Loss Functions in Diffusion Models: A Comparative Study

Dibyanshu Kumar (✉)[0009−0007−2542−4781], Philipp Väth[0000−0002−8247−7907], and Magda Gregorová[0000−0002−1285−8130]

Center for Artificial Intelligence and Robotics, Technical University of Applied Sciences Würzburg-Schweinfurt, Franz-Horn-Straße 2, Würzburg, Germany
kumardibyanshu05@gmail.com, philipp.vaeth@thws.de, magda.gregorova@thws.de

Abstract. Diffusion models have emerged as powerful generative models, inspiring extensive research into their underlying mechanisms. One of the key questions in this area is the loss functions these models shall train with. Multiple formulations have been introduced in the literature over the past several years [4,13,7,11] with some links and some critical differences stemming from various initial considerations. In this paper, we explore the different target objectives and corresponding loss functions in detail. We present a systematic overview of their relationships, unifying them under the framework of the variational lower bound objective. We complement this theoretical analysis with an empirical study providing insights into the conditions under which these objectives diverge in performance and the underlying factors contributing to such deviations. Additionally, we evaluate how the choice of objective impacts the model's ability to achieve specific goals, such as generating high-quality samples or accurately estimating likelihoods. This study offers a unified understanding of loss functions in diffusion models, contributing to more efficient and goal-oriented model designs in future research.

Keywords: Diffusion Model · Loss Functions · Generative Modeling.

1 Introduction

Diffusion models [4] have become a cornerstone of generative modeling in recent years, demonstrating remarkable capabilities in generating high-quality data. Given a sample $\mathbf{x}_0 \sim q(\mathbf{x})$ from a data distribution, the forward process in diffusion models incrementally corrupts the data by adding small amounts of Gaussian noise over multiple steps T. This process is defined as $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \alpha\mathbf{x}_{t-1}, \sigma^2\boldsymbol{I})$, where α controls the scaling of the data $\mathbf{x}_{t-1}$, and σ controls the magnitude of the added noise. The objective is then to learn the reverse process $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ which enables the generation of new samples by starting from pure Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and iteratively denoising it to recover realistic data. This framework of probabilistic modeling allows diffusion models to capture complex data distributions, making them highly effective for a wide range of generative tasks.

The class of diffusion models has seen several notable contributions, particularly in the development of training objectives. In score-based modeling [13], the reverse process is learned by minimizing a denoising score-matching objective. Ho et al. [4], in their work on DDPM, generated high quality images by adopting noise prediction $\boldsymbol{\epsilon}$ as the primary objective. Variational Diffusion Models (VDM) [7] used the training objective, formulated in terms of the Signal-to-Noise Ratio (SNR), which achieved the best likelihood estimation. Additionally, the authors of Progressive Distillation [11] modeled the rate of change in data distribution over time, presenting a novel loss function that combines the data representation $\mathbf{x}$ and the noise component $\boldsymbol{\epsilon}$. This objective was instrumental in reducing the number of sampling steps required to generate high-quality samples. These advancements highlight the critical role of loss function design in improving the performance and efficiency of diffusion models.

Existing research has explored the theoretical equivalence of various training objectives used in diffusion models. For instance, [15] established connections between score matching and diffusion-based generative frameworks by leveraging stochastic differential equations to model the forward process, thereby aligning it with continuous distributions that evolve over time. Similarly, [7] introduced the Evidence Lower Bound (ELBO) objective for diffusion, inspired by Variational Autoencoders [8]. More recently, [6] demonstrated that diffusion model objectives are fundamentally equivalent and closely related to the ELBO framework. However, while these works highlight the theoretical equivalence of the loss functions, they lack a structured analysis of their formulations under a single framework. Furthermore, there is no empirical study investigating whether the mathematical equivalence between objectives persists when training diffusion models with deep neural networks. Therefore, there is only limited understanding of how these loss formulations differ in terms of performance. This gap highlights the need for a systematic exploration of outcomes of these theoretical connections.

In this study, we conduct a comprehensive comparison of different training objectives, specifically the weighted and the ELBO objectives, formulated for four different target predictions of the diffusion models: data $\mathbf{x}$, noise $\boldsymbol{\epsilon}$, rate of change in the data distribution $\mathbf{v}$ and score $\mathbf{s}$. We derive the negative ELBO loss in terms of these targets and establish mathematical relationships with the most commonly used diffusion loss functions. These relationships help us to design experiments that evaluate whether the theoretical equivalence between these objectives holds in practice when used for training over the same datasets. Our experiments highlight the differences and similarities in the theoretical foundations and practical behavior of these loss functions, particularly in terms of loss convergence during training and the quality of generated samples. We explore the loss behavior across different diffusion timesteps, providing insights into the mechanisms that drive their performance and functionality. Additionally, we compare the outcomes of these training objectives in terms of data density estimation and sample quality, offering a comprehensive understanding of their roles in optimizing diffusion models.

The paper is structured as follows: section 2 provides the background on diffusion models. In section 3 we introduce the various target predictions used in diffusion models, derive the loss functions under different framework, and show the relation between them. In section 4 we describe the experiments we perform and give insights on the results obtained. Finally, we conclude by summarizing our findings and suggesting directions for future research. The code used in this study is available at: `https://github.com/dibyanshu100/LFDM`.

2 Model

In this section, we provide an overview of the forward and reverse processes used in diffusion models.

2.1 Forward diffusion process

The forward process in diffusion models is a Markov process, where the information in a given data $\mathbf{x}$ is progressively destroyed by adding noise in a series of timesteps, producing intermediate latent variables, denoted as $\mathbf{z}_t$, where $t \in [0, 1]$ represents the corresponding timestep. To achieve this, a schedule is used to define the amount of noise to be added and the signal to be removed at each timestep, regulated by parameters α_t and σ_t. The distribution of latent variables and the Markov transition distribution in the forward process is defined as follows:

$$\begin{aligned} q(\mathbf{z}_t \mid \mathbf{x}) &= \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \boldsymbol{I}) \\ q(\mathbf{z}_t \mid \mathbf{z}_s) &= \mathcal{N}(\mathbf{z}_t; \alpha_{t|s} \mathbf{z}_s, \sigma_{t|s}^2 \boldsymbol{I}) \end{aligned} \tag{1}$$

where $0 \leq s \leq t \leq 1$, $\alpha_{t|s} = \frac{\alpha_t}{\alpha_s}$ and $\sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2$

The scheduling parameters, α_t and σ_t are strictly positive, smooth, monotonically decreasing and increasing functions of time respectively. Based on this we can define the Signal-to-Noise ratio SNR(t) as:

$$\mathrm{SNR}(t) = \frac{\alpha_t^2}{\sigma_t^2} \tag{2}$$

As time t progresses, the SNR decreases. This implies that for $s < t$, we have $\mathrm{SNR}(s) > \mathrm{SNR}(t)$. At t=0 the data is least noisy and at t=1 there is no more signal left in the data, hence $q(\mathbf{z}_1 \mid \mathbf{x}) = \mathcal{N}(\mathbf{z}_1; 0, \boldsymbol{I})$.

The choice of schedule significantly impacts the performance of diffusion models, and there are several ways of noise scheduling. DDPM [4] employed a linear schedule to add noise over 1000 discrete timesteps. Nichol and Dhariwal [9], used cosine scheduling and found it to perform better due to its smooth transition between low and high levels of noise. In VDM [7], the authors learned the forward noise schedule, moreover they demonstrate that increasing the number of timesteps resulted in a decrease in loss, thereby achieving good results with a continuous-time model. The schedules used in the above mentioned works were variance preserving ($\alpha_t^2 = 1 - \sigma_t^2$), which ensures that the variance of the data remains constant throughout the forward process. Alternatively, in the case of variance-exploding schedules [15,14], $\alpha_t^2 = 1$. It was demonstrated by Kingma et al. [7] that the variance preserving and variance exploding formulations can be considered equivalent in continuous time.

2.2 Reverse generative process

The reverse diffusion process $q(\mathbf{z}_s \mid \mathbf{z}_t)$ is also a Markov chain with Gaussian transition probability and aims to recover the original data $\mathbf{x}$ from the noisy data $\mathbf{z}_t$. Since the true reverse process $q(\mathbf{z}_s \mid \mathbf{z}_t)$ is intractable, it is approximated with a learned distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_s \mid \mathbf{z}_t)$. This forms a hierarchical generative model that samples a sequence of latent variables $\mathbf{z}_t$, with time progressing from t=1 to t=0, gradually denoising the data over T steps to recover the original distribution. For discrete time case number of steps T is finite and is discretized into uniform timesteps of width $1/T$, with $s(i) = \frac{i-1}{T}$ and $t(i) = \frac{i}{T}$,

The overall reverse process is defined as,

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{z}_1) p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}_0) \prod_{i=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) d\mathbf{z} \tag{3}$$

To approximate the true data distribution we need to minimize the negative log likelihood. However, that is intractable and we minimize the tractable negative variational lower bound also called negative evidence lower bound (NELBO) instead, which is standard in latent variable models and is expressed as,

$$\begin{aligned} -\log p_{\boldsymbol{\theta}}(\mathbf{x}) \leq \mathrm{NELBO}(\mathbf{x}) = & \underbrace{D_{\mathrm{KL}}\left(q(\mathbf{z}_1 \mid \mathbf{x}) \,\|\, p(\mathbf{z}_1)\right)}_{\text{Prior Loss}} + \\ & \underbrace{\mathbb{E}_{q(\mathbf{z}_0 \mid \mathbf{x})}\left[-\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}_0)\right]}_{\text{Reconstruction Loss}} + \\ & \underbrace{\sum_{i=1}^{T} \mathbb{E}_{q(\mathbf{z}_{t(i)} \mid \mathbf{x})} D_{\mathrm{KL}}\left[q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\right]}_{\text{Diffusion Loss } (L_T(\mathbf{x}))} \end{aligned} \tag{4}$$

Based on the assumptions of the forward process, $\mathbf{z}_0$ is nearly identical to $\mathbf{x}$ because only a small amount of noise is added, making the reconstruction loss in the equation (4) negligible and therefore can be dropped from the objective in practice. Moreover, as discussed in section 2.1, $q(\mathbf{z}_1 \mid \mathbf{x})$ approaches a pure Gaussian distribution at the end of the forward process which matches our fixed prior $p(\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_1; 0, \boldsymbol{I})$. As a result, the KL divergence $D_{\mathrm{KL}}\big(q(\mathbf{z}_1 \mid \mathbf{x}) \,\|\, p(\mathbf{z}_1)\big)$ tends to zero, hence this term is also dropped. The remaining term is the diffusion loss $L_T(\mathbf{x})$ which depends on the number of timesteps T determining the depth of the generative model.

3 Loss formulations

In the previous section, we defined the NELBO objective (4). For the denoising model there are several options for the target prediction in addition to the data $\mathbf{x}$. For example, some approaches focus on predicting the noise $\boldsymbol{\epsilon}$ added during the forward process [4,12,9]. Another approach predicts the rate of change in the data distribution over time, also known as $\mathbf{v}$-prediction [11]. Some methods target the score function $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ [13,15], which is the gradient of the log-probability density of the data.

For each of these targets, we can derive the NELBO loss formulation (L) from equation (4). In addition, other loss formulations are also proposed in the literature, typically designed to prioritize perceptual sample quality or computational efficiency. We call these weighted loss functions ($\mathcal{L}$) as they can all be shown as a weighted function of the NELBO where the weight $w(t)$ is a suitable chosen weighting function.

$$\mathcal{L} = w(t)L \tag{5}$$

In the following sections, we explore the various target predictions and corresponding loss formulations in detail. We present a systematic review of these relationships, unifying them under the framework of the NELBO objective. Specifically, we derive the NELBO in terms of these alternative targets and show that all the different objectives, whether predicting the original data $\mathbf{x}$, noise $\boldsymbol{\epsilon}$, rate of change in the data distribution $\mathbf{v}$, or score $\mathbf{s}$ can be expressed as weighted functions of the NELBO. For clarity, we refer to different target objectives as $\mathbf{x}$-space, $\boldsymbol{\epsilon}$-space, $\mathbf{v}$-space, and $\mathbf{s}$-space throughout this paper.

3.1 x-space

As shown in section 2.2, the NELBO reduces to diffusion loss which is the last term of equation (4). This can be further simplified to the following form (a detailed derivation of these steps is provided in appendix B.1).

$$L_T(\mathbf{x}) = \frac{T}{2}\, \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}),i\sim \mathrm{U}\{1,T\}} \left[\left(\mathrm{SNR}(s(i)) - \mathrm{SNR}(t(i))\right) \|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\|_2^2 \right] \tag{6}$$

where $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$ is the prediction of the original data $\mathbf{x}$ by our denoising model given the noisy data $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ at timestep t.

For the continuous-time case, $T \to \infty$. Here, the timestep t is treated as a continuous variable, and the transition process is referred to as the continuous-time diffusion process [7]. In this setting, equation (6) transforms into the following form,

$$L(\mathbf{x}) = -\mathbb{E}_{t\sim\mathcal{U}(0,1),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\mathrm{SNR}'(t)\, \|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\|_2^2 \right] \tag{7}$$

where we prove that for cosine noise schedule, $\mathrm{SNR}'(t) = \frac{-\pi\alpha_t}{\sigma_t^3}$ (see appendix B.1). Note that we use $\mathrm{U}\{1, T\}$ to denote sampling from a discrete uniform distribution, while $\mathcal{U}(0, 1)$ denotes sampling from a continuous uniform distribution in the continuous-time setting.

Moreover we can define the weighted loss as,

$$\mathcal{L}(\mathbf{x}) = -\mathbb{E}_{t\sim\mathcal{U}(0,1),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[w_{\mathbf{x}}(t)\,\mathrm{SNR}'(t)\,\|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2\right] \tag{8}$$

By choosing $w_{\mathbf{x}}(t) = -\frac{1}{\mathrm{SNR}'(t)}$, this further simplifies as an expected value of the mean squared error between the original data and the predicted data,

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2\right] = w_{\mathbf{x}}(t)L(\mathbf{x}) \tag{9}$$

3.2 $\boldsymbol{\epsilon}$-space

The $\boldsymbol{\epsilon}$-space loss formulation is one of the most commonly used objective in diffusion models, as proposed in DDPM[4]. Instead of directly reconstructing the original data $\mathbf{x}$, we model the noise component $\boldsymbol{\epsilon}$ that was added to the data in every time step during the forward diffusion process. The authors of the paper claimed that this approach simplifies the learning task, as the prediction of noise aligns with the stochastic nature of the diffusion process.

We derive the NELBO loss in $\boldsymbol{\epsilon}$-space, as detailed in the appendix B.2,

$$L(\boldsymbol{\epsilon}) = -\mathbb{E}_{t,\epsilon}\left[\frac{\mathrm{SNR}'(t)}{\mathrm{SNR}(t)}\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2\right] \tag{10}$$

The loss proposed in DDPM is different from the NELBO loss. They used the weighted $\boldsymbol{\epsilon}$ loss, which implies that the model learns to predict the noise sampled from the unit Gaussian and not the scaled noise which was added to the original data $\mathbf{x}$ at every timestep during the forward diffusion process. The weighted $\boldsymbol{\epsilon}$-loss is given as below and can be seen as a weighted function of (10) with weight $w_{\boldsymbol{\epsilon}}(t) = -\frac{\mathrm{SNR}(t)}{\mathrm{SNR}'(t)}$:

$$\mathcal{L}(\boldsymbol{\epsilon}) = \mathbb{E}_{t,\epsilon}\left[\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2\right] = w_{\boldsymbol{\epsilon}}(t)L(\boldsymbol{\epsilon}) \tag{11}$$

3.3 v-space

The v-space loss, introduced in [11], combines the data $\mathbf{x}$ and noise $\boldsymbol{\epsilon}$. This formulation is particularly beneficial for model distillation to reduce the number of sampling steps, as while sampling the standard noise objective becomes unstable when the SNR approaches zero. In such cases α_t tends to zero, leading to instability in reconstructing the data from the predicted noise as $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t) = \frac{1}{\alpha_t}\left(\mathbf{z}_t - \sigma_t\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t)\right)$. The authors showed that this issue has less impact in conventional diffusion models, where clipping the reconstructed data in the desired range and using a large number of sampling steps can mitigate errors but becomes a key factor in distillation, where efficient sampling is essential in a limited number of steps.

This approach expresses the noisy data using an angular parameter ϕ_t, where $\mathbf{z}_{\phi_t} = \cos(\phi_t)\mathbf{x} + \sin(\phi_t)\boldsymbol{\epsilon}$, and $\phi_t = \arctan\left(\frac{\sigma_t}{\alpha_t}\right)$. The target $\mathbf{v}_{\phi_t}$ is then calculated as $\mathbf{v}_{\phi_t} = \frac{d\mathbf{z}_{\phi_t}}{d\phi_t} = \cos(\phi_t)\boldsymbol{\epsilon} - \sin(\phi_t)\mathbf{x}$, which represents the instantaneous direction and rate of change required to transform the noisy data $\mathbf{z}_{\phi_t}$ along a circular trajectory parameterized by the angle ϕ_t.

The NELBO loss in $\mathbf{v}$-space as derived in appendix B.3 is given as,

$$L(\mathbf{v}) = -\mathbb{E}_{t,\epsilon}\left[\frac{\sigma_t^2}{\alpha_t^2 + \sigma_t^2}\,\mathrm{SNR}'(t)\,\|\mathbf{v} - \hat{\mathbf{v}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\|_2^2\right] \tag{12}$$

The weighted $\mathbf{v}$ loss can be formulated as the weighted function of NELBO with weight $w_{\mathbf{v}}(t) = -\frac{(\alpha_t^2+\sigma_t^2)}{\sigma_t^2\mathrm{SNR}'(t)}$

$$\mathcal{L}(\mathbf{v}) = \mathbb{E}_{t,\epsilon}\big[\|\mathbf{v} - \hat{\mathbf{v}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\|_2^2\big] = w_{\mathbf{v}}(t)L(\mathbf{v}) \tag{13}$$

3.4 s-space

Score modeling, introduced by Song et al. [13] uses denoising score matching [17] to approximate the score function and then use a neural network to learn it. The idea behind score matching is to add a small amount of noise to the data, which makes the score calculation tractable, and therefore learn the score of perturbed distribution instead of the original distribution, which is expressed as $\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t \mid \mathbf{x})$. The authors demonstrate that minimizing the denoising score matching objective across multiple noise scales enables high quality sample generation. This theory is closely aligned with the diffusion process, as both approaches aim to refine the noisy data toward its original distribution.

In [15], the authors bridged the gap between score modeling and diffusion models by proposing score based modeling using stochastic differential equations (SDE). They showed that the forward diffusion process can be interpreted as a discretization of a continuous-time SDE, and the reverse process corresponds to solving the reverse-time SDE using the learned score function.

We demonstrate here that the NELBO loss can once again be formulated in $\mathbf{s}$-space and can be expressed as below (see in appendix B.4),

$$L(\mathbf{s}) = -\mathbb{E}_{t,\epsilon}\left[\frac{\sigma_t^4}{\alpha_t^2}\mathrm{SNR}'(t)\,\|\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t \mid \mathbf{x}) - \hat{\mathbf{s}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\|_2^2\right] \tag{14}$$

The weighted loss in $\mathbf{s}$-space can be formulated as the weighted function of NELBO with weight $w_{\mathbf{s}}(t) = -\frac{\alpha_t^2}{\sigma_t^4\mathrm{SNR}'(t)}$

$$\mathcal{L}(\mathbf{s}) = \mathbb{E}_{t,\epsilon}\big[\|\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t \mid \mathbf{x}) - \hat{\mathbf{s}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\|_2^2\big] = w_{\mathbf{s}}(t)L(\mathbf{s}) \tag{15}$$

Furthermore, for Gaussian noise perturbation the score simplifies to:

$$\mathbf{s} = \nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t \mid \mathbf{x}) = -\frac{(\mathbf{z}_t - \alpha_t\mathbf{x})}{\sigma_t^2} \tag{16}$$

3.5 Equivalence of loss functions

All the NELBO loss formulations across different parameter spaces are derived from same equation (7), therefore, they are fundamentally equivalent. However, to derive the weighted loss from the NELBO loss, we need to apply different weights, which essentially removes the SNR scalings associated with the ℓ_2 difference between the target and the prediction. As a result, these weighted loss functions are not equivalent, even though they are all initially derived from the same NELBO formulation.

To establish a relationship between the weighted loss formulations, we rescale the weighted loss in $\boldsymbol{\epsilon}$, $\mathbf{v}$ and $\mathbf{s}$ space to make it equal to $\mathcal{L}(\mathbf{x})$. We call them rescaled loss $\widetilde{\mathcal{L}}$ that is equivalent across all targets and can be easily obtained using the weights derived in the previous sections,

$$\widetilde{\mathcal{L}}(\boldsymbol{\epsilon}) = \frac{\sigma_t^2}{\alpha_t^2}\mathcal{L}(\boldsymbol{\epsilon}) = \mathcal{L}(\mathbf{x}) \tag{17}$$

$$\widetilde{\mathcal{L}}(\mathbf{v}) = \frac{\sigma_t^2}{\alpha_t^2 + \sigma_t^2}\mathcal{L}(\mathbf{v}) = \mathcal{L}(\mathbf{x}) \tag{18}$$

$$\widetilde{\mathcal{L}}(\mathbf{s}) = \frac{\sigma_t^4}{\alpha_t^2}\mathcal{L}(\mathbf{s}) = \mathcal{L}(\mathbf{x}) \tag{19}$$

Table 1 summarizes all the loss formulations across different targets for the denoising model.

Table 1: Overview of all the loss formulations across different scenarios. While the NELBO and the rescaled loss are equivalent and comparable, the weighted losses are not equivalent and are expected to exhibit different empirical performance.

Target	**NELBO loss** (L)	**Weighted loss** $(\mathcal{L})$	**Rescaled loss** $(\widetilde{\mathcal{L}})$
$\mathbf{x}$	$-\mathbb{E}\big[\text{SNR}'(t)\,\Vert\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}\Vert_2^2\big]$	$\mathbb{E}\big[\Vert\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}\Vert_2^2\big]$	$\mathbb{E}\big[\Vert\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}\Vert_2^2\big]$
$\boldsymbol{\epsilon}$	$-\mathbb{E}\big[\frac{\text{SNR}'(t)}{\text{SNR}(t)}\Vert\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\Vert_2^2\big]$	$\mathbb{E}\big[\Vert\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\Vert_2^2\big]$	$\mathbb{E}\big[\frac{\sigma_t^2}{\alpha_t^2}\Vert\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\Vert_2^2\big]$
$\mathbf{v}$	$-\mathbb{E}\big[\frac{\sigma_t^2}{\alpha_t^2+\sigma_t^2}\,\text{SNR}'(t)\,\Vert\mathbf{v} - \hat{\mathbf{v}}_{\boldsymbol{\theta}}\Vert_2^2\big]$	$\mathbb{E}\big[\Vert\mathbf{v} - \hat{\mathbf{v}}_{\boldsymbol{\theta}}\Vert_2^2\big]$	$\mathbb{E}\big[\frac{\sigma_t^2}{\alpha_t^2+\sigma_t^2}\Vert\mathbf{v} - \hat{\mathbf{v}}_{\boldsymbol{\theta}}\Vert_2^2\big]$
$\mathbf{s}$	$-\mathbb{E}\big[\frac{\sigma_t^4}{\alpha_t^2}\text{SNR}'(t)\,\Vert\mathbf{s} - \hat{\mathbf{s}}_{\boldsymbol{\theta}}\Vert_2^2\big]$	$\mathbb{E}\big[\Vert\mathbf{s} - \hat{\mathbf{s}}_{\boldsymbol{\theta}}\Vert_2^2\big]$	$\mathbb{E}\big[\frac{\sigma_t^4}{\alpha_t^2}\Vert\mathbf{s} - \hat{\mathbf{s}}_{\boldsymbol{\theta}}\Vert_2^2\big]$

Although researchers have claimed that some training objectives outperform others in diffusion models, the reasons behind these differences remain unclear and are often attributed to empirical observations rather than theoretical foundations. For instance, while some works prefer more complex weights for loss functions [1,5,3], others [4,9] find that simpler objectives (e.g. ℓ_2 loss between target and prediction) perform just as well or even better in practice. This discrepancy raises questions about the fundamental role of loss formulations in training diffusion models and whether the observed performance gaps are due to the loss functions themselves or other factors such as model architecture, training dynamics, or noise schedules.

In our theoretical analysis, we formulated the NELBO loss for different denoising models. Specifically, we showed that different formulations of the learned model (i.e. predicting original data $\mathbf{x}$, noise $\boldsymbol{\epsilon}$, rate of change of data distribution $\mathbf{v}$ and score function $\mathbf{s}$) can be mapped to one another, and their corresponding NELBO objectives are mathematically interchangeable. We also formulated the relations between the weighted loss formulations.

In principle, the mathematical equivalence we established should hold when we train the various denoising models with equivalent loss formulations under similar conditions (e.g., dataset, model architecture etc.). In the next section, we outline the experiments conducted to validate this hypothesis and provide a detailed analysis of the results obtained.

4 Experiments

In this section, we outline the experimental setup used to conduct our tests and present the results obtained from these experiments. Additionally, we give a detailed analysis and insights into the findings.

4.1 Experimental setup

To conduct our experiments, we first work with 2-dimensional synthetic datasets that we generated ourselves. These datasets are well-suited for detailed analysis as it is easy to plot numerous examples and visually analyze the complete data manifold. To ensure the generalizability of our findings, we select four distinct 2D datasets with 100K samples each. These datasets are: Cluster data, Ring data, Swiss roll data and Waves data, the scatter plots of these datasets can be seen in fig. 1. In fig. 2 we show the effect of gaussian noise added in the forward process for all these datasets.

We also perform experiments on a high-dimensional image dataset, CIFAR-10, which is a publicly available dataset that contains 32x32 color images across 10 classes. While we present results on an image dataset, our main focus is not on extensive image generation experiments but understanding the behavior of different loss formulations. However, this work sets the foundation for future research to explore their impact on image data more deeply.

We used a variance-preserving cosine schedule for the forward process, combined with a continuous-time reverse model $T \rightarrow \infty$. Hence, the noisy data at time $t \sim \mathcal{U}(0, 1)$, is given as $\mathbf{z}_t = \cos(0.5\pi t)\mathbf{x} + \sin(0.5\pi t)\boldsymbol{\epsilon}$. To ensure comparability across experiments, for the 2D datasets, we modeled the reverse process using a simple feedforward neural network architecture consisting of 7 fully connected layers followed by a ReLU activation. In addition, we maintained consistent training dynamics for all datasets. For the image dataset, we used an architecture inspired by diffusers UNet model [10].

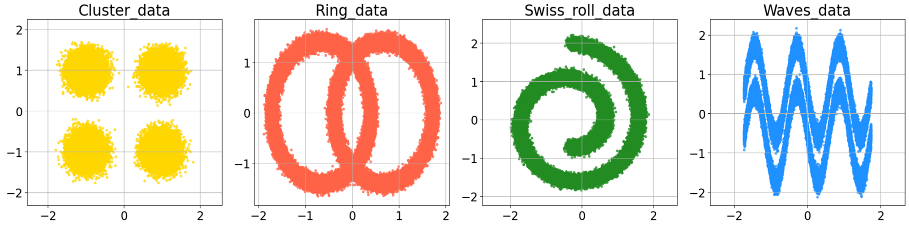

Fig. 1: Scatter plot for 2D datasets

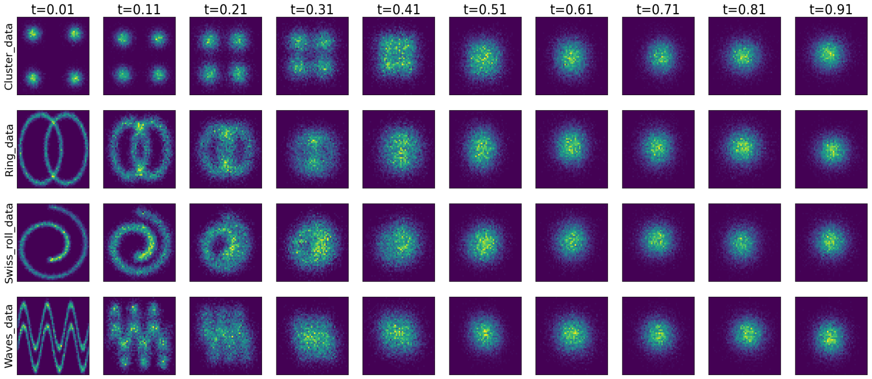

Fig. 2: Effects of adding cosine scheduled Gaussian noise in the forward process

4.2 Experimental results and discussion

In our analysis, we examine the performance of diffusion model trained with various loss formulations from three key perspectives: (i) loss convergence over epochs indicating the training and stability efficiency, (ii) the quality of generated samples that reveals how well the model produces realistic and high fidelity samples, and (iii) loss behavior at different timesteps t that give insights into how different loss formulations influence the reverse diffusion process over time. Due to space limitations, we present some results only for the ring data, while results for other datasets follow similar patterns and are provided in the appendix C for completeness.

Loss convergence vs epochs: We begin by training the denoising model using the NELBO loss formulations L for different target predictions. Given their theoretical equivalence as discussed in section 3, we expect them to behave similarly in experiments. Fig. 3 illustrates the NELBO test loss for different datasets. The loss curves for predictions in the $\mathbf{v}$ and $\boldsymbol{\epsilon}$ space are close, and so is loss in $\mathbf{x}$ and $\mathbf{s}$ space. However, these two groups differ significantly for all scenarios indicating a discrepancy in their training dynamics.

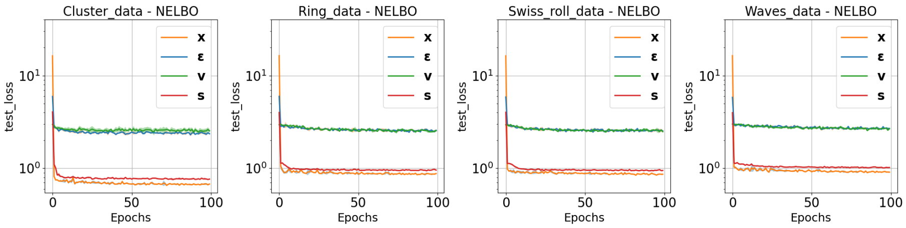

Fig. 3: NELBO test loss for different datasets

We attribute these differences to the different SNR scaling of the targets within the NELBO formulation which is inversely proportional to the weighting function and is given by $\frac{1}{w(t)}$. These scaling factors control how much each timestep contributes to the overall loss, and therefore have an impact on how the model learns during training. As shown in fig. 4, the scaling for $\boldsymbol{\epsilon}$ and $\mathbf{v}$ space are substantially higher in the early time steps, when the noise added to the data is minimal. While $\mathbf{x}$ space also exhibits large initial scaling, its decay is more gradual over time. In contrast, the $\mathbf{s}$ space has higher scaling at later timesteps. This pattern suggests that excessively high scaling at early timesteps, when noise levels are low, may negatively impact the model's overall likelihood performance.

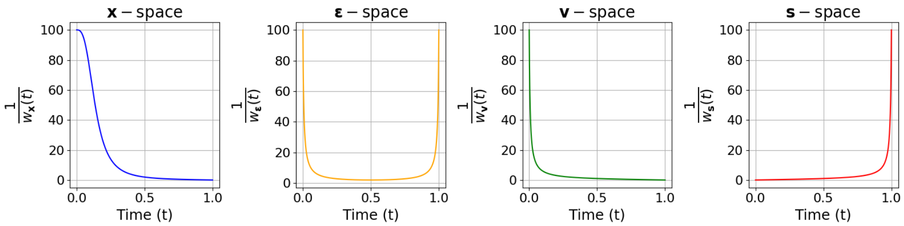

Fig. 4: SNR scalings $(\frac{1}{w(t)})$ with respect to timesteps for various NELBO formulations

Next, we train the model using the weighted loss formulations $\mathcal{L}$ for different datasets as shown in fig. 5 (left). As outlined in section 3.5, the weighted loss formulations are not equivalent and therefore not comparable. To address this, we rescale the weighted test loss as defined in equations (17), (18), and (19). This gives the rescaled loss, $\tilde{\mathcal{L}}$, which is mathematically equivalent to $\mathcal{L}(\mathbf{x})$. The rescaled loss is plotted in fig. 5 (right), where we observe that, after rescaling, the loss curves are very close to each other. This confirms that the mathematical equivalency holds. Moreover, this indicates that the weighted loss formulation is more stable compared to the NELBO formulations, as there are no additional factors influencing the training dynamics.

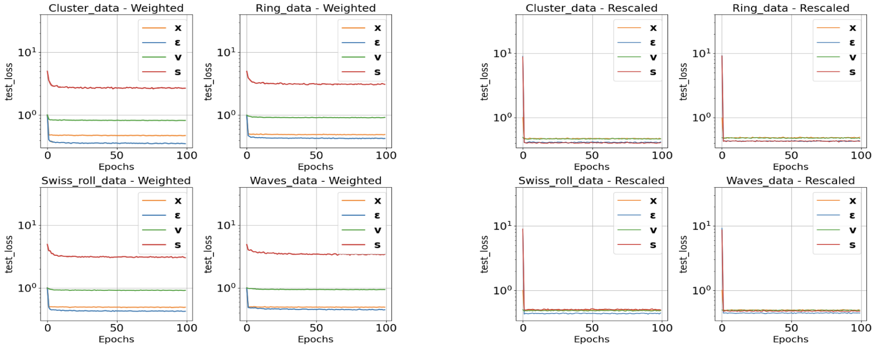

Fig. 5: The weighted test loss $\mathcal{L}$ (left), is not directly comparable across different target predictions. However, the rescaled test loss $\tilde{\mathcal{L}}$ (right), is comparable and demonstrates the mathematical equivalence discussed in section 3.5.

Generated samples: The quality of generated samples shows a different trend compared to loss convergence, indicating that better likelihood estimation does not necessarily correlate with better sample generation, as also discussed in [16]. To analyze the discrepancy, we compare the sample quality using moment-based metrics. Specifically, we measure the mean distance (Euclidean distance between dataset means) and covariance distance (Frobenius norm of the difference between covariance matrices) between real and generated samples. The results are shown in table 2. We see that although the NELBO is better for $\mathbf{x}$ and $\mathbf{s}$ space the sample quality is better for $\boldsymbol{\epsilon}$ and $\mathbf{v}$ space. Moreover, the sample quality for weighted and NELBO loss are similar in most of the cases as also illustrated in fig. 6 which shows 2K generated samples for the ring dataset using weighted and NELBO loss formulations, respectively. This suggests that while the scaling in the NELBO loss functions influences how the model converges, it has little effect on the quality of the generated samples.

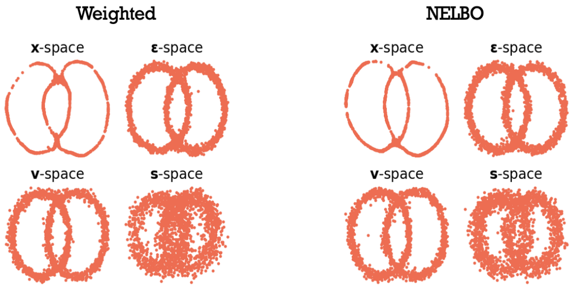

Fig. 6: Comparing 2K samples generated after 512 sampling steps from model trained using weighted loss formulation (left) and NELBO formulation (right)

Table 2: Comparison of NELBO and weighted loss formulations for 2D datasets

Data	**Loss Form**	**NELBO Loss** (L)			**Weighted loss** ($\mathcal{L}$)		
		Nelbo↓	**Mean dist.**↓	**Covar dist.**↓	**Loss**↓	**Mean dist.**↓	**Covar dist.**↓
Cluster data	$\mathbf{x}$	0.6777	0.1754	0.5746	0.4754	0.4300	0.2715
	$\boldsymbol{\epsilon}$	2.3636	0.0364	0.0706	0.3522	0.0634	0.1430
	$\mathbf{v}$	2.5396	0.0307	0.0409	0.8264	0.0389	0.0363
	$\mathbf{s}$	0.7657	0.2279	0.1498	2.6934	0.2688	0.1633
Ring data	$\mathbf{x}$	0.8785	0.2744	0.3807	0.4932	0.2807	0.3974
	$\boldsymbol{\epsilon}$	2.5700	0.0914	0.1107	0.4266	0.0983	0.0366
	$\mathbf{v}$	2.5452	0.0459	0.0718	0.9227	0.0453	0.0088
	$\mathbf{s}$	0.9577	0.2254	0.1563	3.0981	0.2220	0.1637
Swiss data	$\mathbf{x}$	0.8640	0.1133	0.2645	0.4934	0.5256	0.6875
	$\boldsymbol{\epsilon}$	2.5324	0.0689	0.0941	0.4261	0.0857	0.0824
	$\mathbf{v}$	2.4861	0.0418	0.0598	0.9171	0.0427	0.0269
	$\mathbf{s}$	0.9493	0.1266	0.1972	3.0274	0.0893	0.1227
Waves data	$\mathbf{x}$	0.9104	0.1593	0.5559	0.4939	0.1869	0.6911
	$\boldsymbol{\epsilon}$	2.6805	0.0405	0.0757	0.4500	0.0748	0.0778
	$\mathbf{v}$	2.6873	0.0447	0.0738	0.9411	0.0131	0.0271
	$\mathbf{s}$	1.0210	0.0353	0.1369	3.4165	0.0178	0.1676

Loss vs timesteps: In fig. 7, we illustrate the generation of samples using different numbers of sampling steps in the reverse process for the model trained with the weighted loss. The results are similar to those observed with the NELBO loss (see appendix C.1). It can be seen in the image that for the $\mathbf{x}$-space, sample quality declines with more sampling steps but outperforms other objectives with fewer steps, effectively capturing data structure and scale. In contrast, the $\boldsymbol{\epsilon}$-space produces poorer samples with fewer steps, and the sample quality gradually increases. The $\mathbf{v}$-space, captures the data structure well even with fewer sampling steps and the sample quality continues to improve with more steps. The quality of samples generated in the $\mathbf{s}$-space is not good, however, it improves with the number of steps.

One of the reasons for this difference is the loss behavior of various target predictions across timesteps. We visualize the weighted train loss across timesteps for all target predictions on the ring dataset, as shown in Figure (8). Similar trends are observed for other datasets, with corresponding graphs provided in the appendix C.3. In the $\mathbf{x}$-space, the model predicts the original data point at each timestep during the reverse diffusion. As noise increases in the forward process fig. 2, the SNR drops significantly, making prediction harder and resulting in higher losses at later timesteps. In contrast, for $\boldsymbol{\epsilon}$-space, the task is to predict the noise that was added to the data at each timestep. As more noise is introduced, predicting it becomes progressively easier. The $\mathbf{v}$-space formulation as shown in section 3.3 interpolates between data $\mathbf{x}$ and noise $\boldsymbol{\epsilon}$, weighted by

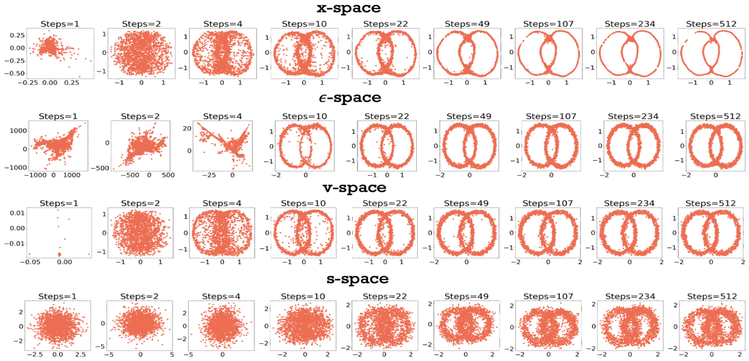

Fig. 7: Generated samples for ring data for different number of sampling steps from model trained on weighted loss $\mathcal{L}$

time dependent functions, requiring the model to find a balance between the two. In $\mathbf{s}$-space the loss is significantly higher in the starting timesteps due to the sensitivity of score matching to noise variance (σ_t), as seen in equation (16). At early timesteps, σ_t^2 becomes negligible and the score function becomes very large in magnitude as $\mathbf{s} \propto \frac{1}{\sigma^2}$, leading to a significant rise in the loss.

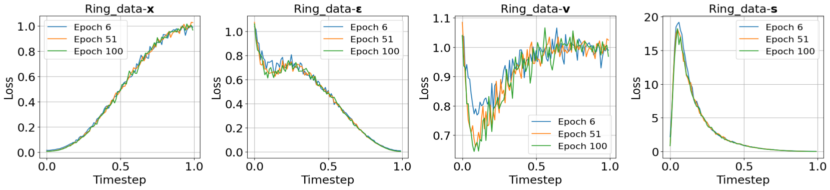

Fig. 8: Behavior of weighted train loss with respect to timesteps for ring data at different epochs

4.3 Results on image dataset

The results of different loss formulations in the image dataset are presented in table 3. We do not include results for score-based metrics because accurately computing them for continuous-time diffusion models in high-dimensional image space requires modeling reverse and forward Stochastic Differential Equations (SDEs), which is beyond the scope of this study and left for future research.

To evaluate the models, we used the NELBO loss to measure how well the model approximates the data likelihood, and the Frechet Inception Distance (FID) to assess the quality of generated samples, which were produced using 500 reverse diffusion steps. We found that the NELBO formulation in $\mathbf{x}$-space has the best performance both in sample quality and probability density estimation. For $\boldsymbol{\epsilon}$ and $\mathbf{v}$ space we found that the weighted loss formulation has better FID scores compared to NELBO. This again indicates that a more accurate likelihood estimation does not necessarily correspond to better sample quality. The images generated from these experiments are provided in the appendix D.

Table 3: Comparison of NELBO and weighted loss formulations for CIFAR10

Data	**Loss Form**	**NELBO Loss** (L)		**Weighted loss** ($\mathcal{L}$)	
		Nelbo↓	**FID↓**	**Loss↓**	**FID↓**
CIFAR10	$\mathbf{x}$	0.0907	17.35	0.0544	19.08
	$\boldsymbol{\epsilon}$	0.8499	39.77	0.0605	20.03
	$\mathbf{v}$	0.8576	36.84	0.1188	31.21

5 Conclusion

In this work, we explored both the theoretical foundations and empirical behavior of various target prediction in diffusion models, with a focus on their corresponding loss formulations under the NELBO and weighted loss frameworks. By systematically deriving and relating the loss functions for different target predictions, that is data $\mathbf{x}$, noise $\boldsymbol{\epsilon}$, rate of change of data distribution $\mathbf{v}$, and score $\mathbf{s}$, we established a unified understanding of how these objectives are connected at a theoretical level.

We designed experiments to evaluate whether the mathematical equivalence of these objectives translates into similar empirical performance. Our results show that, despite theoretical equivalence, practical performance can differ significantly in certain scenarios. In particular, we observed variation in loss convergence, likelihood estimation, and sample quality across loss formulations. Among the NELBO variants, the formulation in the $\mathbf{x}$-space yielded the best likelihood estimates. The quality of generated samples was found to be comparable across both NELBO and weighted loss formulations in most cases for 2D datasets. In contrast, for image data, the weighted loss showed improved performance in the $\boldsymbol{\epsilon}$ and $\mathbf{v}$-spaces.

While our analysis is primarily conducted on 2D synthetic datasets, the insights gained offer a foundation for more extensive experiments on high dimensional image data. These findings highlight the importance of the choice of training objective in diffusion models and its impact on both model performance and sample quality. Overall, our study provides insights into the practical consequences of loss formulations and lays the groundwork for further research on optimizing training objectives in diffusion models.

Acknowledgments. This research is supported by the Center for Artificial Intelligence (CAIRO) at the Technical University of Applied Sciences Würzburg-Schweinfurt (THWS), Würzburg, Germany.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., Yoon, S.: Perception prioritized training of diffusion models. In: Proceedings of the IEEE/CVF Conference on CVPR (2022)
2. Efron, B.: Tweedie's formula and selection bias. Journal of the American Statistical Association (2011)
3. Hang, T., Gu, S., Li, C., Bao, J., Chen, D., Hu, H., Geng, X., Guo, B.: Efficient diffusion training via min-snr weighting strategy. In: Proceedings of the IEEE/CVF international conference on computer vision (2023)
4. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems (2020)
5. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems (2022)
6. Kingma, D., Gao, R.: Understanding diffusion objectives as the elbo with simple data augmentation. Advances in Neural Information Processing Systems (2024)
7. Kingma, D., Salimans, T., Poole, B., Ho, J.: Variational diffusion models. Advances in neural information processing systems (2021)
8. Kingma, D.P., Welling, M., et al.: An introduction to variational autoencoders. Foundations and Trends® in Machine Learning (2019)
9. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International conference on machine learning (2021)
10. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015. Springer (2015)
11. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)
12. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
13. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems (2019)
14. Song, Y., Ermon, S.: Improved techniques for training score-based generative models. Advances in neural information processing systems (2020)
15. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
16. Theis, L., Oord, A.v.d., Bethge, M.: A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844 (2015)
17. Vincent, P.: A connection between score matching and denoising autoencoders. Neural computation (2011)