IIDiff: Learning Cross-Domain Frequency Transitions with Diffusion Mixture-of-Experts

Xin Xue, Yirong Xue, Xingcheng Fu, Yisen Gao, Lanhao Li, Lijun Sun, Yunhao Fu, Haoyi Zhou *
Beihang University, Beijing 100191, China
Guangxi Normal University, Guangxi 541006, China
The Hong Kong University of Science and Technology, Hong Kong 999077, China
haoyi@buaa.edu.cn

Abstract

Diffusion models have shown strong performance in time-series generation, while joint generation across heterogeneous domains remains challenging because their spectral recovery trajectories evolve during denoising and vary across domains. Existing frequency-aware Mixture-of-Experts (MoE) models provide a promising basis for specialization, but integrating them with diffusion models requires addressing two coupled challenges. For intra-domain aspect, the absence of frequency transition compliance leads to imbalance pattern learning across denoising steps and degrade model performance. For the inter-domain aspect, the domain-specific transition variation bring difficulties for domain adaptation. To address these challenges, We propose IIDiff, a frequency-guided diffusion MoE framework that couples progressive frequency specialization with dynamic transition matching. Progressive frequency specialization organizes experts with different effective bandwidths to model intra-domain spectral recovery. Dynamic transition matching extracts a lightweight statistic from the current spectrum to coordinate experts according to inter-domain transition variation. Experiments on 12 real-world datasets across 4 domains show competitive results, confirming its strong domain adaptability and ability to generate high-fidelity cross-domain time-series.

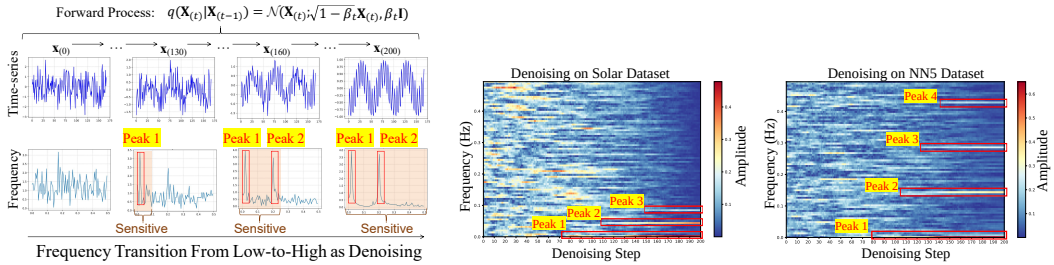
1 Introduction

In recent years, diffusion models have emerged as a dominant generative paradigm in time-series generation [10, 26, 17]. Their ability to model complex temporal dynamics with high fidelity has driven significant advances in various time-series tasks, from forecasting to imputation [7, 1]. However, existing diffusion models still face challenges in handling cross-domain generation. Cross-domain generation refers to using the same model to learn and generate time-series of different domains [10, 5]. As a result, it requires robust domain adaptability, making it remains a challenge for existing diffusion models.

A promising direction for improving domain adaptability is to integrate the Mixture-of-Experts (MoE) framework. Diffusion-based MoE models have demonstrated enhanced effectiveness and adaptability in cross-domain settings, with successful applications in computer vision [23] and natural language processing [27]. These approaches further show that MoE architectures can facilitate knowledge specialization across different domains [24, 14]. For time-series data, frequency signals plays a crucial role in maintaining domain-specific latent patterns, motivating the development of frequency-based MoE methods [16, 15]. However, integrating MoE into diffusion models remains challenging. During denoising, domains-specific frequency transition dynamics makes it difficult for

*Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Intra-domain: Frequency transition as denoising.

(b) Inter-domain: Transition speed differs across domains.

Figure 1: Motivating frequency trajectories. **(a) Intra-domain Frequency Transition Compliance.** The sensitive frequency during denoising shows a transition from low frequency to high, which causes imbalanced frequency pattern learning across denoising steps and degrades model performance, calling for transition compliance mechanism. **(b) Inter-domain Domain-wise Transition Adaptation.** The transition speed varies across diverse domains. Diffusion model transition to the frequency component near 0.1 Hz for solar data at about 150 steps (Peak 3 in the left), but earlier for NN5 data at about 100 steps (Peak 2 in the right), demonstrating the requirement for domain adaptation.

routing mechanisms to reliably allocate experts, resulting in **imbalanced** frequency learning **within domains** and **misaligned** expert transitions **across domains**.

For time-series generation in across domains scenarios, it is crucial to consider intra- and inter-domain issues simultaneously. **For the intra-domain aspect**, existing methods [2, 5] fail to account for inherent frequency transition in diffusion processes, causing imbalanced frequency pattern learning across different denoising steps, thereby degrading model performance. As visualized in Fig. 1(a), diffusion models recover time-series signals in a low-to-high frequency order during denoising, which means models are more sensitive in low-frequency components in early stages and shift to high-frequency details gradually. This leads to an imbalanced learning of high-frequency and low-frequency signals, resulting in underperformance without proper ***transition compliance*** approaches. **For the inter-domain aspect**, static routing mechanism ignores that the transition speed differs across domains, causing poor domain adaptability. As the instance shown in Fig. 1(b), solar data with concentrated low-frequency components recover of frequency peak near to 0.1 Hz (peak 3) at about 150 denoising steps, while NN5 data with distributed frequency ranges start recovering the peak near to 0.1 Hz (peak 2) at about 100 steps, which is much earlier than solar data. Existing MoEs lack mechanisms to quantify domain-specific transition variation for ***domain adaptation***, resulting in misleading routing and poor cross-domain generalization.

These interwoven challenges collectively led to complex and variable frequency transition problems in the denoising process of cross-domain data, which became the key bottleneck for achieving the effectiveness of cross-domain time series generation. To address these challenges, we propose **IIDiff**, a novel and well-motivated framework designed to adapt **I**ntra- and **I**nter-domain frequency transition in **Diff**usion model. It's core innovations lie in two complementary mechanisms, namely the progressive frequency specializing and dynamic transition matching. The progressive frequency specializing allows experts specialize in different frequency bands and be activated progressively across denoising steps, faithfully following the natural low-to-high frequency recovery order of diffusion, avoiding effectively alleviating imbalanced training caused by uncoordinated frequency sensitivity. The dynamic transition matching extracts a compact Gaussian-inspired statistic from the current spectrum, which indirectly models domain-specific transition differs, and uses it to align the transition speed across domains and coordinate experts for adaptive generalization without domain-specific heuristics. Together, the two components connect expert specialization and routing to the evolving spectral state of the generated time-series as denoising.

In summary, our contributions are threefold:

- **Frequency-transition perspective.** We formulate cross-domain time-series diffusion from the perspective of spectral recovery, distinguishing the evolution of spectral content during denoising from its variation across domains.

- **Transition-aware diffusion MoE.** We introduce IIDiff by coupling progressive frequency specialization with dynamic transition matching, enabling expert coordination for intra-domain spectral recovery and inter-domain transition variation.
- **Multi-level empirical analysis.** Extensive experiments across 12 datasets show the effectiveness of IIDiff and provide empirical support for the proposed frequency-transition perspective. The results demonstrate overall improvements in distributional and spectral fidelity, while mechanism analyses further support recovery-conditioned multi-scale modeling.

2 Related Works

Cross-domain time-series generation demands strong domain adaptability, involving issues within and across domains. This section reviews the relevant progress and identifies critical research gaps.

Mixture-of-Experts for time-series. Mixture-of-Experts (MoE) methods have gained widespread adoption in time-series analysis due to their ability to decompose complex tasks into specialized subtasks, enabling effective modeling via multiple expert networks. Recent studies have enhanced MoE's performance by incorporating domain-specific characteristics into routing mechanisms. Moirai-MoE [14] uses learned token and cluster routing with sparse transformer feed-forward experts during foundation pretraining, while Time-MoE similarly applies sparse experts to forecasting [20]. Their routing is designed for temporal representations rather than the evolving spectrum of a noisy generative state. From another perspective, frequency-based MoE approaches stand out as a prominent branch. By decomposing the time-series according to frequency patterns, the frequency-based MOE methods, such as FreqMoE [16] and MoFE-Time [15], facilitate flexible integration with frequency information with multiple experts. Other methods decompose tasks from the perspective of variables [9, 19], which focuses on modeling domain correlations, catering to diverse scenarios. However, existing MoE methods fail to integrate the generative models well. The core gap lies in reconciling the static frequency decomposition with the dynamic frequency transition in diffusion processes, which is a challenge our work aims to resolve.

Diffusion Methods for Time-series. Denoising diffusion probabilistic models have emerged as a powerful generative paradigm, achieving significant breakthroughs in time-series analysis, including forecasting [7, 12],imputation [1, 21], anomaly detection [22] and generation [4, 10, 26]. Notably, diffusion models, such as TimeDP [10], Diffusion-TS [26] and Winformer [3] have demonstrated promising results in cross-domain time-series generation. However, existing methods do not explicitly coordinate multiple effective spectral states as denoising. FeRA [24] estimates two-dimensional latent band energy and adapts a pretrained text-to-image model through frequency-specific LoRA experts, but it does not model temporal spectral transitions domains. Winformer [3] replaces part of point-wise attention with window-wise alignment to adapt periodic matching across domains, but it does not include an explicit mechanism for estimating the spectral recovery state of the current noisy sample as denoising progresses. These methods lack domain-wise specialization, limiting their ability to capture fine-grained features across diverse domains.

3 Preliminary

In this section, we first formally define the problem of cross-domain time-series generation, and then describe how denoising diffusion probabilistic model solves time-series generation in a regular way.

3.1 Problem Definition

The cross-domain time-series generation is to leverage a unified generative model to conduct conditional time-series generation across diverse domain-specific conditions.

Let $\mathcal{D} = \{D_1, \ldots, D_J\}$ be J datasets, a single prompt-conditioned denoiser is jointly trained on a mixture of all J datasets, rather than training a separate model for each source. Every training example is a complete length-L series $\mathbf{X}_j \in \mathbb{R}^{L \times d}$. Following TimeDP [10], a learned assignment network maps a small set of clean reference series to soft assignments over a frozen orthogonally initialized prototype bank, thereby forming the prompt $\mathbf{C}_j$. At inference, the prompt is constructed

without updating the model parameters. The reference series identify the requested source distribution and are separate from the generated sequences.

Specifically, let $\mathbf{X}_j = \{x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(L)}\}$ be an L-length time-series from the dataset J. The target for cross-domain time-series generation is to generate the a meaningful time-series $\tilde{\mathbf{X}}_j \sim p(\mathbf{X}|\mathbf{C}_j)$ with a domain prompting condition $\mathbf{C}_j$, where $p(\cdot)$ represents the probability function of the real distribution and $\mathbf{X}$ is the united set containing all domains. The main evaluation therefore measures conditional generation for sources included in joint training.

3.2 Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPM) [8] is to utilize the denoising method to progressively remove noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ from the noisy data to form a sample $\tilde{\mathbf{X}}$, which estimates the real data $\mathbf{X}$ under the condition $\mathbf{C}$. During the forward process, by adding the noise step-by-step, we can obtain the noisy data $\mathbf{X}_t$ after t steps from the real data $\mathbf{X}_0 = \mathbf{X}$ as $\mathbf{X}_t = \sqrt{\alpha_t}\mathbf{X}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}$, where $\alpha_t \in (0, 1)$. By recurring the Markov process, we can infer $\mathbf{X}_t = \sqrt{\overline{\alpha}_t}\mathbf{X}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_t$, where $\overline{\alpha}_t = \Pi_{i=1}^{t}\alpha_i$. Considering the Bayes' law, the reverse process is to reconstruct the distribution $p(\mathbf{X}_{t-1}|\mathbf{X}_t, \mathbf{C})$ with parameterized estimation as

$$p_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t, \mathbf{C}) = \mathcal{N}\Big(\mathbf{X}_{t-1}; \mu_\theta(\mathbf{X}_t, t, \mathbf{C}), \Sigma\Big) \quad , \tag{1}$$

where θ represents the parameters. During the reverse process, we can estimate the ϵ_t with $\epsilon_\theta(\mathbf{X}_t, t, \mathbf{C})$ by neural networks. Then we get the $\mu_\theta(\mathbf{X}_t, t, \mathbf{C})$ as

$$\mu_\theta(\mathbf{X}_t, t, \mathbf{C}) = \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{X}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(\mathbf{X}_t, t, \mathbf{C})\Big) \quad , \tag{2}$$

with which we can finally calculate the $p_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t, \mathbf{C})$ to generate the $\mathbf{X}_{t-1}$.

4 Method

We propose a frequency-based diffusion method, named IIDiff for cross-domain time-series generation as illustrated in Fig. 2. The model addresses the intra- and inter-domain transition adaptation challenges with progressive frequency specializing and dynamic transition matching, ensuring robust domain adaptability and high-fidelity generation. In this section, we introduce the technical details of each mechanism and their integration into the overall framework.

4.1 Progressive Frequency Specializing

Our motivating trajectories show that the effective spectral content of a sample can change during denoising. Progressive frequency specialization therefore equips the denoiser with multiple smoothing scales for recovery-conditioned modeling, which dynamically adjust experts' frequency reception fields across denoising steps.

We use aggregation windows of varying sizes to control each expert's effective receptive bandwidth. A larger average-pooling window provides stronger smoothing, whereas a smaller window preserves a broader range of detail. Specifically, let $\mathbf{P}^{(i)}$ be the aggregating window size for the i-th unshared expert, with the aggregating operation defined as

$$\mathbf{Y}_t^{(i)} = \text{AvgPool}_{\text{1D}}(\mathbf{Z}_t, \mathbf{P}^{(i)}) \quad , \tag{3}$$

where $\mathbf{Z}_t$ denotes the time-series token sequence at denoising step t. A length-P average pool is a moving-average low-pass filter with response $H_P(\omega) = P^{-1}\sum_{r=0}^{P-1} e^{-\mathrm{i}\omega r}$, where $|H_P(\omega)|$ indicates how strongly frequency ω is retained. Increasing P narrows the principal pass band and emphasizes coarser low-frequency structure, whereas a smaller P preserves a broader frequency range and finer detail. Different window sizes therefore provide ordered but overlapping frequency ranges because the response has side lobes. This parameter-free and inexpensive design is further discussed in Appendix A.

To ensure the frequency receptive fields expand progressively during the denoising process, we equip S unshared experts with strictly decreasing window sizes $\mathbf{P}^{(1)} > \mathbf{P}^{(2)} > \cdots > \mathbf{P}^{(S)}$. During early

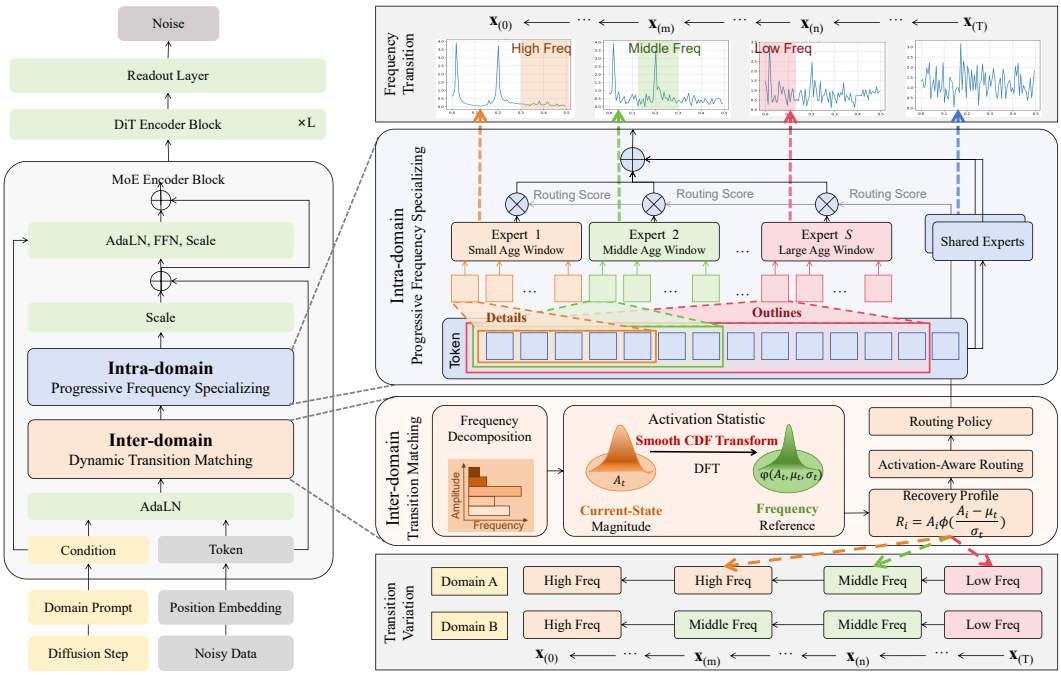


Figure 2: Overall architecture of the IIDiff, addressing the intra- and inter-domain frequency transition misalignment. **Intra-domain:** As the sensitive frequency transfer in low-to-high order, we introduce progressive frequency specializing with experts containing hierarchical aggregating windows, allowing dynamical frequency reception fields adjustment, aligning the denoising process. **Inter-domain:** To adapt the transition variation across domains, we propose a dynamic transition matching to instantaneous monitor the transition process, providing domain-specific routing instructions.

denoising stages, experts with larger windows are assigned higher routing weights, because of their ability to blur high-frequency noise allows the model to focus on recovering core low-frequency outline patterns. As denoising proceeds, the routing weight shifts toward experts with smaller windows, enabling the model to capture fine-grained high-frequency details.

To incorporate conditions, we apply the adaptive layer normalization Ada-LN$(\cdot)$ [18] to the aggregated tokens $\mathbf{Y}_t$, which injects prompting conditions C as

$$\theta_t, \gamma_t = \text{Ada-LN}(\mathbf{Y}_t, \mathbf{C}), \quad \mathbf{Y}_t^{'} = \theta_t \mathbf{Y}_t + \gamma_t \quad . \tag{4}$$

The processed tokens $\mathbf{Y}_t^{'}$ are then fed into attention-based encoders to get the output of the expert. Formally, the j-th unshared expert formulated as

$$\mathcal{E}_{\text{u}}^{(j)} = \text{Attn}(\mathbf{Y}_t^{'}) \quad . \tag{5}$$

Additionally, we retain a small number of shared attention-based experts $\mathcal{E}_{\text{s}}$, which are implemented via vanilla attention, to capture non-stationary temporal patterns that are not frequency-specific. In a summary, the progressive frequency specializing allows adaptive adjusting the receptive fields to capture underlying patterns as sensitive frequency transition during the denoising process.

4.2 Dynamic Transition Matching

From inter-domain aspect, frequency transition speed varies across domains, causing difficulties for cross-domain generation. To address this, we propose dynamic transition matching, which provides a Gaussian-inspired proxy to approximate the reliability of frequency components during the denoising process. It monitors at which stage has the transition currently progressed, enabling adaptive routing across domains.

As stated in Corollary 4.1, the noise in frequency still follows a Gaussian distribution, which is further proved in Appendix A. By linearity, the DFT preserves the additive decomposition of the signal and

noise terms. However, it does not make them identifiable from $\mathbf{X}_t$ alone. The transformed noise term retains Gaussian coefficients under the white-noise assumption, which provides only a statistical cue for routing rather than an exact separation of noise from recovered structure.

Corollary 4.1 (Gaussian noise retains a Gaussian distribution in frequency.)**.** *Let the discrete-time sequence $x(n)$, $n = 0, 1, \ldots, N-1$, be a zero-mean independent and identically distributed (i.i.d.) Gaussian series, i.e., $x(n) \sim \mathcal{N}(0, \sigma^2)$. Applying DFT to $x(n)$ yields the frequency-domain sequence*

$$X(k) = \textit{DFT}\,(x(n)) = \sum_{n=0}^{N-1} x(n) \exp\Big(-j\frac{2\pi}{N} kn \Big) \quad , \tag{6}$$

where $k = 0, 1, \ldots, N-1$. Denote $X(k) = \textit{Re}[X(k)] + j\textit{Im}[X(k)]$, where $\textit{Re}[\cdot]$ and $\textit{Im}[\cdot]$ represent the real part and imaginary part. Then,

$$\textit{Re}[X(k)] \sim \mathcal{N}\left(0, \frac{N\sigma^2}{2}\right) \quad (k \neq 0, N/2) \quad . \tag{7}$$

In particular, when $k = 0$ or $k = N/2$, $\textit{Im}[X(k)] = 0$, and thus $X(k) = \textit{Re}[X(k)] \sim \mathcal{N}(0, N\sigma^2)$.

We extract frequency amplitude features from the noisy data $\mathbf{X}_t$ at denoising step t via DFT. The features $\mathbf{F}_t$ are obtained by

$$\mathbf{a}_t = |\mathrm{DFT}(\mathbf{X}_t)| \in \mathbb{R}^L_{\geq 0}, \tag{8}$$

where $a_{t,\ell}$ denotes the magnitude at frequency position ℓ. The magnitude combines the real and imaginary parts of each coefficient and retains one nonnegative value per frequency position. Because the structured and stochastic contributions overlap in $\mathbf{X}_t$ and cannot be separated exactly, we characterize relative spectral activation. Let $h = \lfloor \rho L \rfloor$, and let $\mathcal{I}_t$ denote the index set of the $L - h$ largest entries of $\mathbf{a}_t$. We fit a Gaussian reference distribution $\mathcal{N}(\mu_t, \sigma_t^2)$ to their amplitudes, where

$$\mu_t = \frac{1}{|\mathcal{I}_t|} \sum_{\ell \in \mathcal{I}_t} a_{t,\ell}, \qquad \sigma_t^2 = \frac{1}{|\mathcal{I}_t| - 1} \sum_{\ell \in \mathcal{I}_t} (a_{t,\ell} - \mu_t)^2 . \tag{9}$$

These statistics define a Gaussian reference distribution $\mathcal{N}(\mu_t, \sigma_t^2)$ over the active magnitudes. We then construct the routing feature

$$S_{t,\ell} = a_{t,\ell}\, \Phi\left(\frac{a_{t,\ell} - \mu_t}{\sigma_t + \varepsilon_{\mathrm{num}}} \right), \qquad \ell = 1, \ldots, L, \tag{10}$$

where Φ is the standard Gaussian cumulative distribution function and $\varepsilon_{\mathrm{num}} > 0$ ensures numerical stability. The Gaussian CDF provides a smooth location-scale weight, while multiplication by $a_{t,\ell}$ retains the relative strength and original position of each component. Accordingly, $\mathbf{S}_t = (S_{t,1}, \ldots, S_{t,L})$ summarizes the strength and locations of spectral activations for routing.

The reference set and its statistics are recomputed for every state, allowing $\mathbf{S}_t$ to vary across samples, time steps, and domains. This construction provides an adaptive coordinate for expert coordination. Additionally, we also discussed about another from of estimation, which estimate the noise to infer the current stage, as reported in Appendix G. The following Routing mechanism maps $\mathbf{S}_t$ to the expert weights.

4.3 Routing Experts with Transition Masking

The final MoE routing policy utilize the dynamic transition matching to instruct the activation of the progressive frequency experts, addressing the intra- and inter-domain transition misalignment.

With the transition masking $\mathbf{S}_t$ for the frequency components $\mathbf{F}_t$, we project the frequency and its masking with a router layer $g_\phi(\cdot)$, which returns the weights of each expert for soft activation as

$$\mathbf{W}_t = \mathrm{Softmax}\Big(g_\phi(\mathbf{F}_t, \mathbf{S}_t) \Big) \in \mathbb{R}^S \quad , \tag{11}$$

where S is the number of the unshared experts. The i-th expert receives weight $\mathbf{W}_t^{(i)}$ for the proportion of this expert's output $\mathcal{E}_{\mathrm{u}}^{(i)}(\cdot;\cdot)$ in the final decision-making. Thus, the output of the unshared experts are collected as the weighted average of the results of these experts. Additionally,

Table 1: Main results for cross-domain generation. Best are **bold** and second best are underlined.

		Ours	TimeDP	Diffusion-TS	TimeGAN	GT-GAN	TimeVAE
Maximum Mean Discrepancy	Electricity	**0.001**$_{\pm 0.001}$	**0.001**$_{\pm 0.001}$	0.003$_{\pm 0.002}$	0.367$_{\pm 0.255}$	0.254$_{\pm 0.166}$	0.577$_{\pm 0.006}$
	Solar	**0.032**$_{\pm 0.005}$	0.041$_{\pm 0.011}$	0.050$_{\pm 0.012}$	0.628$_{\pm 0.053}$	0.578$_{\pm 0.039}$	0.353$_{\pm 0.014}$
	Wind	0.036$_{\pm 0.014}$	**0.025**$_{\pm 0.017}$	0.035$_{\pm 0.006}$	0.213$_{\pm 0.017}$	0.170$_{\pm 0.040}$	0.170$_{\pm 0.004}$
	Traffic	**0.083**$_{\pm 0.010}$	**0.083**$_{\pm 0.034}$	0.111$_{\pm 0.031}$	0.567$_{\pm 0.057}$	0.538$_{\pm 0.078}$	0.218$_{\pm 0.007}$
	Taxi	**0.092**$_{\pm 0.014}$	0.095$_{\pm 0.023}$	0.131$_{\pm 0.014}$	0.275$_{\pm 0.054}$	0.319$_{\pm 0.032}$	0.139$_{\pm 0.007}$
	Pedestrian	**0.037**$_{\pm 0.010}$	0.044$_{\pm 0.020}$	0.071$_{\pm 0.019}$	0.090$_{\pm 0.030}$	0.112$_{\pm 0.019}$	0.065$_{\pm 0.002}$
	Air	0.012$_{\pm 0.001}$	**0.011**$_{\pm 0.003}$	0.022$_{\pm 0.011}$	0.120$_{\pm 0.045}$	0.211$_{\pm 0.041}$	0.089$_{\pm 0.016}$
	Temperature	0.224$_{\pm 0.021}$	**0.219**$_{\pm 0.022}$	0.241$_{\pm 0.049}$	0.926$_{\pm 0.042}$	0.809$_{\pm 0.081}$	1.002$_{\pm 0.014}$
	Rain	**0.035**$_{\pm 0.016}$	0.057$_{\pm 0.039}$	0.079$_{\pm 0.058}$	0.329$_{\pm 0.285}$	0.111$_{\pm 0.109}$	0.292$_{\pm 0.019}$
	NN5	**0.155**$_{\pm 0.008}$	0.164$_{\pm 0.010}$	0.186$_{\pm 0.043}$	0.874$_{\pm 0.088}$	0.632$_{\pm 0.074}$	0.821$_{\pm 0.061}$
	Fred-MD	**0.002**$_{\pm 0.001}$	**0.002**$_{\pm 0.001}$	0.006$_{\pm 0.002}$	0.043$_{\pm 0.021}$	0.133$_{\pm 0.102}$	0.059$_{\pm 0.008}$
	Exchange	**0.137**$_{\pm 0.013}$	0.151$_{\pm 0.024}$	0.206$_{\pm 0.035}$	0.530$_{\pm 0.154}$	0.475$_{\pm 0.116}$	0.543$_{\pm 0.149}$
K-L Divergence	Electricity	**0.006**$_{\pm 0.003}$	0.012$_{\pm 0.016}$	0.315$_{\pm 0.247}$	0.488$_{\pm 0.175}$	0.407$_{\pm 0.079}$	0.734$_{\pm 0.023}$
	Solar	**0.011**$_{\pm 0.004}$	0.016$_{\pm 0.005}$	0.066$_{\pm 0.055}$	0.612$_{\pm 0.447}$	0.120$_{\pm 0.041}$	0.260$_{\pm 0.016}$
	Wind	0.196$_{\pm 0.031}$	0.152$_{\pm 0.034}$	0.548$_{\pm 0.372}$	1.924$_{\pm 1.233}$	**0.107**$_{\pm 0.016}$	0.484$_{\pm 0.015}$
	Traffic	0.013$_{\pm 0.009}$	**0.009**$_{\pm 0.003}$	0.120$_{\pm 0.074}$	1.305$_{\pm 0.320}$	1.409$_{\pm 0.251}$	0.211$_{\pm 0.014}$
	Taxi	**0.004**$_{\pm 0.001}$	0.011$_{\pm 0.004}$	0.075$_{\pm 0.034}$	0.650$_{\pm 0.180}$	0.950$_{\pm 0.197}$	0.110$_{\pm 0.020}$
	Pedestrian	**0.011**$_{\pm 0.011}$	0.014$_{\pm 0.010}$	0.133$_{\pm 0.069}$	0.417$_{\pm 0.181}$	0.411$_{\pm 0.096}$	0.065$_{\pm 0.005}$
	Air	0.029$_{\pm 0.012}$	**0.027**$_{\pm 0.016}$	0.106$_{\pm 0.079}$	0.348$_{\pm 0.093}$	0.578$_{\pm 0.049}$	0.164$_{\pm 0.012}$
	Temperature	0.175$_{\pm 0.028}$	**0.171**$_{\pm 0.073}$	0.342$_{\pm 0.131}$	8.892$_{\pm 2.681}$	3.174$_{\pm 2.685}$	2.183$_{\pm 0.110}$
	Rain	**0.012**$_{\pm 0.007}$	0.013$_{\pm 0.012}$	0.061$_{\pm 0.072}$	0.506$_{\pm 0.174}$	0.432$_{\pm 0.099}$	0.160$_{\pm 0.022}$
	NN5	0.059$_{\pm 0.009}$	**0.054**$_{\pm 0.014}$	0.165$_{\pm 0.076}$	4.928$_{\pm 4.112}$	1.386$_{\pm 0.520}$	1.337$_{\pm 0.220}$
	Fred-MD	**0.201**$_{\pm 0.027}$	0.203$_{\pm 0.035}$	0.835$_{\pm 0.554}$	0.512$_{\pm 0.290}$	0.380$_{\pm 0.070}$	0.346$_{\pm 0.041}$
	Exchange	**1.641**$_{\pm 0.137}$	1.866$_{\pm 0.132}$	2.337$_{\pm 0.714}$	8.861$_{\pm 3.397}$	7.201$_{\pm 4.380}$	10.404$_{\pm 1.434}$

there are shared experts allowing the capturing of the temporal non-stationarity in time-series. The final result is the combination of the unshared experts and N shared experts $\mathcal{E}_{\text{s}}(\cdot;\cdot)$. Thus, the results of the MoE mechanism is

$$\mathcal{E}(\mathbf{Z}_t;\mathbf{C}) = \sum_{i=1}^{S} \mathbf{W}_t^{(i)} \mathcal{E}_{\text{u}}^{(i)}(\mathbf{Z}_t;\mathbf{C}) + \sum_{j=1}^{N} \mathcal{E}_{\text{s}}^{(j)}(\mathbf{Z}_t;\mathbf{C}) \quad . \tag{12}$$

where $\mathbf{Z}_t$ are embedded tokens from the noisy data $\mathbf{X}_t$ and $\mathbf{C}$ is the domain prompting condition.

Finally, a conventional denoising processing is performed. With attention-based encoder layer $\text{Enc}(\cdot;\cdot)$ and MLP-based readout layer $\text{Rd}(\cdot)$, the predicted temporal noise $\epsilon_\theta(\mathbf{X}_t, t)$ is achieved by

$$\epsilon_\theta(\mathbf{X}_t, t, \mathbf{C}) = \text{Rd}\Big(\text{Enc}\big(\mathcal{E}(\mathbf{Z}_t;\mathbf{C});\mathbf{C}\big)\Big) \quad . \tag{13}$$

As in standard diffusion training, the parameters minimize an entrywise noise-prediction loss

$$\mathcal{L}_p = \mathbb{E} \sum_{a,b} \Big| [\epsilon_t - \epsilon_\theta(\mathbf{X}_t, t, \mathbf{C})]_{a,b} \Big|^p, \qquad p \in \{1, 2\}. \tag{14}$$

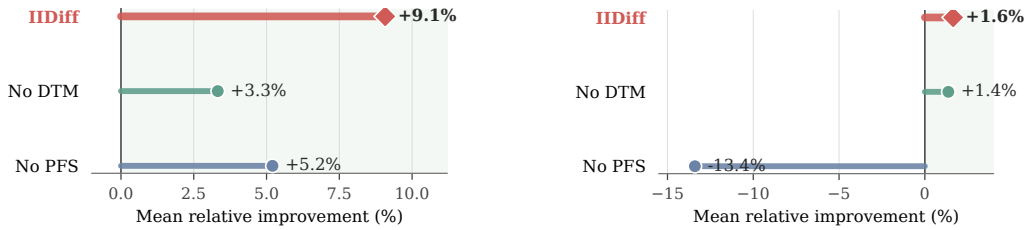
The choice $p = 1$ gives the L1 objective, while $p = 2$ is the squared Frobenius-norm objective. The learned reverse transition in Eq. 1 then generates complete time-series samples.

5 Experiment

In this section, we validate the effectiveness of our model on 12 real-world datasets and further verifies the core hypotheses through visualization. Extended results and cost analysis are in Appendix E to F.

5.1 Settings

The key experimental settings are briefly introduced as below, with more details in Appendix B to D.



(a) MMD relative improvement.

(b) Marginal-KL relative improvement.

Figure 3: Mean relative improvement over the same-backbone vanilla DiT. For each metric and dataset, improvement is $(s_{\text{vanilla}} - s_{\text{variant}})/s_{\text{vanilla}} \times 100\%$. The markers report the mean percentage over 12 datasets.

Datasets. Extensive experiments are conducted on 12 real-world time-series datasets covering 4 practical domains. All datasets are preprocessed into non-overlapping univariate sequence slices with a fixed length of 168 time steps. Following previous work[10], all datasets are concatenated and used for joint model training, ensuring the model learns general domain-adaptive features. More details of the datasets could be found in Appendix B.

Baselines. We select representative methods for cross-domain time-series generation as baselines, covering three core technical paradigms: (1) GAN-based methods including TimeGAN [25] and GT-GAN [11]. (2) VAE-based methods including TimeVAE [6] and TimeVQVAE [13]. (3) Diffusion-based methods (recent SOTAs), including TimeDP [10], Diffusion-TS [26], TimeDiT[4] and Winformer [3]. Due to space limitation, results of partial baselines are shown in Appendix E.

Metrics. We employ two core metrics to assess model performance, including Maximum Mean Discrepancy (MMD) and Kullback-Leibler Divergence (KL). **MMD** maps time-series into a high-dimensional feature space with kernel functions, then calculates the mean difference of the mapped features to accurately measure the distributional gap between two datasets. **KL Divergence** directly quantifies the relative entropy between two probability distributions, reflecting the dissimilarity of underlying data generation mechanisms. The formal definition of metrics is in Appendix C.

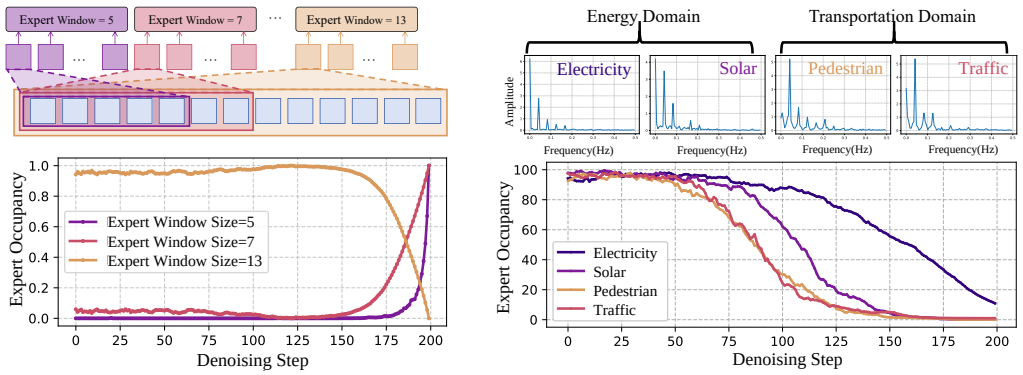
Implementation Experiments are conducted on an NVIDIA V100 GPU (32GB). The results are averaged by 5 runs with different random seeds. More details about the environment and hyper-parameters are in Appendix D and codes are in supplementary materials.

5.2 Main Results

Our model demonstrates superior performance in cross-domain time-series generation across 12 datasets from 4 domains. It outperforms 5 SOTA baselines, including GAN, VAE, and diffusion-based methods, in 9 datasets under the MMD metric. These results validate that the progressive frequency specializing and the dynamic transition matching solve the frequency transition problem intra- and inter- domain, aligning with the diffusion process's low-to-high frequency recovery order properly. The model outperformances across traffic, weather, industrial, and financial domains highlights strong domain adaptability. The metrics confirm that our model generates synthetic time-series with closer distributional similarity to real data, addressing the core challenge of frequency transition for cross-domain generation tasks.

5.3 Ablation Results

The ablation study compares the results of our IIDiff, without progressive frequency specializing (W.O.PFS), dynamic transition matching (W.O.DTM) or without both (W.O.PFS & W.O.DTM). Experimental results are partially shown in Fig. 3, suggesting that each component capture complementary aspects of distribution alignment.



(a) Intra-domain: Experts activation follows transition. (b) Inter-domain: Routing adapts domain transition variation.

Figure 4: **Visualization of experts occupancy. (a) Intra-domain transition compliance:** The smaller the window ($p = 5, 7$), the later the experts will be occupied than the larger ones ($p = 13$). indicating that our hierarchical experts are activated following the transition rules in denoising process, which demonstrates the rationality of the design. **(b) Inter-domain transition adaptation:** The expert ($p = 13$) show varying speeds of occupancy decline across the 4 datasets, demonstrating our routing mechanism can adjust the strategy to adapt the domain-specific transition differences.

5.4 Analysis of Experts Occupancy

Q1: How does the expert occupancy change as frequency transition? Our model leverages progressive frequency specializing, which are diffusion-native by providing hierarchical frequency reception fields to follow the low-to-high frequency transition. During the denoising process, we visualized the occupancy changes of every experts as in Fig.4 (a), averaging on the 12 datasets. From the figure, we can infer that the occupancy of experts with smaller window ($p = 5, 7$) increases with denoising processing, while the larger ones ($p = 13$) decrease. Additionally, the occupancy increases for expert with smallest window ($p = 5$) is slower than the second small one ($p = 7$), indicating that higher frequency components are sensitive at the later part of the denoising process. These phenomena indicate that as denoising, the progressive frequency specializing follows the rule of frequency transition, demonstrating that our model conforms to the natural process of diffusion.

Q2: How does the expert occupancy changing differ across domain? We introduce dynamic transition matching to enhance the domain adaptation by adapting transition differs across domains. In Fig. 4 (b), we visualized the expert occupancy on diverse datasets to explore how our model acts with the inter-domain frequency transition differs. As shown in the above 4 subplots of Fig. 4 (b), the frequency components from different domains show differences, leading to variation in frequency transition speed in denoising process. However, our routing mechanism guided by dynamic transition masking can adaptively sensing the transition speed instantaneously and active experts properly. As shown in the bottom sub-figure of Fig. 4 (b), the occupancy of an expert (window size $p = 13$) has different decrease speeds across domains, indicating that our model can adaptively adjust routing policy to fit the different transition speed across diverse domains.

6 Conclusion

Cross-domain time-series generation faces critical barriers from intra-domain dynamic frequency transition and inter-domain varying transition speeds. To address this, we propose IIDiff, a diffusion MoE framework with two complementary core mechanisms. The progressive frequency specializing dynamically adjust frequency receptive fields to align with the frequency transition, resolving intra-domain imbalance learning. The dynamic transition matching enables adaptive response to inter-domain speed differences. Extensive experiments on 12 real-world datasets across 4 domains validate IIDiff's superiority, demonstrating robust domain adaptability and high-fidelity generation capability.

Acknowledgement

This work was supported by the grants from the Natural Science Foundation of China (U2541212, 62572035). This work was also supported by Beijing Natural Science Foundation (L248032). Thanks for the computing infrastructure provided by Beijing Advanced Innovation Center for Big Data and Brain Computing. The corresponding author is Haoyi Zhou.

References

[1] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=hHiIbk7ApW.

[2] Khaled Alkilane, Yihang He, Bai Zhang, and Der-Horng Lee. Mixture-of-diffusers: Dual-stage diffusion model for improved time series generation, 2025. URL https://openreview.net/forum?id=lcmd2Qdrsv.

[3] Anonymous. Winformer: Transcending pairwise similarity for time-series generation. In *Submitted to The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DY6dnkRKo7. under review.

[4] Defu Cao, Wen Ye, Yizhou Zhang, and Yan Liu. Timedit: General-purpose diffusion transformers for time series foundation model, 2025. URL https://arxiv.org/abs/2409.02322.

[5] Tianyu Chen, Haoyi Zhou, Ying Li, Hao Wang, Zhenzhe Zhang, Tianchen Zhu, Shanghang Zhang, and Jianxin Li. Freqmoe: Dynamic frequency enhancement for neural PDE solvers. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 7356–7364. ijcai.org, 2025. doi: 10.24963/IJCAI.2025/818. URL https://doi.org/10.24963/ijcai.2025/818.

[6] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *CoRR*, abs/2111.08095, 2021. URL https://arxiv.org/abs/2111.08095.

[7] Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i11.29085. URL https://doi.org/10.1609/aaai.v38i11.29085.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[9] Xiaoyu Huang, Weidong Chen, Bo Hu, and Zhendong Mao. Graph mixture of experts and memory-augmented routers for multivariate time series anomaly detection. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 17476–17484. AAAI Press, 2025. doi: 10.1609/AAAI.V39I16.33921. URL https://doi.org/10.1609/aaai.v39i16.33921.

[10] Yu-Hao Huang, Chang Xu, Yueying Wu, Wu-Jun Li, and Jiang Bian. Timedp: Learning to generate multi-domain time series with domain prompts. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 17520–17527. AAAI Press, 2025. doi: 10.1609/AAAI.V39I17.33926. URL https://doi.org/10.1609/aaai.v39i17.33926.

[11] Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. GT-GAN: general purpose time series synthesis with generative adversarial networks. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/f03ce573aa8bce26f77b76f1cb9ee979-Abstract-Conference.html.

[12] Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/5a1a10c2c2c9b9af1514687bc24b8f3d-Abstract-Conference.html.

[13] Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent, editors, *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 7665–7693. PMLR, 2023. URL https://proceedings.mlr.press/v206/lee23d.html.

[14] Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=SrEOUSyJcR.

[15] Yiwen Liu, Chenyu Zhang, Junjie Song, Siqi Chen, Sun Yin, Zihan Wang, Lingming Zeng, Yuji Cao, and Junming Jiao. Mofe-time: Mixture of frequency domain experts for time-series forecasting models. *CoRR*, abs/2507.06502, 2025. doi: 10.48550/ARXIV.2507.06502. URL https://doi.org/10.48550/arXiv.2507.06502.

[16] Ziqi Liu. Freqmoe: Enhancing time series forecasting through frequency decomposition mixture of experts. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Mohammad Emtiyaz Khan, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025*, volume 258 of *Proceedings of Machine Learning Research*, pages 3430–3438. PMLR, 2025. URL https://proceedings.mlr.press/v258/liu25i.html.

[17] Xiangkai Ma, Xiaobin Hong, Wenzhong Li, and Sanglu Lu. UTSD: unified time series diffusion model. *CoRR*, abs/2412.03068, 2024. doi: 10.48550/ARXIV.2412.03068. URL https://doi.org/10.48550/arXiv.2412.03068.

[18] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.org/10.1109/ICCV51070.2023.00387.

[19] Xiaohou Shi, Ke Li, Aobo Liang, and Yan Sun. Time tracker: Mixture-of-experts-enhanced foundation time series forecasting model with decoupled training pipelines. *CoRR*, abs/2505.15151, 2025. doi: 10.48550/ARXIV.2505.15151. URL https://doi.org/10.48550/arXiv.2505.15151.

[20] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=e1wDDFmlVu.

[21] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24804–24816, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/cfe8504bda37b575c70ee1a8276f3486-Abstract.html.

[22] Lulu Wang and Chengqing Li. Adjacent neighborhood transformer-based diffusion model for anomaly detection under incomplete industrial data sources. In Wolfgang Nejdl, Sören Auer, Meeyoung Cha, Marie-Francine Moens, and Marc Najork, editors, *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, WSDM 2025, Hannover, Germany, March 10-14, 2025*, pages 577–585. ACM, 2025. doi: 10.1145/3701551.3703565. URL https://doi.org/10.1145/3701551.3703565.

[23] Yujie Wei, Shiwei Zhang, Hangjie Yuan, Yujin Han, Zhekai Chen, Jiayu Wang, Difan Zou, Xihui Liu, Yingya Zhang, Yu Liu, and Hongming Shan. Routing matters in moe: Scaling diffusion transformers with explicit routing guidance. *CoRR*, abs/2510.24711, 2025. doi: 10.48550/ARXIV.2510.24711. URL https://doi.org/10.48550/arXiv.2510.24711.

[24] Bo Yin, Xiaobin Hu, Xingyu Zhou, Peng-Tao Jiang, Yue Liao, Junwei Zhu, Jiangning Zhang, Ying Tai, Chengjie Wang, and Shuicheng Yan. Fera: Frequency-energy constrained routing for effective diffusion adaptation fine-tuning, 2025. URL https://arxiv.org/abs/2511.17979.

[25] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5509–5519, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c9efe5f26cd17ba6216bbe2a7d26d490-Abstract.html.

[26] Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=4h1apFjO99.

[27] Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tieyuan Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, Xiaolu Zhang, Zhenzhong Lan, Junbo Zhao, Da Zheng, Chongxuan Li, Jianguo Li, and Ji-Rong Wen. Llada-moe: A sparse moe diffusion language model. *CoRR*, abs/2509.24389, 2025. doi: 10.48550/ARXIV.2509.24389. URL https://doi.org/10.48550/arXiv.2509.24389.

A Theories and Proofs

A.1 Proof of Corollary 4.1

Proof of Corollary 4.1. Using Euler's formula $e^{-j\theta} = \cos\theta - j\sin\theta$, decompose $X(k)$ into its real and imaginary parts:

$$\begin{aligned} \text{Re}[X(k)] &= \sum_{n=0}^{N-1} x(n)\cos\left(\frac{2\pi}{N}kn\right) \quad , \\ \text{Im}[X(k)] &= -\sum_{n=0}^{N-1} x(n)\sin\left(\frac{2\pi}{N}kn\right) \quad . \end{aligned} \tag{15}$$

It can be observed that both $\text{Re}[X(k)]$ and $\text{Im}[X(k)]$ are linear combinations of the time-domain Gaussian random variables $x(n)$, with weights $\cos\left(\frac{2\pi}{N}kn\right)$ and $-\sin\left(\frac{2\pi}{N}kn\right)$. As the property of the Gaussian distribution, any linear combination of Gaussian random variables is also Gaussian-distributed. Since $x(n) \sim \mathcal{N}(0, \sigma^2)$, and both $\text{Re}[X(k)]$ and $\text{Im}[X(k)]$ are linear combinations of $x(n)$, they both follow Gaussian distributions. That is, there exist $\mu_{\text{Re}}, \sigma^2_{\text{Re}}, \mu_{\text{Im}}, \sigma^2_{\text{Im}}$ such that:

$$\text{Re}[X(k)] \sim \mathcal{N}(\mu_{\text{Re}}, \sigma^2_{\text{Re}}) \quad , \quad \text{Im}[X(k)] \sim \mathcal{N}(\mu_{\text{Im}}, \sigma^2_{\text{Im}}) \quad . \tag{16}$$

Since $x(n)$ is zero-mean, which means $\mathbb{E}[x(n)] = 0$, we can apply the linearity of expectation:

$$\mathbb{E}[\text{Re}[X(k)]] = \sum_{n=0}^{N-1} \mathbb{E}[x(n)]\cos\left(\frac{2\pi}{N}kn\right) = 0. \tag{17}$$

Similarly, $\mathbb{E}[\text{Im}[X(k)]] = 0$, indicating that both $\text{Re}[X(k)]$ and $\text{Im}[X(k)]$ are zero-mean Gaussian variables. Since $x(n)$ is i.i.d., the variance satisfies $\text{Var}(aY) = a^2\text{Var}(Y)$, and the variance of a linear combination of independent variables equals the linear combination of their variances. Thus:

$$\begin{aligned} \text{Var}(\text{Re}[X(k)]) &= \sigma^2 \sum_{n=0}^{N-1} \cos^2\left(\frac{2\pi}{N}kn\right) \quad , \\ \text{Var}(\text{Im}[X(k)]) &= \sigma^2 \sum_{n=0}^{N-1} \sin^2\left(\frac{2\pi}{N}kn\right) \quad . \end{aligned} \tag{18}$$

Using the trigonometric identity $\cos^2\theta + \sin^2\theta = 1$ and orthogonality $\sum_{n=0}^{N-1}\cos\left(\frac{2\pi}{N}kn\right)\sin\left(\frac{2\pi}{N}kn\right) = 0$, we obtain:

$$\sum_{n=0}^{N-1} \cos^2\left(\frac{2\pi}{N}kn\right) = \sum_{n=0}^{N-1} \sin^2\left(\frac{2\pi}{N}kn\right) = \frac{N}{2} \quad (k \neq 0, N/2) \quad . \tag{19}$$

Substituting this into the variance expressions gives $\sigma^2_{\text{Re}} = \sigma^2_{\text{Im}} = \frac{N\sigma^2}{2}$, meaning the variances of the two parts are equal.

Consider the special cases, where $k = 0$ or $k = \frac{N}{2}$:

Special case 1: When $k = 0$, $e^{-j0} = 1$, so $X(0) = \sum_{n=0}^{N-1} x(n)$. In this case, $\text{Im}[X(0)] = 0$, and the variance is

$$\text{Var}(X(0)) = \sum_{n=0}^{N-1} \text{Var}(x(n)) = N\sigma^2 \quad , \tag{20}$$

which means $X(0) \sim \mathcal{N}(0, N\sigma^2)$.

Special case 2: When N is even and $k = N/2$, $e^{-j\pi n} = (-1)^n$, so we have:

$$X(N/2) = \sum_{n=0}^{N-1} x(n)(-1)^n \quad . \tag{21}$$

Here, $\text{Im}[X(N/2)] = 0$, and similarly $X(N/2) \sim \mathcal{N}(0, N\sigma^2)$.

□

A.2 Intermediate Diffusion States

For the forward process,

$$\mathbf{X}_t = \sqrt{\bar{\alpha}_t}\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{22}$$

Linearity of the DFT yields

$$\mathcal{F}(\mathbf{X}_t) = \sqrt{\bar{\alpha}_t}\mathcal{F}(\mathbf{X}_0) + \sqrt{1-\bar{\alpha}_t}\mathcal{F}(\boldsymbol{\epsilon}). \tag{23}$$

Thus the noise *term* retains jointly Gaussian real and imaginary coefficients, while the complete intermediate state generally follows a data-dependent spectral distribution because it also contains the signal term. Amplitude ranking provides a coarse routing cue because weak signals may occur at low amplitude, broadband signals may occupy many bins, and pure-noise coefficients may be large. The learned router uses this aggregate feature to coordinate four experts while retaining all frequency bins.

A.3 Moving-Average Frequency Response

Ignoring boundary padding, a length-P moving average has impulse response $h_P[n] = 1/P$ for $n = 0, \ldots, P-1$. Its discrete-time Fourier response is

$$H_P(\omega) = \frac{1}{P}\sum_{n=0}^{P-1} e^{-\mathrm{i}\omega n} = e^{-\mathrm{i}\omega(P-1)/2}\frac{\sin(P\omega/2)}{P\sin(\omega/2)}. \tag{24}$$

Increasing P narrows the main lobe and increases temporal smoothing. The response also contains sidelobes, so AvgPool is not an ideal low-pass filter and its experts do not correspond to disjoint bands. Zero padding further changes the response near sequence boundaries. These properties motivate describing the experts as complementary smoothing scales or receptive frequency ranges.

A.4 Sampling and Finite-Window Scope

The DFT is used as a descriptor of the sampled sequence. IIDiff cannot recover information already aliased during acquisition when a continuous signal was sampled above the Nyquist limit. The stride-one average-pooling experts introduce no downsampling aliasing. Windowed DFTs or learnable filterbanks could replace these operators without changing the routing principle.

B Datasets and Experimental Protocol

In this section, we provide a detailed description of the datasets, preprocessing procedures, domain construction, and experimental protocols used in our experiments. The goal of our experimental setting is to evaluate whether a unified diffusion framework can learn transferable generation patterns across diverse time-series domains with heterogeneous frequency characteristics.

B.1 Dataset Collection

In this paper, we conduct the experiments following the setting of the TimeDP[2] and Winformer[3]. The experiments contain 12 real-world datasets from four domains, including energy, economic, weather and transportation. These datasets exhibit substantially different temporal dynamics, sampling frequencies, periodic patterns, and frequency-domain characteristics, making them suitable for evaluating cross-domain generation capability. The pre-processed datasets are open-sourced by TimeDP[4]. We list the details of the datasets in Table 2.

B.2 Cross-Domain Construction

Unlike conventional single-domain time-series generation, our setting jointly trains a unified generative model on datasets from multiple domains simultaneously.

[2]https://github.com/YukhoY/TimeDP

[3]https://openreview.net/forum?id=DY6dnkRKo7

[4]https://huggingface.co/datasets/YukhoW/TimeDP-Data/blob/main/TimeDP-Data.zip

Table 2: Details for the real-world datasets.

Dataset	Electricity	Solar	Wind	Air	Temperature	Rain
Domain	energy	energy	energy	weather	weather	weather
Variables	321	137	1	270	422	422
Sampling interval	1 hour	1 hour	4 second	1 hour	1 day	1 day
Dataset	Traffic	Taxi	Pedestrian	NN5	Fred-MD	Exchange
Domain	transportation	transportation	transportation	economic	economic	economic
Variables	963	1214	1	111	107	8
Sampling interval	1 hour	30 minutes	1 hour	1 day	1 month	1 day

Formally, let $D = D(1), D(2), \ldots, D(J)$, denote the collection of datasets from J domains. Each domain contains time-series with distinct temporal and frequency characteristics.

The purpose of this setting is to evaluate whether the proposed framework can learn temporal representations across heterogeneous domains, adapt expert routing according to domain-specific frequency transition dynamics and maintain generation fidelity under diverse spectral distributions. It should be noted that our setting focuses on multi-domain conditional generation rather than strict domain adaptation under unseen target domains. The model is jointly optimized on all domains while conditioned using domain prompts.

C Metric

In this paper, we adopt two metrics in measuring the quality of the generated time-series data. Firstly, we define a real time-series data with L length and D channels as $\mathbf{X} = [\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(L)}] \in \mathbb{R}^{L \times D}$, and the synthetic data is $\hat{\mathbf{X}} = [\hat{\mathbf{X}^{(1)}}, \hat{\mathbf{X}^{(2)}}, ..., \hat{\mathbf{X}}^{(L)}] \in \mathbb{R}^{L \times D}$. Then we formulate the metrics of Maximum Mean Discrepancy and Kullback-Leibler Divergence as follow.

(1) Maximum Mean Discrepancy (MMD). We report the biased empirical squared MMD with an RBF kernel k.

$$\begin{aligned}\widehat{\mathrm{MMD}}_b^2 =& \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} k(\mathbf{X}_i, \mathbf{X}_{i'}) + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} k(\widehat{\mathbf{X}}_j, \widehat{\mathbf{X}}_{j'}) \\ &- \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(\mathbf{X}_i, \widehat{\mathbf{X}}_j).\end{aligned} \tag{25}$$

The implementation uses the scikit-learn default $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2/D)$, where D is the flattened feature dimension.

(2) Kullback-Leibler Divergence (KL). We flatten the real values, construct a 50-bin histogram, and use its bin edges for the generated values. After normalization, let p_r and q_r denote the real and generated masses in bin r. The reported marginal KL is

$$L_{\mathrm{KL}} = \sum_{r=1}^{50} p_r \log\left(\frac{p_r}{q_r}\right). \tag{26}$$

The implementation adds 10^{-9} to the generated histogram before normalization to avoid division by zero. Values outside the range of the real-data bin edges are not assigned to a bin by the histogram routine.

D Implementation Details

In this section, we report the detailed implementation, including hyper-parameters and environments.

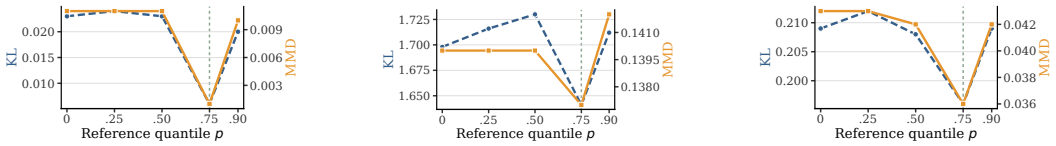
D.1 Hyper parameters

Our method is a transformer-based denoising model, consisting of several encoder blocks and a conditioning block. We have listed the detailed hyper-parameters of each component, which are shown in Table 3.

Table 3: Table of the Hyper-parameters.

Catergory	Module	Name	Value
Neural Denoising Network	Conditioning block	Dimension	32
		Channel	1
		Latent	16
	MoE encoder block	Channel	1
		Hidden size	512
		Attention heads	16
		MLP ratio	4.0
		Unshared Experts	4
		Aggregating sizes	5, 7, 9, 13
		Shared Experts	1
	DiT encoder block	Encoder blocks	5
		Channel	1
		Attention heads	16
		MLP ratio	4.0
Diffusion Forward/Reverse Process		Noising/Denoising steps	200
		Loss type	L1 loss
Training Setting		Batch size	128
		Learning rate	2e-4
		Train steps	50, 000

We conducted a sensitivity analysis of the hyperparameter ρ, and the results (Fig. 5) show that, although these datasets exhibit different noise characteristics, the optimal ρ consistently lies around 0.75 in diverse datasets (e.g. electricity, exchange, wind), and moderate deviations from this value only lead to marginal performance degradation. This behavior suggests that ρ controls the trade-off between preserving signal components and estimating noise statistics, rather than strictly corresponding to a theoretical noise ratio. The results indicates that our model does not require precise tuning of ρ when applied to different domains, as the signal-to-noise ratio in real-world datasets is common. Instead, selecting ρ within a reasonable range is sufficient to ensure robust performance.



(a) Analysis on electricity data. (b) Analysis on exchange data. (c) Analysis on wind data.

Figure 5: Parameter test of ρ on diverse dataset.

D.2 Hard-wares and Environments

The experiments are conducted with a NIVDIA V100 GPU, with 32GB memory. Our model relies on public environment libraries, including CUDA, Python, PyTorch, etc. The framework is based on the source code of TimeDP[5]. Our codes are reported in the supplementary material. More detailed environment and installation tips are reported in the ReadMe file in codes folder, which is provided in supplementary materials.

[5]https://github.com/microsoft/TimeCraft/tree/main/TimeDP

E Extended Experimental Comparisons

To further validate the effectiveness of our proposed model and achieve a more comprehensive comparison, we supplement additional baseline methods including the newly released method Winformer, TimeDiT[4] and Time-VQVAE [13].

Table 4: Comparative results for extended baselines. Lower values are better. Best results are **bold**, and second-best results are underlined. Ties at the reported precision receive the same marking.

Method	IIDiff		Winformer		TimeDiT		Time-VQVAE	
Metric	MMD	KL	MMD	KL	MMD	KL	MMD	KL
Electricity	$\mathbf{0.001}_{\pm 0.001}$	$\mathbf{0.006}_{\pm 0.003}$	$\mathbf{0.001}_{\pm 0.002}$	$\underline{0.008}_{\pm 0.010}$	$\underline{0.002}_{\pm 0.000}$	$0.065_{\pm 0.015}$	$0.152_{\pm 0.024}$	$0.280_{\pm 0.051}$
Solar	$\mathbf{0.032}_{\pm 0.005}$	$\mathbf{0.011}_{\pm 0.004}$	$0.035_{\pm 0.004}$	$\underline{0.013}_{\pm 0.005}$	$\underline{0.033}_{\pm 0.003}$	$0.018_{\pm 0.002}$	$0.437_{\pm 0.020}$	$0.865_{\pm 0.108}$
Wind	$0.036_{\pm 0.014}$	$\underline{0.196}_{\pm 0.031}$	$\underline{0.034}_{\pm 0.014}$	$0.202_{\pm 0.044}$	$\mathbf{0.017}_{\pm 0.000}$	$\mathbf{0.165}_{\pm 0.008}$	$0.131_{\pm 0.014}$	$0.483_{\pm 0.066}$
Traffic	$0.083_{\pm 0.010}$	$\underline{0.013}_{\pm 0.009}$	$\mathbf{0.071}_{\pm 0.005}$	$\mathbf{0.011}_{\pm 0.002}$	$\underline{0.075}_{\pm 0.003}$	$\underline{0.013}_{\pm 0.001}$	$0.213_{\pm 0.016}$	$0.178_{\pm 0.026}$
Taxi	$0.092_{\pm 0.014}$	$\underline{0.004}_{\pm 0.001}$	$\underline{0.085}_{\pm 0.010}$	$0.005_{\pm 0.003}$	$\mathbf{0.082}_{\pm 0.003}$	$\mathbf{0.003}_{\pm 0.001}$	$0.128_{\pm 0.004}$	$0.110_{\pm 0.026}$
Pedestrian	$\mathbf{0.037}_{\pm 0.010}$	$\underline{0.011}_{\pm 0.011}$	$\underline{0.040}_{\pm 0.008}$	$\mathbf{0.009}_{\pm 0.004}$	$\underline{0.040}_{\pm 0.003}$	$0.021_{\pm 0.002}$	$0.067_{\pm 0.007}$	$0.405_{\pm 0.051}$
Air	$\underline{0.012}_{\pm 0.001}$	$\underline{0.029}_{\pm 0.012}$	$\mathbf{0.011}_{\pm 0.002}$	$\mathbf{0.026}_{\pm 0.011}$	$0.017_{\pm 0.001}$	$0.057_{\pm 0.011}$	$0.028_{\pm 0.002}$	$0.054_{\pm 0.012}$
Temperature	$\mathbf{0.224}_{\pm 0.021}$	$\mathbf{0.175}_{\pm 0.028}$	$0.230_{\pm 0.021}$	$\underline{0.176}_{\pm 0.027}$	$\underline{0.225}_{\pm 0.004}$	$0.178_{\pm 0.027}$	$0.323_{\pm 0.008}$	$0.735_{\pm 0.066}$
Rain	$\mathbf{0.035}_{\pm 0.016}$	$\underline{0.012}_{\pm 0.007}$	$\underline{0.036}_{\pm 0.016}$	$\mathbf{0.011}_{\pm 0.003}$	$0.061_{\pm 0.004}$	$0.052_{\pm 0.007}$	$0.074_{\pm 0.007}$	$0.047_{\pm 0.018}$
NN5	$0.155_{\pm 0.008}$	$0.059_{\pm 0.009}$	$\underline{0.147}_{\pm 0.008}$	$\underline{0.045}_{\pm 0.007}$	$\mathbf{0.142}_{\pm 0.002}$	$\mathbf{0.035}_{\pm 0.009}$	$0.327_{\pm 0.012}$	$1.063_{\pm 0.274}$
Fred-MD	$\mathbf{0.002}_{\pm 0.001}$	$\mathbf{0.201}_{\pm 0.027}$	$\mathbf{0.002}_{\pm 0.001}$	$\mathbf{0.201}_{\pm 0.014}$	$\mathbf{0.002}_{\pm 0.001}$	$\underline{0.252}_{\pm 0.027}$	$\underline{0.008}_{\pm 0.002}$	$0.831_{\pm 0.077}$
Exchange	$\mathbf{0.137}_{\pm 0.013}$	$\underline{1.641}_{\pm 0.137}$	$\mathbf{0.137}_{\pm 0.012}$	$\mathbf{1.621}_{\pm 0.126}$	$\underline{0.170}_{\pm 0.000}$	$2.001_{\pm 0.014}$	$0.342_{\pm 0.050}$	$5.052_{\pm 1.385}$
Avg. Dec.	-	-	1.51%	-0.25%	-14.17%	-127.94%	-1479.66%	-1908.22%

Overall, IIDiff achieves competitive or superior performance across most datasets and metrics, particularly on datasets with heterogeneous temporal dynamics and diverse spectral distributions. Compared with conventional diffusion transformers, the proposed framework demonstrates stronger robustness under cross-domain generation settings, suggesting that explicitly modeling frequency transition dynamics is beneficial for multi-domain time-series generation. Compared with vanilla diffusion transformer architectures such as TimeDiT, IIDiff introduces adaptive frequency-specialized routing mechanisms that better align with the intrinsic low-to-high frequency recovery behavior during denoising. Conventional transformers rely on fixed receptive fields throughout diffusion steps, while IIDiff progressively adjusts expert activation according to transition stages and domain characteristics, enabling more flexible temporal reconstruction across different frequency scales. Compared with Winformer, the proposed framework focuses on a different aspect of temporal modeling. Winformer enhances periodic perception inside attention mechanisms, whereas IIDiff explicitly models transition adaptation during denoising through hierarchical experts and transition-aware routing. The two approaches are complementary rather than mutually exclusive.

F Computational Cost

For a further analysis on computational cost, we provide both theoretical and empirical analysis as below.

Theoretical analysis. Let L denote the sequence length, d the hidden dimension, m the number of routed experts, s the number of shared experts, and k_i the average-pooling window of the i-th routed expert. The routing module computes a DFT and sorts the spectral amplitudes, both requiring $O(L \log L)$ operations per input channel. The linear routing projection from the length-L spectral feature to m expert weights requires $O(Lm)$ operations. For the i-th routed expert, stride-one average pooling requires $O(Ldk_i)$ operations. Each routed or shared attention expert has the standard attention complexity $O(Ld^2 + L^2 d)$, including the linear projections and attention computation. Therefore, when all m routed experts and s shared experts are evaluated, the total complexity of one MoE block is

$$O\left(L \log L + Lm + Ld \sum_{i=1}^{m} k_i + (m+s)(Ld^2 + L^2 d)\right), \tag{27}$$

where channel and batch dimensions are omitted for clarity. In our main configuration, $m = 4$, $s = 1$, and top-$k = m$, so the routed mixture is dense and all four routed experts are evaluated. Consequently, the dominant additional cost relative to a single-attention block comes from evaluating multiple attention experts, while the FFT-based routing and average-pooling operations contribute lower-order overhead for the evaluated sequence length.

Empirical analysis. we provide a direct empirical comparison with vanilla DiT (Table 5), demonstrating that IIDiff maintains competitive processing speeds. IIDiff slightly increases parameters (1.1M→1.3M), and the training speed only sees a slight 11% reduction (12.70 → 11.31 batch/s).

Given the consistent and significant improvements in generation quality, this represents a highly favorable efficiency-performance trade-off. IIDiff is compatible with standard MoE optimizations such as expert parallelism, and the computational cost can be effectively scaled with parallel execution. Moreover, IIDiff is not tied to diffusion and can be extended to faster paradigms (e.g. flow matching), so the overhead is modular rather than intrinsic.

Table 5: Empirical computation cost.

Model	Training speed (batch/s)	Sampling speed (batch/s)	Parameter count
DiT	12.70	10.17	1.1M
Ours	11.31	8.81	1.3M

G Another Mode: Routing with noise

The main table uses the highest-25% implementation described in Appendix D. We separately train the configurable lowest-75% mode, which means modeling the noise to infer the current denoising state, and report its five-seed result below.

Here, lowest-75% refers to coefficient activation rather than frequency position. The mode selects the 75% of real DFT coefficients with the smallest absolute values, estimates their Gaussian location and scale, and applies the resulting Gaussian-shaped transform to construct the routing descriptor.

Relation to step estimation. The evaluated mode supplies this low-activation descriptor directly to the router. The same Gaussian reference statistics can parameterize a calibrated scalar estimate $\widehat{\nu}(\mathbf{X}_t)$ of the effective noise content. Comparing this estimate with the forward schedule $\nu(\tau) = \sqrt{1-\bar{\alpha}_\tau}$ gives

$$\widehat{t}(\mathbf{X}_t) = \arg\min_{\tau \in \{1,\dots,T\}} \left| \widehat{\nu}(\mathbf{X}_t) - \nu(\tau) \right|. \tag{28}$$

The estimated step or a soft distribution over neighboring steps can then serve as a scalar recovery coordinate. This provides a feasible extension of the Gaussian-reference construction evaluated by the lowest-75% experiment.

Evaluation and results. MMD and KL are computed separately on all 12 sources. Table 6 reports the corresponding source-wise means and standard deviations. The macro mean assigns equal weight to each source.

IIDiff improves MMD on 9 of the 12 sources and KL on 10, with mean source-wise reductions of 5.16% and 5.15%, respectively. It does not improve MMD on Air, Electricity, or Wind, and does not improve KL on NN5 or Wind.

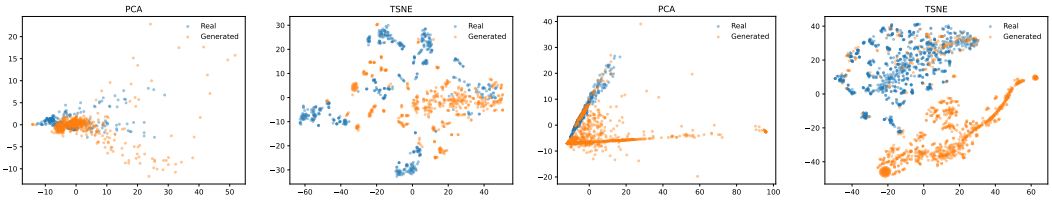
H PCA and t-SNE Visualizations

Figure 6 provides qualitative PCA and t-SNE comparisons. Solar is shown as a representative source, and the plots serve as qualitative diagnostics.

In the PCA projections, IIDiff shows overlap between real and generated samples in the central region, together with a broader generated spread in peripheral regions. TimeDP exhibits more visibly separated dominant directions. The t-SNE projections show partial local mixing for both methods, while several neighborhoods remain dominated by either real or generated samples. Each projection is fitted separately to the concatenated real and generated samples of one method, so coordinates and spatial scales are not directly comparable across panels.

Table 6: Final five-seed result of the lowest-75% reference mode. Results are reported as mean ± standard deviation to three decimal places, and lower values are better. The best result for each source and metric is shown in **bold** according to the unrounded mean.

Dataset	IIDiff (Mode 2)		TimeDP	
Metrics	MMD	KL	MMD	KL
Electricity	$0.001_{\pm 0.001}$	$\mathbf{0.009}_{\pm 0.008}$	$\mathbf{0.001}_{\pm 0.001}$	$0.012_{\pm 0.016}$
Solar	$\mathbf{0.031}_{\pm 0.001}$	$\mathbf{0.013}_{\pm 0.005}$	$0.041_{\pm 0.011}$	$0.016_{\pm 0.005}$
Wind	$0.036_{\pm 0.015}$	$0.201_{\pm 0.039}$	$\mathbf{0.025}_{\pm 0.017}$	$\mathbf{0.152}_{\pm 0.034}$
Traffic	$\mathbf{0.072}_{\pm 0.004}$	$\mathbf{0.008}_{\pm 0.003}$	$0.083_{\pm 0.034}$	$0.009_{\pm 0.003}$
Taxi	$\mathbf{0.086}_{\pm 0.013}$	$\mathbf{0.008}_{\pm 0.006}$	$0.095_{\pm 0.023}$	$0.011_{\pm 0.004}$
Pedestrian	$\mathbf{0.039}_{\pm 0.007}$	$\mathbf{0.013}_{\pm 0.009}$	$0.044_{\pm 0.020}$	$0.014_{\pm 0.010}$
Air	$0.012_{\pm 0.004}$	$\mathbf{0.026}_{\pm 0.009}$	$\mathbf{0.011}_{\pm 0.003}$	$0.027_{\pm 0.016}$
Temperature	$\mathbf{0.218}_{\pm 0.017}$	$\mathbf{0.147}_{\pm 0.032}$	$0.219_{\pm 0.022}$	$0.171_{\pm 0.073}$
Rain	$\mathbf{0.027}_{\pm 0.003}$	$\mathbf{0.012}_{\pm 0.002}$	$0.057_{\pm 0.039}$	$0.013_{\pm 0.012}$
NN5	$\mathbf{0.157}_{\pm 0.011}$	$0.063_{\pm 0.017}$	$0.164_{\pm 0.010}$	$\mathbf{0.054}_{\pm 0.014}$
Fred-MD	$\mathbf{0.001}_{\pm 0.000}$	$\mathbf{0.194}_{\pm 0.009}$	$0.002_{\pm 0.001}$	$0.203_{\pm 0.035}$
Exchange	$\mathbf{0.140}_{\pm 0.008}$	$\mathbf{1.766}_{\pm 0.196}$	$0.151_{\pm 0.024}$	$1.866_{\pm 0.132}$
Macro mean	**0.068**	**0.205**	0.074	0.212



(a) IIDiff PCA (b) IIDiff t-SNE (c) TimeDP PCA (d) TimeDP t-SNE

Figure 6: Representative Solar distribution visualizations. PCA and t-SNE compare real and generated embeddings for IIDiff and TimeDP.

I Joint Training

The main setting uses one model across heterogeneous sources, allowing shared parameters to capture temporal structure that may be useful across domains. We compare the jointly trained model with source-specific models to examine the effect of this shared training on two low-data sources while retaining source-conditioned generation. Table 7 controls architecture, seed, optimizer, and generation protocol. The 12-source model receives 10,000 balanced-mixture updates, corresponding to about 833 expected updates per source, while each single-source model receives 1,000 updates. This one-seed analysis characterizes the two tested sources under the specified configuration.

Table 7: Joint-versus-single-source diagnostic.

Source	Joint MMD	Single MMD	Joint KL	Single KL
Fred-MD	0.00207	0.00281	0.20420	0.70071
NN5	0.15769	0.18141	0.05320	0.26490

J Zero-shot Learning.

Source prompts provide an interface for conditioning the shared model on source-specific context. We evaluate whether a frozen model can use prompts constructed from a small number of clean examples to generate sequences for sources excluded from training. Stock and Web are held out during training, and ten clean examples are used only to construct prompts without parameter fine-tuning. This

one-seed 10-shot experiment provides an exploratory assessment of prompt-based transfer to the two held-out sources. For a separate scope test, Stock and Web are excluded from training. Ten clean examples are used only to construct prompts for a frozen model, with no parameter fine-tuning.

Table 8: Exploratory prompting on two sources excluded from training.

Held-out source	IIDiff MMD	TimeDP MMD	IIDiff KL	TimeDP KL
Stock	0.06099	0.09721	0.66768	0.77299
Web	0.04423	0.04833	0.24337	0.44608

K Routing Stability

We perturb the router input at 6 diffusion steps and 5 SNRs using 5 checkpoints on four representative sources (256 sequences per condition). Table 9 summarizes stability by SNR. Comparisons are against the unperturbed route at the same diffusion step.

Table 9: Routing stability under input perturbations, aggregated over the evaluated diffusion steps.

SNR (dB)	Cosine ↑	Top-1 agreement ↑	L1 change ↓	Tail relative change ↓
30	0.99982	0.99606	0.00202	0.00430
20	0.99760	0.98197	0.00753	0.02029
10	0.97847	0.94727	0.02611	0.11438
5	0.95787	0.92337	0.04202	0.28474
0	0.92362	0.88164	0.06272	0.67767

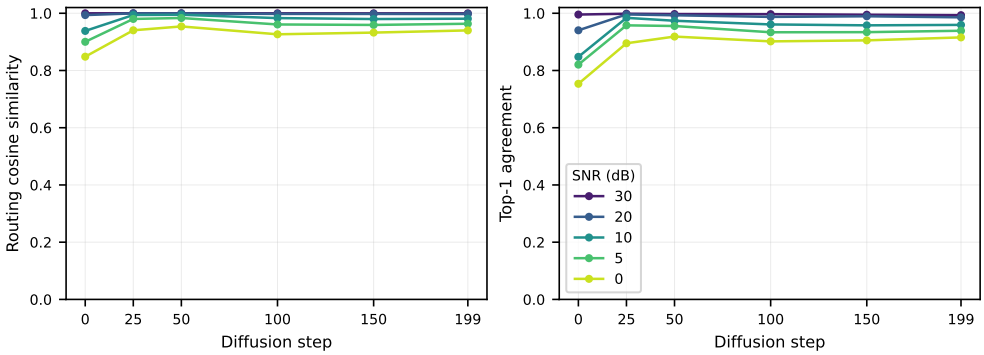
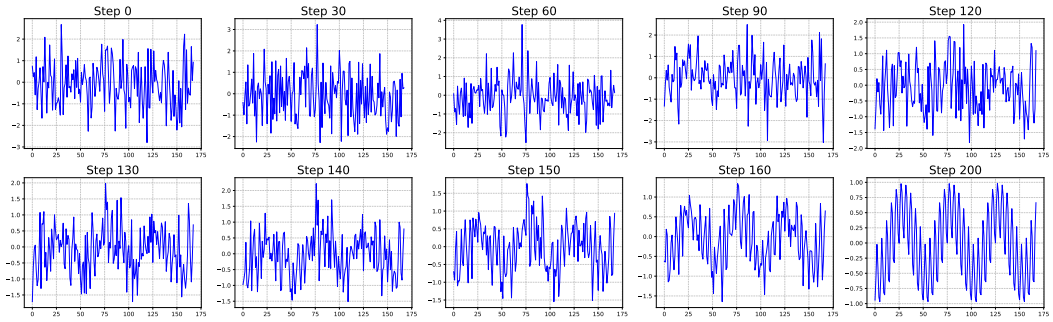


Figure 7: Routing stability across SNRs and diffusion steps. Each curve averages five checkpoints, four representative sources, and 256 sequences per condition.

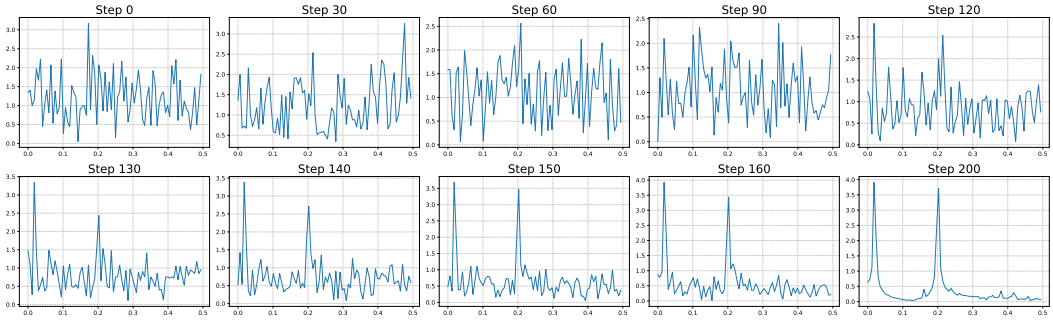
Figure 7 shows that the dominant route is stable under moderate perturbations across the evaluated steps, while sensitivity increases as the SNR decreases from 5 to 0 dB. Across the unperturbed diffusion trajectory, agreement between the current-state route and the clean-state route decreases from 0.9844 at step 0 to 0.4613 at step 199. This observation indicates that the routing weights vary with the current denoising state under the evaluated conditions.

L Visualization

We constructed a dataset containing a signal with only two frequency-domain peaks for denoising network training, so as to facilitate the observation of the frequency transition phenomenon. The variations in the time-domain and frequency-domain signals during the denoising process are illustrated in Fig. 8, which also serves as the original version of the time-domain and frequency-domain images presented in Fig. 1(a). It can be observed that as the denoising process proceeds, the recovery of the frequency-domain signal transitions from low frequencies to high frequencies.



(a) Denoising process on constructed series.



(b) Frequency transition as denoising.

Figure 8: Frequency transition observed on the constructed time-series data, which is the original inspiration of our work. It can be observed that as the denoising process proceeds, the recovery of the frequency-domain signal transitions from low frequencies to high frequencies.

As denoising proceeds, the time-series gradually evolves from Gaussian noise into structured temporal signals. More importantly, the corresponding frequency-domain visualization reveals a clear low-to-high frequency recovery process, where low-frequency components emerge earlier during denoising, while high-frequency details are reconstructed progressively in later stages. This observation provides the motivation for the proposed progressive frequency specializing mechanism. The visualization results are also consistent with the expert occupancy analysis in Fig. 4. Experts with larger aggregation windows are activated more frequently during early denoising stages because they mainly preserve coarse low-frequency structures. As denoising continues, the occupancy gradually shifts toward experts with smaller aggregation windows, indicating that the model progressively focuses on reconstructing finer high-frequency details.

We further observe that different datasets exhibit different routing transition speeds. Datasets with dominant low-frequency periodic patterns, such as Solar, maintain larger-window expert occupancy for longer stages, while datasets with more distributed frequency components, such as Traffic and NN5, activate smaller-window experts earlier. This behavior suggests that the proposed routing mechanism can adaptively respond to domain-specific transition characteristics during denoising. Overall, the visualization results provide empirical evidence that diffusion denoising exhibits structured frequency transition behavior, and that the proposed IIDiff framework successfully aligns expert routing with both denoising stages and domain-specific spectral dynamics.

M Limitation

Despite the promising performance achieved in our method, there still exist limitations worth exploring in future work. Our current approach introduces additional computational overhead when modeling cross-domain frequency transition adaptation, which inevitably increases inference cost compared with lightweight baseline methods. Future work may focus on optimizing computational

efficiency for cross-domain generation, such as designing more compact frequency modeling modules, simplifying the cross-domain adaptation paradigm, or integrating lightweight network architectures and sparse sampling strategies for the diffusion process. In this way, we can further reduce time and resource consumption while maintaining the original performance gain, making the method more applicable to real-world deployment scenarios.

N Declaration of LLM usage

In this work, large language models (LLMs) were only used for language polishing, grammatical revision, and academic expression refinement.

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: The contributions are list in introduction, an both contributions and scope are include in abstract and introduction.
   Guidelines:
   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: The limitation is shown in Appendix M.
   Guidelines:
   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]
   Justification: The full set of assumptions and a complete (and correct) proof are included in Appendix A.
   Guidelines:
   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [Yes]
   Justification: We include the codes in supplementary materials and the details for dataset, metric, implementation, ablation study and hyperparameter analysis in Appendix B to D.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes]
   Justification: We include the codes in supplementary materials and instructions for implementation.
   Guidelines:
   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not

including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?
   Answer: [Yes]
   Justification: We include the details for dataset, metric, implementation, ablation study and hyperparameters in Appendix B to D.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [Yes]
   Justification: We reported the error bars, as mean$\pm$std, in all of the result tables.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.
8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes]
   Justification: We report the resources in Appendix D and F.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have make sure to preserve anonymity, conforming with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**
    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
    Answer: [Yes]
    Justification: All of the assets are released with CC-BY 4.0, and we cite the source of them in Appendix and supplementary material.
    Guidelines:
    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.
13. **New assets**
    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
    Answer: [N/A]
    Justification: The paper does not release new assets.
    Guidelines:
    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.
14. **Crowdsourcing and research with human subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [N/A]
    Justification: The paper does not involve crowdsourcing nor research with human subjects.
    Guidelines:
    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.