.0 on the last channel. **(b)**: **Abstract Weights** before and after adding CDL on PEMS dataset. We compute the *Max Eigenvalue* and the column-wise *Entropy* of the weight matrix to quantify the change. Both the higher, the better.

respective adjustments in Figure 4. We highlight three representative examples: variates 2 and 3 exhibit strong temporal and contextual similarity, while variate 1 shows distinct patterns—indicating weak correlation with the others, as shown in the top part of Figure 4. In the raw Variate Embedding generated using a Channel-Independent way, the model fails to capture inter-series dependencies, resulting in a poorly structured embedding space and a high MSE of 0.249. After applying CoIn, which explicitly models inter-channel interactions, the embeddings of variates 2 and 3 are drawn closer together while variate 1 remains separated, reflecting their underlying relationships and reducing MSE to 0.213. The incorporation of GloT (Global Token) successfully represents the global information and forms a highly structured embedding space. This leads to a higher model performance and reduces MSE to 0.211. Finally, the introduction of CDL (Channel Decoupling Loss) enhances clustering structure, strategically positioning the Global Token (in the middle of all the blue variates) and further lowering the MSE to 0.203.

**Robustness to channel noise of different methods** To test the robustness of different methods to channel noise, we introduce noise progressively during training by adding perturbations to the last channel of the PEMS dataset. This formulated as $\hat{X}_{N,:} = X_{N,:} + \beta \cdot \text{noise}$, where $X_{N,:}, \hat{X}_{N,:} \in \mathbb{R}^{1\times T}$ is the last channel, $noise \in \mathbb{R}^{1\times T}$ is Gaussian noise with mean 0 and standard deviation 1, $\beta \in \mathbb{R}^1$ controls the noise intensity. Figure 5 (**a**) reveals that attention is highly sensitive to noise. While GloT and CDL help alleviate this issue, some degradation remains. In contrast, CoIn achieves a balanced integration of **Channel-Independence** and **Channel-Dependence** by reprogramming the information flow in attention. This enhances robustness against channel noise, as it prevents noisy channels from directly interfering with others, while still capturing inter-channel dependencies effectively. Meanwhile, by preserving and redistributing global information, GloT enhances the model's robustness against channel noise. CDL successfully counteracts noise by constraining the model to capture inter-channel correlations.

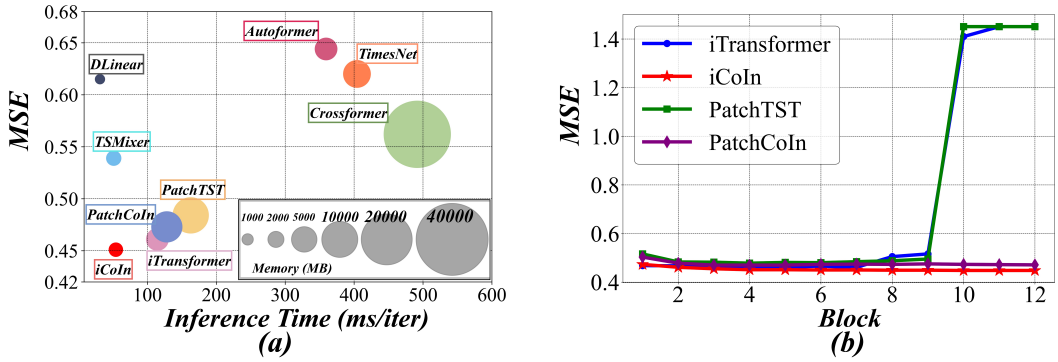


Figure 6: Efficiency and depth scaling of CoIn on Traffic with $T = 96$, $F = 720$, and $B = 32$. **(a)**: Memory and computation time of representative attention-, CoIn-, linear-, and MLP-based forecasters. **(b)**: Forecasting performance of PatchTST/iTransformer and their CoIn variants across depths.

**Visualization of model weights** To test whether CDL changes the learned dependency structure, we visualize the **Abstract Weights** in Figure 5 (**b**) using the effective-weight extraction procedure in Appendix B.2. Unlike an attention map, this matrix summarizes the trained model's effective input–output coupling. With CDL, the weights become more structured while remaining distributed across channels. The larger *Max Eigenvalue* reflects a more pronounced dominant interaction mode, whereas the higher column-wise *Entropy* indicates less concentration on a small subset of channels. Together, these changes suggest that CDL does not simply weaken cross-channel interaction; it reorganizes interaction from brittle channel-specific shortcuts toward coherent yet distributed coupling. The same trend is observed on ECL, Solar_Energy, and Traffic in Appendix B.2, while PEMS MSE improves from 0.125 to 0.117.

**Efficiency of CoIn** Figure 6 (**a**) confirms the complexity analysis in Section 3.4. By removing pairwise query–key interactions, CoIn avoids constructing an $N \times N$ attention map and scales linearly with the number of channels. Replacing attention with CoIn therefore reduces memory usage and computation time in iTransformer and PatchTST while maintaining, and often improving, forecasting accuracy. Its competitiveness against DLinear and TSMixer further shows that this efficiency does not require discarding cross-channel dependency modeling.

**Depth scalability of CoIn** Figure 6 (**b**) shows that attention-based variants become less stable as more blocks are stacked, whereas their CoIn counterparts remain substantially more robust across depth. This behavior matches our motivating analysis: repeated unrestricted pairwise attention can amplify cross-channel contamination, while CoIn routes shared context through channel-wise gates. Controlling this information path therefore improves robustness and makes deeper multivariate forecasting encoders more practical.

## 5 Conclusion

This work treats channel noise as a failure of unregulated information flow in multivariate forecasting. **ResNo** addresses the three failure modes identified in our motivating analysis: CoIn limits direct noise propagation at the *mechanism level*, GloT preserves global context at the *representation level*, and CDL regularizes brittle cross-channel coupling at the *learning level*. Across nine benchmark datasets, ResNo achieves consistently strong accuracy; on the two highest-dimensional benchmarks, PEMS and Traffic, average MSE decreases from 0.101 to 0.086 and from 0.428 to 0.409 relative to iTransformer. The ablation, noise-injection, embedding, effective-weight, efficiency, and depth-scaling results support the same conclusion: ***useful cross-channel dependencies should be preserved, but the pathways through which they are propagated must be controlled***. ResNo therefore provides a robust and efficient alternative to unrestricted attention for multivariate forecasting. Future work will study non-Gaussian, time-varying, and naturally occurring channel corruptions that better reflect sensor failures in real deployments.

## REFERENCES


Sotiris Anagnostidis, Luca Biggio, Lorenzo Noci, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=FxVH7iToXS.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Peng Chen, Yingying ZHANG, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=lJkOCMP2aW.

Si-An Chen, Chun-Liang Li, Sercan O Arik, Nathanael Christian Yoder, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=wbpxTuXgm0.

Razvan-Gabriel Cirstea, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting–full version. *arXiv preprint arXiv:2204.13767*, 2022.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Jinliang Deng, Feiyang Ye, Du Yin, Xuan Song, Ivor Tsang, and Hui Xiong. Parsimony or capability? decomposition delivers both in long-term time series forecasting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=wiEHZSV15I.

Luo Donghao and Wang Xue. Moderntcn: A modern pure convolution structure for general time series analysis. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vpJMJerXHU.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Dazhao Du, Bing Su, and Zhewei Wei. Preformer: Predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

Michael Geurts, George E. P. Box, and Gwilym M. Jenkins. Time series analysis: Forecasting and control. *Journal of Marketing Research*, pp. 269, 1977. doi: 10.2307/3150485. URL http://dx.doi.org/10.2307/3150485.

Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. In *ICLR*, 2023. URL https://openreview.net/pdf?id=QAV2CcLEDh.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in neural information processing systems*, 34:15908–15919, 2021.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *arXiv preprint arXiv:2404.14197*, 2024. URL https://arxiv.org/abs/2404.14197.

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 *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xOzlW2vUYc.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Romain Ilbert, Ambroise Odonnat, Vasilii Feofanov, Aladin Virmaux, Giuseppe Paolo, Themis Palpanas, and Ievgen Redko. SAMformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=8kLzL5QBh2.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

Yuxin Jia, Youfang Lin, Jing Yu, Shuo Wang, Tianhao Liu, and Huaiyu Wan. Pgn: The rnn's new successor is effective for long-range time series forecasting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

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.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018.

Chenghan Li, Mingchen Li, and Ruisheng Diao. TVNet: A novel time series analysis method based on dynamic convolution and 3d-variation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MZDdTzN6Cy.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv preprint arXiv:2308.11200*, 2023. URL https://arxiv.org/abs/2308.11200.

Shengsheng Lin, Weiwei Lin, Xinyi HU, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=clBiQUgj4w.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. In *Proceedings of the Forty-first International Conference on Machine Learning (ICML)*, 2024b.

Fuqiang Liu, Sicong Jiang, Luis Miranda-Moreno, Seongjin Choi, and Lijun Sun. Adversarial vulnerabilities in large language models for time series forecasting. In *Neurips Safe Generative AI Workshop 2024*, 2024a. URL https://openreview.net/forum?id=6Jbzz0L9z0.

Linbo Liu, Youngsuk Park, Trong Nghia Hoang, Hilaf Hasson, and Luke Huan. Robust multivariate time-series forecasting: Adversarial attacks and defense mechanisms. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ctmLBs8lITa.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=AyajSjTAzmg.

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

Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

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

PeiSong Niu, Tian Zhou, Xue Wang, Liang Sun, and Rong Jin. Attention as robust representation for time series forecasting. *arXiv preprint arXiv:2402.05370*, 2024.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4454–4458, 2022.

William Toner and Luke Nicholas Darlow. An analysis of linear time series forecasting models. In *Proceedings of the Forty-first International Conference on Machine Learning (ICML)*, 2024. URL https://openreview.net/forum?id=xl82CcbYaT.

José F. Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: A survey. *Big Data*, pp. 3–21, 2021. doi: 10.1089/big.2020.0159. URL http://dx.doi.org/10.1089/big.2020.0159.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021a. URL https://proceedings.mlr.press/v139/touvron21a.html.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021b.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 32–42, 2021c.

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

Xingyu Wang, Hui Liu, Junzhao Du, Zhihan Yang, and Xiyao Dong. Clformer: Locally grouped auto-correlation and convolutional transformer for long-term multivariate time series forecasting. *Engineering Applications of Artificial Intelligence*, 121:106042, 2023.

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

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

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *arXiv preprint arXiv:2403.11144*, 2024d.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022. URL https://arxiv.org/abs/2202.01381.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf.

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.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with 10k parameters. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bWcnvZ3qMb.

Yingnan Yang, Qingling Zhu, and Jianyong Chen. Vcformer: Variable correlation transformer with inherent lagged correlation for multivariate time series forecasting. *arXiv preprint arXiv:2405.11470*, 2024.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=bGs1qWQ1Fx.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=iif9mGCTfy.

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filternet: Harnessing frequency filters for time series forecasting. *arXiv preprint arXiv:2411.01623*, 2024.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive Tokens for Efficient Vision Transformer . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10799–10808. IEEE Computer Society, June 2022. doi: 10.1109/CVPR52688.2022.01054. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01054.

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 *Proceedings of the Forty-first International Conference on Machine Learning (ICML)*, 2024. URL https://openreview.net/forum?id=87CYNyCGOo.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8980–8987, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317/26089.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning, 2021. URL https://openreview.net/forum?id=lE1AB4stmX.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, pp. 40770–40803. PMLR, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:11106–11115, 2022. doi: 10.1609/aaai.v35i12.17325. URL http://dx.doi.org/10.1609/aaai.v35i12.17325.

Xin Zhou, Weiqing Wang, Wray Buntine, Shilin Qu, Abishek Sriramulu, Weicong Tan, and Christoph Bergmeir. Scalable transformer for high dimensional multivariate time series forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3515–3526, 2024.

# A EXPERIMENTAL DETAILS

## A.1 DATASETS

We elaborate on the datasets employed in this study with the following details.

1. **ETT (Electricity Transformer Temperature)** Zhou et al. (2022) [2] comprises two hourly-level datasets (ETTh) and two 15-minute-level datasets (ETTm). Each dataset contains seven oil and load features of electricity transformers from July 2016 to July 2018.
2. **Traffic** (Wu et al., 2021) [3] describes the road occupancy rates. It contains the hourly data recorded by the sensors of San Francisco freeways from 2015 to 2016.
3. **Electricity** (Wu et al., 2021) [4] collects the hourly electricity consumption of 321 clients from 2012 to 2014.
4. **Weather** (Wu et al., 2021) [5] includes 21 indicators of weather, such as air temperature and humidity. Its data is recorded every 10 min for 2020 in Germany.
5. **Solar_Energy** Lai et al. (2018) [6] records the solar power production of 137 PV plants in 2006, which is sampled every 10 minutes.
6. **PEMS** (Liu et al., 2022) [7] contains public traffic network data in California collected in 5-minute windows.

We follow the same data processing and train-validation-test set split protocol used in iTransformer (Liu et al., 2024b), where the train, validation, and test datasets are strictly divided according to chronological order to make sure no data leakage issues. We fix the length of the lookback series as $T = 96$ for all datasets, and the prediction length $F \in \{12, 24, 48, 96\}$ for PEMS(07) datasets, and $F \in \{96, 192, 336, 720\}$ for others. Other details of these datasets are concluded in Table 6.

Table 6: Detailed dataset descriptions. *Channels* denotes the number of channels in each dataset. *Dataset Split* denotes the total number of time points in (Train, Validation, Test) split respectively. *Prediction Length* denotes the future time points to be predicted, and four prediction settings are included in each dataset. *Granularity* denotes the sampling interval of time points.

| Dataset | Channels | Prediction Length | Dataset Split | Granularity | Domain |
|---|---|---|---|---|---|
| ETTh1, ETTh2 | 7 | {96, 192, 336, 720} | (8545, 2881, 2881) | Hourly | Electricity |
| ETTm1, ETTm2 | 7 | {96, 192, 336, 720} | (34465, 11521, 11521) | 15min | Electricity |
| Weather | 21 | {96, 192, 336, 720} | (36792, 5271, 10540) | 10min | Weather |
| ECL | 321 | {96, 192, 336, 720} | (18317, 2633, 5261) | Hourly | Electricity |
| Traffic | 862 | {96, 192, 336, 720} | (12185, 1757, 3509) | Hourly | Transportation |
| Solar-Energy | 137 | {96, 192, 336, 720} | (36601, 5161, 10417) | 10min | Energy |
| PEMS07 | 883 | {12, 24, 48, 96} | (16911, 5622, 5622) | 5min | Transportation |

[2] https://github.com/zhouhaoyi/ETDataset
[3] http://pems.dot.ca.gov
[4] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
[5] https://www.bgc-jena.mpg.de/wetter/
[6] https://github.com/thuml/iTransformer
[7] https://pems.dot.ca.gov/

### A.2 Implementation Details

We trained our **ResNo** model using the MSE loss function and employed the ADAM optimizer. For evaluation purposes, we used two key performance metrics: the mean square error (MSE) and the mean absolute error (MAE). We initialized the random seed as $rs = 2025$. The dimension of the Layer is set to $D = 512$. The batch size $B = 32$ for the Traffic&Solar&PEMS&Weather due to their large scale will cause **Out of Memory** when employed with large batch size, and $bs = 128$ for others. The learning rate is searched from $lr \in \{0.001, 0.0005, 0.0003, 0.0001, 0.0005, 0.00001\}$. The number of *Encoder Block* is searched from $S \in \{2, 3, 4, 5, 6\}$ except $S = 10$ for the Traffic dataset due to its large scale. Our implementation was carried out in PyTorch and executed on a single NVIDIA GeForce RTX 4090 with 24G VRAM. To foster reproducibility, we make our code and training scripts available in this **Anonymous Repo**[8].

### A.3 Baselines

We choose **ELEVEN** well-acknowledged and state-of-the-art models for comparison to evaluate the effectiveness of our proposed **ResNo** for multivariate time series forecasting. All the compared multivariate forecasting baseline models that we reproduced are implemented based on the benchmark of **Time series Lab** (Wang et al., 2024c) Repository [9], which is fairly built on the configurations provided by each model's original paper or official code. Those that have not yet been included in **Time series Lab** are directly reproduced from their official code repositories. Their information is summarized as follows:

#### A.3.1 Transformer-based methods

**iTransformer** (Liu et al., 2024b) inverts the structure of Transformer without modifying any existing modules by encoding individual series into variate tokens. These tokens are utilized by the attention mechanism to capture multivariate correlations, and FFNs are adopted for each variate token to learn nonlinear representations. The official implementation is available at https://github.com/thuml/iTransformer.

**Crossformer** (Zhang & Yan, 2023) introduces a cross-modal transformer that combines both visual and textual representations for multimodal learning tasks. By leveraging cross-modal attention mechanisms, it effectively improves the interaction between different modalities. The official implementation is available at https://github.com/Thinklab-SJTU/Crossformer.

**PatchTST** (Nie et al., 2023) divides time series data into subseries-level patches to extract local semantic information and adopts a channel-independence strategy where each channel shares the same embedding and Transformer weights across all the series. The official implementation is available at https://github.com/yuqinie98/PatchTST.

#### A.3.2 Linear-based or MLP-based methods

**DLinear** (Zeng et al., 2023) utilizes a simple yet effective one-layer linear model to capture temporal relationships between input and output sequences. The official implementation is available at https://github.com/cure-lab/LTSF-Linear.

**FITS** (Xu et al., 2024) performs time series analysis through interpolation in the complex frequency domain, enjoying low cost with 10k parameters. The official implementation is available at https://github.com/VEWOXIC/FITS.

**TSMixer** (Chen et al., 2023) introduces a novel time-series modeling architecture that combines transformer and MLP-based approaches to efficiently capture temporal dependencies across various time-series tasks. The method demonstrates superior performance by using a simple yet effective architecture for modeling time-series data. The official implementation is available at https://github.com/google-research/google-research/tree/master/tsmixer.

[8] https://anonymous.4open.science/r/ResNo-laoda24/

[9] https://github.com/thuml/Time-Series-Library

**FreTS** (Yi et al., 2023b) introduces a novel direction of applying MLPs in the frequency domain to effectively capture the underlying patterns of time series, benefiting from global view and energy compaction. The official implementation: https://github.com/aikunyi/FreTS.

#### A.3.3 CNN-BASED METHODS

**TimesNet** (Wu et al., 2023) transforms 1D time series into a set of 2D tensors based on multiple periods to analyse temporal variations. The above transformation allows the 2D-variations to be easily captured by 2D kernels by encoding the intraperiod- and interperiod-variations into the columns and rows of the 2D tensors, respectively. The official implementation is available at https://github.com/thuml/TimesNet.

**ModernTCN** (Donghao & Xue, 2024) proposes a novel approach for temporal convolutional networks (TCNs), focusing on improving their performance in modeling long-range dependencies. The method introduces enhancements that allow it to handle large-scale sequential data more efficiently. The official implementation: https://github.com/luodhhh/ModernTCN.

#### A.3.4 FILTER-BASED METHODS

**FilterNet** (Yi et al., 2024) introduces a novel deep learning framework for time series forecasting, which employs a filtering mechanism to improve model robustness and prediction accuracy. The method enhances traditional models by integrating an adaptive filter to capture complex temporal patterns. The official implementation: https://github.com/aikunyi/FilterNet.

#### A.3.5 GNN-BASED METHODS

**CrossGNN** (Huang et al., 2023) presents a graph neural network model designed for cross-domain graph learning, enhancing the ability to transfer knowledge across different graph structures. The method integrates cross-domain attention to align graph representations effectively for improved performance. The official implementation: https://github.com/hqh0728/CrossGNN.

# B VISUALIZATIONS

## B.1 VISUALIZATION OF SERIES EMBEDDING

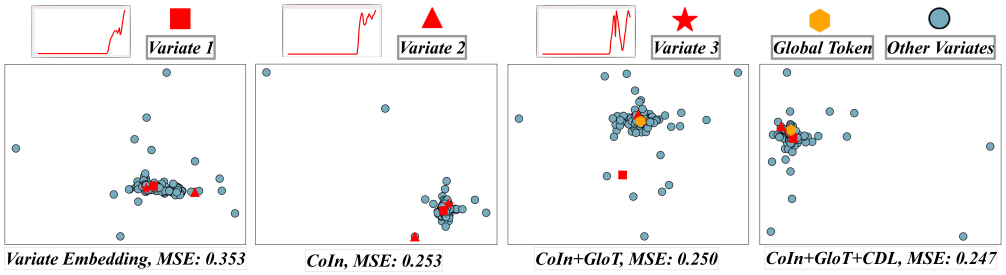


Figure 7: T-SNE visualization of the impact of CoIn and CDL on series embeddings. We choose the $input-96-forecast-720$ task on **Solar_Energy** dataset. Three example variates are highlighted: variates 1&2&3 exhibited a similar inter-channel correlation.

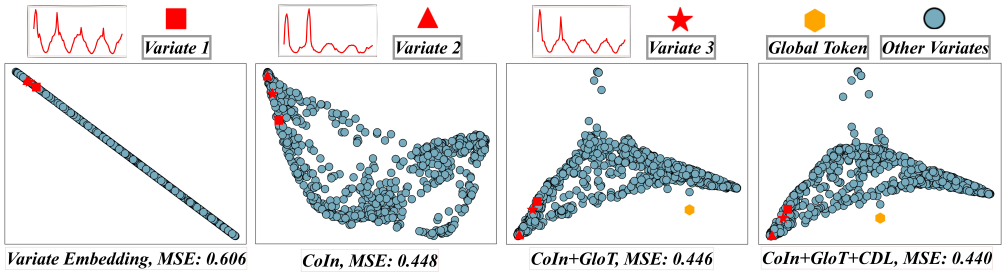


Figure 8: T-SNE visualization of the impact of CoIn and CDL on series embeddings. We choose the $input-96-forecast-720$ task on **Traffic** dataset. Three example variates are highlighted: variates 1&3 exhibited a stronger correlation, whereas variate 2, though similar, was less closely associated with them.

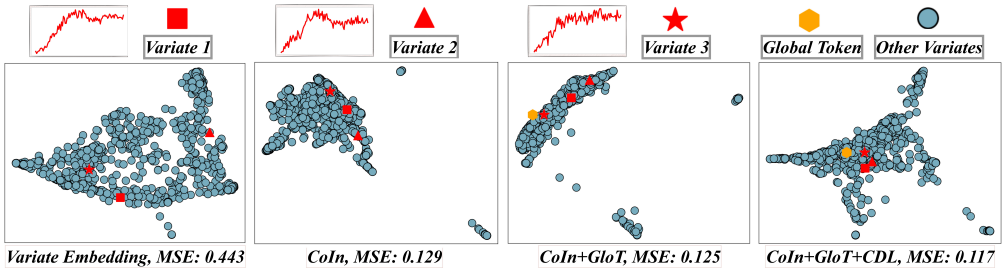


Figure 9: T-SNE visualization of the impact of CoIn and CDL on series embeddings. We choose the $input-96-forecast-96$ task on **PEMS(07)** dataset. Three example variates are highlighted: variates 1&2&3 exhibited an indistinguishable inter-channel correlation.

We additionally provide t-SNE visualizations of series embeddings on the **Solar_Energy**, **Traffic**, and **PEMS** datasets, beyond the main results reported on the ECL dataset.

It is evident that simple Variate Embedding, due to its channel-independent nature, fails to capture inter-series dependencies effectively. Although **CoIn** partially captures these correlations, it remains insufficient in modeling more intricate interdependencies. As shown in Figure 9, for example, variates 1, 2, and 3 in the PEMS dataset are embedded by CoIn into relatively close positions in the latent space. However, these variates are expected to exhibit even stronger mutual correlations, suggesting that they should be placed even closer together. The introduction of **CDL** enables the model to extract and represent such complex relationships more accurately. Furthermore, the proposed **Global Token** accurately captures and encodes global information across all variates, further enhancing the model's ability to represent inter-series interactions holistically.

### B.2 VISUALIZATION OF MODEL WEIGHT

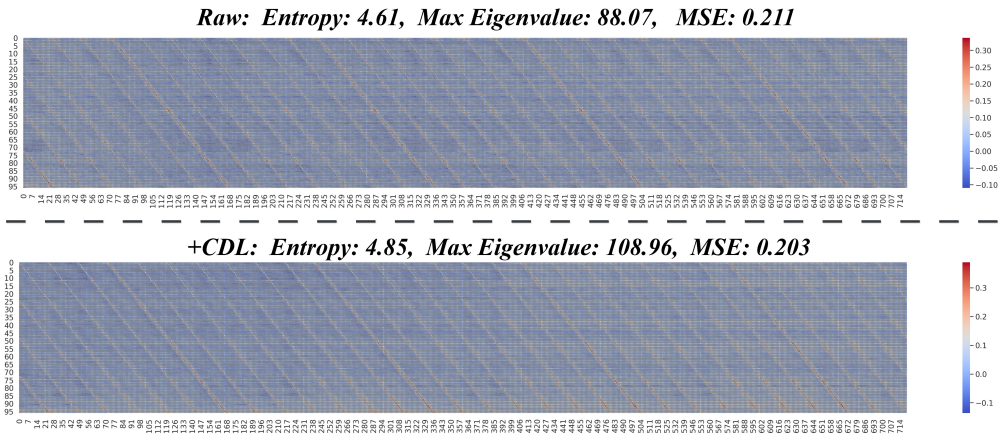


Figure 10: Impact of 'Channel Decoupling Loss' (CDL) on model weight. We choose the $input-96-forecast-720$ task on **ECL** dataset.

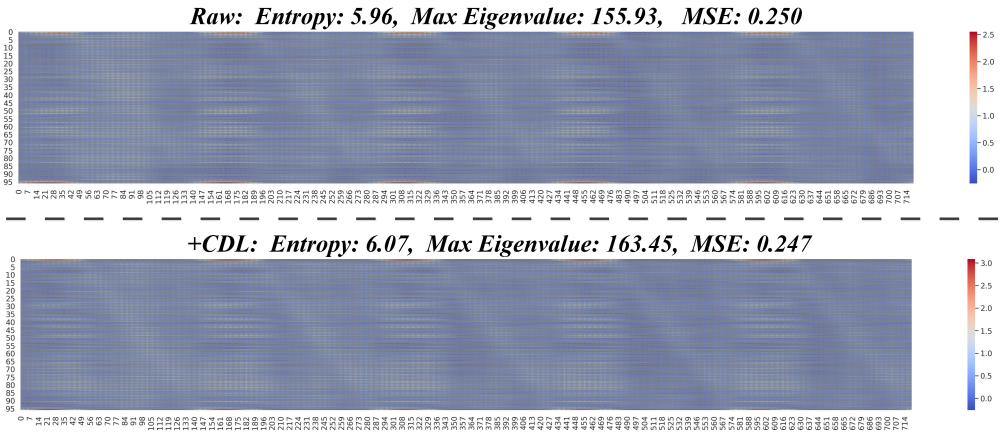


Figure 11: Impact of 'Channel Decoupling Loss' (CDL) on model weight. We choose the $input-96-forecast-720$ task on **Solar_Energy** dataset.

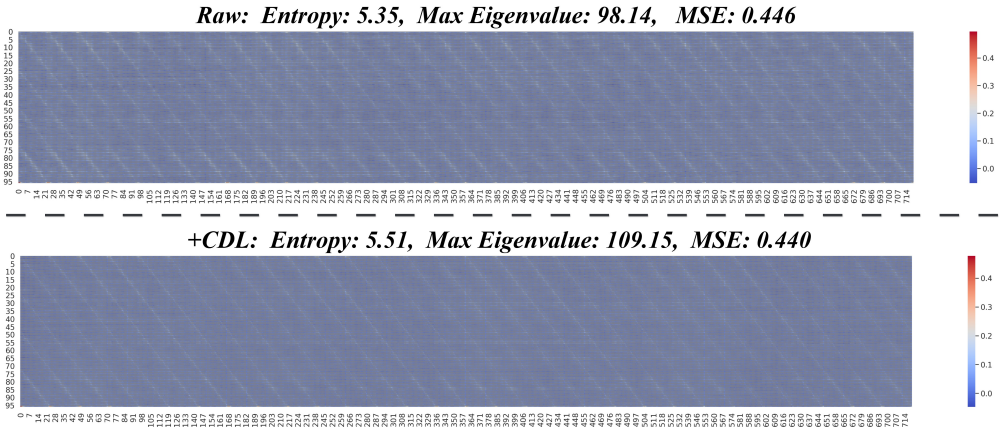


Figure 12: Impact of 'Channel Decoupling Loss' (CDL) on model weight. We choose the $input-96-forecast-720$ task on **Traffic** dataset.

In addition to the PEMS dataset used in the main text, we also provide visualizations of the learned weights on three additional representative datasets: **ECL**, **Solar_Energy**, and **Traffic**.

The method used for getting the weight follows the approach outlined in **Analysis of linear model** (Toner & Darlow, 2024). Plotting the learned matrices as in Figure 10- 12 requires us to first convert each trained model into the form $f(\vec{x}) = A\vec{x} + \vec{b}$. To do this we note that $f(\vec{0}) = A\vec{0} + \vec{b} = \vec{b}$. Thus, the bias can be found by passing the zero vector into the trained model. We can determine $A$ similarly. Let $\vec{e_i}$ denote the $i$-th coordinate vector, that is $\vec{e_i}$ is the vector which is 1 at position $i$ and zero elsewhere. Then $f(\vec{e_i}) = A\vec{e_i} + \vec{b} = A_{\cdot,i} + \vec{b}$ where $A_{\cdot,i}$ is the $i$-th column of $A$. Hence, given that we have already computed the bias term, we may derive $A$ simply by passing through each coordinate vector $\vec{e_i}$ and subtracting $\vec{b}$.

We also calculate the *Max Eigenvalue* and the column-wise *Entropy* of the weight matrix to indicate and quantify the change. The introduction of CDL leads to more structured and interpretable model weights across the aforementioned datasets. Moreover, both the *Entropy* and the *Max Eigenvalue* increase, suggesting an enhancement in the model's representational capacity.

### B.3 Visualization of Multivariate Correlations

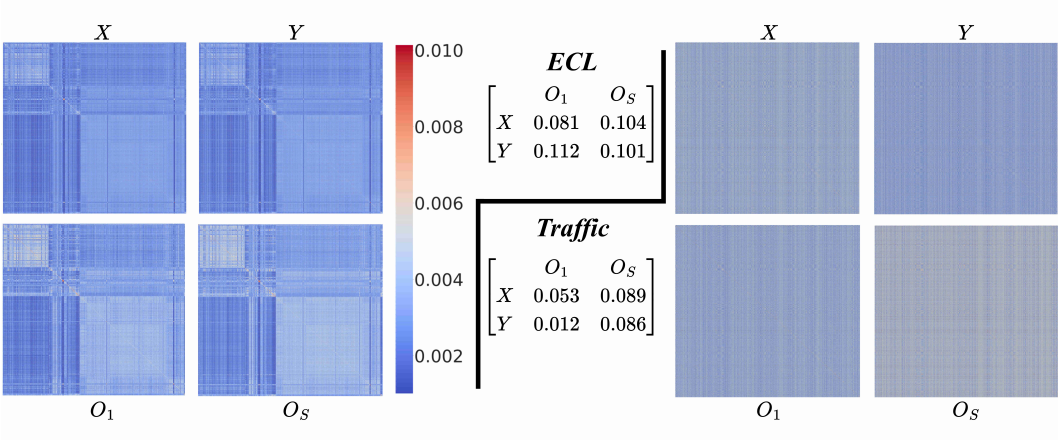


Figure 13: Multivariate correlations of raw time series and the learned embedding by CoIn. $X$ is the raw input, $Y$ is the target. $O_1$ is the output embedding of the first encoder block, and $O_S$ is the last encoder block. We use the Frobenius norm $\|A-B\|_F = \sqrt{\sum_{i,j}(A_{ij}-B_{ij})^2}$ to quantify the similarity. A lower value indicates a higher similarity.

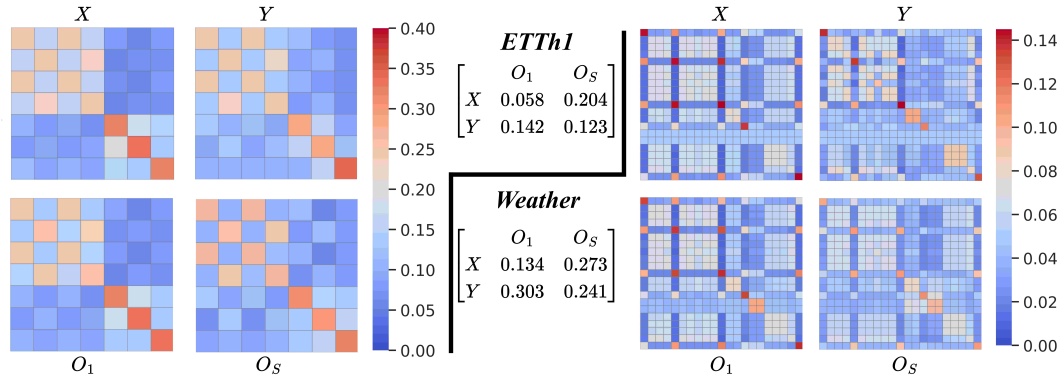


Figure 14: Multivariate correlations of raw time series and the learned embedding by CoIn. $X$ is the raw input, $Y$ is the target. $O_1$ is the output embedding of the first encoder block, and $O_S$ is the last encoder block. We use the Frobenius norm $\|A-B\|_F = \sqrt{\sum_{i,j}(A_{ij}-B_{ij})^2}$ to quantify the similarity. A lower value indicates a higher similarity.

We provide visualizations of multivariate correlations on four datasets, as shown in Figure 13 and Figure 14. We calculate the value of the correlation map by:

$$\bar{x}_{i,:} = \frac{1}{T}\sum_{t=1}^{T} x_{i,t}, \quad \tilde{x}_{i,t} = x_{i,t} - \bar{x}_{i,:}, \quad \tilde{\rho}_{ij} = \frac{\sum_{t=1}^{T} \tilde{x}_{i,t}\,\tilde{x}_{j,t}}{\sqrt{\sum_{t=1}^{T}\tilde{x}_{i,t}^2}\sqrt{\sum_{t=1}^{T}\tilde{x}_{j,t}^2} + \epsilon}, \quad \epsilon = 1\times 10^{-8},$$
$$\mathrm{cor}_{ij} = \frac{\exp\left(\tilde{\rho}_{ij}\right)}{\sum_{k=1}^{N}\exp\left(\tilde{\rho}_{ik}\right)}. \tag{B.1}$$

These results demonstrate that CoIn effectively captures inter-series dependencies. Moreover, the encoder layer built based on CoIn exhibits highly interpretable feature maps. Notably, in the shallow layers, the correlation maps of the learned representation closely resemble the input sequence $X$, while in deeper layers, the representations progressively align with the target outputs $Y$. This suggests that CoIn progressively transforms input features toward task-relevant representations, enabling both effective modeling and interpretability.

# C Additional Results

## C.1 Error Bar Test

Table 7: Robustness of **ResNo** on multivariate time series forecasting. The results are obtained from five random seeds $rs \in \{2024, 2025, 2026, 2027, 2028\}$. Prediction lengths $F \in \{12, 24, 48, 96\}$ for PEMS and $F \in \{96, 192, 336, 720\}$ for others).

| Dataset | ECL | | ETTh1 | | ETTh2 | |
|---|---|---|---|---|---|---|
| Horizon | MSE | MAE | MSE | MAE | MSE | MAE |
| 96 | 0.144±0.0001 | 0.236±0.0016 | 0.376±0.0005 | 0.396±0.0005 | 0.293±0.0004 | 0.343±0.0004 |
| 192 | 0.157±0.0002 | 0.252±0.0002 | 0.434±0.0016 | 0.429±0.0018 | 0.374±0.0005 | 0.392±0.0004 |
| 336 | 0.171±0.0004 | 0.267±0.0004 | 0.469±0.0031 | 0.445±0.0021 | 0.382±0.0005 | 0.409±0.0003 |
| 720 | 0.202±0.0002 | 0.295±0.0003 | 0.474±0.0077 | 0.470±0.0045 | 0.413±0.0006 | 0.434±0.0004 |
| Dataset | PEMS | | Traffic | | Weather | |
| Horizon | MSE | MAE | MSE | MAE | MSE | MAE |
| 96/12 | 0.057±0.0001 | 0.150±0.0002 | 0.381±0.0011 | 0.249±0.0014 | 0.169±0.0016 | 0.208±0.0018 |
| 192/24 | 0.072±0.0001 | 0.169±0.0003 | 0.400±0.0003 | 0.259±0.0002 | 0.217±0.0006 | 0.253±0.0003 |
| 336/48 | 0.094±0.0004 | 0.194±0.0006 | 0.413±0.0004 | 0.266±0.0002 | 0.273±0.0011 | 0.294±0.0005 |
| 720/96 | 0.117±0.0002 | 0.215±0.0004 | 0.440±0.0005 | 0.282±0.0004 | 0.352±0.0012 | 0.346±0.0005 |

We report the standard deviation of **ResNo** performance across five runs with different random seeds in Table 7, indicating that ResNo's performance is stable.

## C.2 Efficiency Analysis

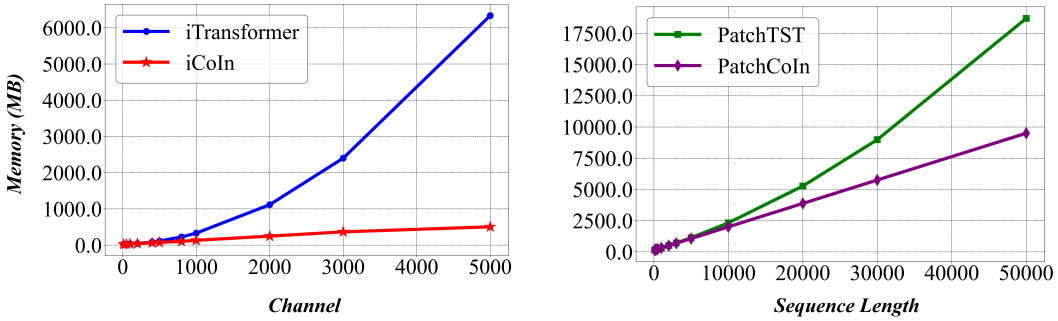


Figure 15: Memory Usage comparison of CoIn and attention. We select Channel-Dependent iTransformer and Channel-Independent PatchTST as examples. Specifically, we compare the memory consumption of CoIn and attention under varying numbers of channels (with $B = 4$, $T = 96$, $F = 720$) and input sequence lengths (with $B = 4$, $N = 1$, $F = 720$).

We compare the memory consumption of CoIn and attention under varying numbers of channels (iTransformer&iCoIn) and input sequence lengths (PatchTST&PatchCoIn). The results in Figure 15 further confirm the high efficiency of the proposed CoIn.

# AI Use Statement

Large language models, including Claude Code, were used only for AI-assisted drafting, editing, and literature checks. They did not influence experimental design, execution, analysis, or interpretation. All scientific decisions, analyses, and results were produced and fully verified solely by the authors.dw