SFDIFF : DIFFUSION MODEL WITH SELF-GENERATION FOR PROBABILISTIC FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have emerged as an effective approach for time-series probabilistic forecasting, aiming to generate future observations based on historical data through a denoising process. In this paper, we introduce self-generation technique designed to enhance the performance of conditional generation in time-series forecasting. Self-generation involves synthesizing not only future observations, but also historical data itself conditioned on the given historical context. While noise is often introduced during the observation process, our method can reduce the amount of noise in observed historical data, thereby enhancing forecasting accuracy. Additionally, to further boost forecasting performance, we incorporate classifier-free generation methods into conditional generation for time-series forecasting. In the experiment, we demonstrate that our method outperforms other condition generation methods.

1 INTRODUCTION

Time series forecasting is a critical problem in the fields of machine learning and deep learning, focusing on predicting future observations based on historical data. This process requires learning the relationship patterns between past and future data during training and, at inference time, reconstructing patterns that best fit the given historical data from the learned relationships. Time-series forecasting problems are essential across many domains, spanning fields such as physics, climate, healthcare, and finance (Lim & Zohren, 2021; Torres et al., 2021a; Masini et al., 2023).

Over the years, numerous deep learning methods have been proposed to address time-series forecasting problems (Lim & Zohren, 2021; Torres et al., 2021b; Miller et al., 2024). Among these, diffusion-based conditional generative methods have shown strong forecasting performance. (Rasul et al., 2021; Tashiro et al., 2021; Yan et al., 2021) These methods involve training neural networks to approximate the score values ($\nabla_{\mathbf{x}_t^{\text{pred}}} \log p(\mathbf{x}_t^{\text{pred}}|\mathbf{x}^{\text{hist}})$) on diffusion step t, where $p(\mathbf{x}^{\text{pred}}|\mathbf{x}^{\text{hist}})$ represents the conditional distribution of future observations given historical data. Using these trained networks, future observations are synthesized through a denoising process, including a reverse Stochastic Differential Equation (SDE) process or ancestral sampling, conditioned on the historical data.

However, due to measurement errors or the occurrence of anomalies, the conditional historical observations may contain noise, which can negatively impact forecasting performance (Rožanec et al., 2021). To address this, we propose Self-Generation, which extends the denoising process of score-based generation to both future observations and conditional historical data, effectively reducing the inherent noise in the conditional inputs. Specifically, our training objective is not to approximate $\nabla_{\mathbf{x}_t^{\text{pred}}} \log p(\mathbf{x}_t^{\text{pred}}|\mathbf{x}^{\text{hist}})$ but rather $\nabla_{\mathbf{x}_t^{\text{total}}} \log p(\mathbf{x}_t^{\text{total}}|\mathbf{x}^{\text{hist}})$, where $\mathbf{x}^{\text{total}}$ represents an union of two time-series $\mathbf{x}^{\text{pred}}$ and $\mathbf{x}^{\text{hist}}$.

Modeling the total sequence within the diffusion process has two major advantages. First, high-frequency anomalies in the data are effectively mitigated after a few forward diffusion steps (Choi et al., 2022; Yang et al., 2023). Second, during the generation process, predictions and conditions are interdependently generated. This interdependence enables the reverse diffusion process to minimize the impact of anomalies in the historical data, as predictions are generated using predominantly non-anomalous conditions and purified conditions informed by the generated predictions. To further leverage the noise-reduction capabilities of Self-Generation and focus on prediction generation,

we introduce an imbalanced weighting scheme in the loss function between the past and future components, demonstrating the importance of our careful training design.

To further enhance the forecasting performance of our score-based conditional generation, we integrate classifier-free generation introduced by Ho & Salimans (2022), with score-based conditional generation for time-series forecasting into a unified framework. Our results show that incorporating classifier-free methods into conditional generation significantly reduces forecasting errors, and this positive impact is further amplified when combined with Self-Generation. Finally, in the experimental section, we demonstrate that our proposed method achieves state-of-the-art forecasting performance across 2 toy datasets and 5 real datasets, outperforming 12 baseline methods. To summarize, our contributions can be outlined as follows:

1. We propose Self-Generation as a novel approach for score-based conditional generation in time-series forecasting. Self-Generation reduces noise in conditional historical observations by synthesizing both future and historical observations through a denoising process, thereby enhancing forecasting performance.
2. We theoretically show how generating the entire time-series (rather than only the future part) within the diffusion process enables a noise-purification mechanism, and formally derive a corresponding total-sequence score-matching objective. In Section 3, we present Theorems 1–2 to justify this extended generation strategy and detail the resulting loss function, which emphasizes accurate predictions informed by denoised historical conditions.
3. To further enhance the forecasting performance of Self-Generation, we adapt the classifier-free generation approach to suit time-series forecasting scenarios.
4. Out of 2 toy examples and 4 real datasets, our score-based conditional generative method with Self-Generation achieves state-of-the-art performance in all cases, compared to 12 baselines, including methods based on variational autoencoder (VAE), diffusion, and gaussian process (GP).

2 Preliminary and Problem Statement

2.1 Diffusion models

Generative models aim to synthesize realistic data, such as images, by learning the underlying probability distribution of the data (Oussidi & Elhassouny, 2018; Harshvardhan et al., 2020; Cao et al., 2024). Among various generative approaches, diffusion models have gained prominence defeating generative adversarial network (GAN), in terms of generating high-quality images with more stable training (Dhariwal & Nichol, 2021; Song et al., 2020; Ho et al., 2020; Cao et al., 2024). Diffusion models operate through following two-step process: i) **Noising** step, which means gradually adding noise to an image, transforming it into Gaussian noise, ii) **Denoising** step, which means recovering the original image from the noisy version, where the noise is sampled from a specific distribution, typically a normal distribution (Yang et al., 2023).

Initially, the denoising process was designed to reverse the noising process by adding noise in the opposite direction at each step. This process is derived from minimizing the Kullback-Leibler (KL) divergence between the joint probability of noising and denoising step, leading to an inequality involving the negative log-likelihood, similar to the variational autoencoder (VAE) framework. This approach is called Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020).

Given original image $\mathbf{x} \sim p(\mathbf{x})$ and the length of noising and denoising step T, DDPMs add noise to the image according to the transition kernel: $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$, where $t \in \{1, 2, ..., T\}$ and $\beta_t \in (0, 1)$ is a hyperparameter. With sufficiently large T, $\mathbf{x}_t$ converges to a normal distribution. DDPMs then train a corresponding learnable denoising kernel $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(t, \mathbf{x}_t), \Sigma(t, \mathbf{x}_t))$, where the denoising process aims to reverse the added noise.

As a follow-up research, Song et al. (2020) have generalized diffusion models from discrete-time processes to continuous Stochastic Differential Equation (SDE) formulations, introducing Variance Exploding (VE), Variance Preserving (VP), and sub-VP processes. In this framework, the noising and denoising processes of diffusion models are reinterpreted as forward and reverse SDEs, respectively:

$$d\mathbf{x} = \mathbf{f}(t, \mathbf{x})dt + g(t)d\mathbf{w}$$
$$d\mathbf{x} = [\mathbf{f}(t, \mathbf{x}) - g(t)^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})]dt + g(t)d\bar{\mathbf{w}}$$

, where $t \in [0, 1]$, f is an affine and $\mathbf{w}, \bar{\mathbf{w}}$ represent forward and backward Brownian motion, respectively. Among these, the VP process is particularly notable for its connection to DDPMs, where: $\mathbf{f}(t, \mathbf{x}) = -\frac{1}{2}\beta(t)\mathbf{x}, g(t) = \sqrt{\beta(t)}$. They demonstrate that score based generative models train score network $s_\theta(\cdot, \cdot)$ to learn a gradient of log likelihood, score function, by using following score matching loss:

$$L_{SM}(\theta) = \mathbb{E}_{t,\mathbf{x}_t}[\lambda(t)||s_\theta(t, \mathbf{x}_t) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)||^2],$$

where $\mathbf{x}_t \sim p(\mathbf{x}_t)$. However, directly using score matching loss is computationally prohibitive since calculating exact score function of $\mathbf{x}_t$ needs statistical method (Hyvärinen, 2005; Song et al., 2020). Thanks to specific formulation of $\mathbf{f}$ and g, we can derive a following denoising score matching loss, which can be calculated by using given formula (Vincent, 2011; Øksendal, 2014):

$$L_{DSM}(\theta) = \mathbb{E}_{t,\mathbf{x},\mathbf{x}_t}[\lambda(t)||s_\theta(t, \mathbf{x}_t) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x})||^2],$$

where $\mathbf{x} \sim p(\mathbf{x}), \mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x})$. We can directly derive the equivalence between $L_{SM}(\theta)$ and $L_{DSM}(\theta)$ by considering the structure of the forward SDE. The drift term $\mathbf{f}(\cdot, \cdot)$ is affine and the diffusion term $g(\cdot)$ depends solely on the diffusion step. This results in the conditional probability $p(\mathbf{x}_t|\mathbf{x})$ being represented as a Gaussian distribution, $\mathcal{N}(\mathbf{x}_t; \mu_t(\mathbf{x}), \sigma_t)$ (Øksendal, 2014). Therefore, we can compute the gradient of log likelihood, $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x})$, as: $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}) = -(\mathbf{x}_t - \mathbf{x})/\sigma_t^2 = -\epsilon/\sigma_t$, where the reparametrization trick is used on $\mathbf{x}_t = \mu_t(\mathbf{x}_t) + \sigma_t\epsilon$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

By linking the SDE and ODE formulations, Song et al. (2021) proposed setting $\lambda(t) = g^2(t)$ to ensure the following inequality:

$$-\mathbb{E}_{\mathbf{x}}[\log p(\mathbf{x})] \leq L_{SM}(\theta) + C_1,$$

where $L_{SM}(\theta) = \mathbb{E}_{t,\mathbf{x}_t}\left[g^2(t)\|s_\theta(t, \mathbf{x}_t) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)\|_2^2\right]$ and C_1 is a constant. Based on this, we adopt $g^2(\cdot)$ as the default weighting in our experiments.

Once the score network is trained, diffusion models proceed with the denoising step. At this stage, there are two main sampling strategies: the predictor-corrector (PC) sampler and a deterministic sampler based on the probability flow ordinary differential equation (ODE). In here, we explain PC sampler that is used in our experiment. The PC sampler works by first estimating the next step using a known numerical SDE solver, which is called *predictor*. Then refining the estimate with a score-based MCMC strategy, which is named of *corrector*. A representative example of predictor is an Euler-Maruyama sampling predictor, which is a discretization of backward SDE:

$$\mathbf{x}_{t-1} = [\mathbf{f}(t, \mathbf{x}_t) - g(t)^2 s_\theta(t, \mathbf{x}_t)]\Delta t + g(t)\Delta w,$$

, where $t \in [1, 0]$, Δt is a time interval and $\Delta w \sim \mathcal{N}(\mathbf{0}, \Delta t\mathbf{I})$.

Song et al. (2020) achieved state-of-the-art results through extensive hyperparameter tuning of various SDEs, predictors and correctors. However, for our experiments, we adopt the VP SDE and use an Euler-Maruyama sampling predictor without corrector, which is a default setting of it (Song et al., 2020). This allows us to isolate the performance of SFdiff from other factors, ensuring that other control variables remain fixed.

2.2 Time-series Forecasting

Time-series forecasting involves predicting future values based on historical data (Lim & Zohren, 2021; Torres et al., 2021b; Miller et al., 2024). Specifically, given a historical sequence $\mathbf{x}^{1:N}$, the task is to forecast the future sequence $\mathbf{x}^{N+1:N+T}$, where N represents the length of the historical data, and T represents the length of the prediction. Each data point $\mathbf{x}$ belongs to $\mathbb{R}^d$. For clarity, we define $\mathbf{x}^{\text{hist}}$ by a sequence of history, $\mathbf{x}^{1:N}$, $\mathbf{x}^{\text{pred}}$ by a future values, $\mathbf{x}^{N+1:N+T}$, and $\mathbf{x}^{\text{total}}$, a total sequence $\mathbf{x}^{1:N+T}$. Time-series forecasting has been widely researched improve the accuracy of future predictions. However, the complex, intertwined characteristics of time-series data make it difficult to fully capture and understand its underlying patterns.

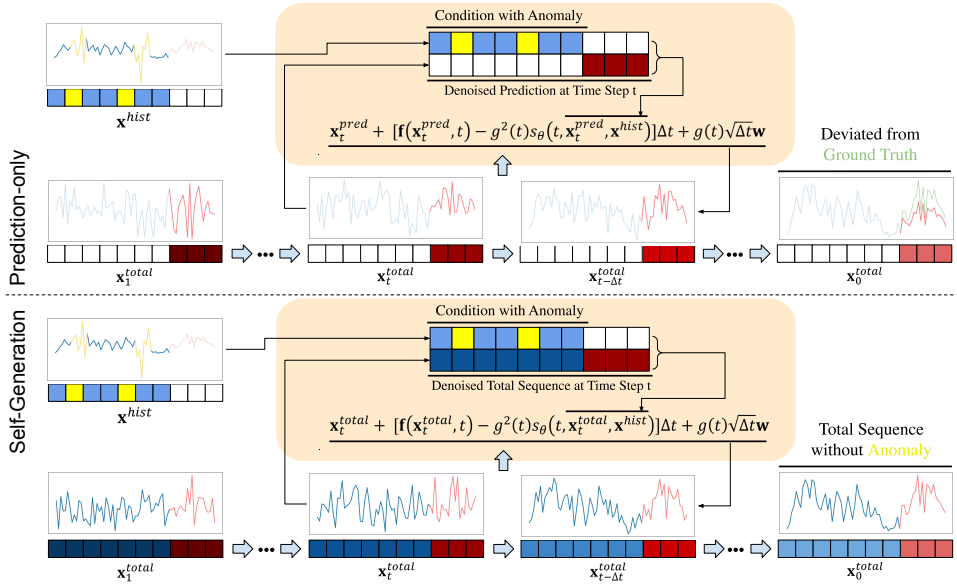

Figure 1: **Overall visualization of the sampling process of SFdiff.** Comparison between prediction-only generation (top) and self-generation (bottom). Even when anomalies (highlighted in yellow) exist in the historical data, SFdiff iteratively applies reverse diffusion to produce an anomaly-free sequence containing both historical and robust future observations, whereas prediction-only generation deviates from the ground truth. Although reverse SDE is shown here, other sampling strategies can be used interchangeably.

To address this challenge, researchers have increasingly turned to generative models, which aim to model the conditional likelihood of time-series data and provide a more comprehensive understanding of its structure. As a result, many time-series diffusion models were appeared, which generally aim to learn conditional distribution of prediction given history sequence, $p(\mathbf{x}^{\text{pred}}|\mathbf{x}^{\text{hist}})$ (Rasul et al., 2021; Tashiro et al., 2021). We provide a detailed explanation of their contributions and the rationale behind their target selection in Section 6. Therefore, those who apply DDPM methods to forecasting problem optimize the following equation:

$$L_{SM}^{\text{pred}}(\theta) = \mathbb{E}_{t,\mathbf{x}_t^{\text{pred}}}\left[\lambda(t)\|s_\theta(t,\mathbf{x}_t^{\text{pred}},\mathbf{x}^{\text{hist}}) - \nabla_{\mathbf{x}_t^{\text{pred}}}\log p(\mathbf{x}_t^{\text{pred}}|\mathbf{x}^{\text{hist}})\|_2^2\right].$$

3 Proposed Method

In this section, we analyze existing prediction methods and propose a novel self-generation approach that reconstructs the entire sequence consisting of purified condition given noised history data.

3.1 Diffusion Model with Self-Generation

Current methods train diffusion models by optimizing the conditional probability $p(\mathbf{x}^{\text{pred}}|\mathbf{x}^{\text{hist}})$ through well-known DDPM loss (Rasul et al., 2021; Tashiro et al., 2021; Kollovieh et al., 2023b):

$$L_{DDPM}^{pred}(\theta) = \mathbb{E}_{t,\epsilon,\mathbf{x}^{pred}}[\lambda(t)||\epsilon - \epsilon_\theta(t,\mathbf{x}_t^{pred},\mathbf{x}^{hist})||^2]$$

Song et al. (2021) proved that the target value of DDPM ($\log p(\mathbf{x}^{\text{pred}}|\mathbf{x}^{\text{hist}})$) can be optimized in the perspective of score-based approach:

$$-\mathbb{E}_{\mathbf{x}^{\text{total}}}\log p(\mathbf{x}^{\text{pred}}|\mathbf{x}^{\text{hist}}) \leq \frac{1}{2}\cdot L_{SM}^{\text{pred}}(\theta) + C_1,$$

where

$$L_{SM}^{\text{pred}}(\theta) = \mathbb{E}_{t,\mathbf{x}^{\text{hist}},\mathbf{x}_t^{\text{pred}}}\big[\lambda(t)\|s_\theta(t,\mathbf{x}_t^{\text{pred}},\mathbf{x}^{\text{hist}}) - \nabla_{\mathbf{x}_t^{\text{pred}}}\log p(\mathbf{x}_t^{\text{pred}}|\mathbf{x}^{\text{hist}})\|_2^2\big].$$

While intuitive, this approach is sensitive to noisy conditions: if $\mathbf{x}^{\text{hist}}$ contains adversarial noise, the resulting diffusion model may produce degraded samples. To address this, we propose SFdiff : Diffusion Model with Self-Generation for Probabilistic Forecasting, which generates the entire time-series instead of only synthesizing prediction sequence. SFdiff learns mathematically same conditional distribution $p(\mathbf{x}^{\text{total}}|\mathbf{x}^{\text{hist}}) = p(\mathbf{x}^{\text{pred}}|\mathbf{x}^{\text{hist}})$ by optimizing the following inequality:

$$-\mathbb{E}_{\mathbf{x}^{\text{total}}}\log p(\mathbf{x}^{\text{total}}|\mathbf{x}^{\text{hist}}) \le \frac{1}{2}\cdot L_{SM}^{\text{total}}(\theta) + C_1,$$

where

$$L_{SM}^{\text{total}}(\theta) = \mathbb{E}_{t,\mathbf{x}^{\text{hist}},\mathbf{x}_t^{\text{total}}}\big[\lambda(t)\|s_\theta(t,\mathbf{x}_t^{\text{total}},\mathbf{x}^{\text{hist}}) - \nabla_{\mathbf{x}_t^{\text{total}}}\log p(\mathbf{x}_t^{\text{total}}|\mathbf{x}^{\text{hist}})\|_2^2\big].$$

Generating the total time-series offers two key advantages:

1. **Noise Purification:** Diffusion models inherently denoise conditions. By learning the conditional distribution of the total time-series, the model can generate a purified sequence consistent with the full distribution.
2. **Interdependent Predictions:** During total sequence generation, predictions are both influenced by and influence historical data through the diffusion process.

The following theorem demonstrates how the self-generation preserve robustness between noisy and clean conditions:

Theorem 3.1 (Self-generation yields smaller sensitivity)**.** *Assume (A1)–(A3). Let* $H(t) := \int_t^1 g(s)^2\,ds$ *and* $G := H(0) = \int_0^1 g(s)^2\,ds > 0$*. Then for noised input condition* $\boldsymbol{c}_s \equiv \boldsymbol{x}_s^{hist}$*,*

(prediction-only) $$\|\mathbf{x}_0^{pred} - \mathbf{x}_0^{pred\,\prime}\| \le L\int_0^1 g(s)^2\,\|\mathbf{c}_s - \mathbf{c}_s'\|\,ds \;\le\; L\,G\sup_{s\in[0,1]}\|\mathbf{c}_s - \mathbf{c}_s'\|,$$

(total-sequence) $$\|\mathbf{x}_0^{pred} - \mathbf{x}_0^{pred\,\prime}\| \le L\int_0^1 g(s)^2\,e^{-m_x\,H(s)}\,\|\mathbf{c}_s - \mathbf{c}_s'\|\,ds \;=\; \frac{L}{m_x}\big(1-e^{-m_x G}\big)\sup_s\|\mathbf{c}_s - \mathbf{c}_s'\|.$$

Consequently,

$$\frac{L}{m_x}\big(1-e^{-m_x G}\big) \;<\; L\,G \quad\Rightarrow\quad \|\mathbf{x}_0^{pred} - \mathbf{x}_0^{pred\,\prime}\| \;<\; \|\mathbf{x}_0^{pred} - \mathbf{x}_0^{pred\,\prime}\|_{\textit{pred-only}}.$$

Thus the total-sequence *conditional score produces forecasts with strictly smaller sensitivity to condition perturbations than the* prediction-only *score.*

On the Theorem 3.1, we distinguish $\mathbf{x}^{\text{hist}}$ from condition and total sequence by denoting that of condition as $\mathbf{c}$. The above theorem shows that when noise is injected to history data, self-generation always takes lower bound than prediction-only method.

It is well known that directly computing $L_{SM}^{\text{pred}}(\theta)$ and $L_{SM}^{\text{total}}(\theta)$ is computationally prohibitive due to the need for statistical methods (Hyvärinen, 2005; Song et al., 2020). Therefore, we derive the denoising score-matching losses to train the score network s_θ and guarantee its convergence:

Theorem 3.2. *For each* $L_{SM}^{pred}(\theta)$ *and* $L_{SM}^{total}(\theta)$*, its denoising score matching are represented as follows:*

$$L_{DSM}^{pred}(\theta) = \mathbb{E}_{t,\mathbf{x}^{total},\mathbf{x}_t^{total}}[\lambda(t)||s_\theta(t,\mathbf{x}_t^{pred},\mathbf{x}^{hist}) - \nabla_{\mathbf{x}_t^{pred}} logp(\mathbf{x}_t^{total}|\mathbf{x}^{total})||_2^2]$$
$$L_{DSM}^{total}(\theta) = \mathbb{E}_{t,\mathbf{x}^{total},\mathbf{x}_t^{total}}[\lambda(t)||s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}) - \nabla_{\mathbf{x}_t^{total}} logp(\mathbf{x}_t^{total}|\mathbf{x}^{total})||_2^2]$$

Therefore, these models aim same conditional score function since $\nabla_{\mathbf{x}_t^{total}} logp(\mathbf{x}_t^{total}|\mathbf{x}^{total}) = \nabla_{[\mathbf{x}_t^{hist},\mathbf{x}_t^{pred}]} logp(\mathbf{x}_t^{total}|\mathbf{x}^{total})$.

Beyond using $L_{DSM}^{\text{total}}(\theta)$, we place additional emphasis on the prediction portion of the sequence. In designing SFdiff, we aim to ensure that it generates a predictive sequence that takes past history into account but is not overly dominated by historical values. To achieve this balance, we introduce a hyperparameter γ to control the influence of the past history. The exact loss function is then defined as:

$$l(\theta) = ||s_\theta(t, \mathbf{x}_t^{\text{total}}, \mathbf{x}^{\text{hist}}) - \nabla_{\mathbf{x}_t^{\text{total}}} \log p(\mathbf{x}_t^{\text{total}} | \mathbf{x}^{\text{total}})||^2,$$
$$L(\theta) = \mathbb{E}_{t, \mathbf{x}^{\text{total}}, \mathbf{x}_t^{\text{total}}}[\lambda(t)||\gamma \mathbf{m} \otimes l(\theta) + (1 - \mathbf{m}) \otimes l(\theta)||_1]$$

, where $\otimes$ is a hadamard product and $\mathbf{m} = \{x_{ij}\}_{(N+T)\times d}$ is a mask vector that $x_{ij} = 1$ if $i \leq N$ and 0 otherwise, dividing the past and future elements in our loss function.

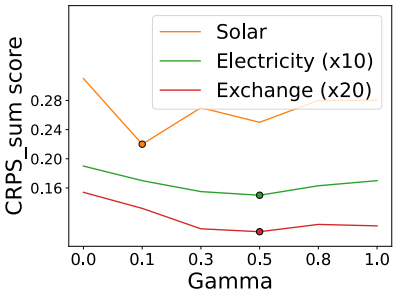

Figure 2: $CRPS_{sum}$ on different γ.

3.2 Training and Inference

For training SFdiff, we employ the Variance Preserving (VP) SDE (Song et al., 2020), which generalizes existing DDPM-based methods (Rasul et al., 2021; Tashiro et al., 2021) to compute $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{x})$. In this phase, a key aspect is controlling the parameter γ. Setting $\gamma = 0.0$ (equivalent to DDPM) ignores past information, while $\gamma = 1.0$ fully incorporates it. Neither extreme is optimal. As shown in Figure 2 and Figure 3, SFdiff achieves the best results on the Solar and Electricity datasets when $\gamma = 0.1$ and $\gamma = 0.5$, respectively.

After training the model, we generate total sequence from history condition by using well-known PC sampling procedure and its default setting (Song et al., 2020). Furthermore, to facilitate self-generation technique, we adapt classifier-free guidance(CFG) to our framework. However, up to our survey, CFG is confined to DDPM and there was no adaptation of CFG to score-based diffusion models. Therefore, we briefly introduced CFG in Section C and now explain its usage in SFdiff.

From a score matching perspective, CFG using score function can be understood as $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{c}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{c}) + w\nabla_{\mathbf{x}_t} \log p(\mathbf{c}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{c}) + w\nabla_{\mathbf{x}_t}(\log p(\mathbf{x}_t|\mathbf{c}) - \log p(\mathbf{x}_t)) = (1 + w)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{c}) - w\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$. And this formulation leads to the generalized score function used in CFG: $\tilde{s}_\theta(\mathbf{x}_t, \mathbf{c}) = (1 + w)s_\theta(\mathbf{x}_t, \mathbf{c}) - ws_\theta(\mathbf{x}_t, \mathbf{0})$, where $\mathbf{0}$ means zero padding. We use this generalized CFG sampling. As the formulation shows, CFG should train both conditional and unconditional sampling to single model. In line with Ho & Salimans (2022), we adopt a proportional training strategy, where with probability p_{cond} (setting 0.2 as default value), the model trains the conditional score network $s_\theta(\mathbf{x}_t, \mathbf{c})$, and with probability $1 - p_{\text{cond}}$, it trains the unconditional score network $s_\theta(\mathbf{x}_t, \mathbf{0})$.

Comparing self-generation with prior prediction-only generation methods reveals interesting insights. Intuitively, CFG applied to prediction generation with potentially noisy conditions may amplify undesirable influences, degrading performance. In contrast, self-generation benefits from CFG by generating predictions jointly with a denoised historical sequence. As shown in Table 1, CFG negatively impacts prediction-only generation by exacerbating noise-related effects. However, combined with Table 2, self-generation yields overall improved results, with minimal performance degradation observed for the Solar dataset, which is the only reduced result among dataset.

Table 1: $CRPS_{sum}$ comparison between prediction CFG results on prediction generation and total generation.

	Exchange	Electricity	Solar
L^{pred}	.006±.001	.021±.001	.287±.020
L_{CFG}^{pred}	.008±.001	.026±.001	.451±.011
L^{total}	.006±.000	.018±.001	.250±.007
L_{CFG}^{total}	.005±.000	.015±.000	.277±.006

4 Experiments

In this section, we present the results of experiments conducted to evaluate the performance of our proposed model.

4.0.1 EXPERIMENTAL SETUPS

Our experiments consist of two stages: (1) assessing whether our generative framework effectively reduces noise in the conditions using toy datasets, and (2) evaluating our model's performance on real-world time-series datasets.

To verify noise reduction in the conditions, we utilize two toy datasets: the 2D oscillator ODE and the 3D harmonic ODE. The corresponding ODE formulations are as follows:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} y \\ y(1-x)^2 - x \end{bmatrix} \text{ and } \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} -y \\ x - z \\ y \end{bmatrix},$$

where the initial conditions are $[2.0, 0.0]$ and $[0.5, 0.5, 0.5]$, respectively. We generate the ODE trajectories using the well-known `scipy` package with a time interval of 0.1.

For real dataset experiments, we use our model on 5 widely-used time-series forecasting datasets: Exchange (Lai et al., 2017), Solar (Lai et al., 2017), Electricity[1], Taxi[2], Wikipedia[3]. We give detailed description of these datasets in Table 5, including dimension, total number of timesteps, domain and frequency data of each dataset. We also report hyperparameters setting in Table 5, the history and prediction lengths, the number of diffusion steps, and the number of iterations, etc. Here, we point out that we follow the common practice of training based on iteration count and saving checkpoints every 5,000 steps, as done in other diffusion models (Ho et al., 2020; Song et al., 2020).

After training our model on the selected real datasets, we evaluate its performance against a wide range of baseline models. These baselines include: i) classical multivariate methods such as VAR, VAR-Lasso (Lütkepohl, 2005), GARCH (van der Weide, 2002), and VES (Hyndman et al., 2008); ii) RNN-based methods like Vec-LSTM-ind-scaling, Vec-LSTM-lowrank-Copula, GP-scaling, and GP-Copula (Salinas et al., 2019); iii) Transformer-based models, specifically Transformer-MAF (Rasul et al., 2020); and iv) VAE and diffusion-based models, including KVAE (Fraccaro et al., 2017), TimeGrad (Rasul et al., 2021), and CSDI (Tashiro et al., 2021). A description of these baseline models can be found in Appendix D.

For evaluation, we use the sum of continuous ranked probability score ($CRPS_{sum}$), a widely recognized metric for probabilistic forecasting. CRPS measures the compatibility between the cumulative distribution function (CDF) F and an observation x as $CRPS(F, x) = \int (F(z) - \mathbb{I}(x \leq z))^2 dz$, where $\mathbb{I}$ is an indicator function. To approximate CDF, we use an empirically estimated CDF $\hat{F} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(x_i \leq z)$, where x_i are samples from F. Then we compute the sum of CRPS over all features, denoted as $CRPS_{sum}$,

$$CRPS_{sum}(F, x) = \frac{CRPS(F, \sum_i x_{i,t})}{\sum_{i,t} |x_{i,t}|}$$

, where $\sum_{i,t} |x_{i,t}|$ means the summation of all target features at time t. For other detailed descriptions of experimental setup, we refer to Section B.

4.1 EXPERIMENTS ON TOY DATASETS

We utilize two toy datasets: the 2D oscillator dataset and the 3D harmonic dataset. During training, we augment the given trajectories by randomly adding noise sampled from $\mathcal{N}(0, \frac{1}{2}\mathbf{I})$ to half of the trajectory to promote robust training. Each trajectory is uniformly divided into segments of length 72. For testing, we introduce more intense noise, sampled from $\mathcal{N}(0, \mathbf{I})$, to $\frac{1}{8}$ of the condition portion of the test samples to evaluate the purification effectiveness of our model.

Figure 3 illustrates that our model effectively purifies noisy conditions, significantly reducing large anomalous values. Notably, the bottom row of the figure shows that the parameter γ in our loss

[1] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[2] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[3] https://github.com/mbohlkeschneider/gluon-ts/tree/mv_release/datasets

Table 2: $CRPS_{sum}$ results on evaluation datasets. The best scores are in boldface.

	Exchange	Solar	Electricity	Taxi	Wiki
VES	.005±.000	.900±.003	.880±.004	-	-
VAR	.005±.000	.830±.006	.039±.001	-	-
VAR-Lasso	.012±.000	.510±.006	.025±.000	-	3.10±.004
GARCH	.023±.000	.880±.002	.190±.001	-	-
KVAE	.014±.002	.340±.025	.051±.019	-	.095±.012
Vec-LSTM ind-scaling	.008±.001	.391±.017	.025±.001	.506±.005	.133±.002
Vec-LSTM low-copula	.007±.000	.319±.011	.064±.008	.326±.007	.241±.033
GP scaling	.009±.000	.368±.012	.022±.000	.183±.395	1.48±1.03
GP copula	.007±.000	.337±.024	.025±.002	.208±.183	.086±.004
Transformer MAF	.005±.003	.301±.014	.021±.000	.179±.002	.063±.003
TimeGrad	.006±.001	.287±.020	.021±.001	.114±.020	.049±.002
CSDI	.007±.001	.298±.004	.017±.000	.123±.003	.047±.003
SFdiff	.006±.000	**.250±.007**	.018±.001	.122±.001	.052±.000
SFdiff-CFG	**.005±.000**	.277±.006	**.015±.000**	**.092±.001**	**.046±.001**

function plays a crucial role in achieving successful purification. When $\gamma \approx 1$, the synthesized time-series closely follow the noisy conditions, whereas when $\gamma \approx 0$, the model struggles to generate meaningful conditions. Thus, controlling γ is essential not only for generating accurate predictions but also for mitigating out-of-distribution values, a process we refer to as purification.

4.2 Experiments on real datasets

We present the $CRPS_{sum}$ performance of SFdiff and other baseline models in Table 2. We evaluate SFdiff with 5 different seeds, and we report both the mean and standard deviation. As shown in the table, SFdiff consistently outperforms all competing models across every dataset, including other diffusion-based forecasting models. Notably, while diffusion-based forecasting models like TimeGrad and CSDI perform comparably on certain datasets, SFdiff consistently delivers superior results across a wide range of data complexities, from relatively low-dimensional datasets (e.g., Exchange) to high-dimensional ones (e.g., Wiki).

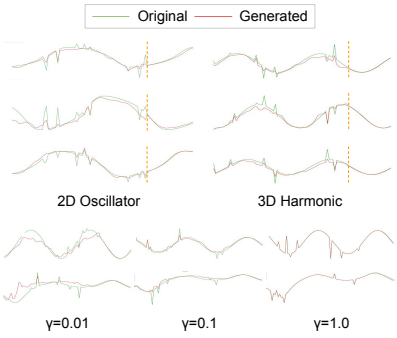

Figure 3: Generated total time-series. The orange dots divide history and prediction part.

5 Ablation Experiments

In this section, we conduct sensitivity studies about weight of classifier-free guidance. As shown in the Table 3, CFG results getting deteriorated as the weight of CFG (w) getting stronger.

Next, we present ablation studies conducted across several datasets to analyze the impact of varying the diffusion steps in SFdiff. We experiment with different numbers of diffusion steps: 50, 100, 200, 250, 500, and report the corresponding $CRPS_{sum}$ results.

	Original	$CFG_{0.01}$	$CFG_{0.1}$
Exchange	.006±.000	**.005±.000**	.006±.000
Electricity	.018±.001	**.015±.000**	.016±.000
Solar	**.250±.007**	.277±.006	.300±.002

Table 3: Detailed dataset descriptions.

As indicated by the results, there are optimal "sweet spots" for the number of steps depending on the dataset. For example, SFdiff requires relatively fewer diffusion steps on datasets like Exchange and Electricity, whereas it benefits from higher steps on

Table 4: Results of ablation study varying the number of sampling steps

	50	100	200	250	500
Exchange	0057±.0003	**.0054±.0002**	.0057±.0002	.0059±.0004	.0057±.0002
Electricity	.0168±.0003	**.0165±.0005**	.0168±.0007	.0166±.0005	.0166±.0002
Solar	.4540±.0125	.2829±.0090	.2501±.0070	.2313±.0059	**.2155±.0089**

the Solar dataset to achieve the best performance. However, since lots of diffusion steps increase sampling time of SFdiff, we compromise them by hyperparameters in Table 5 in Appendix B.

We also point out that an notable distinction of SFdiff, compared to other diffusion-based forecasting models such as CSDI (Tashiro et al., 2021) and TimeGrad (Rasul et al., 2021), is its ability to adjust the number of sampling steps without the need for additional training at each specific step. This flexibility offers a significant advantage, as it allows SFdiff to adapt more efficiently across varying datasets and conditions, without incurring extra computational costs for retraining.

6 Related work

This section briefly reviews diffusion-based time-series forecasting models, categorizing them based on their target score objectives.

Existing diffusion-based forecasting models are broadly divided into two categories: models targeting the prediction sequence score function, $\nabla_{\mathbf{x}_t^{\text{pred}}} \log p(\mathbf{x}_t^{\text{pred}} | \mathbf{x}^{\text{hist}})$, and those modeling the entire sequence score, $\nabla_{\mathbf{x}_t^{\text{total}}} \log p(\mathbf{x}_t^{\text{total}} | \mathbf{x}^{\text{hist}})$.

TimeGrad (Rasul et al., 2021) and CSDI (Tashiro et al., 2021) belong to the first category. TimeGrad generates predictions autoregressively, predicting one step ahead iteratively, whereas CSDI generates the entire prediction sequence in a single step. Although one-shot generation can be efficient, it may introduce higher variance in samples, prompting CSDI to stabilize performance by averaging multiple samples.

In the second category, models like TSDiff (Kollovieh et al., 2023b) generate the complete sequence, leveraging history-guided sampling to enhance conditional generation. Additionally, Lim et al. (2023) and Lim et al. (2024) propose autoregressive generation in a latent space to handle irregularly sampled data effectively, offering improved modeling of complex time dependencies.

Our proposed method, SFdiff, combines advantages from both categories, integrating predictive accuracy and guidance mechanisms within a unified framework for robust and flexible time-series forecasting.

7 Conclusion

We propose the Self-Generation framework, leveraging diffusion models to robustly forecast time-series data despite anomalous inputs. Self-Generation effectively purifies noisy conditions by generating the entire sequence, balancing historical and future components. Moreover, we introduce classifier-free guidance into diffusion-based forecasting, significantly enhancing predictive accuracy. Extensive experiments demonstrate our model consistently outperforms 12 baselines across two toy examples and four real-world datasets. **Limitations**. While our approach achieves state-of-the-art forecasting performance, it requires careful tuning of the hyperparameter γ to balance historical and predictive sequences effectively. Furthermore, the computational cost associated with diffusion models can be substantial, especially for large-scale applications. **Societal Impacts**. Our work presents positive societal impacts by improving predictive accuracy in critical domains such as healthcare and finance, potentially aiding better-informed decisions. However, misuse in sensitive areas, such as privacy-sensitive data forecasting, might lead to ethical concerns. **Safeguards**. Since our work primarily focuses on synthetic and publicly available datasets, specific safeguards for high-risk misuse scenarios were not required. However, we emphasize careful ethical consideration for future extensions involving sensitive or confidential data.

REFERENCES

Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11472–11481, 2022.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021. URL https://arxiv.org/abs/2105.05233.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning, 2017. URL https://arxiv.org/abs/1710.05741.

GM Harshvardhan, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285, 2020.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. URL https://arxiv.org/abs/2204.03458.

Robin John Hyndman, Anne B Koehler, J Keith Ord, and Ralph David Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, 2008. ISBN 9783540719168.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

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, 2023a. URL https://arxiv.org/abs/2307.11494.

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, 2023b. URL https://arxiv.org/abs/2307.11494.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, 2021. URL https://arxiv.org/abs/2009.09761.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. *CoRR*, abs/1703.07015, 2017. URL http://arxiv.org/abs/1703.07015.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Haksoo Lim, Minjung Kim, Sewon Park, and Noseong Park. Regular time-series generation using sgm, 2023. URL https://arxiv.org/abs/2301.08518.

Haksoo Lim, Minjung Kim, Sewon Park, Jaehoon Lee, and Noseong Park. TSGM: Regular and irregular time-series generation using score-based generative models, 2024. URL https://openreview.net/forum?id=nFG1YmQTqi.

Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Number 978-3-540-27752-1 in Springer Books. Springer, September 2005. ISBN ARRAY(0x505580b8). doi: 10.1007/978-3-540-27752-1. URL https://ideas.repec.org/b/spr/sprbok/978-3-540-27752-1.html.

Ricardo P Masini, Marcelo C Medeiros, and Eduardo F Mendes. Machine learning advances for time series forecasting. *Journal of economic surveys*, 37(1):76–111, 2023.

John A. Miller, Mohammed Aldosari, Farah Saeed, Nasid Habib Barna, Subas Rana, I. Budak Arpinar, and Ninghao Liu. A survey of deep learning and foundation models for time series forecasting, 2024. URL https://arxiv.org/abs/2401.13912.

Byeonghu Na, Yeongmin Kim, HeeSun Bae, Jung Hyun Lee, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Label-noise robust diffusion models, 2024. URL https://arxiv.org/abs/2402.17517.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications (Universitext)*. Springer, 6th edition, January 2014. ISBN 3540047581. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/3540047581.

Achraf Oussidi and Azeddine Elhassouny. Deep generative models: Survey. In *2018 International conference on intelligent systems and computer vision (ISCV)*, pp. 1–8. IEEE, 2018.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multi-variate probabilistic time series forecasting via conditioned normalizing flows. *CoRR*, abs/2002.06103, 2020. URL https://arxiv.org/abs/2002.06103.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. *CoRR*, abs/2101.12072, 2021. URL https://arxiv.org/abs/2101.12072.

Jože Rožanec, Elena Trajkova, Klemen Kenda, Blaž Fortuna, and Dunja Mladenić. Explaining bad forecasts in global time series models. *Applied Sciences*, 11(19), 2021. ISSN 2076-3417. doi: 10.3390/app11199243. URL https://www.mdpi.com/2076-3417/11/19/9243.

David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *CoRR*, abs/1910.03002, 2019. URL http://arxiv.org/abs/1910.03002.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020. URL https://arxiv.org/abs/2011.13456.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models, 2021. URL https://arxiv.org/abs/2101.09258.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. *CoRR*, abs/2107.03502, 2021. URL https://arxiv.org/abs/2107.03502.

José F. Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: A survey. *Big Data*, 9(1):3–21, 2021a. doi: 10.1089/big.2020.0159. URL https://doi.org/10.1089/big.2020.0159. PMID: 33275484.

José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. *Big Data*, 9(1):3–21, 2021b.

Roy van der Weide. Go-garch: A multivariate generalized orthogonal garch model. *Journal of Applied Econometrics*, 17(5):549–564, 2002. ISSN 08837252, 10991255. URL http://www.jstor.org/stable/4129271.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *CoRR*, abs/2106.10121, 2021. URL https://arxiv.org/abs/2106.10121.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.