SimDiff: Simpler Yet Better Diffusion Model for Time Series Point Forecasting

Hang Ding[1,*] **Xue Wang**[2,*], **Tian Zhou**[3,4,*], **Tao Yao**[1,†]
[1]Shanghai Jiao Tong University
[2]Alibaba Group US
[3]DAMO Academy, Alibaba Group
[4]Hupan Lab
dearsloth@sjtu.edu.cn, taoyao@sjtu.edu.cn

Abstract

Diffusion models have recently shown promise in time series forecasting, particularly for probabilistic predictions. However, they often fail to achieve state-of-the-art point estimation performance compared to regression-based methods. This limitation stems from difficulties in providing sufficient contextual bias to track distribution shifts and in balancing output diversity with the stability and precision required for point forecasts. Existing diffusion-based approaches mainly focus on full-distribution modeling under probabilistic frameworks, often with likelihood maximization objectives, while paying little attention to dedicated strategies for high-accuracy point estimation. Moreover, other existing point prediction diffusion methods frequently rely on pre-trained or jointly trained mature models for contextual bias, sacrificing the generative flexibility of diffusion models.

To address these challenges, we propose SimDiff, a single-stage, end-to-end framework. SimDiff employs a single unified Transformer network carefully tailored to serve as both denoiser and predictor, eliminating the need for external pre-trained or jointly trained regressors. It achieves state-of-the-art point estimation performance by leveraging intrinsic output diversity and improving mean squared error accuracy through multiple inference ensembling. Key innovations, including normalization independence and the median-of-means estimator, further enhance adaptability and stability. Extensive experiments demonstrate that SimDiff significantly outperforms existing methods in time series point forecasting.

Code — https://github.com/Dear-Sloth/SimDiff/tree/main

1 Introduction

Time series forecasting plays a crucial role across various real-world domains, including economics (Friedman 1962), retail sales prediction (Böse et al. 2017; Courty and Li 1999; Qiu et al. 2024), and energy management (Gao et al. 2020; Qiu et al. 2025a). The fundamental process of time series forecasting involves generating future sequences based on historical observations, which is intuitively well-suited for the application of diffusion models.

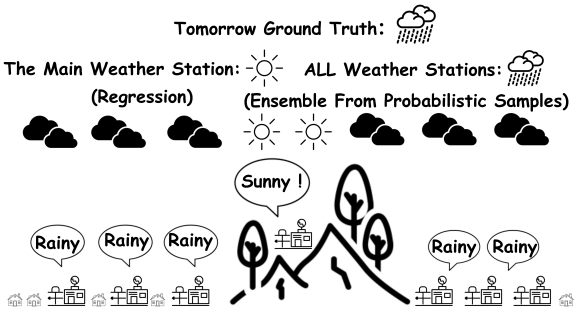


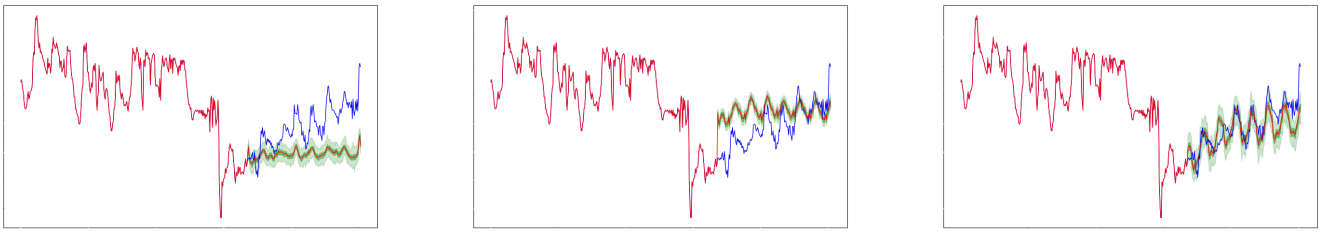
Figure 1: Trustworthy Forecasting by Ensembling Diverse Probability Samples

Two inseparable questions continue to hinder diffusion-based forecasting: *(i) how to inject sufficient contextual bias from past observations to obtain a stable and faithful predictive distribution? and (ii) how to reconcile the inherent trade-off between output diversity and point-forecast accuracy*?

Early likelihood-driven approaches such as TimeGrad (Rasul et al. 2021) and CSDI (Tashiro et al. 2021) maximize log-probability and therefore produce richly diverse samples. However, real-world time series often exhibit pronounced distribution drift between historical and future windows, something the likelihood objective neither detects nor corrects. These models also neglect to make targeted adjustments to the distribution drift of the data, thus often failing to capture the true underlying dynamics of time series. This limitation leads to suboptimal probabilistic performance and fails to provide a solid foundation for accurate point forecasting. As a result, training becomes unstable, sampling variance explodes, and the outputs are "too diverse to be useful," delivering poor Mean Squared Error (MSE) or Mean Absolute Error (MAE) scores. Consequently, these probabilistic models are seldom compared against strong point-forecasting baselines.

To tame this instability, TimeDiff (Shen and Kwok 2023) and mr-Diff (Shen, Chen, and Kwok 2024) prepend a *pre-trained* autoregressive predictor whose outputs serve as the initial trajectory $y_{0:m}$, while directly optimizing performance metrics such as MSE/MAE on the training data. This stitched design stabilizes optimization and improves point prediction

*Equal contribution
† Corresponding author



(a) CSDI (b) TimeDiff (c) SimDiff

Figure 2: Visualizations on *ETTh1* by (a) *CSDI*, (b) *TimeDiff*, and (c) *SimDiff*. CSDI only shows 90% interval due to the existence of extreme samples.

accuracy, but it also fixes the diffusion process to a deterministic baseline, constraining the model's inherent ability to explore the full range of possible distributions. As a result, distributional coverage shrinks sharply, and the system reintroduces the maintenance burden of an extra model. These models essentially become similar to regression models, capturing less of the true data distribution.

TMDM (Li et al. 2024) goes a step further by jointly training a mature transformer predictor (e.g., Autoformer (Wu et al. 2021) or Non-stationary Transformer (Liu et al. 2023)) and a conditional diffusion model within a Bayesian ELBO framework. This hybrid design mitigates variance and drift, reaching state-of-the-art point metrics among probabilistic forecasters. However, it still relies on an embedded regressor, converges slowly and entails high inference cost.

These observations underscore the need for a *truly simple* diffusion model—one that handles distribution drift, preserves sample diversity, and achieves strong point accuracy *without* auxiliary predictors. To this end, we propose **SimDiff**, a simple yet effective diffusion model for time-series forecasting. SimDiff exploits the inherent generative nature of diffusion models to enhance point estimation and remove dependence on pre-trained or jointly trained predictors. Specifically, we address two core questions:

1) Can the generative nature of diffusion models be harnessed to enhance point estimation through ensemble methods? This fundamental question explores how to leverage the generative capabilities of diffusion models to improve point predictions without sacrificing diversity, a gap not effectively addressed in prior work.

2) Is it possible to train a purely end-to-end diffusion model without relying on mature regression models to provide the necessary contextual bias? This inquiry challenges the prevailing reliance on pre-trained or jointly trained external predictors, seeking a simpler yet effective diffusion-based framework.

Our main contributions are summarized as follows:

- We propose **SimDiff**, the first fully end-to-end diffusion model achieving stable SOTA results in time series point forecasting. It employs a unified network as both denoiser and forecaster, greatly simplifying model design.
- We introduce **Normalization Independence (N.I.)**, a diffusion-specific technique that better captures data distributions and mitigates temporal drift.
- We design a simple yet efficient transformer backbone, offering clear empirical validation and practical design insights for future studies.
- **SimDiff** matches leading probabilistic models (in CRPS, CRPS-sum) without explicit design by leveraging the inherent generative nature of diffusion. Building on this, we propose a **Median-of-Means (MoM)** estimator that aggregates probabilistic samples to deliver clear **SOTA results in point forecasting**. The simplicity of SimDiff further enables much faster inference than existing diffusion-based models, underscoring its superior **efficiency**.

By addressing the aforementioned challenges, SimDiff bridges the gap between probabilistic diversity and accurate point prediction, leveraging the full potential of diffusion models in time series forecasting and offering a practical and powerful framework for future research in this domain.

2 SimDiff: Simpler yet Better Diffusion Model

We consider the following problem: given a sequence of multivariate time series observations $\mathbf{X} = x_{-L+1} : x_0$, where each x_t at time step t is a vector of dimension M, our goal is to forecast H future values $\mathbf{Y} = x_1 : x_H$. This predictive task is addressed using our proposed *SimDiff* model, which is detailed in Figure 3.

Diffusion and Denoising For Time Series

Forward Diffusion Process. The forward diffusion process for time series forecasting in *SimDiff* follows the methodology of the classical condtional denoising diffusion probabilistic models(DDPMs) (Ho, Jain, and Abbeel 2020), as is also demonstrated in the former diffusion-TS models (Tashiro et al. 2021; Shen and Kwok 2023; Shen, Chen, and Kwok 2024; Rasul et al. 2021)

In particular, the forward diffusion step for Y at step k is given by:

$$Y_k = \sqrt{\bar{\alpha}_k} Y_0 + \sqrt{1 - \bar{\alpha}_k}\epsilon, \quad k = 1, \ldots, K, \qquad (1)$$

where the noise matrix ϵ is sampled from $\mathcal{N}(0, I)$ with the same size as Y,and Y is an M-dimensional vector with a prediction horizon of H.

Backward Denoising Process. The backward denoising process aims to reconstruct the future time series Y through

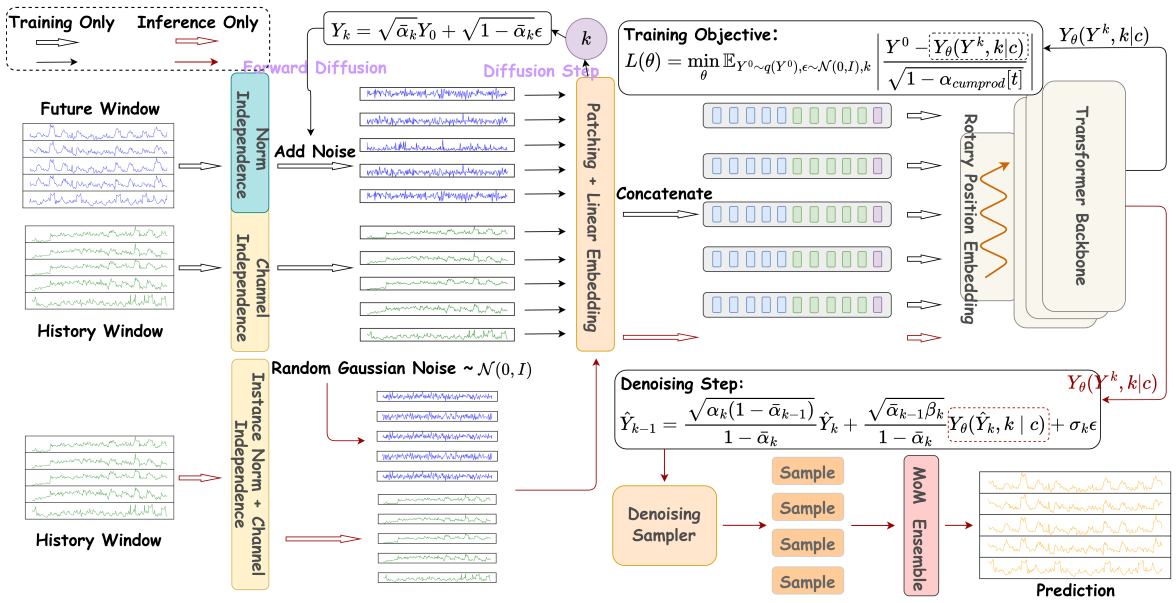


Figure 3: **SimDiff**: We have developed a streamlined end-to-end patch-based transformer diffusion model for time series forecasting tasks. Key components of our design include Normalization Independence, MoM ensembling, and the incorporation of RoPE.

a denoising transformer backbone. Each denoising step k is formulated as:

$$p_\theta(Y_{k-1} \mid Y_k, c) = \mathcal{N}\big(Y_{k-1};\, \mu_\theta(Y_k, k \mid c),\, \sigma_k^2 I\big)\,, \quad k = K, \ldots, 1, \tag{2}$$

where θ includes all parameters of the unified conditional denoising transformer, c is a condition derived from the past observations via conditional network, and the mean $\mu_\theta(Y_k, k \mid c)$ is computed as:

$$\mu_\theta(Y_k, k \mid c) = \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k} Y_k + \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k} Y_\theta(Y_k, k \mid c). \tag{3}$$

The denoising objective for learning θ is to minimize the gap between the predicted time series and the future ground truth, which will be introduced in detail later.

During inference, the initialization starts from $\hat{Y}_K \sim \mathcal{N}(0, I)$. For each denoising step k, the update is:

$$\hat{Y}_{k-1} = \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k} \hat{Y}_k + \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k} Y_\theta(\hat{Y}_k, k \mid c) + \sigma_k \epsilon. \tag{4}$$

where $\epsilon \sim \mathcal{N}(0, I)$ when $k > 1$, and $\epsilon = 0$ otherwise.

By iteratively applying these steps, We can then reconstruct the future time series from the noise.

Normalization Independence

Why. Past and future segments of a time series rarely share the same level or scale; normalising both with statistics of the past therefore biases the model when distribution drift occurs. Prior diffusion work nonetheless applies $\mathbf{X}_{\text{norm}} = (\mathbf{X} - \mu_{\mathbf{X}})/\sigma_{\mathbf{X}}$ and $\mathbf{Y}_{\text{norm}} = (\mathbf{Y} - \mu_{\mathbf{X}})/\sigma_{\mathbf{X}}$, implicitly assuming stationarity.

Algorithm 1: N.I. in training vs. inference

Require: past $\mathbf{X}$, (future $\mathbf{Y}$ for training), learnable $(\boldsymbol{\gamma}, \boldsymbol{\beta})$

Compute μ_X, σ_X from $\mathbf{X}$

Normalize $\mathbf{X}$: $\mathbf{X}_{\text{norm}} = \boldsymbol{\gamma} \cdot \dfrac{\mathbf{X} - \mu_X}{\sigma_X} + \boldsymbol{\beta}$

if training **then**

  Compute μ_Y, σ_Y from $\mathbf{Y}$

  Normalize $\mathbf{Y}$: $\mathbf{Y}_{\text{norm}} = \dfrac{\mathbf{Y} - \mu_Y}{\sigma_Y}$

  Corrupt $\mathbf{Y}_{\text{norm}}$ with diffusion noise; optimize loss

else

  Sample standard Gaussian noise $\boldsymbol{\epsilon}$

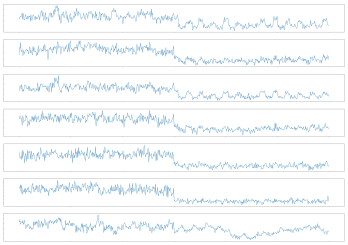
  Cond-DDPM denoising on $\boldsymbol{\epsilon}$ conditioned on $\mathbf{X}_{\text{norm}}$, producing $\widehat{\mathbf{Y}}_{\text{norm}}$

  De-normalize: $\widehat{\mathbf{Y}} = \sigma_X \cdot \dfrac{\widehat{\mathbf{Y}}_{\text{norm}} - \boldsymbol{\beta}}{\boldsymbol{\gamma}} + \mu_X$

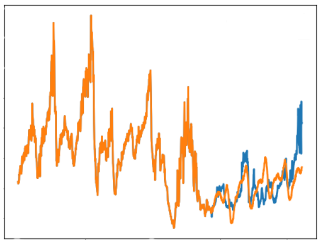
end if

What. NI breaks this coupling. *Past* samples are instance-normalised and rescaled by learnable $(\boldsymbol{\gamma}, \boldsymbol{\beta})$; *future* targets are normalised with their *own* statistics independently *only during training*. At test time, the model makes predictions from standard Gaussian noise, then de-normalises them using only past statistics and the learned affine parameters. This simple change stabilises training, and lets the network learn to infer future scale shifts from the past without the risk of data leakage.

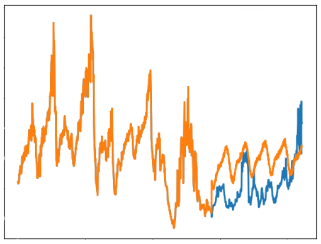
NI adds only a lightweight affine layer with negligible cost, but markedly improves robustness to distribution drift by better aligning training data with the diffusion's Gaussian



(a) Synthetic Data with Out of Distribution Drift



(b) Norm. Independence



(c) Z-Score Normalization

Figure 4: Normalization Independence: Enhancing Robust Training by Mitigating O.O.D.

prior (Fig. 4). Detailed ablations can be found in the later section.

Transformer Denoising Network

Our *SimDiff* model uses a transformer-based denoising network specifically designed for time series forecasting, moving away from complex stacked CNNs or U-Nets. This architecture leverages transformer's strengths in capturing temporal dependencies, essential for unstationary time series analysis.

Patch-based Tokenization. We utilize patching (Nie et al. 2023; Zhang and Yan 2023) to convert time series into overlapping tokens, with each patch acting as a token for local dependencies. A dense MLP transforms these patches into token embeddings, and diffusion timesteps are similarly processed into a time token, which is concatenated with the original tokens.

Rotary Position Embedding (RoPE). To better capture temporal order in long-term forecasting, we employ Rotary Position Embedding (RoPE) (Su et al. 2023). By encoding relative positional information through rotational transformations, RoPE preserves temporal dependencies and strengthens the attention mechanism's ability to focus across time, enhancing modeling of dynamic patterns.

Channel Independence and No Skip Connections. Skip connections, as used in U-ViT (Bao et al. 2023), help preserve spatial features but can amplify noise in time series, distorting diffusion distributions and degrading performance. To mitigate this, *SimDiff* removes skip connections for more stable modeling. Additionally, it employs channel independence (Nie et al. 2023), processing each channel separately to enhance efficiency and reduce complexity. This increases data volume, improves distribution learning, and enables global attention to focus on essential temporal patterns for more accurate forecasting.

These designs above enable *SimDiff* to balance simplicity and depth, ensuring robust and efficient time series prediction. More analysis on design choices can be found in the supplementary materials.

Median of Means Ensemble

Diffusion models inherently explore a wide range of possible probability traces, where extreme values are often unavoidable. To mitigate the influence of these outliers while still faithfully capturing the overall distribution trend, we intuitively need a proper strategy to derive a stable final estimate from the sampled probabilities. Therefore, we introduce the MoM estimator to transition from efficient distribution prediction to precise point estimation.

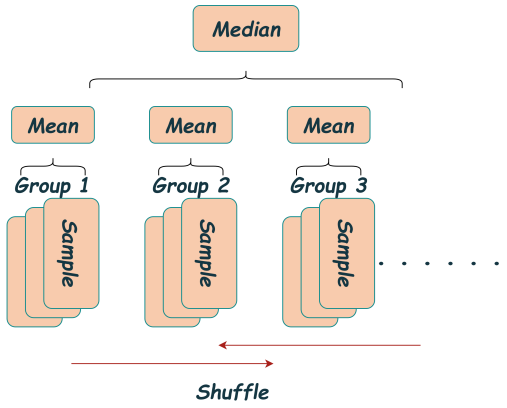


Figure 5: The MoM Ensemble

Originally a statistical method, we reintroduced MoM in our model as a reliable approach to estimate true values from multiple probabilistic samples. The MoM estimator divides a dataset of size n into K subsamples of size B, computes their means $\hat{\mu}_1, \ldots, \hat{\mu}_K$, and takes the median. To improve robustness, this process is repeated R times with shuffled data. The final estimator is the average of the R medians (just as the Figure 5 shows):

$$\hat{\mu}_{\text{MoM}} = \frac{1}{R}\sum_{r=1}^{R} \text{median}(\hat{\mu}_1^{(r)}, \ldots, \hat{\mu}_K^{(r)}) \tag{5}$$

Loss Function for Robust Training

Different from the literature, we choose a weighted mean absolute error (MAE) loss as the denoising objective.The loss function is expressed as:

$$L(\theta) = \min_{\theta} \mathbb{E}_{Y^0 \sim q(Y^0), \epsilon \sim \mathcal{N}(0,I), k} \left| \frac{Y^0 - Y_\theta(Y^k, k|c)}{\sqrt{1 - \alpha_{cumprod}[k]}} \right| \tag{6}$$

where Y_0 represents the target values at the initial timestep, $Y(\theta; k, c)$ denotes the model output,and $\alpha_{\text{cumprod}}[t]$ refers to

the cumulative product of $1-\alpha$ up to timestep t, which adjusts the normalization factor to account for the accumulated noise reduction over the diffusion steps.This scaling is critical as it allows the model to focus learning on periods with higher noise levels, ensuring robustness and accuracy in the denoising performance across varying conditions in the diffusion process.

3 Experiments

Probabilistic Forecasting Underpins Accurate Point Estimates

Diffusion models excel at modelling predictive distributions, a property that naturally supports accurate point forecasts. Estimating a probability distribution is inherently more challenging than predicting a single point, as it requires capturing both central tendencies and distributional boundaries. Effective point prediction thus indicates how well the model understands these distributional margins—a model capable of accurate distribution modelling is also more likely to produce precise point predictions. To quantitatively evaluate distributional quality, we primarily rely on the Continuous Ranked Probability Score (CRPS) and its aggregate variant, CRPS-sum, because they measure the full distance between predicted and empirical distributions.

Method	*ELEC.*		*TRAFFIC*		*TAXI*		*WIKI.*	
	C.	C.S.	C.	C.S.	C.	C.S.	C.	C.S.
GP-Copula	0.77	0.024	0.48	0.078	0.54	0.208	0.71	0.086
LSRP	0.45	0.024	0.43	0.078	0.49	0.175	0.59	0.078
LSTMMAF	0.41	0.023	0.45	0.069	0.46	0.161	0.55	0.067
CSDI	0.37	0.029	0.32	0.053	-	-	-	-
D3VAE	0.33	0.030	0.29	0.049	**0.35**	**0.130**	0.52	0.069
LDT	0.27	0.021	0.23	<u>0.040</u>	<u>0.36</u>	0.131	0.46	0.063
TMDM	<u>0.24</u>	0.023	<u>0.21</u>	0.042	0.44	0.172	<u>0.43</u>	<u>0.060</u>
MG-TSD	0.25	0.023	0.38	0.044	<u>0.36</u>	**0.130**	0.46	0.063
TSDiff	<u>0.24</u>	<u>0.020</u>	0.34	0.046	0.40	0.155	0.45	0.066
TimeGrad	0.34	0.025	0.35	0.050	0.40	0.137	0.55	0.064
SSSD	0.35	0.026	0.33	0.047	0.39	0.133	0.53	0.065
SimDiff	**0.22**	**0.019**	**0.16**	**0.039**	0.42	0.166	**0.41**	**0.057**

Table 1: Testing CRPS and CRPS-Sum in the multivariate setting. Best: **bold**, the second best: <u>underlined</u>. - denotes out of GPU memory.

Results. Table 1 demonstrates that SimDiff achieves strong probabilistic forecasting performance across datasets despite not being explicitly optimized for this probabilistic task. Instead, it leverages the inherent generative capacity of diffusion through a single, carefully designed diffusion–Transformer framework. This effective distributional fit establishes a solid foundation for the point forecasting gains discussed in Section 3. Compared to baselines, SimDiff robustly adapts to diverse series, effectively balancing prediction diversity and precision under dynamic distribution shifts.

In summary, SimDiff's robust probabilistic performance supports its SOTA point forecasting accuracy, illustrating that reliable distribution modelling and precise point estimation can coexist within a simple, end-to-end diffusion framework.

Accurate Long-Term Time Series Point Forecasting

Building on this strong distributional grasp, our Median of Means (MoM) estimator effectively translates probabilistic forecasts into robust and precise point predictions.

Results. Table 2 presents the MSE in the multivariate setting, where the proposed *SimDiff* model achieves the best performance on 6 out of 9 datasets, with particularly notable improvements on challenging datasets such as Norpool and ETTh1. Even on the remaining 3 datasets, *SimDiff* secures the second-best rankings, demonstrating its robustness across diverse data types. On large and complex datasets like Traffic and Electricity, where other diffusion models tend to underperform, *SimDiff* also achieves state-of-the-art or comparable results. Quantitatively, *SimDiff* reduces the MSE by an average of **8.3%** across all datasets compared to other diffusion models like mr-Diff, showing substantial improvements. These results underscore *SimDiff*'s consistent superiority over other competitive models, as reflected in its rankings. The full experiment settings can be found in the appendix material.

Standard Diffusion Transformer Suffices for Accurate Point Forecasting

In this section, we demonstrate that our simpler yet meticulously modified *SimDiff* model effectively leverages diffusion in end-to-end training, accurately capturing distribution shifts while maintaining a balance between sample diversity and accuracy, thereby improving performance through ensembling.

Baseline diffusions illustrate two extremes. TimeDiff (Shen and Kwok 2023) conditions on a *pre-trained* autoregressive model; this curbs variance but injects bias and limits flexibility. TimeGrad (Rasul et al. 2021) and CSDI (Tashiro et al. 2021), in contrast, optimize likelihood only: they generate highly diverse but poorly aligned samples, suffer from training instability and insufficient contexts, and often produce outliers that hurt point forecasts.

SimDiff's single Transformer denoiser avoids both pitfalls. A fully end-to-end training process allows the model to harness diffusion's strengths. By addressing the instability and misaligned distributions observed in TimeGrad and CSDI with tailored design and robust training objectives, our approach ensures that the samples are both diverse and meaningful for ensembling.

Table 3 reports single-shot MSE and ensemble MSE_E (underline indicates improvement); the right-most column lists sample variance averaged over features and horizons. SimDiff attains lower error and controlled variance, confirming that its samples capture temporal structure more faithfully than those of the baselines. Appendix material and Section 2 detail why the one-stage design produces richer samples than pre-training pipelines and how our design mitigates OOD drift, reduces bias, and stabilizes optimization. Together, these components let SimDiff deliver the most reliable distribution estimates and the strongest point forecasts after ensembling. In the following sections, we will delve deeper into how these modules contribute to our model's performance.

Method	*NorPool*	*Caiso*	*Traffic*	*Electricity*	*Weather*	*Exchange*	*ETTh1*	*ETTm1*	*Wind*	Rank
OURS	**0.534**$_{(1)}$	0.106$_{(2)}$	0.383$_{(2)}$	**0.145**$_{(1)}$	0.299$_{(2)}$	**0.015**$_{(1)}$	**0.394**$_{(1)}$	**0.322**$_{(1)}$	**0.880**$_{(1)}$	**1.33**
mr-Diff	0.645$_{(4)}$	0.127$_{(5)}$	0.474$_{(8)}$	0.155$_{(5)}$	**0.296**$_{(1)}$	0.016$_{(2)}$	0.411$_{(5)}$	0.340$_{(4)}$	0.881$_{(2)}$	4.00
TimeDiff	0.665$_{(6)}$	0.136$_{(8)}$	0.564$_{(10)}$	0.193$_{(7)}$	0.311$_{(4)}$	0.018$_{(8)}$	0.407$_{(3)}$	0.336$_{(2)}$	0.896$_{(3)}$	5.67
TimeGrad	1.152$_{(22)}$	0.258$_{(20)}$	1.745$_{(24)}$	0.736$_{(23)}$	0.392$_{(16)}$	0.079$_{(22)}$	0.993$_{(24)}$	0.874$_{(23)}$	1.209$_{(23)}$	21.89
TMDM	0.681$_{(8)}$	0.214$_{(14)}$	0.513$_{(9)}$	0.267$_{(14)}$	0.403$_{(18)}$	0.023$_{(13)}$	0.535$_{(13)}$	0.436$_{(14)}$	0.901$_{(5)}$	12.00
CSDI	1.011$_{(21)}$	0.253$_{(19)}$	-	-	0.356$_{(11)}$	0.077$_{(21)}$	0.497$_{(9)}$	0.529$_{(19)}$	1.066$_{(12)}$	16.00
SSSD	0.872$_{(14)}$	0.195$_{(11)}$	0.642$_{(13)}$	0.255$_{(13)}$	0.349$_{(10)}$	0.061$_{(18)}$	0.726$_{(20)}$	0.464$_{(15)}$	1.188$_{(21)}$	15.00
D3VAE	0.745$_{(12)}$	0.241$_{(18)}$	0.928$_{(19)}$	0.286$_{(17)}$	0.375$_{(13)}$	0.200$_{(24)}$	0.504$_{(11)}$	0.362$_{(10)}$	1.118$_{(17)}$	15.67
CPF	1.613$_{(25)}$	0.383$_{(22)}$	1.625$_{(23)}$	0.793$_{(24)}$	1.390$_{(25)}$	0.016$_{(2)}$	0.730$_{(21)}$	0.482$_{(17)}$	1.140$_{(19)}$	19.78
PSA-GAN	1.501$_{(24)}$	0.510$_{(24)}$	1.614$_{(22)}$	0.535$_{(22)}$	1.220$_{(23)}$	0.018$_{(8)}$	0.623$_{(19)}$	0.537$_{(20)}$	1.127$_{(18)}$	20.00
N-Hits	0.716$_{(10)}$	0.131$_{(6)}$	0.386$_{(4)}$	0.152$_{(4)}$	0.323$_{(6)}$	0.017$_{(7)}$	0.498$_{(10)}$	0.353$_{(8)}$	1.033$_{(9)}$	7.11
FiLM	0.723$_{(11)}$	0.179$_{(10)}$	0.628$_{(13)}$	0.210$_{(10)}$	0.327$_{(7)}$	0.016$_{(2)}$	0.426$_{(7)}$	0.347$_{(6)}$	0.984$_{(6)}$	8.00
Depts	0.662$_{(5)}$	0.106$_{(2)}$	1.019$_{(21)}$	0.319$_{(19)}$	0.761$_{(22)}$	0.020$_{(11)}$	0.579$_{(15)}$	0.380$_{(11)}$	1.082$_{(14)}$	13.33
NBeats	0.832$_{(13)}$	0.141$_{(9)}$	**0.373**$_{(1)}$	0.269$_{(15)}$	1.344$_{(24)}$	0.016$_{(2)}$	0.586$_{(17)}$	0.391$_{(12)}$	1.069$_{(13)}$	11.78
Scaleformer	0.983$_{(17)}$	0.207$_{(13)}$	0.618$_{(12)}$	0.195$_{(8)}$	0.462$_{(19)}$	0.036$_{(15)}$	0.613$_{(18)}$	0.481$_{(16)}$	1.359$_{(24)}$	15.78
PatchTST	0.547$_{(2)}$	0.110$_{(4)}$	0.385$_{(3)}$	0.147$_{(2)}$	0.302$_{(3)}$	0.016$_{(2)}$	0.405$_{(2)}$	0.337$_{(3)}$	1.017$_{(8)}$	3.22
FedFormer	0.873$_{(15)}$	0.205$_{(12)}$	0.591$_{(11)}$	0.238$_{(12)}$	0.342$_{(9)}$	0.133$_{(23)}$	0.541$_{(14)}$	0.426$_{(13)}$	1.113$_{(16)}$	13.89
Autoformer	0.940$_{(16)}$	0.226$_{(16)}$	0.688$_{(18)}$	0.201$_{(9)}$	0.360$_{(12)}$	0.056$_{(17)}$	0.516$_{(12)}$	0.565$_{(21)}$	1.083$_{(15)}$	15.11
Pyraformer	1.008$_{(20)}$	0.273$_{(21)}$	0.659$_{(15)}$	0.273$_{(16)}$	0.394$_{(17)}$	0.032$_{(14)}$	0.579$_{(15)}$	0.493$_{(18)}$	1.061$_{(11)}$	16.33
Informer	0.985$_{(18)}$	0.231$_{(17)}$	0.664$_{(16)}$	0.298$_{(18)}$	0.385$_{(14)}$	0.073$_{(20)}$	0.775$_{(23)}$	0.673$_{(22)}$	1.168$_{(20)}$	18.67
Transformer	1.005$_{(19)}$	0.206$_{(13)}$	0.671$_{(17)}$	0.328$_{(20)}$	0.388$_{(15)}$	0.062$_{(19)}$	0.759$_{(22)}$	0.992$_{(24)}$	1.201$_{(22)}$	19.00
SCINet	0.613$_{(3)}$	**0.095**$_{(1)}$	0.434$_{(7)}$	0.171$_{(6)}$	0.329$_{(8)}$	0.036$_{(15)}$	0.465$_{(8)}$	0.359$_{(9)}$	1.055$_{(10)}$	7.44
NLinear	0.707$_{(9)}$	0.135$_{(7)}$	0.430$_{(6)}$	0.147$_{(2)}$	0.313$_{(5)}$	0.019$_{(10)}$	0.410$_{(4)}$	0.349$_{(7)}$	0.989$_{(7)}$	6.33
DLinear	0.670$_{(7)}$	0.461$_{(23)}$	0.389$_{(5)}$	0.215$_{(11)}$	0.488$_{(20)}$	0.022$_{(12)}$	0.415$_{(6)}$	0.345$_{(5)}$	0.899$_{(4)}$	10.33
LSTMa	1.481$_{(23)}$	0.217$_{(16)}$	0.966$_{(20)}$	0.414$_{(21)}$	0.662$_{(21)}$	0.403$_{(25)}$	1.149$_{(25)}$	1.030$_{(25)}$	1.464$_{(25)}$	22.33

Table 2: Testing MSE in multivariate settings. The number in brackets indicates the rank. Best: **bold**, the second best: underlined. - denotes out of GPU memory. The results are the average of 5 runs. Our results are stable with a variance of 5 runs less than 1e-5.

Models	$SimDiff$ (1 Stage, Point)			$TimeDiff$ (2 Stage, Point)			$CSDI$ (1 Stage, Prob)			$TimeGrad$ (1 Stage, Prob)		
Datasets	MSE	MSE_E	$Var.$	MSE	MSE_E	$Var.$	MSE	MSE_E	$Var.$	MSE	MSE_E	$Var.$
$ETTh1$	0.408	0.394	**0.012**	0.407	0.405	0.00081	0.497	0.494	0.017	0.993	0.990	0.013
$ETTm1$	0.333	0.322	**0.012**	0.336	0.342	0.00072	0.529	0.527	0.039	0.874	0.891	0.061
$Weather$	0.317	0.299	**0.005**	0.311	0.309	0.00064	0.356	0.359	0.029	0.392	0.393	0.034
$NorPool$	0.548	0.534	**0.011**	0.665	0.670	0.00062	1.011	1.210	0.084	1.152	1.189	0.085
$Wind$	0.901	0.880	**0.018**	0.896	0.891	0.00074	1.066	1.013	0.012	1.209	1.201	0.037

Table 3: Assessing the Diversity of Samples From Our Proposed Method

Significance of Training-Specific Normalization Independence

N.I.	*NorPool*	*Elec.*	*Traffic*	*ETTm1*	*Weather*	*Wind*	*Exchange*
✓	**0.534**	**0.145**	**0.383**	**0.322**	**0.299**	**0.880**	**0.015**
✗	0.555	0.151	0.389	0.327	0.328	0.891	0.019

Table 4: Ablation study showing the essential role of N.I.

We conduct an ablation study to verify the impact of our proposed N.I. . This technique is designed to mitigate out-of-distribution (OOD) issues by reducing the distributional bias between past observations and future targets.

The results, as shown in Table 4, indicate that N.I. consistently improves the model's performance across various datasets, especially for the dataset with severe O.O.D. like *Weather* and *NorPool*, as also can be seen in Figure 4.

These results underscore the significance Normalization Independence. During training we "Gaussianise" each segment with its *own* statistics, which stabilises optimisation, while a learnable affine layer on the past sequence predicts the future shift / scale at test time. This simple change cuts bias, handles OOD drift, and lowers MSE—ultimately contributing to the success of our ensemble strategy. Also, we provide a preliminary proof in the appendix material.

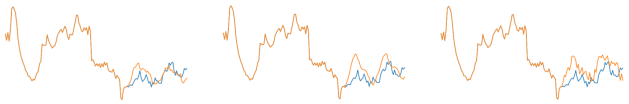
Ensemble Enhances SimDiff as an Effective Point Estimator

We ablated three strategies—single-sample inference, simple averaging, and our Median-of-Means ensemble—to gauge their effect on accuracy and stability. All three ensembles lift SimDiff's accuracy, but MoM delivers the largest gain. Simple averaging smooths away high-frequency detail, whereas MoM effectively captures the true distribution of the data, retaining subtle temporal patterns rather than a smoothed trajectory (Figure 6). MoM's robustness to outliers and heavy-tailed

Ensemble	*ETTh1*	*Weather*	*Wind*	*Caiso*
MoM	**0.394**	**0.299**	**0.880**	**0.106**
Avg.	0.398	0.305	0.887	0.109
1 Inf.	0.408	0.317	0.901	0.110

Table 5: Impact of Various Ensemble Methods on Boosting Effectiveness

noise significantly improves prediction stability, leading to the lowest MSE across all datasets. Theoretically, MoM also offers stronger statistical guarantees, providing tighter concentration bounds in finite-sample regimes, as formally proven in Appendix material. By replacing the single deterministic



(a) MoM (b) Simple Avg. (c) 1 Inference

Figure 6: Power of MoM

pass used in pretrain–conditioned diffusions (Shen and Kwok 2023; Shen, Chen, and Kwok 2024) with MoM ensemble, SimDiff exploits the full probalilistic trace of diffusion while retaining numerical stability.

Inference Efficiency

As shown in Table 6, the simple and effective design of our model brings over 90% improvement in inference speed. *SimDiff* ranks first in single-sample inference time among all evaluated diffusion models, underscoring the efficiency of our single-stage Transformer architecture.

	$H = 96$	$H = 168$	$H = 192$	$H = 336$	$H = 720$
SimDiff	**0.22**	**0.24**	**0.30**	**0.33**	**0.46**
TimeDiff	4.73	5.21	5.84	6.91	8.40
mr-Diff	7.02	7.41	8.10	8.95	10.92
CSDI	67.02	89.36	108.41	295.15	379.80
TMDM	111.23	152.92	187.24	252.18	483.39
SSSD	135.92	204.25	236.32	368.29	886.56
TimeGrad	294.85	573.05	621.70	1071.45	2312.26

Table 6: Inference time (in ms) of various time series diffusion models in single inference process with different prediction horizons (H) on the **ETTh1**.

This efficiency is particularly notable when compared to the next fastest model, TimeDiff. While TimeDiff reduces some computational load by pre-training a regressor, it still relies on a more complex U-Net for its denoising block. In contrast, our streamlined Transformer design avoids this architectural overhead. Consequently, even when *SimDiff* performs multiple inference passes for our MoM ensemble, the total computational time remains highly competitive, striking an balance between accuracy and practical efficiency.

Further analyses, including ablations on model components and parameter sensitivity, can be found in the Appendix.

4 Related Works

Diffusion Models for Time Series Forecasting

Diffusion models, initially developed for generation (DDPM (Ho, Jain, and Abbeel 2020)), are now applied to time series forecasting. Early autoregressive models like TimeGrad (Rasul et al. 2021) suffered from cumulative errors and slow inference. Non-autoregressive approaches (CSDI (Tashiro et al. 2021), SSSD (Alcaraz and Strodthoff 2022)) address these issues but still experience boundary artifacts and computational overhead. Recent methods such as TimeDiff (Shen and Kwok 2023) and mr-Diff (Shen, Chen, and Kwok 2024) use conditional and multiresolution strategies to enhance accuracy. However, their reliance on pretrained bases like DLinear (Zeng et al. 2023) limits adaptability to complex series and reduces generative flexibility.

Transformers and Diffusion Transformers

Transformers for Time Series Forecasting. Transformers significantly improve forecasting through long-range dependency modeling, such as Informer (Zhou et al. 2021), Autoformer (Wu et al. 2021), PatchTST (Nie et al. 2023), and FEDformer (Zhou et al. 2022b).

Diffusion Transformers. Recent transformer-based diffusion models, such as U-ViT (Bao et al. 2023) and DiT (Peebles and Xie 2023), improve scalability and generative performance. However, designed primarily for static vision data, they face challenges capturing temporal dependencies in time series.

Other Deep Learning Models

Other deep learning approaches include basis-expansion methods (FiLM (Zhou et al. 2022a), NBeats (Oreshkin et al. 2019), DUET (Qiu et al. 2025c),DAG (Qiu et al. 2025d),DBLoss (Qiu et al. 2025b)), hybrid recursive convolutional models (SCINet (Liu et al. 2022)), CNN-based approaches (TimesNet (Wu et al. 2022), DeepGLO (Sen, Yu, and Dhillon 2019)), and RNN-based models (LSTNet (Lai et al. 2018), DeepAR (Salinas et al. 2020)). These models balance interpretability, scalability, and accuracy differently, each with inherent trade-offs.

5 Conclusion

In this work, we present *SimDiff*, a simple yet effective diffusion model for time series point forecasting. SimDiff addresses the twin challenges of providing sufficient contextual bias for stability and accuracy and balancing diversity with accuracy, integrating a tailored Transformer within a diffusion framework *without relying on any external pre-trained or jointly trained models*. Its fully end-to-end training achieves state-of-the-art point forecasting and competitive probabilistic performance. Normalization independence and the Median-of-Means estimator further improve robustness to noise and distribution shifts. Owing to its simplicity, SimDiff also delivers faster inference than prior diffusion-based models. We hope these results stimulate further research into diffusion techniques for time series forecasting across diverse domains.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) grants W2441021, 72371172, 72342023, and 71929101. We also thank all authors for their sincere contributions and discussions.

References

Alcaraz, J. M. L.; and Strodthoff, N. 2022. Diffusion-based time series imputation and forecasting with structured state space models. Technical report, arXiv.

Bao, F.; Nie, S.; Xue, K.; Cao, Y.; Li, C.; Su, H.; and Zhu, J. 2023. All are Worth Words: A ViT Backbone for Diffusion Models. In *CVPR*.

Böse, J.-H.; Flunkert, V.; Gasthaus, J.; Januschowski, T.; Lange, D.; Salinas, D.; Schelter, S.; Seeger, M.; and Wang, Y. 2017. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, 10(12): 1694–1705.

Courty, P.; and Li, H. 1999. Timing of seasonal sales. *The Journal of Business*, 72(4): 545–572.

Friedman, M. 1962. The interpolation of time series by related series. *J. Amer. Statist. Assoc*.

Gao, J.; Song, X.; Wen, Q.; Wang, P.; Sun, L.; and Xu, H. 2020. RobustTAD: Robust time series anomaly detection via decomposition and convolutional neural networks. *KDD Workshop on Mining and Learning from Time Series (KDD-MileTS'20)*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 6840–6851. Curran Associates, Inc.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR Conference on Research & Development in Information Retrieval*.

Li, Y.; Chen, W.; Hu, X.; Chen, B.; baolin sun; and Zhou, M. 2024. Transformer-Modulated Diffusion Models for Probabilistic Multivariate Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2023. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. arXiv:2205.14415.

Nie, Y.; H. Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; and Yang, B. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. In *Proc. VLDB Endow.*, 2363–2377.

Qiu, X.; Li, Z.; Qiu, W.; Hu, S.; Zhou, L.; Wu, X.; Li, Z.; Guo, C.; Zhou, A.; Sheng, Z.; Hu, J.; Jensen, C. S.; and Yang, B. 2025a. TAB: Unified Benchmarking of Time Series Anomaly Detection Methods. In *Proc. VLDB Endow.*, 2775–2789.

Qiu, X.; Wu, X.; Cheng, H.; Liu, X.; Guo, C.; Hu, J.; and Yang, B. 2025b. DBLoss: Decomposition-based Loss Function for Time Series Forecasting. In *NeurIPS*.

Qiu, X.; Wu, X.; Lin, Y.; Guo, C.; Hu, J.; and Yang, B. 2025c. DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting. In *SIGKDD*, 1185–1196.

Qiu, X.; Zhu, Y.; Li, Z.; Cheng, H.; Wu, X.; Guo, C.; Yang, B.; and Hu, J. 2025d. DAG: A dual causal network for time series forecasting with exogenous variables. *arXiv preprint arXiv:2509.14933*.

Rasul, K.; Seward, C.; Schuster, I.; and Vollgraf, R. 2021. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191.

Sen, R.; Yu, H.-F.; and Dhillon, I. S. 2019. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32.

Shen, L.; Chen, W.; and Kwok, J. 2024. Multi-Resolution Diffusion Models for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Shen, L.; and Kwok, J. 2023. Non-autoregressive Conditional Diffusion Models for Time Series Prediction. arXiv:2306.05043.

Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; and Liu, Y. 2023. RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv:2104.09864.

Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Neural Information Processing Systems*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Neural Information Processing Systems*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In *AAAI Conference on Artificial Intelligence*.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *The Eleventh International Conference on Learning Representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Sun, L.; Yao, T.; Jin, R.; et al. 2022a. FiLM: Frequency improved Legendre memory model for long-term time series forecasting. In *Neural Information Processing Systems*.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022b. FEDformer: frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*.