MDN: A MULTI-SCALE DISENTANGLEMENT NETWORK FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Although multi-scale modeling methods have achieved success in time series forecasting, they often overlook cross-scale entanglement and directly model cross-scale interactions among entangled features, allowing entanglement to interfere with interaction modeling. Extracting disentangled multi-scale features before modeling interactions can mitigate such interference. However, extracting such features faces two challenges: (1) Non-linear cross-scale entanglement is difficult to disentangle effectively. (2) Scale-shared information confounds the disentanglement process. To this end, we propose a **M**ulti-scale **D**isentanglement **N**etwork (MDN) for time series forecasting. Specifically, MDN first employs a Product-of-Experts-based separation module to infer and remove a robust scale-shared feature, thereby reducing interference from scale-shared information. Then, a mutual information (MI)-based disentanglement constraint is designed to minimize pairwise MI upper bounds among entangled features, thereby mitigating non-linear cross-scale entanglement. Finally, MDN sequentially models the intra-scale and inter-scale interactions among the disentangled features. Extensive experiments on 7 real-world datasets demonstrate that MDN achieves state-of-the-art performance against 14 competitive baselines. Code is available at: `https://anonymous.4open.science/r/MDN-6BF2`.

1 INTRODUCTION

Time series are ubiquitous in the modern world, capturing the dynamics of diverse real-world systems. Time series forecasting, the task of predicting future values from historical data, plays a critical role in various real-world applications, e.g., traffic flow forecasting (Liu et al., 2023), stock price forecasting (Fan & Shen, 2024), and weather forecasting (Wu et al., 2023).

Time series are fundamentally characterized by a multi-scale structure, exhibiting various temporal patterns at different scales (Chen et al., 2023; Wang et al., 2024a). For example, driven by periodic human activities, the traffic flow exhibits clear multi-scale temporal patterns, from daily patterns (e.g., morning and evening rush hours) to weekly patterns (e.g., high-traffic weekdays and low-traffic weekends). Such multi-scale temporal patterns exhibit two important characteristics. First, multi-scale temporal patterns within the same time series often overlap (Chen et al., 2024; Wu et al., 2022), leading to cross-scale entanglement (Hu et al., 2025; Liu et al., 2025; Gao et al., 2025), which manifests as statistical dependence (Kim & Mnih, 2018). For example, a sudden power outage is reflected in both hourly and daily electricity consumption, introducing redundant information across scales. From a statistical perspective, such entanglement manifests as statistical dependence among multi-scale representations. Second, multi-scale temporal patterns exhibit intricate cross-scale interactions (e.g., causality) (Shang et al., 2024; Chen et al., 2025). For example, in the temperature series, the large-scale decadal warming shifts the underlying climate distribution, thereby contributing to the increased frequency and persistence of small-scale extreme heat events (Horton et al., 2016). Therefore, accurate time series forecasting depends on meticulous multi-scale modeling that can properly characterize temporal patterns across different scales.

Motivated by this, a growing number of multi-scale modeling methods have recently emerged in the time series forecasting community (Wang et al., 2023; Zhang et al., 2024). The majority of methods follow a four-step paradigm: multi-scale feature extraction, feature interaction modeling, cross-scale information aggregation, and final forecasting (Shang et al., 2026; Huang et al., 2025).

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.

Nevertheless, most existing methods overlook cross-scale entanglement and directly model the interactions between inherently entangled multi-scale temporal features. As a result, redundant and mixed information arising from cross-scale entanglement may be mistakenly modeled as meaningful interactions, causing the model to capture spurious cross-scale interactions. Recent studies have shown that learning disentangled features can decompose complex representations into independent factors, thereby reducing the interference induced by entanglement (Zhan et al., 2026; Liu et al., 2025; Ma et al., 2019). Inspired by this, a natural approach to mitigating cross-scale entanglement is to extract disentangled multi-scale temporal features before modeling their cross-scale interactions.

Despite this promise, extracting disentangled multi-scale temporal features for time series forecasting faces two major challenges: (1) **Non-Linear Cross-Scale Entanglement** The entanglement of multi-scale temporal features tends to be non-linear, which stems from the inherent complex dynamics in time series. For example, in a power outage, the daily drop in electricity consumption is not a simple linear aggregation of hourly ones, as it is modulated by non-linear human compensatory behavior. Existing methods (Liu et al., 2025; Woo et al., 2022; Xia et al., 2026) typically impose constraints based on linear correlation metrics (e.g., cosine similarity) to achieve feature disentanglement, which inherently fail to address the non-linear entanglement of multi-scale temporal features. (2) **Scale-Shared Information Interference** The temporal feature at each scale contains three components, including scale-shared information (e.g., the overall magnitude level of the time series), scale-specific information (e.g., large-scale long-term trends or small-scale short-term fluctuations), and entanglement-induced noise that should be suppressed. Since scale-shared information manifests as cross-scale dependencies, directly disentangling original multi-scale temporal features would mistakenly treat scale-shared information as entanglement-induced noise, thereby making it difficult to suppress entanglement-induced noise and clearly separate scale-specific information.

Motivated by these observations, we propose MDN, a **M**ulti-scale **D**isentanglement **N**etwork for time series forecasting. To the best of our knowledge, MDN is the first work that employs a mutual information-based disentanglement constraint to extract disentangled multi-scale temporal features for time series forecasting. The main contributions are summarized as follows:

- We propose a mutual information (MI)-based scale-specific feature disentanglement (SSFD) module. SSFD employs MI as an information-theoretic measure of entanglement beyond linear correlation and introduces a learnable estimator to approximate the upper bound of MI between each pair of scale-specific temporal features. The estimated MI upper bound is then minimized during training, which can reduce non-linear statistical dependence among scale-specific temporal features, thereby mitigating non-linear cross-scale entanglement and facilitating the extraction of disentangled multi-scale temporal features.
- We design a Product-of-Experts (PoE)-based scale-shared feature separation (SSFS) module. SSFS first treats each scale as an expert to generate a scale-shared feature candidate and corresponding uncertainty, and then fuses these candidates according to their uncertainties to obtain a robust scale-shared temporal feature. This feature is subsequently separated from each scale, which can enable subsequent disentanglement to focus on scale-specific components, thereby mitigating scale-shared information interference.
- We compare MDN with 14 competitive baselines on 7 real-world datasets. Extensive experimental results demonstrate that MDN achieves state-of-the-art (SOTA) performance, highlighting its effectiveness in modeling interactions between disentangled multi-scale temporal features.

2 Related Work

Disentangled Representation Learning for Time Series Forecasting. Disentangled representation learning (DRL) aims to obtain representations that can identify and disentangle the underlying factors hidden in the observed data. Owing to its advantages in improving the model explainability, robustness, and generalization capacity, DRL has been widely applied in various fields (e.g., recommendation systems (Wang et al., 2022) and deepfake detection (Li et al., 2025)). Recently, DRL has also attracted increasing attention from the time series forecasting community. Existing methods mainly leverage disentangled representation learning to separate intertwined temporal factors (e.g., seasonal and trend representations, time-invariant and time-varying components). For example, CoST (Woo et al., 2022) disentangles the time series into seasonal and trend representations

using time domain and frequency domain contrastive objectives, respectively. TimeEmb (Xia et al., 2026) disentangles the time series into a time-invariant component captured by a learnable embedding bank and a time-varying component modeled via frequency filtering. Although these methods effectively disentangle intertwined temporal factors, they mainly perform disentanglement at the original temporal scale and overlook the multi-scale structure of time series. Moving beyond single-scale disentanglement, DisMS-TS (Liu et al., 2025) introduces a disentangled multi-scale framework that imposes cosine-similarity-based regularization to reduce redundant information among temporal representations across scales. However, cosine-similarity-based regularization can only capture linear correlations and fails to address inherent non-linear cross-scale entanglement in time series, resulting in incomplete disentanglement. In addition, DisMS-TS separately extracts scale-shared representations at each scale and employs a similarity constraint to align them, which results in inconsistencies among these scale-shared representations and thus fails to effectively mitigate scale-shared information interference.

In contrast, we employ a mutual information-based disentanglement constraint to eliminate statistical dependence among multi-scale temporal features, thereby mitigating the inherent non-linear cross-scale entanglement. In addition, we utilize a Product-of-Experts mechanism to infer a robust scale-shared temporal feature, which is then removed from each scale to counteract scale-shared information interference. Related work on multi-scale time series forecasting appears in Appendix A.

3 Methodology

3.1 Problem Formulation

We first provide the time series forecasting definition. Given the input time series $\mathbf{X}_{1:T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\} \in \mathbb{R}^{T \times N}$, where T is the size of look-back window, N is the number of variates, and $\boldsymbol{x}_t \in \mathbb{R}^N$ represents the observed values at time step t, the task of time series forecasting is to predict the future values $\widehat{\mathbf{Y}}_{T+1:T+H} \in \mathbb{R}^{H \times N}$, where H is the prediction length. Formally, the time series forecasting task can be formulated as:

$$\widehat{\mathbf{Y}}_{T+1:T+H} = f(\mathbf{X}_{1:T}, \theta), \tag{1}$$

where f represents the deep learning model of time series forecasting, and θ denotes all learnable parameters of f.

3.2 Overview

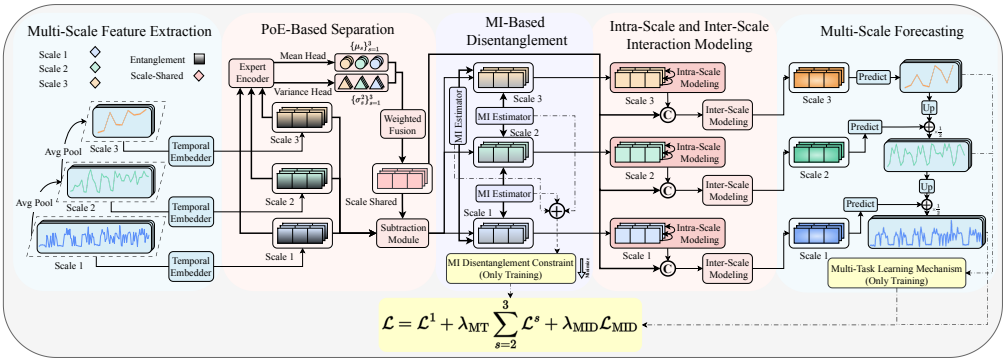


Figure 1: The framework of MDN. For clarity, MDN is illustrated with 3 scales.

Fig. 1 illustrates the overall framework of MDN. The core of MDN is to extract disentangled multi-scale temporal features for time series forecasting by separating a scale-shared feature and minimizing mutual information among scale-specific features. In doing so, we first extract preliminary multi-scale temporal features from the input time series and then infer a robust scale-shared temporal feature via a Product-of-Experts (PoE) mechanism. This feature is then subtracted from the temporal feature at each scale to obtain residual scale-specific temporal features, thereby mitigating

scale-shared information interference for subsequent disentanglement. Since these scale-specific temporal features are still entangled, we further impose a mutual information (MI)-based disentanglement constraint among these features to extract disentangled multi-scale temporal features. After that, we sequentially model the intra-scale and inter-scale interactions among these disentangled multi-scale temporal features to enhance their expressiveness. Finally, we generate forecasts at different scales and progressively fuse them from large-scale to small-scale. In addition, we employ a multi-task learning mechanism to supervise these forecasts, which improves the generalization and robustness of MDN.

3.3 Multi-Scale Feature Extraction (MSFE) Module

Time series are inherently characterized by multi-scale structures, where observations at different scales often exhibit distinct temporal patterns. To characterize such multi-scale structures, we introduce the MSFE module to construct multi-scale subsequences from the input time series and extract preliminary multi-scale temporal features from them. Specifically, given the input time series $\mathbf{X}_{1:T}$, we regard it as the subsequence at the smallest scale and iteratively apply average pooling along the temporal dimension to construct multi-scale subsequences. Each subsequence is then embedded along the temporal dimension to extract preliminary multi-scale temporal features. This process can be formulated as follows:

$$\begin{aligned} \mathbf{X}^1 &= \mathbf{X}_{1:T}, \\ \mathbf{X}^s &= \mathrm{AvgPool}\left(\mathbf{X}^{s-1}; w_s\right), \quad s = 2, \ldots, S, \\ \mathbf{H}^s &= \mathrm{Embed}_s\left(\mathbf{X}^s\right), \quad s = 1, \ldots, S, \end{aligned} \tag{2}$$

where S is the number of scales, w_s is the average pooling window size for constructing the subsequence at the s-th scale, $\mathbf{X}^s \in \mathbb{R}^{T_s \times N}$ denotes the subsequence at the s-th scale, T_s is the length of the subsequence $\mathbf{X}^s$, with $T_s = \left\lfloor \frac{T_{s-1}}{w_s} \right\rfloor$ for $s = 2, \ldots, S$ and $T_1 = T$, and $\mathbf{H}^s \in \mathbb{R}^{N \times D}$ denotes the extracted preliminary temporal feature at the s-th scale. However, multi-scale temporal patterns within the same time series often overlap, causing these preliminary multi-scale temporal features to be entangled and thus requiring subsequent disentanglement.

3.4 PoE-Based Scale-Shared Feature Separation (SSFS) Module

After extracting preliminary multi-scale temporal features, a straightforward approach to obtaining disentangled multi-scale features is to impose a disentanglement constraint on these preliminary features. However, we argue that this direct approach would suffer from scale-shared information interference during the disentanglement process. Directly imposing a disentanglement constraint on these features would not only increase the difficulty of reducing entanglement-induced noise but also mistakenly suppress scale-shared temporal patterns that are important for subsequent cross-scale interaction modeling. Therefore, we propose the SSFS module to extract a robust scale-shared temporal feature from all scales via a PoE mechanism and separate it from the preliminary temporal feature at each scale to mitigate scale-shared information interference. Specifically, a shared expert encoder first produces the expert feature $\mathbf{E}^s = \phi(\mathbf{H}^s)$ for each scale. Then, the shared mean and variance heads estimate $\boldsymbol{\mu}_s = \mathrm{Linear}_\mu(\mathbf{E}^s)$ and $\boldsymbol{\sigma}_s^2 = \mathrm{Softplus}(\mathrm{Linear}_\sigma(\mathbf{E}^s)) + \epsilon$, respectively, where Softplus ensures positive variance estimates and ϵ provides numerical stability. Accordingly, the s-th expert defines a Gaussian distribution parameterized with mean $\boldsymbol{\mu}_s \in \mathbb{R}^{N \times D}$ and variance $\boldsymbol{\sigma}_s^2 \in \mathbb{R}^{N \times D}$. These Gaussian experts are then aggregated through inverse-variance weighted fusion to infer a robust scale-shared temporal feature, which is subtracted from the preliminary temporal feature at each scale. This process can be formulated as follows:

$$\mathbf{P} = \frac{\sum_{s=1}^{S} \boldsymbol{\sigma}_s^{-2} \odot \boldsymbol{\mu}_s}{\sum_{s=1}^{S} \boldsymbol{\sigma}_s^{-2}}, \qquad \mathbf{H}_{\mathrm{res}}^s = \mathbf{H}^s - \mathbf{P}, \quad s = 1, \ldots, S, \tag{3}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{P} \in \mathbb{R}^{N \times D}$ is the inferred scale-shared temporal feature, and $\mathbf{H}_{\mathrm{res}}^s \in \mathbb{R}^{N \times D}$ denotes the residual scale-specific temporal feature at the s-th scale.

By explicitly separating scale-shared information before disentanglement, SSFS mitigates scale-shared information interference, allowing the disentanglement constraint to focus on reducing entanglement-induced noise. In addition, SSFS prevents scale-shared temporal patterns from being

mistakenly suppressed and preserves them as important global context for subsequent cross-scale interaction modeling.

3.5 MI-Based Scale-Specific Feature Disentanglement (SSFD) Module

Although the SSFS module separates the scale-shared temporal feature from preliminary multi-scale temporal features, the obtained scale-specific temporal features are still entangled. Notably, such entanglement tends to be non-linear due to the inherent complex dynamics in time series. Conventional disentanglement constraints based on linear correlation metrics (e.g., cosine similarity) mainly capture linear correlation but fail to identify non-linear statistical dependence, thereby remaining insufficient to address non-linear cross-scale entanglement. To this end, in the SSFD module, we employ mutual information (MI) as a measure of the entanglement among scale-specific temporal features and design an MI-based disentanglement constraint that minimizes the upper bound of MI between each pair of these features. However, directly computing MI between high-dimensional temporal features is intractable. Therefore, we adopt Contrastive Log-ratio Upper Bound (CLUB) (Cheng et al., 2020) as the estimator to approximate the MI upper bound between scale-specific temporal features. Specifically, CLUB estimates an MI upper bound by contrasting the conditional log-likelihood of positive pairs sampled from the joint distribution with that of negative pairs sampled from the product of marginal distributions. Since the true conditional distribution is unavailable, CLUB introduces a variational distribution to approximate it and yields the empirical upper-bound estimator formulated as follows:

$$\widehat{I}^{\mathrm{CLUB}}_{(i,j)} = \frac{1}{M}\sum_{m=1}^{M} \log q_{\psi_{(i,j)}}\left(\mathbf{u}^j_m \mid \mathbf{u}^i_m\right) - \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M} \log q_{\psi_{(i,j)}}\left(\mathbf{u}^j_n \mid \mathbf{u}^i_m\right), \tag{4}$$

where $\widehat{I}^{\mathrm{CLUB}}_{(i,j)}$ denotes the MI upper bound estimated by CLUB between the scale-specific temporal features $\mathbf{H}^i_{\mathrm{res}}$ and $\mathbf{H}^j_{\mathrm{res}}$, (i,j) denotes a scale-index pair with $1 \le i < j \le S$, $q_{\psi_{(i,j)}}$ denotes the variational distribution parameterized by $\psi_{(i,j)}$ for the scale-index pair (i,j), $\mathbf{u}^s_m \in \mathbb{R}^D$ denotes the m-th sample drawn from $\mathbf{H}^s_{\mathrm{res}}$, and M is the number of samples. Notably, $(\mathbf{u}^i_m, \mathbf{u}^j_m)$ forms a positive pair with the same sample index, while $(\mathbf{u}^i_m, \mathbf{u}^j_n)$ forms a negative pair when $m \neq n$. In practice, $q_{\psi_{(i,j)}}$ is parameterized by two lightweight multilayer perceptrons (MLPs) that predict the mean and variance of a Gaussian conditional distribution, respectively.

Based on the CLUB estimator, we design an MI-based disentanglement constraint by summing the estimated MI upper bounds over non-duplicated scale-index pairs according to the symmetry of MI, which can be formulated as follows:

$$\mathcal{L}_{\mathrm{MID}} = \sum_{1 \le i < j \le S} \widehat{I}^{\mathrm{CLUB}}_{(i,j)}. \tag{5}$$

This constraint is later incorporated into the overall training loss as a regularization term. By minimizing $\mathcal{L}_{\mathrm{MID}}$, SSFD reduces the non-linear statistical dependence among scale-specific temporal features, thereby mitigating non-linear cross-scale entanglement and facilitating the extraction of disentangled multi-scale temporal features for subsequent interaction modeling. More details about the training strategy of the constraint can be found in Appendix B.

3.6 Intra-Scale and Inter-Scale Interaction Modeling

Capturing temporal dynamics and inter-variate correlations is essential for accurate time series forecasting. To this end, we propose the IM module to enhance disentangled multi-scale temporal features by sequentially modeling intra-scale interactions and inter-scale interactions, which can capture multivariate correlations and enhance temporal dynamics, respectively. Specifically, for each scale, we first apply an attention mechanism along the variate dimension to model intra-scale interactions and capture multivariate correlations. Then, we use an MLP to fuse the scale-shared temporal feature, which serves as global context, with the scale-specific temporal feature at each scale, thereby modeling inter-scale interactions and enhancing temporal dynamics. This process can be formulated as follows:

$$\begin{aligned} \widetilde{\mathbf{H}}^s &= \mathrm{LN}\left(\mathbf{H}^s_{\mathrm{res}} + \mathrm{Attn}_s\left(\mathbf{H}^s_{\mathrm{res}}\right)\right), \quad s = 1, \ldots, S, \\ \mathbf{H}^s_{\mathrm{IM}} &= \mathrm{LN}\left(\widetilde{\mathbf{H}}^s + \mathrm{MLP}_s\left(\left[\widetilde{\mathbf{H}}^s; \mathbf{P}\right]\right)\right), \quad s = 1, \ldots, S, \end{aligned} \tag{6}$$

where $\mathrm{LN}(\cdot)$ denotes layer normalization, $\mathrm{Attn}_s(\cdot)$ denotes the attention mechanism along the variate dimension for intra-scale interaction modeling at the s-th scale, $[\cdot;\cdot]$ denotes feature concatenation along the feature dimension, $\mathrm{MLP}_s(\cdot)$ denotes the fusion network for inter-scale interaction modeling at the s-th scale, and $\mathbf{H}^s_{\mathrm{IM}} \in \mathbb{R}^{N\times D}$ denotes the enhanced temporal feature at the s-th scale.

By sequentially modeling intra-scale and inter-scale interactions, IM enhances the expressiveness of disentangled multi-scale temporal features for subsequent forecasting.

3.7 Multi-Scale Forecasting and Loss Function

After obtaining enhanced multi-scale temporal features $\{\mathbf{H}^s_{\mathrm{IM}}\}_{s=1}^{S}$, we first generate an initial forecast at each scale using a scale-wise linear predictor. Since larger-scale forecasts contain more macroscopic temporal information (e.g., trend), we recursively fuse forecasts from large-scale to small-scale, so that larger-scale forecasts can provide useful guidance for smaller-scale forecasts. This process can be formulated as follows:

$$\begin{aligned} \widehat{\mathbf{Y}}^s &= \mathrm{Linear}_s\left(\mathbf{H}^s_{\mathrm{IM}}\right), \quad s = 1,\dots,S, \\ \widehat{\mathbf{Y}}^s_{\mathrm{fuse}} &= \frac{1}{2}\left(\widehat{\mathbf{Y}}^s + \mathrm{Up}\left(\widehat{\mathbf{Y}}^{s+1}_{\mathrm{fuse}}\right)\right), \quad s = S-1,\dots,1, \end{aligned} \tag{7}$$

where $\widehat{\mathbf{Y}}^S_{\mathrm{fuse}} = \widehat{\mathbf{Y}}^S$ is the initial case of the recursive forecast fusion, $\mathrm{Linear}_s(\cdot)$ denotes the scale-wise linear predictor at the s-th scale, $\widehat{\mathbf{Y}}^s, \widehat{\mathbf{Y}}^s_{\mathrm{fuse}} \in \mathbb{R}^{H_s\times N}$ denote the initial forecast and fused forecast at the s-th scale, respectively. H_s is the forecast length at the s-th scale, with $H_1 = H$ and $H_s = \left\lfloor \frac{H_{s-1}}{w_s} \right\rfloor$ for $s = 2,\dots,S$, where w_s is the corresponding average pooling window size. $\mathrm{Up}(\cdot)$ denotes the upsampling operation that aligns the forecast length from H_{s+1} to H_s. The final forecast is obtained at the smallest scale, i.e., $\widehat{\mathbf{Y}}_{T+1:T+H} = \widehat{\mathbf{Y}}^1_{\mathrm{fuse}}$.

Time series forecasting models are prone to overfitting due to insufficient training data. To improve the generalization of MDN, we employ a multi-task learning mechanism by supervising forecasts at all scales, where scale-wise supervision signals are constructed by average downsampling the ground truth future sequence $\mathbf{Y}_{T+1:T+H}$, which can be formulated as follows:

$$\mathbf{Y}^s = \mathrm{AvgDown}\left(\mathbf{Y}_{T+1:T+H}, H_s\right), \quad s = 2,\dots,S, \tag{8}$$

where $\mathbf{Y}^s \in \mathbb{R}^{H_s\times N}$ denotes the supervision signal at the s-th scale, with $\mathbf{Y}^1 = \mathbf{Y}_{T+1:T+H}$. Conventional methods usually adopt only a time-domain loss. Since the time-domain loss overlooks the autocorrelation within the future sequence (Wang et al., 2025), we further introduce a frequency-domain loss to align the spectra of forecasts and ground truths, thereby mitigating the influence of autocorrelation. Specifically, we employ the Mean Squared Error (MSE) in the time domain and the Mean Absolute Error (MAE) in the frequency domain. The loss at the s-th scale can be formulated as follows:

$$\mathcal{L}^s = (1-\alpha)\ \mathrm{MAE}\Big(\mathcal{F}\Big(\widehat{\mathbf{Y}}^s_{\mathrm{fuse}}\Big), \mathcal{F}(\mathbf{Y}^s)\Big) + \alpha\ \mathrm{MSE}\Big(\widehat{\mathbf{Y}}^s_{\mathrm{fuse}}, \mathbf{Y}^s\Big), \quad s = 1,\dots,S, \tag{9}$$

where $\mathcal{F}(\cdot)$ denotes the fast Fourier transform (FFT) and $\alpha \in [0,1]$ balances the time-domain and frequency-domain losses. Finally, the overall training loss is formulated as:

$$\mathcal{L} = \mathcal{L}^1 + \lambda_{\mathrm{MT}} \sum_{s=2}^{S} \mathcal{L}^s + \lambda_{\mathrm{MID}} \mathcal{L}_{\mathrm{MID}}, \tag{10}$$

where $\mathcal{L}^1$ is the forecasting loss at the original scale, λ_{MT} controls the contribution of auxiliary losses at larger scales, and λ_{MID} controls the contribution of the MI-based disentanglement loss.

4 Experiments

4.1 Experimental Setup

Datasets. Following previous works (Xia et al., 2026; Wang et al., 2024b), we conduct experiments on 7 real-world datasets, including 4 ETT datasets (ETTh1, ETTh2, ETTm1, and ETTm2), Traffic,

Electricity, and Weather datasets.

Baselines. We compare MDN with 14 SOTA baselines across 2 representative categories: (1) **Multi-scale models**: MICN (Wang et al., 2023), TimeKAN (Huang et al., 2025), MSGNet (Cai et al., 2024), TimeMixer (Wang et al., 2024b), TimesNet (Wu et al., 2022), and AMD (Hu et al., 2025); (2) **Single-scale models**: TimePro (Ma et al., 2025), Time-TK (Zhang et al., 2026), DUET (Qiu et al., 2025), iTransformer (Liu et al., 2024), PatchTST (Nie et al., 2022), DLinear (Zeng et al., 2023), Crossformer (Zhang & Yan, 2023), and CMoS (Si et al., 2025).

Experimental Settings. All experiments are implemented in PyTorch and conducted on a single NVIDIA GeForce RTX 5090 GPU. We use MSE and MAE as evaluation metrics, where lower values indicate better performance. We use Adam as the optimizer and adopt early stopping to avoid overfitting. To ensure a fair comparison, we evaluate all baselines with the look-back window length $T = 96$ and prediction length $H \in \{96, 192, 336, 720\}$ across all datasets. The hyperparameters are tuned using the Neural Network Intelligence (NNI) toolkit based on validation performance. More details about datasets, baselines, and experimental settings are provided in Appendices C.1, C.3, and C.4, respectively.

4.2 Main Results

Table 1: Full results of long-term time series forecasting. The best results are **bolded** and the second best results are underlined. For a fair comparison, baselines marked with * are reproduced under our experimental settings using their official implementations, while the results of the remaining baselines are taken from published studies. More details are provided in Appendix C.4.3.

Methods		MDN Ours		TimeKAN* ICLR 2025		AMD* AAAI 2025		TimeMixer ICLR 2024		MSGNet AAAI 2024		MICN ICLR 2023		TimesNet ICLR 2023		Time-TK WWW 2026		TimePro ICML 2025		CMoS ICML 2025		DUET KDD 2025		iTransformer ICLR 2024		PatchTST ICLR 2023		Crossformer ICLR 2023		DLinear AAAI 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	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.365**	**0.390**	0.370	0.395	0.383	0.394	0.381	0.401	0.390	0.411	0.421	0.431	0.384	0.402	0.370	0.393	0.375	0.398	0.396	0.403	0.377	0.393	0.394	0.409	0.414	0.419	0.423	0.448	0.386	0.400
	192	**0.413**	**0.420**	0.414	**0.420**	0.436	0.423	0.440	0.433	0.443	0.442	0.474	0.487	0.436	0.429	0.423	0.421	0.427	0.429	0.432	0.428	0.429	0.425	0.448	0.441	0.460	0.445	0.471	0.474	0.437	0.432
	336	**0.450**	**0.440**	0.454	0.441	0.477	0.444	0.501	0.462	0.482	0.469	0.569	0.551	0.491	0.469	0.465	0.444	0.472	0.450	0.481	0.454	0.471	0.446	0.491	0.464	0.501	0.466	0.570	0.546	0.481	0.459
	720	**0.455**	0.459	0.459	**0.458**	0.479	0.464	0.501	0.482	0.496	0.488	0.770	0.672	0.521	0.500	0.470	0.462	0.476	0.474	0.482	0.482	0.496	0.480	0.519	0.502	0.500	0.488	0.653	0.621	0.519	0.516
ETTh2	96	**0.278**	**0.330**	0.291	0.341	0.285	0.337	0.292	0.343	0.329	0.371	0.299	0.364	0.340	0.374	0.293	0.340	0.293	0.345	0.309	0.351	0.296	0.345	0.297	0.349	0.292	0.345	0.745	0.584	0.333	0.387
	192	**0.352**	**0.377**	0.375	0.392	0.387	0.394	0.378	0.397	0.402	0.414	0.441	0.454	0.402	0.414	0.368	0.388	0.367	0.394	0.396	0.404	0.368	0.389	0.380	0.400	0.388	0.405	0.877	0.656	0.477	0.476
	336	**0.394**	**0.410**	0.423	0.435	0.416	0.427	0.432	0.434	0.440	0.445	0.654	0.567	0.452	0.452	0.410	0.423	0.419	0.431	0.431	0.437	0.411	0.422	0.428	0.432	0.427	0.436	1.043	0.731	0.594	0.541
	720	**0.394**	**0.424**	0.461	0.459	0.423	0.441	0.454	0.458	0.480	0.477	0.956	0.716	0.462	0.468	0.418	0.438	0.427	0.445	0.431	0.446	0.412	0.434	0.427	0.445	0.447	0.458	1.104	0.763	0.831	0.657
ETTm1	96	**0.306**	**0.344**	0.322	0.361	0.322	0.359	0.328	0.363	0.319	0.366	0.316	0.362	0.338	0.375	0.315	0.354	0.326	0.364	0.354	0.381	0.324	0.354	0.336	0.370	0.329	0.365	0.404	0.426	0.345	0.372
	192	**0.354**	**0.372**	0.358	0.378	0.367	0.381	0.364	0.384	0.377	0.397	0.363	0.390	0.374	0.387	0.356	0.378	0.367	0.383	0.390	0.396	0.369	0.379	0.381	0.395	0.380	0.394	0.450	0.451	0.380	0.389
	336	**0.384**	**0.394**	0.387	0.403	0.400	0.402	0.390	0.404	0.417	0.422	0.408	0.426	0.410	0.411	0.393	0.402	0.402	0.409	0.423	0.418	0.404	0.402	0.417	0.418	0.400	0.410	0.532	0.515	0.413	0.413
	720	**0.440**	**0.431**	0.445	0.436	0.465	0.437	0.458	0.445	0.487	0.463	0.481	0.476	0.478	0.450	0.453	0.439	0.469	0.446	0.481	0.445	0.463	0.437	0.487	0.456	0.475	0.453	0.666	0.589	0.474	0.453
ETTm2	96	**0.169**	**0.247**	0.174	0.257	0.180	0.265	0.178	0.259	0.182	0.266	0.179	0.275	0.187	0.267	0.173	0.253	0.178	0.260	0.186	0.270	0.174	0.255	0.185	0.271	0.193	0.280	0.287	0.366	0.193	0.292
	192	**0.234**	**0.290**	0.239	0.300	0.244	0.304	0.242	0.303	0.248	0.306	0.307	0.376	0.249	0.309	0.238	0.298	0.242	0.303	0.248	0.307	0.243	0.302	0.251	0.312	0.246	0.307	0.414	0.492	0.284	0.362
	336	**0.293**	**0.329**	0.301	0.340	0.302	0.340	0.304	0.342	0.312	0.346	0.325	0.388	0.321	0.351	0.298	0.339	0.303	0.342	0.308	0.344	0.304	0.341	0.314	0.350	0.314	0.351	0.597	0.542	0.369	0.427
	720	**0.388**	**0.387**	0.397	0.398	0.403	0.397	0.395	0.397	0.414	0.404	0.502	0.490	0.408	0.403	0.395	0.395	0.400	0.399	0.409	0.400	0.399	0.397	0.411	0.405	0.410	0.405	1.730	1.042	0.554	0.522
Weather	96	0.161	**0.200**	0.162	0.208	0.181	0.227	0.165	0.212	0.163	0.212	0.161	0.229	0.172	0.220	0.173	0.213	0.166	0.207	0.170	0.217	0.163	0.202	0.174	0.214	0.177	0.218	**0.158**	0.230	0.196	0.255
	192	0.213	**0.247**	0.208	0.249	0.227	0.264	0.209	0.253	0.211	0.254	0.220	0.281	0.219	0.261	0.222	0.257	0.216	0.254	0.216	0.257	0.218	0.252	0.221	0.254	0.225	0.259	**0.206**	0.277	0.237	0.296
	336	0.270	**0.290**	**0.263**	**0.290**	0.281	0.302	0.264	0.293	0.273	0.299	0.278	0.331	0.280	0.306	0.277	0.296	0.273	0.296	0.270	0.294	0.274	0.294	0.278	0.296	0.278	0.297	0.272	0.335	0.283	0.335
	720	0.346	**0.340**	0.342	0.342	0.356	0.349	0.342	0.345	0.351	0.348	**0.311**	0.356	0.365	0.359	0.351	0.346	0.351	0.346	0.348	0.344	0.349	0.343	0.358	0.347	0.354	0.348	0.398	0.418	0.345	0.381
Traffic	96	**0.378**	**0.252**	0.578	0.379	0.509	0.333	0.462	0.285	0.608	0.349	0.519	0.309	0.593	0.321	0.392	0.263	0.407	0.269	0.576	0.342	0.407	**0.252**	0.395	0.268	0.544	0.359	0.522	0.290	0.650	0.396
	192	**0.408**	0.264	0.551	0.368	0.503	0.328	0.473	0.296	0.634	0.371	0.537	0.315	0.617	0.336	0.415	0.273	0.424	0.276	0.596	0.361	0.431	**0.262**	0.417	0.276	0.540	0.354	0.530	0.293	0.598	0.370
	336	**0.416**	**0.269**	0.563	0.363	0.515	0.331	0.498	0.296	0.669	0.388	0.534	0.313	0.629	0.336	0.430	0.279	0.443	0.286	0.630	0.371	0.456	**0.269**	0.433	0.283	0.551	0.358	0.558	0.305	0.605	0.373
	720	**0.449**	**0.287**	0.610	0.382	0.545	0.347	0.506	0.313	0.729	0.420	0.577	0.325	0.640	0.350	0.463	0.297	0.483	0.311	0.667	0.390	0.509	0.292	0.467	0.302	0.586	0.375	0.589	0.328	0.645	0.394
Electricity	96	**0.136**	**0.227**	0.175	0.267	0.166	0.258	0.153	0.244	0.165	0.274	0.164	0.269	0.168	0.272	0.147	0.240	0.139	0.234	0.179	0.262	0.145	0.233	0.148	0.240	0.188	0.280	0.219	0.314	0.197	0.282
	192	**0.152**	**0.242**	0.183	0.274	0.178	0.266	0.166	0.256	0.185	0.292	0.177	0.285	0.184	0.289	0.162	0.253	0.156	0.249	0.186	0.269	0.163	0.248	0.162	0.253	0.193	0.285	0.231	0.322	0.196	0.285
	336	**0.167**	**0.259**	0.197	0.287	0.194	0.282	0.184	0.275	0.197	0.304	0.193	0.304	0.198	0.300	0.177	0.269	0.172	0.267	0.202	0.285	0.175	0.262	0.178	0.269	0.211	0.302	0.246	0.337	0.209	0.301
	720	**0.195**	**0.286**	0.237	0.320	0.233	0.314	0.226	0.313	0.231	0.332	0.212	0.321	0.220	0.320	0.209	0.298	0.209	0.299	0.247	0.321	0.204	0.291	0.225	0.317	0.253	0.335	0.280	0.363	0.245	0.333

Table 1 reports the results of long-term time series forecasting. We can observe that: (1) MDN achieves SOTA performance in the vast majority of forecasting cases. Specifically, MDN ranks first in 50 cases and second in 3 cases across the 56 forecasting cases evaluated. We attribute this superior performance to MDN's coordinated framework for multi-scale disentanglement, in which the PoE-based separation mechanism first isolates scale-shared information, enabling the MI-based disentanglement constraint to effectively mitigate non-linear cross-scale entanglement. (2) Although Time-TK is the strongest single-scale baseline through multi-offset interactions, it lacks explicit multi-scale modeling. MDN instead captures disentangled multi-scale dynamics, reducing MSE and MAE by 3.33% and 2.75% on average, respectively. (3) TimeKAN is the strongest multi-scale baseline by modeling multi-frequency dynamics through cascaded decomposition and multi-order KANs. However, it does not explicitly reduce dependence among scale representations, whereas MDN applies an MI-based disentanglement constraint, reducing MSE and MAE by 8.27% and 7.51% on average, respectively. (4) Electricity exhibits strong cross-scale entanglement due to recurring consumption patterns across resolutions. Nevertheless, MDN reduces MSE and MAE by 2.98% and 1.93% on average relative to the best results among all baselines, with the MSE gain increasing from 2.16% at $H = 96$ to 4.41% at $H = 720$, indicating greater benefits for long-term forecasting. More results and analysis are provided in Appendix D.2.

4.3 Ablation Study

Table 2: Ablation study results on ETTh2 under different prediction lengths. The best results are **bolded**.

Variant	$H = 96$		$H = 192$		$H = 336$		$H = 720$	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o MI	0.284	0.334	0.361	0.383	0.405	0.416	0.402	0.430
-v Cos	0.283	0.333	0.356	0.380	0.400	0.415	0.404	0.430
w/o Sep	0.283	0.331	0.356	0.380	0.399	0.412	0.404	0.433
-v Stoch	0.281	0.331	0.358	0.381	0.415	0.420	0.404	0.430
w/o Intra	0.282	**0.330**	0.356	0.378	0.405	0.415	0.418	0.438
w/o Inter	0.281	0.332	0.358	0.382	0.399	0.415	0.409	0.432
w/o Fusion	0.286	0.333	0.363	0.383	0.404	0.416	0.402	0.427
w/o MTL	0.291	0.340	0.363	0.385	0.403	0.415	0.411	0.435
MDN	**0.278**	**0.330**	**0.352**	**0.377**	**0.394**	**0.410**	**0.394**	**0.424**

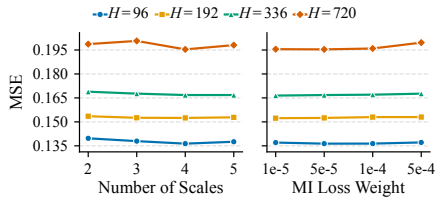


Figure 2: Hyperparameter sensitivity of the number of scales and MI loss weight on the Electricity dataset.

We conduct a comprehensive ablation study to validate the effectiveness of key designs of MDN from 3 perspectives. The experimental results on the ETTh2 dataset are reported in Table 2. **Multi-scale disentanglement.** We assess the contributions of the 2 complementary disentanglement modules (i.e., the SSFS and SSFD modules) by designing the following 4 variants: (1) Removing scale-shared feature separation (**w/o Sep**). (2) Replacing the PoE mechanism for scale-shared feature extraction with the stochastic pooling strategy used in SOFTS (Han et al., 2024) (**-v Stoch**). (3) Removing the MI-based disentanglement constraint (**w/o MI**). (4) Replacing the MI-based disentanglement constraint with the cosine-similarity-based one used in DisMS-TS (Liu et al., 2025) (**-v Cos**). We can observe that: (1) MDN consistently outperforms all 4 variants, validating the effectiveness of scale-shared feature extraction and separation as well as the MI-based disentanglement constraint. (2) On average, **-v Cos** outperforms **w/o MI** but still underperforms MDN, indicating the benefit of explicitly imposing a disentanglement constraint and the superiority of the MI-based constraint in mitigating cross-scale entanglement by characterizing non-linear statistical dependence. **Multi-scale interaction modeling.** We evaluate the contributions of intra-scale and inter-scale interaction modeling by designing the following 2 variants: (1) Removing intra-scale interaction modeling (**w/o Intra**). (2) Removing inter-scale interaction modeling (**w/o Inter**). We can observe that: (1) MDN outperforms both variants, demonstrating the importance of intra-scale and inter-scale interaction modeling for capturing multivariate correlations and multi-scale temporal dynamics, respectively. (2) **w/o Inter** achieves a lower average MSE than **w/o Intra**. This performance gap can be attributed to the strong correlations among variables in the ETTh2 dataset, which make intra-scale interaction modeling particularly important. **Multi-scale forecasting.** We investigate the effectiveness of recursive forecast fusion and multi-task learning mechanism by designing the following 2 variants: (1) Removing recursive forecast fusion (**w/o Fusion**). (2) Removing multi-task learning (**w/o MTL**). We can observe that: (1) MDN outperforms both variants, validating the effectiveness of recursive forecast fusion and the multi-task learning mechanism in leveraging macroscopic temporal information and improving generalization, respectively. (2) At $H = 720$, **w/o MTL** performs substantially worse than **w/o Fusion**, suggesting that improving generalization is particularly important for long-term forecasting. More results about ablation studies are provided in Appendix D.4.

4.4 Hyperparameter Sensitivity

We evaluate the sensitivity of MDN to the number of scales (S) and the MI loss weight (λ_{MID}) on the Electricity dataset. We vary one hyperparameter at a time while keeping all others fixed. The MSE results are shown in Fig. 2. We can observe that: (1) The number of scales (S) affects the diversity of temporal information encoded in multi-scale representations, with an optimum at $S = 4$. A smaller S may provide insufficient multi-scale temporal information, whereas a larger S may introduce redundant patterns, thereby aggravating cross-scale entanglement. (2) The MI loss weight controls the strength of the disentanglement constraint imposed on multi-scale temporal features, with an optimum at $\lambda_{\text{MID}} = 5 \times 10^{-5}$. A smaller λ_{MID} may be insufficient to mitigate cross-scale entanglement, whereas a larger λ_{MID} may suppress important dependencies among scale-specific features, thereby impairing subsequent cross-scale interaction modeling. (3) The relatively

small performance variations indicate that MDN is robust to both hyperparameters within reasonable ranges.

4.5 Visualization of Disentangled Multi-Scale Features

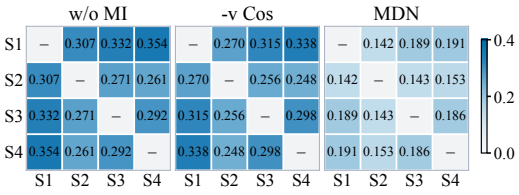


Figure 3: Pairwise PCC heatmaps on ETTm2 with $H = 336$. Lower values indicate weaker linear dependence.

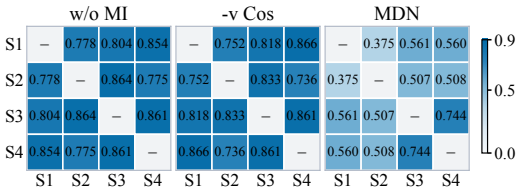


Figure 4: Pairwise HSIC heatmaps on ETTm2 with $H = 336$. Lower values indicate weaker non-linear dependence.

Table 3: Average dependence values on ETTm2 with $H = 336$. Lower values are better. The best results are **bolded**.

Metric	w/o MI	-v Cos	MDN
Pearson correlation coefficient	0.303	0.288	**0.167**
Hilbert-Schmidt independence criterion	0.823	0.811	**0.543**

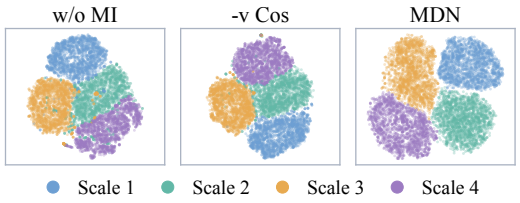


Figure 5: t-SNE projections of multi-scale features on ETTm2 with $H = 336$.

To examine whether MDN effectively mitigates cross-scale entanglement, we compare the multi-scale features extracted by **w/o MI**, **-v Cos**, and MDN on ETTm2 with $H = 336$. First, we measure linear and non-linear statistical dependence using the Pearson correlation coefficient (PCC) and Hilbert-Schmidt independence criterion (HSIC), respectively. Both dependence metrics are computed using the same samples from the test set. Table 3 reports their average values across scale pairs. We can observe that both PCC and HSIC decrease consistently from **w/o MI** to **-v Cos** and then to MDN. Specifically, MDN reduces PCC by 44.88% and HSIC by 34.02% relative to **w/o MI**. Figs. 3 and 4 show the pairwise PCC and HSIC heatmaps across scales, respectively. We can observe that MDN exhibits consistently lower off-diagonal values, indicating weaker linear and non-linear dependence among disentangled features. In addition, the higher HSIC values of **-v Cos** than **w/o MI** for some scale pairs indicate that reducing linear correlation does not necessarily mitigate non-linear dependence. Second, Fig. 5 shows the t-SNE projections of multi-scale features. We can observe that **w/o MI** and **-v Cos** exhibit substantial cross-scale mixing, whereas MDN shows little overlap and clear scale-wise boundaries. Overall, these results demonstrate that MDN can effectively extract disentangled multi-scale features through scale-shared feature separation and an MI-based disentanglement constraint, thereby mitigating non-linear cross-scale entanglement. More experiments are provided in Appendix D.

5 Conclusions

In this paper, we propose MDN, a multi-scale disentanglement network designed to mitigate cross-scale entanglement in time series forecasting. MDN first employs a PoE-based separation module to infer a robust scale-shared feature and separate it from the feature at each scale, thereby alleviating scale-shared information interference. It then imposes an MI-based disentanglement constraint on the resulting residual features to reduce non-linear statistical dependence among them, thus mitigating cross-scale entanglement and yielding disentangled multi-scale features for subsequent interaction modeling and forecasting. Experimental results on 7 real-world datasets demonstrate the effectiveness of MDN, while comprehensive studies validate its key designs.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript writing and revision, LaTeX formatting, and checking mathematical notation, tables, and cross-references. All AI-assisted content was carefully reviewed and verified by the authors. The authors take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT.

We provide the source code and implementation of MDN in an anonymous repository. Detailed descriptions of the datasets, experimental settings, baseline configurations, computing environment, hyperparameter search spaces, and final hyperparameter settings are included in the paper and appendix. We also report robustness results across multiple random seeds. These materials are intended to facilitate reproduction of the reported results.

ETHICS STATEMENT.

This work focuses on methodological research for multivariate time series forecasting using established benchmark datasets in electricity, weather, and traffic domains. It does not involve human subjects, user interaction, or personally identifiable information. We do not foresee direct ethical concerns arising from the proposed methodology. Predictions should nevertheless be carefully validated before deployment in safety-critical or high-stakes applications.

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*, pp. 11141–11149, 2024.

Ling Chen, Donghui Chen, Zongjiang Shang, Binqing Wu, Cen Zheng, Bo Wen, and Wei Zhang. Multi-scale adaptive graph neural network for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):10748–10761, 2023.

Ling Chen, Jiahua Cui, Zongjiang Shang, and Dongliang Cui. TPRNN: A top-down pyramidal recurrent neural network for time series forecasting. *Information Sciences*, 699:121792, 2025.

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. *arXiv preprint arXiv:2402.05956*, 2024.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. CLUB: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning*, pp. 1779–1788, 2020.

Jinyong Fan and Yanyan Shen. StockMixer: A simple yet strong MLP-based architecture for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8389–8397, 2024.

Boshi Gao, Qingjian Ni, Fanbo Ju, Yu Chen, and Ziqi Zhao. A multi-scale representation learning framework for long-term time series forecasting. *arXiv preprint arXiv:2505.08199*, 2025.

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

Radley M Horton, Justin S Mankin, Corey Lesk, Ethan Coffel, and Colin Raymond. A review of recent advances in research on extreme heat events. *Current Climate Change Reports*, 2(4): 242–259, 2016.

Yifan Hu, Peiyuan Liu, Peng Zhu, Dawei Cheng, and Tao Dai. Adaptive multi-scale decomposition framework for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17359–17367, 2025.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. TimeKAN: Kan-based frequency decomposition learning architecture for long-term time series forecasting. *arXiv preprint arXiv:2502.06910*, 2025.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658, 2018.

Kai Li, Wenqi Ren, Jianshu Li, Wei Wang, and Xiaochun Cao. Critical forgetting-based multi-scale disentanglement for deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 424–432, 2025.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. LargeST: A benchmark dataset for large-scale traffic forecasting. *Advances in Neural Information Processing Systems*, 36:75354–75371, 2023.

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 *International Conference on Learning Representations*, pp. 11116–11140, 2024.

Zhipeng Liu, Peibo Duan, Binwu Wang, Xuan Tang, Qi Chu, Changsheng Zhang, Yongsheng Huang, and Bin Zhang. DisMS-TS: Eliminating redundant multi-scale features for time series classification. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10817–10826, 2025.

Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. *Advances in Neural Information Processing Systems*, 32, 2019.

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 *International Conference on Machine Learning*, pp. 42096–42111, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

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 ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1185–1196, 2025.

Zongjiang Shang, Ling Chen, Binqing Wu, and Dongliang Cui. MSHyper: Multi-scale hypergraph transformer for long-range time series forecasting. *arXiv preprint arXiv:2401.09261*, 2024.

Zongjiang Shang, Dongliang Cui, Binqing Wu, and Ling Chen. Multi-scale hypergraph meets LLMs: Aligning large language models for time series analysis. *arXiv preprint arXiv:2602.04369*, 2026.

Haotian Si, Changhua Pei, Jianhui Li, Dan Pei, and Gaogang Xie. CMoS: Rethinking time series prediction through the lens of chunk-wise spatial correlations. In *International Conference on Machine Learning*, pp. 55493–55510, 2025.

Hao Wang, Lichen Pan, Yuan Shen, Zhichao Chen, Degui Yang, Yifei Yang, Sen Zhang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. FreDF: Learning to forecast in the frequency domain. In *International Conference on Learning Representations*, pp. 6893–6922, 2025.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *International Conference on Learning Representations*, 2023.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. *arXiv preprint arXiv:2410.16032*, 2024a.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024b.

Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. Disentangled representation learning for recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):408–424, 2022.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. *arXiv preprint arXiv:2202.01575*, 2022.

Binqing Wu, Weiqi Chen, Wengwei Wang, Bingqing Peng, Liang Sun, and Ling Chen. WeatherGNN: Exploiting meteo-and spatial-dependencies for local numerical weather prediction bias-correction. *arXiv preprint arXiv:2310.05517*, 2023.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Mingyuan Xia, Chunxu Zhang, Zijian Zhang, Hao Miao, Qidong Liu, Yuanshao Zhu, and Bo Yang. Timeemb: A lightweight static-dynamic disentanglement framework for time series forecasting. *Advances in Neural Information Processing Systems*, 38:76227–76258, 2026.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11121–11128, 2023.

Xiaohan Zhan, Yuliang Shi, Jihu Wang, Shijun Liu, Fanyu Kong, and Zhiyong Chen. DHMRec: Collaboration-guided multimodal disentanglement and hierarchical fusion for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 16244–16252, 2026.

Fan Zhang, Shiming Fan, and Hua Wang. Time-TK: A multi-offset temporal interaction framework combining transformer and kolmogorov-arnold networks for time series forecasting. In *Proceedings of the ACM Web Conference*, pp. 7495–7506, 2026.

Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Multi-resolution time-series transformer for long-term forecasting. In *International Conference on Artificial Intelligence and Statistics*, pp. 4222–4230, 2024.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

A Additional Related Work

Multi-Scale Modeling for Time Series Forecasting. Multi-scale modeling has been widely explored in the time series forecasting community due to its ability to capture temporal patterns at different scales. Most multi-scale modeling methods first extract multi-scale features and then integrate them through cross-scale interaction modeling for final forecasting. According to how multi-scale temporal features are extracted, existing multi-scale modeling methods can be divided into downsampling-based and patch-based methods. Downsampling-based methods progressively construct multi-scale subsequences using downsampling operators (e.g., average pooling and convolution), and then extract multi-scale temporal features from them. For example, TimeMixer (Wang et al., 2024b) employs average pooling to construct multi-scale subsequences and decomposes each subsequence into trend and seasonal components, finally mixing them to model cross-scale interactions. Patch-based methods segment time series into semantically richer patches of different lengths to extract multi-scale temporal features. For example, Pathformer (Chen et al., 2024) divides time series into patches of different lengths and introduces adaptive pathways to dynamically adjust the multi-scale modeling for forecasting. Despite their effectiveness in capturing multi-scale temporal patterns, existing multi-scale modeling methods typically overlook the inherent cross-scale entanglement and model cross-scale interactions among entangled multi-scale temporal features directly, which may capture spurious cross-scale interactions.

B Training Strategy for the MI-Based Constraint

We alternately optimize MDN and the CLUB estimators. For each scale pair (i, j) with $i < j$, we employ an independent CLUB estimator $q_{\psi_{ij}}(\mathbf{u}^j \mid \mathbf{u}^i)$, where the residual features of each mini-batch are reshaped into $\mathbf{u}^s \in \mathbb{R}^{M \times D}$ with $M = BN$. During the first E_{w} warm-up epochs, we set $\lambda_{\mathrm{MID}} = 0$ while training the CLUB estimators to approximate the conditional distributions. After warm-up, MDN is updated by minimizing the overall objective with the CLUB estimators fixed. The resulting residual features are then detached, and each CLUB estimator is updated by maximizing its conditional log-likelihood. We perform $K = 5$ CLUB updates after each MDN update. The complete procedure is summarized in Algorithm 1.

Algorithm 1 Alternating Training of MDN and CLUB Estimators

Require: Training data $\mathcal{D}$, MDN parameters θ, CLUB parameters $\{\psi_{ij}\}$, warm-up epochs E_{w}, and CLUB update frequency K

for each epoch e **do**
  Set $\lambda_{\mathrm{MID}}(e) = 0$ if $e < E_{\mathrm{w}}$; otherwise use the target or scheduled weight
  for each mini-batch $\mathcal{B}$ **do**
    Obtain predictions and residual features $\{\mathbf{u}^s\}_{s=1}^{S}$ using MDN
    $\mathcal{L}_{\mathrm{MID}} \leftarrow \sum_{1 \le i < j \le S} \widehat{I}_{\psi_{ij}}(\mathbf{u}^i; \mathbf{u}^j)$
    Update θ using $\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{MID}}(e)\mathcal{L}_{\mathrm{MID}}$, with $\{\psi_{ij}\}$ fixed
    for $k = 1, \ldots, K$ **do**
      Recompute and detach $\{\mathbf{u}^s\}_{s=1}^{S}$
      $\mathcal{L}_{\mathrm{CLUB}} \leftarrow -\sum_{i<j} \frac{1}{M} \sum_{m=1}^{M} \log q_{\psi_{ij}}(\mathbf{u}_m^j \mid \mathbf{u}_m^i)$
      Update $\{\psi_{ij}\}$ by minimizing $\mathcal{L}_{\mathrm{CLUB}}$, with θ fixed
    end for
  end for
end for

For Electricity and Traffic, we use randomly permuted negative samples to approximate the CLUB upper bound, avoiding the quadratic memory cost of enumerating all cross-sample pairs.

C Additional Experimental Setup

C.1 Descriptions of Datasets

Our experiments cover seven multivariate forecasting benchmarks from transformer operation, electricity consumption, meteorological monitoring, and road transportation domains. The ETT benchmark consists of four subsets collected from two electricity transformers. Each subset contains the oil temperature together with six power-load variables. ETTh1 and ETTh2 are sampled hourly, whereas ETTm1 and ETTm2 provide observations at 15-minute intervals. The Electricity dataset comprises hourly consumption series from 321 clients. The Weather dataset contains 21 meteorological variables measured every 10 minutes at a weather station in Germany. The Traffic dataset provides hourly road-occupancy measurements from 862 freeway sensors. The corresponding statistics and forecasting configurations are summarized in Table 4.

Table 4: Statistics and forecasting configurations of the seven datasets.

Dataset	# Timesteps	# Variates	Sampling Interval	Prediction Lengths	Train/Val/Test
ETTh1	17,420	7	1 hour	$\{96, 192, 336, 720\}$	6:2:2
ETTh2	17,420	7	1 hour	$\{96, 192, 336, 720\}$	6:2:2
ETTm1	69,680	7	15 minutes	$\{96, 192, 336, 720\}$	6:2:2
ETTm2	69,680	7	15 minutes	$\{96, 192, 336, 720\}$	6:2:2
Weather	52,696	21	10 minutes	$\{96, 192, 336, 720\}$	7:1:2
Electricity	26,304	321	1 hour	$\{96, 192, 336, 720\}$	7:1:2
Traffic	17,544	862	1 hour	$\{96, 192, 336, 720\}$	7:1:2

To prevent information leakage, all observations are arranged chronologically before partitioning. The four ETT datasets use a training, validation, and test ratio of 6:2:2, while Weather, Electricity, and Traffic use a ratio of 7:1:2. A look-back window of $T = 96$ is adopted for all datasets, and the prediction length is selected from $H \in \{96, 192, 336, 720\}$.

C.2 Descriptions of Evaluation Metrics

We evaluate the final forecast in the time domain using Mean Squared Error (MSE) and Mean Absolute Error (MAE). Given the prediction $\widehat{\mathbf{Y}}_{T+1:T+H} \in \mathbb{R}^{H\times N}$ and its corresponding ground truth $\mathbf{Y}_{T+1:T+H} \in \mathbb{R}^{H\times N}$, we define the forecasting error as $\mathbf{E}_{T+1:T+H} = \widehat{\mathbf{Y}}_{T+1:T+H} - \mathbf{Y}_{T+1:T+H}$. The evaluation metrics are calculated as

$$\begin{aligned} \mathrm{MSE} &= \frac{1}{HN} \left\|\mathbf{E}_{T+1:T+H}\right\|_F^2, \\ \mathrm{MAE} &= \frac{1}{HN} \left\|\mathbf{E}_{T+1:T+H}\right\|_1, \end{aligned} \tag{11}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|_1$ denotes the element-wise ℓ_1 norm. The metrics are averaged over all forecasting instances in the test set, with lower values indicating better forecasting performance.

To quantify the dependence among multi-scale temporal features, we further adopt the Pearson correlation coefficient (PCC) and the Hilbert–Schmidt independence criterion (HSIC). Given two scale-specific features $\mathbf{Z}^s, \mathbf{Z}^{s'} \in \mathbb{R}^{N\times D}$, the PCC between the d-th feature dimensions is defined as

$$\rho_d^{s,s'} = \frac{\mathrm{Cov}(\mathbf{Z}^s_{:,d}, \mathbf{Z}^{s'}_{:,d})}{\sqrt{\mathrm{Var}(\mathbf{Z}^s_{:,d})\,\mathrm{Var}(\mathbf{Z}^{s'}_{:,d})}}. \tag{12}$$

We report the root-mean-square PCC over all feature dimensions:

$$\text{RMS-PCC}(s, s') = \sqrt{\frac{1}{D}\sum_{d=1}^{D}\left(\rho_d^{s,s'}\right)^2}. \tag{13}$$

To measure nonlinear statistical dependence, we compute HSIC with RBF kernels. Let $\mathbf{K}_c = \mathbf{CKC}$ and $\mathbf{L}_c = \mathbf{CLC}$ denote the centered kernel matrices, where $\mathbf{C} = \mathbf{I}_N - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$. The normalized HSIC is defined as

$$\mathrm{nHSIC}(s, s') = \frac{\mathrm{tr}(\mathbf{K}_c\mathbf{L}_c)}{\|\mathbf{K}_c\|_F\|\mathbf{L}_c\|_F}. \tag{14}$$

Lower RMS-PCC and nHSIC values indicate weaker linear and nonlinear dependence

C.3 Descriptions of Baselines

We compare MDN with 14 competitive baselines. Brief descriptions of these baselines are provided below:
(1) **MICN** (Wang et al., 2023): Employs multi-scale hybrid decomposition and local-global convolutions to capture complex temporal patterns at different resolutions.
(2) **TimeKAN** (Huang et al., 2025): Utilizes cascaded frequency decomposition and multi-order Kolmogorov–Arnold networks to capture multi-frequency temporal dynamics.
(3) **MSGNet** (Cai et al., 2024): Leverages frequency-domain analysis to identify periodic patterns and combines attention with adaptive graph convolution to model multi-scale inter-series correlations.
(4) **TimeMixer** (Wang et al., 2024b): Employs decomposable mixing blocks and multiple predictors to model and aggregate temporal patterns across scales.
(5) **TimesNet** (Wu et al., 2022): Transforms one-dimensional time series into multiple two-dimensional tensors according to dominant periods to capture intra- and inter-period variations.
(6) **AMD** (Hu et al., 2025): Decomposes time series into distinct temporal patterns at multiple scales and adaptively aggregates their forecasts through a mixture-of-experts mechanism.
(7) **TimePro** (Ma et al., 2025): Constructs variate- and time-aware hyper-states within a Mamba-based architecture to capture variable relationships and salient temporal information.
(8) **Time-TK** (Zhang et al., 2026): Employs multi-offset token embedding, interactive Kolmogorov–Arnold networks, and temporal interaction to capture dependencies among offset subsequences.
(9) **DUET** (Qiu et al., 2025): Introduces dual clustering over the temporal and channel dimensions to adaptively capture temporal patterns and inter-variable dependencies.
(10) **iTransformer** (Liu et al., 2024): Embeds each variate as a token and applies attention across variates to capture multivariate correlations.
(11) **PatchTST** (Nie et al., 2022): Divides each univariate series into patches and independently processes different variables with a channel-independent Transformer.
(12) **DLinear** (Zeng et al., 2023): Decomposes time series into trend and seasonal components and models them using separate linear layers.
(13) **Crossformer** (Zhang & Yan, 2023): Employs dimension-segment-wise embedding and two-stage attention to model cross-time and cross-variable dependencies.
(14) **CMoS** (Si et al., 2025): Segments time series into temporal chunks and employs correlation mixing to capture diverse chunk-wise spatial correlations.

C.4 Descriptions of Experimental Setting

C.4.1 Computing Environment.

All experiments were conducted on a workstation running Ubuntu 24.04.3 LTS with an Intel Core i9-14900K CPU, 32 CPU cores, and 251 GB of system memory. GPU experiments were performed on an NVIDIA GeForce RTX 5090 with 32 GB memory. The implementation used Python 3.10.20 and PyTorch 2.11.0 with CUDA 12.8.

C.4.2 Hyperparameter Settings of MDN.

We use one CLUB estimator for each pair of scales. Since the final model adopts $S = 4$, 6 CLUB estimators are used to characterize the pairwise dependencies among scale-specific features. Each estimator contains two separate two-layer MLP heads for predicting the conditional mean and log-variance, respectively. The hidden size passed to CLUB is set to $D/2$, corresponding to an intermediate width of $D/4$ in each MLP head. The CLUB estimators are updated five times after each main-model optimization step with a learning rate of 0.01. Positive pairs are constructed using the

same sample index, whereas negative pairs are formed by pairing samples with different indices within each mini-batch.

We denote the number of scales, feature dimension, multi-task loss weight, MI loss weight, learning rate, batch size, and training epochs by S, D, λ_{MT}, λ_{MID}, η, B, and E, respectively. We fix $S = 4$ for all experiments. The search ranges for different dataset groups are summarized in Table 5. For each dataset and prediction length, the final configuration is selected according to the lowest validation error. The selected hyperparameters are reported in Table 6.

Table 5: Hyperparameter search ranges for different datasets.

Dataset	D	λ_{MT}	λ_{MID}	η	B	E
ETTh1	$\{24, 32\}$	$\{.05, .10, .12\}$	$\{10^{-5}, 8\times10^{-5}, 10^{-4}\}$	$\{.040, .050, .055\}$	256	$\{15, 18\}$
ETTh2	$\{16, 24\}$	$\{.08, .10\}$	$\{10^{-4}, 1.5\times10^{-4}, 2\times10^{-4}\}$	$\{.025, .030, .040\}$	$\{256, 384\}$	$\{8, 18, 20\}$
ETTm1	32	$\{.08, .10\}$	$\{3, 4, 6, 7\}\times10^{-4}$	$\{.025, .032\}$	$\{256, 384\}$	$\{12, 15, 20\}$
ETTm2	32	$\{.10, .30\}$	$\{4, 7\}\times10^{-4},\ 1.8\times10^{-3},\ 3\times10^{-3}$	$\{.015, .020, .025\}$	256	$\{10, 18\}$
Electricity	$\{192, 224\}$	.10	5×10^{-5}	$\{.0025, .003, .004, .005\}$	32	30
Weather	$\{40, 56, 72\}$	$\{.25, .35, .40\}$	$\{1.5, 3, 4\}\times10^{-4}$	$\{.0025, .004, .007\}$	$\{64, 128\}$	$\{12, 15, 18\}$
Traffic	$\{192, 240, 256\}$	.10	$\{3, 5, 7\}\times10^{-5}$	$\{.0025, .003\}$	$\{32, 64\}$	$\{60, 75, 80, 85\}$

Table 6: Final hyperparameter configurations used for different datasets and prediction lengths.

Data	H	S	D	λ_{MT}	λ_{MID}	η	B	E
ETTh1	96	4	24	0.10	1e−4	0.0500	256	15
	192	4	24	0.10	1e−4	0.0550	256	18
	336	4	32	0.12	8e−5	0.0500	256	18
	720	4	24	0.05	1e−5	0.0400	256	18
ETTh2	96	4	16	0.08	1.5e−4	0.0300	256	8
	192	4	24	0.10	1e−4	0.0400	384	18
	336	4	16	0.10	2e−4	0.0250	256	20
	720	4	24	0.10	1.5e−4	0.0400	256	20
ETTm1	96	4	32	0.10	3e−4	0.0250	256	15
	192	4	32	0.08	6e−4	0.0250	256	12
	336	4	32	0.10	7e−4	0.0250	256	12
	720	4	32	0.08	4e−4	0.0320	384	20
ETTm2	96	4	32	0.10	7e−4	0.0200	256	18
	192	4	32	0.10	1.8e−3	0.0150	256	18
	336	4	32	0.10	3e−3	0.0200	256	18
	720	4	32	0.30	4e−4	0.0250	256	10
Weather	96	4	56	0.25	1.5e−4	0.0070	128	12
	192	4	72	0.40	4e−4	0.0025	64	12
	336	4	40	0.25	1.5e−4	0.0040	64	15
	720	4	72	0.35	3e−4	0.0025	64	18
Traffic	96	4	192	0.10	5e−5	0.0030	64	85
	192	4	256	0.10	5e−5	0.0025	32	75
	336	4	240	0.10	3e−5	0.0025	32	80
	720	4	256	0.10	7e−5	0.0025	32	60
Electricity	96	4	192	0.10	5e−5	0.0050	32	30
	192	4	224	0.10	5e−5	0.0040	32	30
	336	4	192	0.10	5e−5	0.0025	32	30
	720	4	192	0.10	5e−5	0.0030	32	30

C.4.3 Hyperparameter Settings of Baselines.

The baseline results reported in our experiments come from two sources: results adopted from published benchmark releases and results reproduced by us using the corresponding official implementations. The MICN results are taken from the multi-resolution time-series transformer study (Zhang et al., 2024), while the TimePro and DUET results are obtained from their corresponding official repositories (Ma et al., 2025; Qiu et al., 2025). The results of the other baselines, except for AMD and TimeKAN, are adopted from the official Time-TK repository.

We reproduce AMD using its official implementation because its original configuration uses an input length different from 96, which is adopted by our model and most other baselines. We only change the input length to 96 while keeping its architecture, hyperparameters, and training procedure unchanged. For TimeKAN, we reproduce the results using its official implementation. For all datasets except Traffic, we adopt the dataset-specific default hyperparameters provided in the official repository. Since Traffic is not included in the original TimeKAN benchmark, we use the Electricity configuration, which corresponds to a similarly high-dimensional multivariate setting. Results reproduced by us are marked with *.

D Additional Experimental Results

D.1 Computational Cost

Table 7: Computational cost on ETTh1. MACs, FLOPs, and inference time are reported per sample, while peak GPU memory is measured during training. The MDN parameter count includes 3.816K training-only CLUB parameters.

Model	Params	MACs	FLOPs	Memory	Infer. (ms)
MDN	36.302K	**0.211M**	**0.422M**	776.6	**0.01197**
TimeKAN	16.891K	5.026M	10.051M	461.2	0.03079
AMD	3.176M	22.103M	44.207M	190.0	0.03731

We further compare MDN with two competitive multi-scale forecasting models, TimeKAN and AMD, under an identical computation-cost protocol. All models are evaluated on ETTh1 with an input length of 96 and a prediction horizon of 96, using a batch size of 128 and the same NVIDIA RTX 5090 GPU. The results are summarized in Table 7. MDN exhibits a clear advantage in computational efficiency. It requires only 0.211M MACs and 0.422M FLOPs per sample. TimeKAN and AMD require approximately $23.8\times$ and $104.8\times$ as many operations as MDN, respectively. This remarkably low computational demand also enables fast inference. MDN requires only 0.01197 ms per sample, making it approximately $2.57\times$ and $3.12\times$ faster than TimeKAN and AMD, respectively. These results demonstrate that MDN can effectively model multi-scale temporal patterns without imposing a heavy deployment burden. MDN has a slightly larger parameter count than TimeKAN and a slightly higher peak GPU memory footprint. This additional cost mainly arises from the CLUB estimators introduced to estimate and minimize the mutual information among scale-specific representations. In particular, CLUB introduces 3.816K auxiliary parameters used only during training. These estimators are removed after training, leaving 32.486K parameters in the forecasting network, and do not participate in forecasting. Overall, considering predictive performance, computational complexity, and inference efficiency, MDN provides the best overall trade-off and is the most attractive choice among the three multi-scale models.

D.2 AVERAGE MAIN RESULTS

Table 8: Average results of long-term time series forecasting over four prediction lengths. The best results are **bolded**, and the second-best results are underlined. Baselines marked with * are reproduced using their official implementations.

Methods		MDN Ours		TimeKAN* ICLR 2025		AMD* AAAI 2025		TimeMixer ICLR 2024		MSGNet AAAI 2024		MICN ICLR 2023		TimesNet ICLR 2023		Time-TK WWW 2026	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	Avg	**0.421**	**0.427**	0.424	0.429	0.444	0.431	0.456	0.445	0.453	0.453	0.559	0.535	0.458	0.450	0.432	0.430
ETTh2	Avg	**0.355**	**0.385**	0.388	0.407	0.378	0.400	0.389	0.408	0.413	0.427	0.588	0.525	0.414	0.427	0.372	0.397
ETTm1	Avg	**0.371**	**0.385**	0.378	0.395	0.389	0.395	0.385	0.399	0.400	0.412	0.392	0.414	0.400	0.406	0.379	0.393
ETTm2	Avg	**0.271**	**0.313**	0.278	0.324	0.282	0.327	0.280	0.325	0.289	0.331	0.328	0.382	0.291	0.333	0.276	0.321
Weather	Avg	0.248	**0.269**	0.244	0.272	0.261	0.286	0.245	0.276	0.250	0.278	**0.243**	0.299	0.259	0.287	0.256	0.278
Traffic	Avg	**0.413**	**0.268**	0.576	0.373	0.518	0.335	0.485	0.298	0.660	0.382	0.542	0.316	0.620	0.336	0.425	0.278
Electricity	Avg	**0.163**	**0.254**	0.198	0.287	0.193	0.280	0.182	0.272	0.195	0.301	0.187	0.295	0.193	0.295	0.174	0.265

Methods		TimePro ICML 2025		CMoS ICML 2025		DUET KDD 2025		iTransformer ICLR 2024		PatchTST ICLR 2023		Crossformer ICLR 2023		DLinear AAAI 2023	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	Avg	0.438	0.438	0.448	0.442	0.443	0.436	0.463	0.454	0.469	0.455	0.529	0.522	0.456	0.452
ETTh2	Avg	0.377	0.404	0.392	0.410	0.372	0.398	0.383	0.407	0.389	0.411	0.942	0.684	0.559	0.515
ETTm1	Avg	0.391	0.401	0.412	0.410	0.390	0.393	0.405	0.410	0.396	0.406	0.513	0.495	0.403	0.407
ETTm2	Avg	0.281	0.326	0.288	0.330	0.280	0.324	0.290	0.335	0.291	0.336	0.757	0.611	0.350	0.401
Weather	Avg	0.252	0.276	0.251	0.278	0.251	0.273	0.258	0.278	0.259	0.281	0.259	0.315	0.265	0.317
Traffic	Avg	0.439	0.286	0.617	0.366	0.451	0.269	0.428	0.282	0.555	0.362	0.550	0.304	0.625	0.383
Electricity	Avg	0.169	0.262	0.204	0.284	0.172	0.259	0.178	0.270	0.211	0.301	0.244	0.334	0.212	0.300

Table 8 reports the forecasting results averaged over the four prediction lengths. MDN achieves the lowest MAE on all seven datasets and the lowest MSE on six datasets, ranking first in 13 of the 14 dataset–metric comparisons. This consistent performance benefits from the coordinated use of PoE-based separation and MI-based disentanglement, which alleviate scale-shared interference and non-linear cross-scale dependence, respectively. Time-TK (Zhang et al., 2026) is a competitive single-scale baseline that captures temporal dependencies from multiple offset views but does not explicitly model temporal patterns at different resolutions. By extracting disentangled multi-scale features, MDN reduces MSE and MAE by 3.33% and 2.75% on average relative to Time-TK, respectively. TimeKAN (Huang et al., 2025) is a competitive multi-scale baseline that models multi-frequency dynamics through cascaded decomposition and multi-order Kolmogorov–Arnold networks. However, its residual-based decomposition does not explicitly constrain dependence among scale representations. In comparison, MDN reduces MSE and MAE by 8.27% and 7.51% on average relative to TimeKAN, respectively. Electricity contains recurring consumption patterns across different temporal resolutions, which can cause substantial overlap among multi-scale representations. MDN reduces MSE and MAE by 2.98% and 1.93% on average relative to the best baseline results. Moreover, its MSE reduction increases monotonically from 2.16% at $H = 96$ to 4.41% at $H = 720$, indicating that disentangled multi-scale features become increasingly beneficial for long-term forecasting.

D.3 Prediction Visualization

To qualitatively assess the forecasting performance of MDN, Figure 6 presents its predictions on Electricity under four prediction lengths. All panels visualize the OT variable from the same test window. To avoid cherry-picking, we select a common window whose OT-variable errors are jointly closest to the median across the four settings. MDN consistently captures the recurring consumption patterns as the prediction length increases and preserves the overall temporal dynamics even at $H = 720$, complementing the quantitative results reported in the main paper.

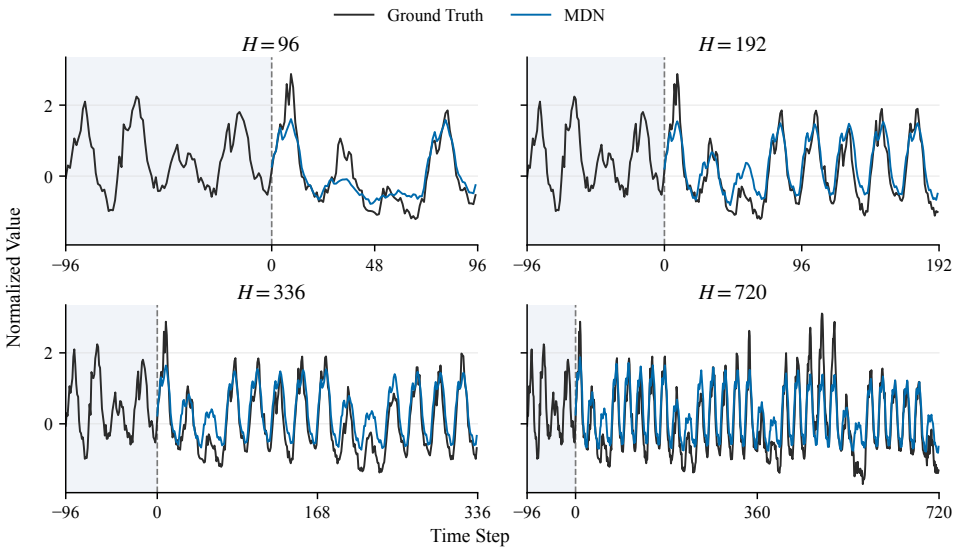


Figure 6: Prediction visualizations of MDN on Electricity under four prediction lengths. All panels show the OT variable from the same test window. The shaded regions denote the input sequence, and the dashed lines mark the forecast starts.

D.4 Full Ablation Results

Table 9 reports the complete ablation results on six datasets. Averaged over the four prediction lengths, MDN obtains the lowest MSE and MAE on every dataset, confirming that the three design groups are complementary. Replacing the MI-based constraint with cosine similarity performs better on average than removing it, but remains inferior to MDN, supporting the benefit of modeling non-linear statistical dependence. The degradations of **w/o Sep** and **-v Stoch** further verify the contributions of scale-shared feature separation and PoE-based extraction, respectively. Both interaction variants reduce forecasting accuracy, while **w/o MTL** causes the largest overall degradation, highlighting the importance of multi-task supervision for generalization.

Table 9: Full ablation study results on six datasets. The best results are **bolded**.

Variation		w/o MI		-v Cos		w/o Sep		-v Stoch		w/o Intra		w/o Inter		w/o Fusion		w/o MTL		MDN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.370	**0.390**	0.369	**0.390**	0.368	**0.390**	0.369	0.391	0.371	0.391	0.375	0.394	0.367	0.392	0.371	0.394	**0.365**	**0.390**
	192	0.418	**0.420**	0.420	**0.420**	0.420	0.421	0.420	0.421	0.417	**0.420**	0.420	**0.420**	0.423	0.424	0.422	0.423	**0.413**	**0.420**
	336	0.456	0.440	0.456	0.440	0.461	0.442	0.458	0.441	0.454	**0.439**	0.462	0.440	**0.450**	0.440	0.467	0.446	**0.450**	0.440
	720	0.473	0.468	0.476	0.470	0.482	0.474	0.474	0.469	0.468	0.464	0.468	0.465	0.469	0.468	0.477	0.472	**0.455**	**0.459**
ETTh2	96	0.284	0.334	0.283	0.333	0.283	0.331	0.281	0.331	0.282	**0.330**	0.281	0.332	0.286	0.333	0.291	0.340	**0.278**	**0.330**
	192	0.361	0.383	0.356	0.380	0.356	0.380	0.358	0.381	0.356	0.378	0.358	0.382	0.363	0.383	0.363	0.385	**0.352**	**0.377**
	336	0.405	0.416	0.400	0.415	0.399	0.412	0.415	0.420	0.405	0.415	0.399	0.415	0.404	0.416	0.403	0.415	**0.394**	**0.410**
	720	0.402	0.430	0.404	0.430	0.404	0.433	0.404	0.430	0.418	0.438	0.409	0.432	0.402	0.427	0.411	0.435	**0.394**	**0.424**
ETTm1	96	0.319	0.352	0.311	0.346	0.310	0.346	0.310	0.346	0.310	**0.343**	0.313	0.347	0.316	0.349	0.313	0.349	**0.306**	0.344
	192	0.358	0.374	0.358	0.374	0.356	0.374	0.358	0.375	0.358	0.374	0.361	0.376	0.361	0.377	0.365	0.379	**0.354**	**0.372**
	336	0.388	0.396	0.389	0.396	0.389	0.396	0.388	0.396	0.389	0.396	0.394	0.399	0.391	0.397	0.398	0.403	**0.384**	**0.394**
	720	0.447	0.433	0.446	0.433	0.443	**0.431**	0.446	0.432	0.445	0.433	0.454	0.435	0.445	0.434	0.466	0.444	**0.440**	**0.431**
ETTm2	96	0.172	0.249	0.175	0.253	0.173	0.249	0.170	0.248	0.173	0.251	0.172	0.251	0.173	0.251	0.176	0.252	**0.169**	**0.247**
	192	0.238	0.292	0.237	0.293	0.242	0.297	0.236	0.293	0.237	0.293	0.240	0.296	0.239	0.296	0.242	0.295	**0.234**	**0.290**
	336	0.298	0.331	0.298	0.332	0.300	0.333	0.295	0.331	0.297	0.332	0.305	0.337	0.301	0.336	0.300	0.336	**0.293**	**0.329**
	720	0.394	0.389	0.391	0.388	0.393	0.389	0.393	0.390	0.390	0.389	0.394	0.389	0.392	0.388	0.396	0.392	**0.388**	**0.387**
Weather	96	0.167	0.207	0.166	0.205	0.170	0.205	0.170	0.210	0.175	0.212	0.168	0.208	0.165	0.202	0.169	0.207	**0.161**	**0.200**
	192	0.219	0.251	0.220	0.252	0.220	0.250	0.216	0.250	0.224	0.255	0.215	0.250	0.219	0.251	0.218	0.252	**0.213**	**0.247**
	336	0.273	**0.290**	0.274	**0.290**	0.277	0.294	0.275	0.293	0.277	0.293	0.274	0.293	0.273	0.292	0.278	0.296	**0.270**	**0.290**
	720	0.350	0.342	0.351	0.343	0.352	0.344	0.352	0.344	0.357	0.348	0.350	0.345	0.349	0.343	0.355	0.347	**0.346**	**0.340**
ECL	96	0.138	0.228	0.139	0.228	0.141	0.230	0.140	0.230	0.157	0.240	0.145	0.236	0.141	0.231	0.141	0.232	**0.136**	**0.227**
	192	**0.152**	**0.242**	**0.152**	**0.242**	0.155	0.244	0.154	0.243	0.166	0.250	0.158	0.248	0.155	0.245	0.157	0.247	**0.152**	**0.242**
	336	**0.167**	0.260	**0.167**	0.260	0.169	0.261	0.168	**0.259**	0.183	0.268	0.172	0.264	0.168	0.260	0.172	0.264	**0.167**	**0.259**
	720	0.197	0.288	0.197	0.288	0.200	0.292	0.197	0.287	0.219	0.300	0.204	0.294	0.201	0.291	0.205	0.296	**0.195**	**0.286**

D.5 Robustness to Random Seeds.

We evaluate the robustness of MDN to random initialization on ETTm1, as a representative dataset of the ETT family, and Electricity, as a representative high-dimensional multivariate dataset. Each experiment is repeated using five random seeds, i.e., $\{2021, 2022, 2023, 2024, 2025\}$. Table 10 reports the mean and sample standard deviation of the resulting performance.

Table 10: Forecasting performance over five random seeds. Results are reported as mean $\pm$ standard deviation.

Dataset	H	MSE	MAE
ETTm1	96	0.3081 ± 0.0015	0.3446 ± 0.0010
	192	0.3548 ± 0.0009	0.3727 ± 0.0005
	336	0.3857 ± 0.0014	0.3949 ± 0.0008
	720	0.4434 ± 0.0017	0.4326 ± 0.0008
Electricity	96	0.1377 ± 0.0008	0.2282 ± 0.0005
	192	0.1527 ± 0.0002	0.2425 ± 0.0003
	336	0.1672 ± 0.0004	0.2595 ± 0.0004
	720	0.1973 ± 0.0012	0.2883 ± 0.0016

The standard deviations remain below 0.0017, and the coefficients of variation are below 0.60% across all settings, demonstrating limited performance variation across random seeds. We further apply the Friedman test by treating the prediction lengths as blocks and the random seeds as conditions. No significant seed effect is detected for either MSE or MAE on ETTm1, with $p = 0.1359$ and $p = 0.2311$, respectively, or on Electricity, with $p = 0.0563$ and $p = 0.1586$, respectively. These results support the robustness of MDN to random initialization.