

Received 28 July 2025, accepted 10 August 2025, date of publication 12 August 2025, date of current version 18 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3598141

RESEARCH ARTICLE

Enhancing Global and Local Context Modeling in Time Series Through Multi-Step Transformer-Diffusion Interaction

DAGYEONG NA¹, JINHO KANG², BYOUNGWOONG KANG³,
AND JUNSEOK KWON^{1,4}, (Member, IEEE)

¹Graduate School of Artificial Intelligence, Chung-Ang University, Seoul 06974, South Korea

²Department of Artificial Intelligence, University of Seoul, Seoul 02504, South Korea

³Graduate School of Artificial Intelligence, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

⁴School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Junseok Kwon (jskwon@cau.ac.kr)

This work was supported in part by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2022-00143911, AI Excellence Global Innovative Leader Education Program); in part by the Chung-Ang University Graduate Research Scholarship in 2024; and in part by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2021-II211341, Artificial Intelligence Graduate School Program (ChungAng University)).

ABSTRACT Multivariate time-series forecasting (MTSF) has become crucial across various domains, with transformer-based architectures emerging as the primary choice because of their excellent ability to capture long-term dependencies. However, these models often fail to represent fine-grained local patterns, which are critical for accurate forecasting. To address this limitation, we propose a novel MTSF method based on the diffusion process, utilizing the transformer as prior knowledge. By conditioning the denoising process on global embeddings derived from the transformer, our approach effectively captures both global and local patterns. However, when transformers are used as conditional priors in the diffusion process, errors in their predictions can propagate, adversely affecting overall performance. Thus, we also introduce a mechanism that mitigates the impact of such errors by enabling effective interactions between transformers and diffusion models. Furthermore, the proposed multi-step denoising process progressively refines temporal patterns, preserving both global structures and local variations to enhance robustness and generalization. Experimental results demonstrate that the proposed approach overcomes the limitations of existing models, consistently outperforming previous state-of-the-art models (TMDM), achieving up to 97.9% lower mean squared error (MSE) on Weather and 96.3% on ETTm2.

INDEX TERMS Multivariate time series forecasting, diffusion, transformer, local-global modeling.

I. INTRODUCTION

Multivariate Time-Series Forecasting (MTSF) has become increasingly important in various domains [1], [2], [3], [4], [5], requiring the ability to capture temporal dependencies between multiple variables. For this task, models based on Convolutional Neural Networks (CNNs) [6], [7] and Recurrent Neural Networks (RNNs) [8], [9], [10] have been introduced. However, they have faced limitations in modeling long-term dependencies in time series. To overcome these

limitations, transformer architectures [11] have emerged as more effective solutions, using self-attention mechanisms to capture long-term dependencies [12], [13], [14], [15].

Transformer-based MTSF models such as PatchTST [15] and iTransformer [16] have leveraged their strengths in long-term dependency forecasting to achieve state-of-the-art (SOTA) performance. However, they struggle to capture local temporal patterns that are not based on the trend features. This limitation arises from the inherent design of transformer-based models, which emphasize global interactions across all positions. Consequently, local patterns may be overshadowed by global features, leading to a loss of critical details (see

The associate editor coordinating the review of this manuscript and approving it for publication was Wojciech Sałabun.

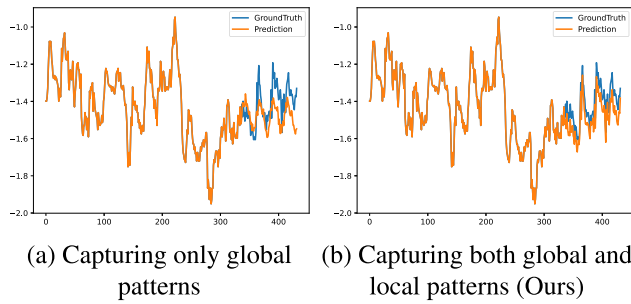


FIGURE 1. Advantages of capturing both global and local temporal dynamics for MTSF. (a) iTransformer [19] predictions on the ETTh1 dataset. Despite iTransformer demonstrating strong performance on the exchange dataset, the local dynamics are not well-reflected. (b) The proposed method effectively captures both the local patterns and global trends and seasonality.

the illustrative example in Figure 1(a)). Local patterns refer to short-term fluctuations and fine-grained variations within a time series, distinguishing them from broader trends or seasonal components. These patterns capture rapid changes and localized deviations, which are crucial for accurately modeling real-world temporal dynamics [17]. In contrast, global patterns refer to broader trends and seasonality, which capture the overarching structure and periodic behaviors in time series data. Since both local and global patterns are essential characteristics of time series data [18], this motivates the need for a model that can jointly capture both aspects, which is conceptually depicted in Figure 1(b) for illustrative purpose.

Diffusion-based models [20] have emerged as a promising approach for time series forecasting (TSF) [21], [22], [23], which allows explicit inference of global and local patterns. These models utilize white Gaussian noise during the diffusion process. However, they often fail to capture the intrinsic characteristics of time-series data, which leads to performance degradation. To address the shortcomings of diffusion-based models in multivariate time series forecasting (MTSF), prior information extracted from transformer-based models, which demonstrates strong performance and serves as a reliable source of prior knowledge, has been incorporated into the diffusion process [24], enhancing the model's ability to capture both global and local temporal dependencies.

However, when transformers are utilized as conditional priors in the diffusion process, errors in their predictions can propagate, negatively affecting overall performance. To address this, we introduce an interaction mechanism that enhances the integration between transformer outputs and the diffusion model. This mechanism consists of three key components: transformer-derived conditional guidance, future-mixup for robust training, and a multi-step denoising strategy that refines temporal patterns. Together, these components help mitigate the influence of prior prediction errors and improve forecasting stability. Figure 2 provides a conceptual illustration of the proposed interaction mechanism. This figure conceptually contrasts the behavior of models with and without interaction, emphasizing how the proposed

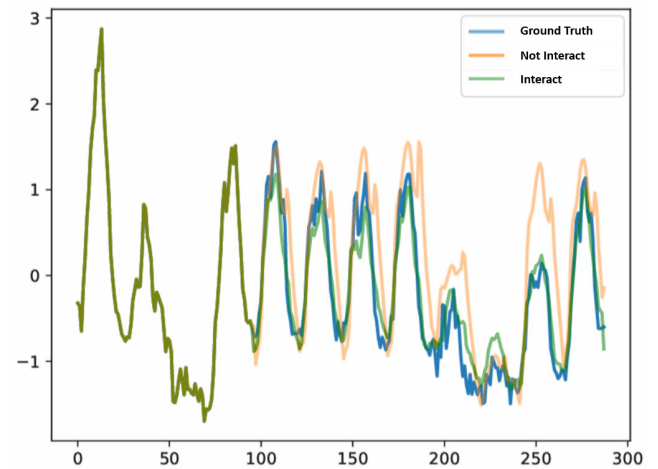


FIGURE 2. Illustration of how interaction mechanisms between transformer and diffusion models affect forecasting predictions. The blue line represents the ground truth, the orange line shows a model output without interaction, and the green line includes the proposed interaction mechanism. Although both models used the same prior, the interactive version aligns more closely with fine-grained local patterns, motivating the need for tighter integration between components.

integration is designed to better preserve local temporal patterns during prediction.

In this study, we propose a novel approach that combines the complementary strengths of the transformer and diffusion models. One of the key challenges in applying diffusion models to time series forecasting is maintaining temporal consistency throughout the generative process. To address this, we follow [24] and redefine the endpoint of the diffusion process as $\hat{\mathbf{y}}_{0:M} = \mathbb{E}[\mathbf{y}_{0:M} | \mathbf{x}_{0:N}]$ is the transformer-derived conditional mean. This anchoring enables the diffusion model to leverage global structural information from the transformer, thereby improving the modeling of future time steps. However, errors in transformer predictions can accumulate over time, leading to a significant degradation in long-term forecasting. Thus, we incorporated future-mixup, as introduced in [23], during training. This technique randomly mixes the ground truth of future predictions with past observations, thereby enhancing training robustness. Instead of solely relying on Transformer outputs, our model introduces interaction techniques that help absorb inaccuracies and improve the robustness of the forecasting process.

In standard diffusion-based models, denoising is typically performed in a single step, using the same number of diffusion steps as in the forward process. However, this often fails to balance noise removal and the preservation of temporal structures, leading to either the loss of fine-grained details or the retention of residual noise. To address this issue, we propose a multi-step denoising strategy that fixes the diffusion step to control the noise magnitude and applies multiple refinement steps during reconstruction. This strategy is intended to enhance both global trends and local patterns, while avoiding overfitting to noise.

Our framework is designed to complement the strengths of transformer models with diffusion-based reconstruction while addressing their limitations in local context modeling through future-mixup and iterative denoising techniques. The main contributions of the proposed method are summarized as follows.

- We propose a novel forecasting framework that integrates global representations from transformer models with local pattern modeling via a diffusion process, enabling accurate multivariate time series forecasting.
- To address error propagation from transformer-based priors, we introduce a future-mixup mechanism that improves the robustness of conditioning and adopts a multi-step denoising process to gradually refine the temporal dynamics.
- Extensive experiments on five real-world benchmarks demonstrate that our method consistently improves the forecasting accuracy over state-of-the-art models, including transformer- and diffusion-based approaches.

II. RELATED WORKS

A. CONVENTIONAL MTSF

Numerous studies on the MTSF have aimed to predict the future values of multiple variables using their historical data and inter-dependencies without external guidance, primarily by developing advanced model architectures and frameworks. Early CNN-based [6], [7] and RNN-based [8], [9], [10] methodologies have been introduced as deep learning approaches to perform MTSF tasks. RNNs modeled time series based on the Markov assumption, whereas CNNs utilized temporal convolutional networks; however, both approaches struggled to capture long-term dependencies. To overcome these limitations, transformer models, which have shown great success in natural language processing [25] and computer vision [26], have been introduced for time-series forecasting.

B. TRANSFORMER-BASED MTSF

Recent studies have applied transformer variants to long-term MTSF [12], [13], [14], [15]. Through attention mechanisms, they effectively captured long-term temporal dependencies, significantly improving forecasting performance. Informer [12] improved the efficiency of long-sequence prediction by employing a ProbSparse self attention mechanism that probabilistically selects dominant queries to reduce the computational complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(L \log L)$. The crossformer [27] recognized the correlations between channels and modeled these interrelationships effectively. Autoformer [13] incorporated a decomposition block to progressively capture long-term trends and seasonality, whereas Fedformer [14] utilized frequency-based decomposition to improve performance. The non-stationary transformer (NSTransformer) [19] extends decomposition approaches by incorporating adaptive stationarization, which explicitly handles non-stationary

data (e.g., shifting trends or volatile variances) through learnable normalization and detrending layers, enabling dynamic adjustment of decomposition weights for unstable real-world patterns. PatchTST [15] leveraged patching to extract local semantic information from time-series data and maximize performance through a channel-independent approach. iTransformer [16] preserved the embedding of each input channel independently and incorporated inter-variate correlations through attention mechanisms.

In contrast to these approaches, the proposed method integrates the long-term trends and seasonality captured by the transformer while addressing its limitations in modeling fine-grained local patterns through the diffusion process.

C. DIFFUSION-BASED MTSF

Diffusion models have been applied to MTSF. TimeGrad [22], a conditional diffusion model, utilized the hidden states of an RNN as a guide and made predictions in an autoregressive manner. However, for a long time series, inference faces challenges owing to autoregressive decoding. To address this, TimeDiff [23] employed a non-autoregressive approach, using future mixup and autoregressive initialization as conditioning. Furthermore, mr-Diff [21] integrated decomposition and multi-resolution techniques to achieve performance improvements. The aforementioned models trained the diffusion process to converge to a standard normal distribution $\mathcal{N}(0, I)$, as the endpoint.

By contrast, the proposed method utilizes the transformer's strong predictive capabilities as prior knowledge, guiding the diffusion process to converge toward its predicted conditional mean.

D. TRANSFORMER+DIFFUSION-BASED MTSF

When using diffusion models alone for MTSF, unstable generation is a critical issue. To address this, previous studies have proposed integrating a transformer as a conditioning mechanism within the diffusion process [24], [28], [29]. However, the existing models that combine transformers with diffusion processes suffer from erroneous generation when the transformer incorrectly predicts the conditional mean.

To mitigate this issue, we applied the *future-mixup* technique during training, which compensates for potential prediction errors in the transformer, leading to more accurate and robust forecasting.

III. PROPOSED METHOD

In this section, we present a unified framework that integrates transformer-based global context modeling with diffusion-based local pattern refinement. As shown in Figure 3, our method introduces a future-mixup strategy to reduce the impact of erroneous transformer priors during training, and a multi-denoising mechanism to progressively reconstruct fine-grained temporal details. Together, these components enable the robust and accurate multivariate time series forecasting by effectively aligning global trends with local variations.

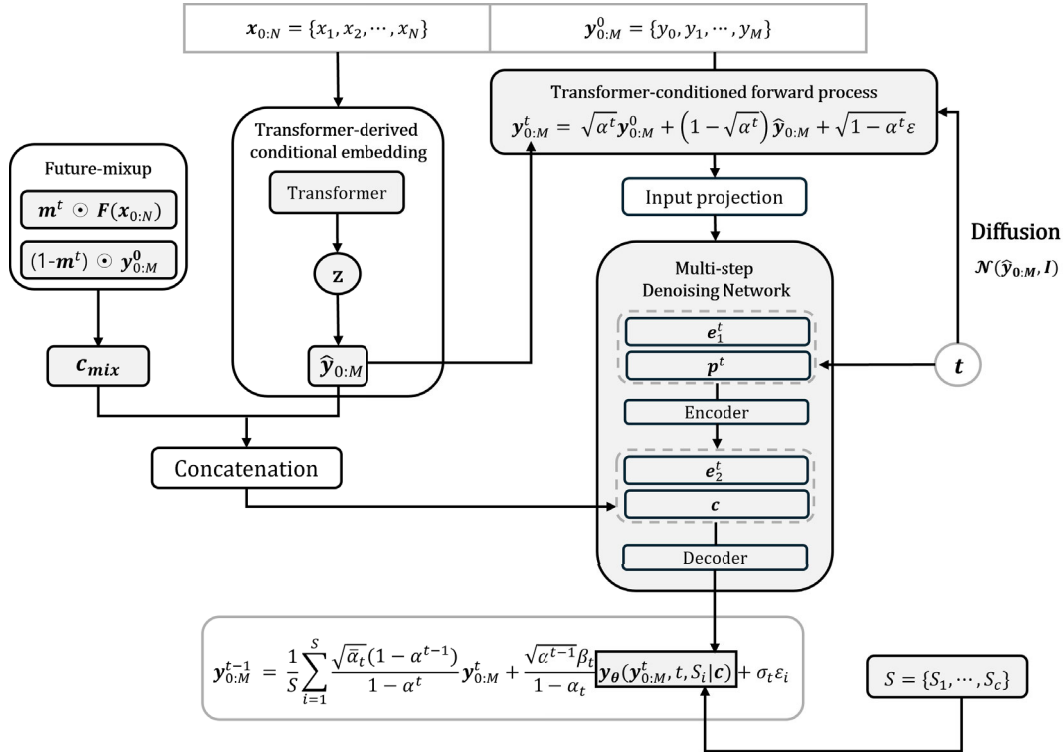


FIGURE 3. Overall structure of the proposed method. The boxes in the diagram represent different components: (1) Transformer-derived conditional embedding serves as prior knowledge in the diffusion process. (2) The Transformer-conditioned forward process leverages this prior to preserve essential patterns and improve forecasting performance. (3) The future-mixup strategy mitigates erroneous conditioning. (4) The multi-step denoising network refines predictions through multiple denoising steps, enhancing reconstruction quality.

A. DENOISING DIFFUSION PROBABILISTIC MODELS

Among the diffusion models, one of the most well-known is the Denoising Diffusion Probabilistic Model (DDPM) [20]. The forward process follows a Markov process, where Gaussian noise is progressively added to the data at each step, transforming the original, $\mathbf{y}_{0:M}^0$, into complete Gaussian noise, $\mathbf{y}_{0:M}^T$, according to an increasing variance schedule, $\beta_1, \dots, \beta_t$ with $\beta_t \in (0, 1)$. The approximate posterior $q(\mathbf{y}_{0:M}^{1:T} | \mathbf{y}_{0:M}^0)$ was computed as follows:

$$q(\mathbf{y}_{0:M}^{1:T} | \mathbf{y}_{0:M}^0) = \prod_{t=1}^T q(\mathbf{y}_{0:M}^t | \mathbf{y}_{0:M}^{t-1}), \quad (1)$$

$$q(\mathbf{y}_{0:M}^t | \mathbf{y}_{0:M}^{t-1}) = \mathcal{N}(\sqrt{1 - \beta^t} \mathbf{y}_{0:M}^{t-1}, \beta^t \mathbf{I}), \quad (2)$$

where T denotes the total number of steps in the forward process.

The reverse process is also a Markov process that reconstructs the data $\mathbf{y}_{0:M}^0$ by gradually removing noise from the corrupted time series $\mathbf{y}_{0:M}^T$. This process can be defined as follows:

$$p_{\theta}(\mathbf{y}_{0:M}^{0:T}) = p(\mathbf{y}_{0:M}^T) \prod_{t=1}^T p_{\theta}(\mathbf{y}_{0:M}^{t-1} | \mathbf{y}_{0:M}^t), \quad (3)$$

$$p_{\theta}(\mathbf{y}_{0:M}^{t-1} | \mathbf{y}_{0:M}^t) = \mathcal{N}(\mu_{\theta}(\mathbf{y}_{0:M}^t, t), \tilde{\beta}^t \mathbf{I}). \quad (4)$$

The final loss function is derived from the observation that training on the variant of the variational bound enhances the sampling quality, where the network ϵ_{θ} is trained to predict the noise.

$$\mathcal{L} = \|\epsilon - \epsilon_{\theta}(\sqrt{\alpha^t} \mathbf{y}_{0:M}^0 + \sqrt{1 - \alpha^t} \epsilon, t)\|^2, \quad (5)$$

where t is uniformly distributed between 1 and T .

B. TRANSFORMER AS PRIOR KNOWLEDGE

Although transformers have achieved significant advancements in time series forecasting, diffusion-based models often lag behind the limitations in generating informative conditional embeddings [22], [30], [31]. To address this, prior works, such as [24], leverage transformer-derived representations to condition the diffusion process. However, inaccurate transformer guidance may degrade performance. We mitigate this issue by adopting the future-mixup strategy [23], which improves the robustness of conditional embeddings and enhances the overall effectiveness of the guided diffusion process.

1) TRANSFORMER-DERIVED CONDITIONAL EMBEDDINGS

Given the transformer architecture $\mathcal{T}$ and historical time series $\mathbf{x}_{0:N}$, we derive a representation $\mathcal{T}(\mathbf{x}_{0:N})$ that serves as

a guide for the true posterior distribution $\mathbf{z}$:

$$q(\mathbf{z}|\mathcal{T}(\mathbf{x}_{0:N})) \sim \mathcal{N}(\tilde{\mu}_z(\mathcal{T}(\mathbf{x}_{0:N})), \tilde{\sigma}_z(\mathcal{T}(\mathbf{x}_{0:N}))), \quad (6)$$

where $\tilde{\mu}_z$ and $\tilde{\sigma}_z$ are modeled using neural networks. Given a well-trained $\mathbf{z}$, we can construct the conditional representation $\hat{\mathbf{y}}_{0:M}$:

$$\hat{\mathbf{y}}_{0:M} \sim \mathcal{N}(\mu_z(\mathbf{z}), \mathbf{I}), \quad (7)$$

where μ_z is modeled as a nonlinear function using neural networks. In (7), the latent variable $\mathbf{z}$ captures the prior knowledge of a well-designed transformer and is later used to generate conditional embedding $\hat{\mathbf{y}}_{0:M}$.

2) FUTURE-MIXUP TECHNIQUE

To mitigate the issue of the transformer failing to provide sufficient guidance, we incorporate a future-mixup strategy. This method integrates the mapping of past information, $\mathcal{F}(\mathbf{x}_{0:N})$, with the future ground truth, $\mathbf{y}_{1:M}^0$. We employed a convolution network, $\mathcal{F}$, to map past information, a commonly used approach in time-series forecasting [32], [33].

During training, this results in the following conditioning:

$$\mathbf{c}_{mix} = \mathbf{m}^t \odot \mathcal{F}(\mathbf{x}_{0:N}) + (1 - \mathbf{m}^t) \odot \mathbf{y}_{1:M}^0, \quad (8)$$

where t represents the diffusion step, $\mathbf{m}^k \in [0, 1]^{d \times H}$ is a mixing matrix in which each element is randomly sampled from a uniform distribution over $[0, 1]$, and $\odot$ denotes the Hadamard product. During inference, as $\mathbf{y}_{1:M}^0$ is no longer available, $\mathbf{c}_{mix}$ is simplified to:

$$\mathbf{c}_{mix} = \mathcal{F}(\mathbf{x}_{0:N}). \quad (9)$$

By incorporating the future-mixup strategy, we correct erroneous conditional information from the transformer, enhancing its guidance in the diffusion process and ultimately improving the time-series forecasting performance.

C. DOUBLE CONDITIONAL DIFFUSION MTSF

1) TRANSFORMER CONDITIONED FORWARD PROCESS

Conventional diffusion-based MTSF models often enforce $\mathbf{y}_{0:M}^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which risks erasing critical temporal dependencies by projecting complex time series distributions into standard Gaussian space. Following [24], we instead anchor the terminal distribution to $\mathcal{N}(\hat{\mathbf{y}}_{0:M}, \mathbf{I})$, where $\hat{\mathbf{y}}_{0:M}$ denotes the transformer-predicted conditional mean of the future time series. This modification allows the denoising process to preserve the high-level structural information embedded in the transformer output while guiding the generation toward a meaningful prior.

The forward process for the conditional distribution is then defined as:

$$q(\mathbf{y}_{0:M}^t | \mathbf{y}_{0:M}^{t-1}, \hat{\mathbf{y}}_{0:M}) = \mathcal{N}\left(\mathbf{y}_{0:M}^t \middle| \sqrt{1 - \beta^t} \mathbf{y}_{0:M}^{t-1} + (1 - \sqrt{1 - \beta^t}) \hat{\mathbf{y}}_{0:M}, \beta^t \mathbf{I}\right), \quad (10)$$

Algorithm 1 Training

Input : $\mathbf{x}_{0:N}$, noise step t , denoising step $S = \{S_1, S_2, \dots, S_c\}$

Output : $\hat{\mathbf{y}}_{0:M}^{t-1}$

Training

- 1: Initialize the parameters;
- 2: **Repeat**
- 3: Draw $\mathbf{y}_{0:M}^0 \sim q(\mathbf{y}_{0:M}^0 | \mathbf{x}_{0:N})$.
- 4: Draw $t \sim \text{Uniform}(1, \dots, T)$.
- 5: Draw $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- 6: Obtain $\hat{\mathbf{y}}_{0:M}$ by using (7).
- 7: Generating diffused samples.

$$\mathbf{y}_{0:M}^t = \sqrt{\alpha^t} \mathbf{y}_{0:M}^0 + (1 - \sqrt{1 - \alpha^t}) \hat{\mathbf{y}}_{0:M} + \sqrt{1 - \alpha^t} \epsilon. \quad (11)$$

- 8: Obtain the diffusion step t embedding $\mathbf{p}^t$ using (III-C2).
- 9: Randomly generate a matrix $\mathbf{m}^t$.
- 10: Obtain the condition $\mathbf{c}_{mix}$ by future-mixup using (8).
- 11: Obtain condition $\mathbf{c}$ based on $\mathbf{c}_{mix}$ and $\hat{\mathbf{y}}_{0:M}$ by (13).
- 12: Use the denoising network to generate a denoised sample with multiple reconstructions using (14).
- 13: Calculate the loss $\mathcal{L}(\theta)$ in (15).
- 14: **Until** converged.

where the diffusion schedule $\{\beta^t\}_{t=1:T} \in (0, 1)$, $\bar{\alpha}^t = 1 - \beta^t$, and $\alpha^t = \prod_{i=1}^t \bar{\alpha}^i$.

To simplify the application, we can sample $\mathbf{y}_{0:M}^t$ from $\mathbf{y}_{0:M}^0$ at an arbitrary time step t , following the approach in (11). Using this formulation, the diffusion process begins with the true data $\mathbf{y}_{0:M}^0$ and gradually transitions to $\hat{\mathbf{y}}_{0:M}$, effectively interpolating the data between them. This approach utilizes prior knowledge of the conditional mean estimation provided by the transformer, preserving essential patterns and enhancing the model's forecasting performance.

2) MULTI-RECONSTRUCTION DENOISING NETWORK

The denoising network processes the output of the forward process $\mathbf{y}_{0:M}^t$. Its architecture follows the design principles of TimeDiff [23], ensuring the efficient processing of temporal features in time series data. Initially, $\mathbf{y}_{0:M}^t$ passes through an input projection layer consisting of multiple convolutional layers, producing an embedding $\mathbf{e}_1^t$. Subsequently, a diffusion-step embedding $\mathbf{p}^t$ is computed using sinusoidal position embedding [11], inspired by transformers:

$$\mathbf{p}^t = \text{SiLU}(\text{FC}(\text{SiLU}(\text{FC}(k_e)))) \quad (12)$$

where k_e denotes the sinusoidal embedding of timestep t . The embedding $\mathbf{p}^t$ is broadcasted over the sequence length and concatenated with $\mathbf{e}_1^t$ along the channel dimension. The concatenated features are then processed by a multilayer convolution-based encoder to generate $\mathbf{e}_2^t$.

The double condition, consisting of $\mathbf{c}_{mix}$ in (8) generated by future-mixup and $\hat{\mathbf{y}}_{0:M}$ obtained from the transformer, was

Algorithm 2 Testing (Inference Phase)**Input :** $\mathbf{x}_{0:N}$, noise step t , denoise step $S = \{S_1, S_2, \dots, S_c\}$ **Output :** $\hat{\mathbf{y}}_{0:M}^{t-1}$ **Training**

```

1:  $\mathbf{y}_{0:M}^T \sim \mathcal{N}(\hat{\mathbf{y}}_{0:M}, \mathbf{I})$ 
2:
3: for  $t = T, \dots, 1$  do
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , if  $t > 1$  else  $\epsilon = 0$ 
5:   Obtain diffusion step  $t$ 's embedding  $\mathbf{p}^t$  using (III-C2).

6:   Obtain condition  $\mathbf{c}_{mix}$  using (9).
7:   Obtain  $\hat{\mathbf{y}}_{0:M}$  by using (7).
8:   Obtain condition  $\mathbf{c}$  based on  $\mathbf{c}_{mix}$  and  $\hat{\mathbf{y}}_{0:M}$  using (13).
9:   Sample  $\hat{\mathbf{y}}_{0:M}^{t-1}$  from equation (14).
10:   $\mathbf{y}_{0:M}^{t-1} = \hat{\mathbf{y}}_{0:M}^{t-1}$ 
11: end for
12: Return  $\mathbf{y}_{0:M}^0$ 

```

TABLE 1. Summary of dataset statistics.

Dataset	Dim	Freq.	Time steps	Pred. steps
Exchange	8	1 Day	7,588	192
ILI	7	1 Week	966	36
ETTm2	7	15 Min	69,680	192
Electricity	321	1 Hour	26,304	192
Weather	21	10 Min	52,695	192

applied along the channel dimension to condition the model.

$$\mathbf{c} = \text{concat}([\mathbf{c}_{mix}, \hat{\mathbf{y}}_{0:M}]). \quad (13)$$

The decoder, which consists of multiple convolution layers, processes $\mathbf{e}_2^t$ along with the conditional input $\mathbf{c}$, concatenates them along the variable dimension. Finally, the denoising network produces the denoised output $\hat{\mathbf{y}}_{0:M}^{t-1}$ through a multi-reconstruction process, which involves performing multiple reconstructions with different step values S_i and averaging the results:

$$\hat{\mathbf{y}}_{0:M}^{t-1} = \frac{1}{S} \sum_{i=1}^S \left(\frac{\sqrt{\alpha_t} (1 - \alpha^{t-1})}{1 - \alpha^t} \mathbf{y}_{0:M}^t + \frac{\sqrt{\alpha^{t-1}} \beta_t}{1 - \alpha_t} \mathbf{y}_\theta(\mathbf{y}_{0:M}^t, t, S_i | \mathbf{c}) + \sigma_t \epsilon_i \right) \quad (14)$$

where $\mathbf{c}$ is obtained using Eq.(13). In our implementation, S_i was randomly selected 10 times within a range of 20-40 steps.

3) MULTI-RECONSTRUCTION TRAINING OBJECTIVE

To ensure robustness and stability in the denoising process, we adopted a multi-reconstruction training strategy. Unlike standard diffusion-based approaches that perform denoising in a single step, our approach averages the multi-step denoising results obtained with different steps S_i . The training objective minimizes the discrepancy between the true data

and multi-reconstruction output:

$$\mathcal{L}(\theta) = \min_{\theta} \mathbb{E} \left[\left\| \mathbf{y}_{0:M}^0 - \frac{1}{S} \sum_{i=1}^S \mathbf{y}_\theta(\mathbf{y}_{0:M}^t, t, S_i | \mathbf{c}) \right\|^2 \right]. \quad (15)$$

Algorithms 1 and 2 outline the training and inference (i.e., testing) procedures of the proposed method, respectively.

IV. EXPERIMENTS**A. EXPERIMENTAL SETUP****1) DATASET**

Our experiment includes five real-world datasets:

- *Exchange dataset* [7]: The exchange dataset contains daily exchange rate records for eight different countries, spanning 1990 to 2016, providing a comprehensive view of currency fluctuations over this period.
- *ILI dataset*: This dataset contains the weekly ratio of influenza-like illness (ILI) patients to the total number of patients, as reported by the United States Centers for Disease Control and Prevention (CDC) from 2002 to 2021.
- *ETT dataset (ETTm2)* [12]: This dataset contains seven factors related to electricity transformers, covering the period July 2016 to July 2018. ETTm2 was recorded at 15 minute intervals. Each time step included six power load features and oil temperature measurements.
- *Weather dataset* [13]: This dataset consisted of 21 meteorological indicators collected every 10 min from the Weather Station of the Max Planck Biogeochemistry Institute in 2020.
- *Electricity dataset* [13]: This dataset records the hourly electricity consumption of 321 clients from 2012 to 2014.

Each dataset was divided into training, validation, and test sets following the protocols in [13] and [14]. The detailed characteristics of each dataset are summarized in Table 1.

2) IMPLEMENTATION DETAILS

In our experiments, we set the number of diffusion steps $T = 100$ and implemented a cosine variance schedule [22], starting from $\beta_1 = 10^{-4}$ to $\beta_t = 10^{-1}$. The batch size was set to 64 and training was conducted with early stopping for a maximum of 100 epochs. The data were pre-processed using instance normalization. Reference [35] specifically, for each variable, the mean of the look-back window was subtracted and the result was divided by the standard deviation. During inference, denormalization was applied after the final prediction was obtained. The model was trained using the AdamW optimizer [36] with a learning rate of 10^{-3} . We adopt DPM-Solver [37] as our inference sampler, which allows for fewer denoising steps compared to ancestral samplers, significantly reducing inference latency while maintaining accuracy. All experiments were repeated 10 times, reporting the 10 times, reporting the mean and standard deviation. We have released the complete source code

TABLE 2. Quantitative comparison of diffusion-based models (D), Transformer-based models (T), and our method in terms of MSE and MAE across five real-world datasets. The best performance is boldfaced, while the second-best is underlined. Ours^{NST} denotes our method integrated with NSTransformer, and Ours^I denotes integration with Informer.

Dataset	Exchange		ILI		ETTm2		Electricity		Weather	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeGrad (D)[22]	2.43±0.229	0.90±0.232	2.65±0.164	1.15±0.172	1.36±0.133	0.74±0.123	0.69±0.188	0.74±0.109	0.90±0.139	0.57±0.136
CSDI (D)[30]	1.67±0.162	0.75±0.058	2.54±0.098	1.21±0.128	1.28±0.074	0.67±0.064	0.56±0.212	0.81±0.150	0.86±0.073	0.56±0.096
SSSD (D)[31]	0.90±0.171	0.86±0.127	2.52±0.118	1.08±0.131	0.97±0.043	0.56±0.060	0.47±0.129	0.60±0.207	0.67±0.159	0.49±0.106
D ³ VAE (D)[34]	0.76±0.118	0.62±0.108	2.44±0.115	1.11±0.127	0.79±0.038	0.46±0.047	0.33±0.194	0.49±0.119	0.43±0.139	0.34±0.131
TimeDiff (D)[23]	0.48±0.095	0.43±0.109	2.46±0.148	1.09±0.064	0.41±0.014	0.42±0.013	0.27±0.024	0.32±0.131	0.36±0.146	0.37±0.052
TMDM (D)[24]	0.26±0.019	0.37±0.015	1.99±0.085	<u>0.85±0.026</u>	0.27±0.023	0.35±0.015	0.19±0.007	<u>0.27±0.008</u>	0.28±0.095	<u>0.25±0.103</u>
Transformer (T)[11]	1.20±0.129	0.84±0.041	4.93±0.277	1.48±0.082	4.76±1.214	1.76±0.256	0.26±0.016	0.36±0.015	0.57±0.045	0.53±0.024
Informer (T)[12]	1.31±0.154	0.85±0.021	5.33±0.177	1.59±0.078	5.74±0.475	1.99±0.181	0.35±0.012	0.43±0.020	0.48±0.078	0.47±0.045
Autoformer (T)[13]	0.44±0.146	0.48±0.082	3.26±0.180	1.25±0.066	0.47±0.032	0.46±0.021	0.22±0.019	0.33±0.015	0.31±0.018	0.37±0.025
NSTransformer (T)[19]	0.25±0.088	0.36±0.091	1.93±0.157	0.87±0.058	0.53±0.042	0.48±0.016	0.18±0.012	0.28±0.012	0.25±0.016	0.29±0.020
Ours ^{NST}	<u>0.21±0.004</u>	<u>0.33±0.004</u>	<u>1.90±0.046</u>	0.86±0.002	<u>0.25±0.006</u>	<u>0.31±0.005</u>	<u>0.18±0.001</u>	0.29±0.002	<u>0.23±0.005</u>	0.28±0.001
Ours ^I	0.17±0.069	0.30±0.067	1.50±0.157	0.78±0.038	0.01±0.005	0.08±0.016	0.09±0.004	0.23±0.004	0.006±0.002	0.05±0.009

for reproducibility at <https://github.com/anonymous1123-n/MTDI.git>

B. COMPARISONS

1) BASELINES

We extensively compared our model with 10 baselines, including diffusion-based models: Timegrad [22], CSDI [30], SSSD [31], D³VAE [34], TimeDiff [23], and TMDM [24], and transformer-based models: Transformer [11], Informer [12], Autoformer [13], and NSTransformer [19].

2) QUANTITATIVE COMPARISON

To evaluate the performance of our model, we applied it to five datasets and compared it to several competitive baselines. The mean squared error (MSE) and mean absolute error (MAE) were used as evaluation metrics. As shown in Table 2, Informer [12] integrated with our framework (Ours^I) achieved state-of-the-art (SOTA) performance in all benchmark datasets. We observe that our method achieves significant improvements on high-resolution datasets, such as ETTm2, Electricity, and Weather, which exhibit both strong global seasonality and are pronounced. This performance can be attributed to the ability of our architecture to combine the long-term modeling capacity of the pre-trained transformer with local refinement through the diffusion mechanism. In contrast, Exchange and ILI have lower input dimensionality and fewer time steps, with less complex temporal dynamics. Consequently, the performance margins between the models tend to be smaller on these datasets. Furthermore, the Electricity dataset (321 channels) poses substantial computational challenges because of its high dimensionality. Traditional diffusion models often struggle in

such settings because they operate repeatedly on raw input sequences. However, our framework mitigates this overhead by conditioning the diffusion process on Transformer-derived global representations, rather than applying attention or denoising directly to all input channels.

Our model consistently improves the performance across diverse transformer-based backbones, including Informer [12], Autoformer [13], and NSTransformer [19], as shown in Table 3. Although TMDM improves forecasting by stabilizing the training through the integration of transformer and Diffusion models, our approach further enhances robustness by mitigating error propagation from the Transformer via *future-mixup* and refining local patterns through *multi-denoising*.

Notably, the largest performance gain was observed when integrated with Informer. This supports our hypothesis that Informer's sparse query mechanism, which emphasizes long-range dependencies, provides effective global guidance to the diffusion process. This global-local separation enhances local pattern modeling during reconstruction. These findings validate the generalizability and adaptability of our framework across various Transformer architectures.

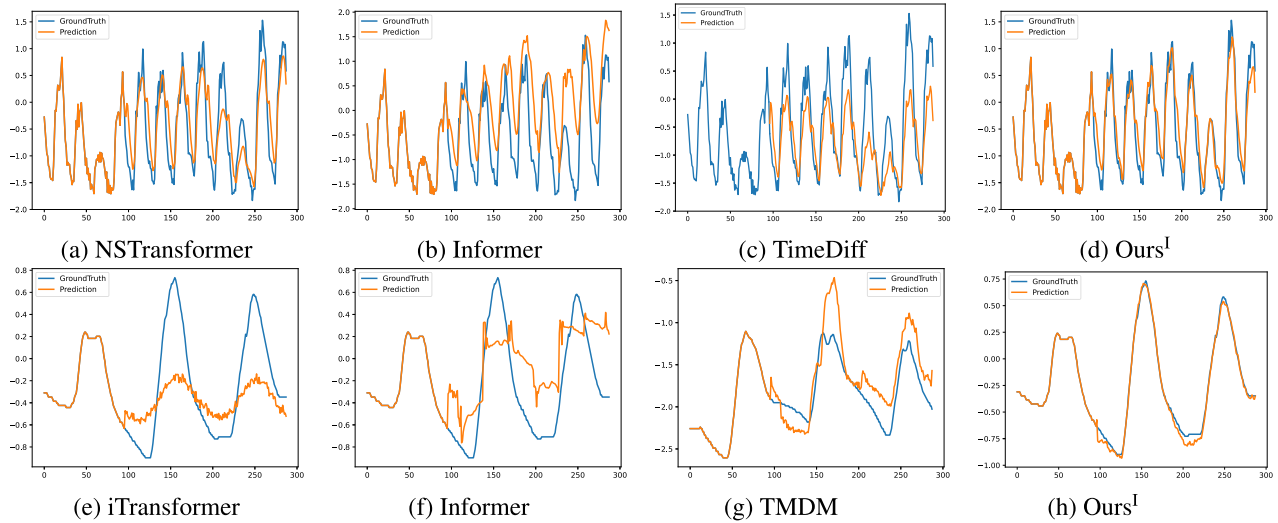
3) QUALITATIVE COMPARISON

Figure 4 presents a qualitative comparison of the NSTransformer [19], Informer [12], iTransformer [16], TimeDiff [23], TMDM [24], and our model on the Electricity and ETTm2 datasets. The first row corresponds to the Electricity dataset, whereas the second row illustrates the results for ETTm2. TMDM was built on the NSTransformer, whereas our model was based on Informer.

As observed in the first row, the comparison models effectively captured overall temporal patterns but failed to accurately model fine-grained local variations, particularly

TABLE 3. Consistent performance improvements when integrating our method with various transformer-based models.

Dataset	Exchange		ETTM2		Electricity	
Method	MSE	MAE	MSE	MAE	MSE	MAE
Informer	1.31 ± 0.154	0.85 ± 0.021	5.74 ± 0.475	1.99 ± 0.181	0.35 ± 0.012	0.43 ± 0.020
+ TMDM	1.12 ± 0.085	0.83 ± 0.014	1.25 ± 0.264	0.90 ± 0.096	0.34 ± 0.010	0.40 ± 0.013
+ Ours	0.17 ± 0.069	0.30 ± 0.067	0.01 ± 0.05	0.08 ± 0.016	0.09 ± 0.004	0.23 ± 0.004
Autoformer	0.44 ± 0.146	0.48 ± 0.082	0.47 ± 0.032	0.46 ± 0.021	0.22 ± 0.019	0.33 ± 0.015
+ TMDM	0.43 ± 0.027	0.47 ± 0.016	0.33 ± 0.028	0.40 ± 0.018	0.20 ± 0.009	0.29 ± 0.010
+ Ours	0.20 ± 0.003	0.32 ± 0.002	0.26 ± 0.001	0.32 ± 0.000	0.22 ± 0.004	0.31 ± 0.003
NSTransformer	0.25 ± 0.088	0.36 ± 0.091	0.53 ± 0.042	0.48 ± 0.016	0.18 ± 0.012	0.28 ± 0.012
+ TMDM	0.26 ± 0.019	0.37 ± 0.015	0.27 ± 0.023	0.35 ± 0.015	0.19 ± 0.007	0.27 ± 0.008
+ Ours	0.21 ± 0.004	0.33 ± 0.004	0.25 ± 0.006	0.31 ± 0.005	0.18 ± 0.001	0.29 ± 0.002

**FIGURE 4.** Qualitative comparison on the electricity dataset. (a) NSTransformer [19], (b) iTransformer [16], (c) TMDM [24] and the proposed method.

in the upper regions of the time series. In contrast, the proposed model demonstrates a superior ability to preserve such local patterns, resulting in more precise forecasts. Similarly, the second row of Figure 4 shows that even iTransformer, the current state-of-the-art Transformer-based model, struggles to accurately predict certain patterns in the ETTm2 dataset. However, our model exhibited a significantly improved prediction capability, closely aligned with the truth of the ground in challenging regions. These results highlight the effectiveness of our approach in capturing both global and local temporal patterns. While the overall improvements in MSE/MAE in Table 2 may appear modest, the qualitative differences in Figure 4 reflect the model's ability to correct localized deviations and capture fine-grained patterns. Such local accuracy improvements are particularly valuable in real-world scenarios where rapid changes may have significant implications.

C. ABLATION STUDY

1) EFFECT OF CONDITIONING STRATEGIES

To evaluate the effectiveness of our model, we conducted an ablation study by comparing three model variants:

(1) the baseline NSTransformer, (2) the NSTransformer with our model architecture incorporating future-mixup, and (3) the NSTransformer with our model architecture integrating both future-mixup and multi-denoising. Table 5 presents the results. As shown in the table, incorporating future-mixup improves performance across all datasets compared to the baseline NSTransformer, demonstrating its effectiveness in mitigating propagation errors and enhancing forecasting robustness. Furthermore, the integration of multi-denoising leads to additional performance gains, achieving the best results across all datasets. This improvement can be attributed to the ability of multi-step denoising to enable fine-grained restoration of temporal structures, preserving local patterns while maintaining global patterns. These findings highlight the importance of both future-mixup and multi-denoising in improving the model's predictive accuracy and generalization ability.

To further analyze the contribution of each component, we conducted two additional sensitivity experiments on Exchange dataset: one varying the number of denoising steps, and the other varying the mixup ratio. First, we evaluated the performance of our model under different denoising

TABLE 4. Ablation study on the number of trainable parameters and inference time (in ms) of various time series diffusion models with different prediction horizons (H) on the univariate ETTh1.

Model	# of trainable params	inference time(ms)		
		$H = 96$	$H = 168$	$H = 192$
Ours [†]	3.4M	302.8	316.0	323.5
TimeDiff	1.7M	16.2	17.3	17.6
TimeGrad	3.1M	870.2	1620.9	1854.5
CSDI	10M	90.4	128.3	142.8
SSSD	32M	418.6	590.2	645.4
TMDM	9.3M	462.8	603.2	646.8

TABLE 5. Ablation study on the contributions of future-mixup and multi-denoising in our method. All experiments are conducted using ours^{NST}, which integrates our framework with NSTransformer.

Dataset	-	+Future-Mixup	+Multi-Denoising
Exchange	0.26	0.22	0.21
ILI	1.99	1.94	1.84
ETTh2	0.27	0.26	0.25

TABLE 6. Impact of number of denoising steps (Mixup ratio = 0.9).

Denoising Steps S	MSE	MAE
10	0.208	0.345
20	0.203	0.338
30	0.158	0.297
40	0.085	0.222

steps $S \in \{10, 20, 30, 40\}$ while fixing the mixup ratio to 0.9. As shown in Table 6, performance improves significantly as the number of denoising steps increases. This highlights the importance of multi-step denoising, which enables progressive refinement and better restoration of fine-grained temporal dynamics. Second, we fixed the number of denoising steps to 30 and varied the mixup ratio from 0.0 to 0.9. The results in Table 7 demonstrate that the use of randomized future mixup improves generalization and robustness. Particularly, the best performance was achieved with a mixup ratio of 0.9, suggesting that random mixup introduces sufficient stochasticity for robust denoising. These results empirically justify our choice to adopt multi-step denoising and randomized future mixup, showing their complementary strengths in enhancing predictive accuracy and temporal consistency.

2) INFERENCE EFFICIENCY

Table 4 summarizes the number of trainable parameters and inference times for the various models across different prediction horizons (H) on the univariate ETTh1 dataset. Although our model has more parameters and a slower inference speed than TimeDiff, it delivers significantly better forecasting accuracy, as shown in Table 8. Moreover, compared to other diffusion-based models such as CSDI, SSSD, and TMDM, our model requires substantially fewer trainable parameters while maintaining superior efficiency. Although the inference speed is slower than that of TimeDiff and CSDI, it is faster

TABLE 7. Impact of mixup ratio (Denoising steps = 30).

Mixup Ratio	MSE	MAE
0.0	0.247	0.367
0.3	0.250	0.377
0.5	0.236	0.365
0.7	0.241	0.364
0.9	0.074	0.209

TABLE 8. Forecasting performance on the exchange dataset across varying horizons ($H=96$ to $H=720$). Results marked with “-” indicate out-of-memory (OOM) errors during evaluation.

Horizon	$H=96$		$H=192$		$H=336$		$H=720$	
Model	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeDiff	7.73	0.36	0.29	0.39	0.48	0.52	0.64	0.62
TMDM	0.12	0.25	0.26	0.37	0.77	0.68	-	-
Ours [†]	0.05	0.16	0.17	0.25	0.23	0.46	0.50	0.56

than that of TMDM and TimeGrad. Thus, this improvement was achieved through a non-autoregressive structure that eliminates step-by-step decoding, as required in TimeGrad, and the integration of DPM-Solver [37], which reduces the number of denoising steps.

3) SCALABILITY ANALYSIS

To evaluate the scalability of our method, we evaluated it on the Exchange dataset across multiple forecasting horizons ($H=96, 192, 336, 720$). As shown in Table 8, our method consistently outperformed TimeDiff and the TMDM across all settings. In the long-horizon scenarios ($H=336$ and $H=720$), our method achieved substantial gains, whereas TMDM fails at $H=720$ because of out-of-memory (OOM) issues. This result highlights the superior memory efficiency and scalability of our framework, particularly under high-dimensional and long-horizon conditions. Additionally, even in shorter horizons (e.g., $H=96$), our method achieves lower MSE and MAE values, indicating that it does not compromise the short-term accuracy for scalability. These findings validate the robustness of the proposed design across diverse forecasting lengths and input complexities.

4) DISTRIBUTIONAL SHIFT

To evaluate the robustness of our model under real-world distributional shift scenarios, we conducted a cross-domain

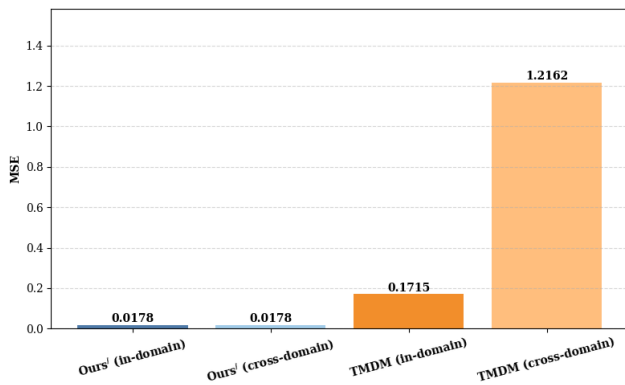


FIGURE 5. Comparison of MSE across different model settings under in-domain and cross-domain scenarios. The proposed model (Ours^I) consistently outperforms TMDM with significantly lower MSE in both domains, demonstrating its robustness and generalization capability.

generalization experiment. Specifically, we trained all models exclusively on the ETTm1 dataset and evaluated them on the ETTm2 dataset without fine-tuning. Although ETTm1 and ETTm2 share similar periodic patterns and overall signal characteristics, they differ slightly in their sampling intervals and local dynamics. As shown in Figure 5, the TMDM exhibits a substantial performance drop when transitioning from in-domain to cross-domain, indicating its vulnerability to even minor domain shifts. In contrast, our model (Ours^I) maintained a consistently low MSE in both domains, suggesting a stronger generalization capability. These results confirm that our framework is better equipped to handle distributional shifts, particularly in time series with closely related structures.

V. CONCLUSION AND DISCUSSION

In this study, we propose a novel methodology that integrates the transformer's ability to capture long-term dependencies with the strength of the diffusion model to capture fine-grained local patterns. To achieve this, we condition the diffusion process on the transformer's conditional mean estimation to ensure that global trends are effectively incorporated. Additionally, we employ the future-mixup strategy to correct errors in the transformer's predictions, thereby enhancing the reliability of the conditional information. Furthermore, we introduce a multi-denoising approach that performs multiple denoising steps with varying step sizes to refine the reconstruction quality and better preserve local structures.

Through this approach, we successfully preserve local features that have been underestimated in traditional prediction-focused time-series research while simultaneously improving forecasting performance. Our method demonstrated superior performance compared to baseline models, demonstrating its effectiveness in time-series forecasting. However, owing to the structural limitations of the diffusion model, the inference process incurs high computational costs and time overhead. Addressing this issue by introducing a lightweight

diffusion model is important for future research. Despite these limitations, our study makes a significant contribution by proposing an effective integration of transformers and diffusion models, while emphasizing the importance of all feature information in time-series analysis to enhance performance.

REFERENCES

- [1] K. Mendis, M. Wickramasinghe, and P. Marasinghe, "Multivariate time series forecasting: A review," in *Proc. ACCV*, 2024, pp. 1–9.
- [2] X. Liu and W. Wang, "Deep time series forecasting models: A comprehensive survey," *Mathematics*, vol. 12, no. 10, p. 1504, May 2024.
- [3] M. Mudelsee, "Trend analysis of climate time series: A review of methods," *Earth-Sci. Rev.*, vol. 190, pp. 310–322, Mar. 2019.
- [4] I. Dimoulkas, P. Mazidi, and L. Herre, "Neural networks for GEFCom2017 probabilistic load forecasting," *Int. J. Forecasting*, vol. 35, no. 4, pp. 1409–1423, Oct. 2019.
- [5] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, N. Q. Tran, and B. Seaman, "Sales demand forecast in e-commerce using a long short-term memory neural network methodology," in *Proc. ICONIP*, 2019, pp. 462–474.
- [6] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 156–165.
- [7] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 95–104.
- [8] S. S. Rangapuram, M. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," in *Proc. NIPS*, vol. 31, 2018, pp. 7785–7794.
- [9] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," in *Proc. NIPS*, 2019, pp. 4652–4663.
- [10] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1181–1191, Jul. 2020.
- [11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, vol. 30, 2017, pp. 5998–6008.
- [12] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI*, vol. 35, 2021, pp. 11106–11115.
- [13] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. NeurIPS*, 2021, pp. 22419–22430.
- [14] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposition transformer for long-term series forecasting," in *Proc. ICML*, 2022, pp. 27268–27286.
- [15] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," 2022, *arXiv:2211.14730*.
- [16] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "ITransformer: Inverted transformers are effective for time series forecasting," 2023, *arXiv:2310.06625*.
- [17] X. Xu, C. Chen, Y. Liang, B. Huang, G. Bai, L. Zhao, and K. Shu, "SST: Multi-scale hybrid mamba-transformer experts for long-short range time series forecasting," 2024, *arXiv:2404.14757*.
- [18] R. Kruger, A. Mueen, and V. M. A. Souza, "Peak prediction in time series: Comparing approaches for energy high-load prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2024, pp. 1–8.
- [19] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," in *Proc. NeurIPS*, 2022, pp. 9881–9893.
- [20] J. Ho, A. N. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NIPS*, 2020, pp. 6840–6851.
- [21] L. Shen, W. Chen, and J. Kwok, "Multi-resolution diffusion models for time series forecasting," in *Proc. ICLR*, 2024.
- [22] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *Proc. ICML*, 2021, pp. 8857–8868.

- [23] L. Shen and J. T. Kwok, "Non-autoregressive conditional diffusion models for time series prediction," in *Proc. ICML*, 2023, pp. 31016–31029.
- [24] Y. Li, W. Chen, X. Hu, B. Chen, and M. Zhou, "Transformer-modulated diffusion models for probabilistic multivariate time series forecasting," in *Proc. ICLR*, 2024.
- [25] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2018, pp. 4171–4186.
- [26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [27] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *Proc. ICLR*, 2023.
- [28] S. Feng, C. Miao, Z. Zhang, and P. Zhao, "Latent diffusion transformer for probabilistic time series forecasting," in *Proc. AAAI*, vol. 38, 2024, pp. 11979–11987.
- [29] D. Cao, W. Ye, Y. Zhang, and Y. Liu, "TimeDiT: General-purpose diffusion transformers for time series foundation model," 2024, *arXiv:2409.02322*.
- [30] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "CSDI: Conditional score-based diffusion models for probabilistic time series imputation," in *Proc. NeurIPS*, 2021, pp. 24804–24816.
- [31] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured state space models," 2022, *arXiv:2208.09399*.
- [32] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.
- [33] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. NIPS*, 2019, pp. 6245–6255.
- [34] Y. Li, X. Lu, Y. Wang, and D. Dou, "Generative time series forecasting with diffusion, denoise, and disentanglement," in *Proc. NeurIPS*, 2023, pp. 23009–23022.
- [35] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *Proc. ICLR*, 2021.
- [36] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
- [37] C. Lu, Y. Zhou, F. Bao, J. F. Chen, C. Li, and J. Zhu, "DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps," in *Proc. NeurIPS*, 2022, pp. 5775–5787.



DAGYEONG NA received the B.Sc. degree from Chung-Ang University, South Korea, in 2024, where she is currently pursuing the M.S. degree in artificial intelligence. Her research interests include time series forecasting and anomaly detection.



JINHO KANG received the B.S. degree in computer science and engineering from the University of Seoul, in 2022, where he is currently pursuing the M.S. degree in artificial intelligence.



BYOUNGWOONG KANG received the B.S. degree in financial engineering from Hanyang University, Seoul, South Korea, in 2023, and the M.S. degree in artificial intelligence from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2025. He is currently an AI Engineer with KT Corporation. His research interests include deep learning, time series forecasting, and tabular data analysis.



JUNSEOK KWON (Member, IEEE) received the B.Sc. degree, the M.Sc. degree in object tracking (supervised by Prof. Kyoung Mu Lee), and the Ph.D. degree in electrical engineering and computer science from Seoul National University, South Korea, in 2006, 2008, and 2013, respectively. He was a Postdoctoral Researcher with the Computer Vision Laboratory, ETH Zurich, from 2013 to 2014, under the supervision of Prof. Luc Van Gool. He is currently an Associate Professor with the School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea. He is working in the field of object tracking to capture the dynamics of cities. His research interests include visual tracking, visual surveillance, and Monte Carlo sampling method and its variants.

...