

# Spiking Denoising Diffusion Probabilistic Models

Jiahang Cao<sup>1\*</sup> Ziqing Wang<sup>1,2\*</sup> Hanzhong Guo<sup>3\*</sup> Hao Cheng<sup>1</sup> Qiang Zhang<sup>1</sup> Renjing Xu<sup>1†</sup>

<sup>1</sup>The Hong Kong University of Science and Technology (Guangzhou)

<sup>2</sup>North Carolina State University, <sup>3</sup>Renmin University of China

{jcao248, hcheng046, qzhang749}@connect.hkust-gz.edu.cn,

guohezhang@ruc.edu.cn, zwang247@ncsu.edu, renjingxu@hkust-gz.edu.cn

## Abstract

Spiking neural networks (SNNs) have ultra-low energy consumption and high biological plausibility due to their binary and bio-driven nature compared with artificial neural networks (ANNs). While previous research has primarily focused on enhancing the performance of SNNs in classification tasks, the generative potential of SNNs remains relatively unexplored. In our paper, we put forward Spiking Denoising Diffusion Probabilistic Models (SDDPM), a new class of SNN-based generative models that achieve high sample quality. To fully exploit the energy efficiency of SNNs, we propose a purely Spiking U-Net architecture, which achieves comparable performance to its ANN counterpart using only 4 time steps, resulting in significantly reduced energy consumption. Extensive experimental results reveal that our approach achieves state-of-the-art on the generative tasks and substantially outperforms other SNN-based generative models, achieving up to 12 $\times$  and 6 $\times$  improvement on the CIFAR-10 and the CelebA datasets, respectively. Moreover, we propose a threshold-guided strategy that can further improve the performances by 2.69% in a training-free manner. The SDDPM symbolizes a significant advancement in the field of SNN generation, injecting new perspectives and potential avenues of exploration. Our code is available at <https://github.com/AndyCao1125/SDDPM>.

## 1. Introduction

Spiking neural networks (SNNs), being regarded as the third generation of neural networks, are potential competitors to artificial neural networks (ANNs) due to their distinguished properties: high biological plausibility, event-driven nature, and low power consumption. In SNNs, all information is represented by binary time series data rather than continuous representation, which allows SNNs

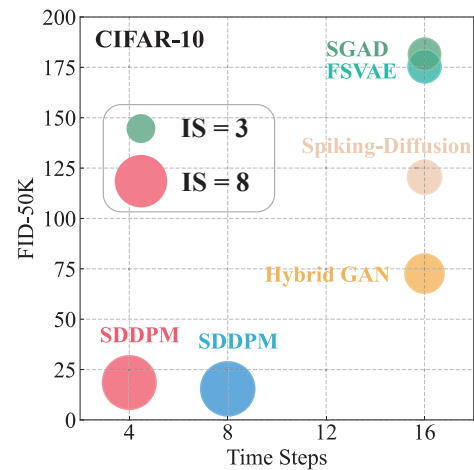


Figure 1. **Comparisons of the SNN-based generative models.** The Fréchet Inception Distance (FID) serves as a measure of image quality, with lower values indicating superior performance. The Inception Score (IS) acts as an alternate measure of model performance, where a higher score is desirable. The size of the markers denotes the IS score. In comparison to other SNN-based generative models, our models demonstrate state-of-the-art performance with fewer time steps.

to adopt low-power accumulation (AC) instead of the traditional high-power multiply-accumulation (MAC), leading to significant energy efficiency gains. Existing works reveal that on specialized hardware, such as Loihi [7] and TrueNorth [1], SNNs can save energy by orders of magnitude compared with ANNs. Additionally, SNNs follow their biological counterparts and inherit complex temporal dynamics from them, endowing SNNs with powerful abilities to extract spatial-temporal features in a variety of tasks, including recognition [10, 53, 61], tracking [57], segmentation [26] and images restoration [31].

However, most of the existing research on SNNs primarily focuses on classification-based tasks, and the regression capability of SNNs has not been well demonstrated, especially in image generation tasks. Spiking-GAN [27] is the first SNN-based image generation model, but its low per-

\*Equal contribution.

†Corresponding author.

formance and limited experimentation on handwritten data hinder a sufficient demonstration of its generative ability on high-dimensional data. Kamata *et al.* [22] propose a fully spiking variational autoencoder (FSVAE) combined with discrete Bernoulli sampling and claim that the quality of the generated images surpasses the ANN-based VAE in the same setting but the quality of the generated images is limited for practical applications. [18,38]. Consequently, it is imperative to develop a generative algorithm capable of producing high-quality images while also reducing energy consumption.

Recently, diffusion models have achieved remarkable success in generation tasks [11,23] since they offer several advantages compared with other deep generative models (DGMs). Firstly, diffusion’s regression loss makes its training more stable than the adversarial loss in the Generative Adversarial Networks (GANs) and therefore the diffusion model is more suitable for large-scale generation tasks. [41]. Secondly, the training objective of diffusion models is derived directly from the likelihood perspective, so the problem of mode collapse can be avoided when the model converges. Furthermore, since the diffusion models can be viewed as a VAE for a given encoder, it is easier to optimize. These advantages provide the impetus for us to investigate the feasibility of incorporating SNNs into the diffusion model, leveraging the generative capabilities of diffusion models along with the energy efficiency inherent in SNNs.

In this work, we propose Spiking Denoising Diffusion Probabilistic Models (SDDPM), a novel category of SNN-based diffusion models exhibiting exceptional image generation capabilities. To fully leverage the energy efficiency of SNNs, we propose the Spiking U-Net architecture that achieves comparable performance to its ANN counterpart while employing only 4 spiking time steps, resulting in significantly reduced energy consumption. Moreover, we employ a pre-spike structure to ensure the accurate transmission of spikes. We also propose training-free threshold guidance, which further enhances the quality of the generated images by adjusting the threshold value of the spiking neurons. Comprehensive experimental results demonstrate that threshold guidances contribute to the facilitation of SDDPM. Our approach is evaluated on four datasets: MNIST, Fashion-MNIST, CIFAR-10, and CelebA. As shown in Fig. 1, we demonstrate that the proposed SDDPM outperforms all SNN-based generative models by a significant margin, requiring only a small number of spiking time steps. We also conduct extensive ablation studies to reveal the effectiveness of each component. To sum up, our contributions lie in four folds:

- To the best of our knowledge, SDDPM is the first work that employs spiking neural networks on diffusion models.

- To fully exploit the energy efficiency of SNN, We design a purely Spiking U-Net that can achieve comparable performance to its ANN counterpart while saving 62.5% energy consumption.
- Extensive experiments show that SDDPM achieves state-of-the-art performances among SNN-based generative models. Specifically, our proposed SDDPM outperforms the SNN-based baselines by up to 1200% and 600% on the CIFAR-10 and CelebA datasets with only 4 spiking time steps.
- We also introduce a threshold-guidance strategy aimed at further enhancing performance, which results in a 2.69% improvement without any additional training.

## 2. Related Work

**Training Methods of Spiking Neural Networks.** Generally, there are two ways to obtain deep SNN models: ANN-to-SNN conversion and direct training. The ANN-to-SNN conversion [5,9,12,32,53] involves converting a pre-trained ANN into an SNN by replacing the ReLU activation layers with spiking neurons, which allows the SNN to simulate the behavior of the original ANN using spiking neurons. This conversion method is generally known to achieve higher accuracy compared to direct training methods. However, the conversion methods typically require a longer training time compared to direct training methods, resulting in the need for more training resources. On the other hand, direct training methods involve training the SNN directly from scratch. Surrogate gradients [30,37,55] are utilized for addressing the non-differentiability problem of spiking neurons, enabling the training of SNNs using gradient-based optimization techniques. In our study, we explore the feasibility of implementing diffusion models in SNNs using the direct train method, aiming to reduce power consumption and investigate the potential generative abilities of SNNs.

**Diffusion Models.** Some research focuses on analyzing the theoretical foundation and formulation of diffusion models [18,47–49]. What’s more, diffusion model is divided into discrete time diffusion and continuous time diffusion [23,25,49] depending on whether the time step in diffusion is sampled from discrete distribution or continuous distribution. Certain solvers [2,13,21,36] have been proposed to expedite the sampling process in diffusion models. Additionally, some studies are dedicated to designing more efficient diffusion networks [3,39].

**Spiking Neural Network in Generative Models.** There has been some prior research investigating the capabilities of SNNs in generative tasks. VDIB [46] is a hybrid variational autoencoder, consisting of an SNN-based encoder and an ANN-based encoder. Hybrid guided-VAE [50] and

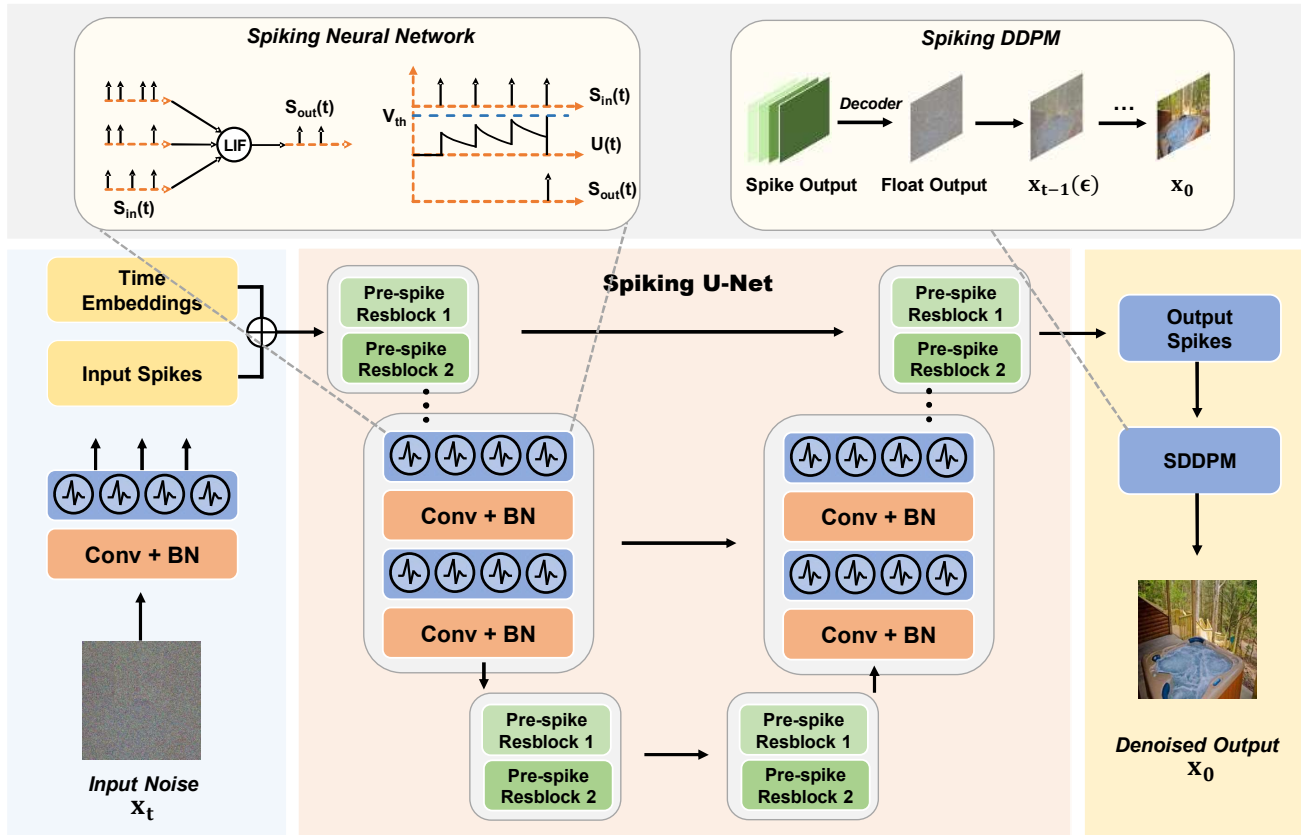


Figure 2. **Illustration of the architecture and pipeline of Spiking Diffusion Models.** The SDDPM architecture is suitable for use on top of any existing diffusion models, where we inherit the most commonly used U-Net backbone and propose the Spiking U-Net. Our network consists of several Pre-spike Resblocks (colored in green), each of which contains spiking neurons (blue) and Conv-BatchNorm layers (orange). Given a random Gaussian noise input  $x_t$ , it is converted into the spikes by an encoding layer and subsequently fed into the Spiking U-Net along with the time embeddings. The network transmits only spikes, represented by 0/1 vector. The output spikes  $S_{out}(t)$  are formed as a result of the accumulation of membrane potentials  $U(t)$  within the neuron under the influence of the consecutive input spikes  $S_{in}(t)$ . Once the membrane potential exceeds the threshold  $V_{th}$ , the neuron will generate a spike. Eventually, the output spikes are passed through a decoding layer to obtain the predicted noise  $\epsilon$ , followed by  $N$  times denoising to restore the image  $x_0$ .

hybrid GAN [43] also adopt SNN-ANN architecture. However, the aforementioned approaches rely on the ANNs, resulting in the entire model not being fully deployed on neuromorphic hardware. Spiking GAN [27] incorporates a fully SNN-based backbone and utilizes a time-to-first-spike coding scheme. Kamata *et al.* [22] introduce a fully spiking variational autoencoder (FSVAE), which samples images according to the Bernoulli distribution. Recently, Feng *et al.* [16] construct a spiking generative adversarial network with attention-scoring decoding for handling complex images, and Liu *et al.* [33] propose a spike-based vector quantized variational autoencoder (VQ-SVAE) to learn a discrete latent space for images. However, the primary limitation of existing spiking generative models is their low performance and poor generated image quality. These drawbacks hinder their competitiveness in the field of generative models, despite their low energy consumption. To tackle this issue, we introduce the Spiking Denoising Diffusion Probabilistic

Model (SDDPM), which not only delivers substantial improvements over existing SNN-based generative models but also preserves the advantages of SNNs.

### 3. Background

#### 3.1. Spiking Neural Network

The spiking neural network is a bio-inspired algorithm that mimics the actual signaling process occurring in brains. Compared to the artificial neural network, it transmits sparse spikes instead of continuous representations, offering benefits such as low energy consumption and robustness. In this paper, we adopt the widely used Leaky Integrate-and-Fire (LIF [6, 20]) model, which effectively characterizes the dynamic process of spike generation and can be defined as:

$$\tau \frac{dV(t)}{dt} = -(V(t) - V_{reset}) + I(t), \quad (1)$$

where  $I(t)$  represents the input synaptic current at time  $t$  to charge up to produce a membrane potential  $V(t)$ ,  $\tau$  is the time constant. When the membrane potential exceeds the threshold  $\vartheta_{\text{th}}$ , the neuron will trigger a spike and resets its membrane potential to a value  $V_{\text{reset}}$  ( $V_{\text{reset}} < \vartheta_{\text{th}}$ ). The LIF neuron achieves a balance between computing cost and biological plausibility.

In practice, the dynamics need to be discretized to facilitate reasoning and training. The discretized version of LIF model can be described as:

$$U[n] = e^{\frac{1}{\tau}} V[n-1] + I[n], \quad (2)$$

$$S[n] = \Theta(U[n] - \vartheta_{\text{th}}), \quad (3)$$

$$V[n] = U[n](1 - S[n]) + V_{\text{reset}} S[n], \quad (4)$$

where  $n$  is the time step,  $U[n]$  is the membrane potential before reset,  $S[n]$  denotes the output spike which equals 1 when there is a spike and 0 otherwise,  $\Theta(x)$  is the Heaviside step function,  $V[n]$  represents the membrane potential after triggering a spike. In addition, we use the ‘‘hard reset’’ method [15] for resetting the membrane potential in Eq. (4), which means that the value of the membrane potential  $V[n]$  after triggering a spike ( $S[n] = 1$ ) will go back to  $V_{\text{reset}} = 0$ .

### 3.2. Diffusion Models and Classifier Guidance

Diffusion models gradually perturb data with a forward diffusion process and then learn to reverse such process to recover the data distribution.

Formally, let  $x_0 \in \mathbb{R}^n$  be a random variable with unknown data distribution  $q(x_0)$ . The forward diffusion process  $\{x_t\}_{t \in [0, T]}$  indexed by time  $t$ , can be represented by the following forward stochastic differential equations (SDE):

$$dx_t = f(t)x_t dt + g(t)d\omega, \quad x_0 \sim q(x_0), \quad (5)$$

where  $\omega \in \mathbb{R}^n$  is a standard Wiener process. Let  $q(x_t)$  be the marginal distribution of the above SDE at time  $t$ . Its corresponding reversal process can be described by another SDE which recovers the data distribution from noise [49]:

$$dx = [f(t)x_t - g^2(t)\nabla_{x_t} \log q(x_t)] dt + g(t)d\bar{\omega}, \quad (6)$$

where  $\bar{\omega} \in \mathbb{R}^n$  is a reverse-time standard Wiener process and this reversal SDE starts from  $x_T \sim q(x_T)$ . In Eq. (6), the only unknown term is the score function  $\nabla_{x_t} \log q(x_t)$ . To estimate this term, prior works [18, 23, 49] employ a noise network  $\epsilon_\theta(x_t, t)$  to estimate scaled score function  $\sigma(t)\nabla_{x_t} \log q(x_t)$  via denoising score matching (DSM) [52], which ensures that the optimal solution satisfies  $\epsilon_\theta(x_t, t) = -\sigma(t)\nabla_{x_t} \log q(x_t)$ , where  $\sigma(t)$  denotes the variance of  $q(x_t|x_0) \sim \mathcal{N}(x_t|a(t)x_0, \sigma^2(t)I)$ , which is related to the notation in Eq. (5) as shown in Eq. (7),

$$f(t) = \frac{d \log a(t)}{dt}, \quad g^2(t) = \frac{d\sigma^2(t)}{dt} - 2\sigma^2(t) \frac{d \log a(t)}{dt}. \quad (7)$$

Hence, sampling can be achieved by discretizing the reverse SDE in Eq. (6) by replacing the  $\nabla_{x_t} \log q(x_t)$  with noise network  $-\frac{\epsilon_\theta(x_t, t)}{\sigma(t)}$ . Furthermore, to enable conditional sampling, such as sampling cat images, we can refine the reverse stochastic differential equation (SDE) presented in Eq. (6) as follows [11]:

$$\epsilon_\theta(x_t, c) = \epsilon_\theta(x_t) - s\sigma(t)\nabla_{x_t} \log p_\phi(c|x_t, t), \quad (8)$$

Here,  $p_\phi(c|x_t, t)$  represents the classifier,  $s$  denotes the temperature controlling the intensity of guidance, and Eq. (8) indicates that a conditional sample can be generated using only a pre-trained noise network and a classifier. Ho *et al.* [19] introduced classifier-free guidance, which significantly enhances the diversity of generated samples. This methodology has found extensive application in practical scenarios [45], as demonstrated by the works of Ho . [19].

Furthermore, it is important to note that the guidance mentioned above is not limited to a specific category, which can be applied to various forms of guidance. For example, in some studies [4, 59], energy-based guidance is proposed to facilitate image translation and molecular design. Additionally, Kim *et al.* [24] introduce discriminator guidance to mitigate estimation bias of the noise network, resulting in state-of-the-art performance on the CIFAR-10 dataset.

## 4. Method

In this section, we introduce our methodologies in three stages. In Sec. 4.1, we introduce our proposed Spiking U-Net and provide a comprehensive explanation of its network architecture. Then, we present the pre-spike residual structure in Sec. 4.2. Eventually, we put forward a threshold-guiding strategy and its corresponding theory in Sec. 4.3. The computational formulations for calculating the energy consumption of the SNNs are given in the Supplementary Material.

### 4.1. Spiking U-Net Structure

The overview of the architecture and sampling pipeline is illustrated in Fig. 2. Spiking U-Net is the main component of the whole SDDPM structure. Unlike previous work [43, 46, 50] that use hybrid architecture consisting of SNN and ANN, we introduce a purely SNN-based structure, thereby fully leveraging the enhanced energy efficiency inherent to SNNs.

The ANN-based U-Net utilized in DDPM [18] is characterized by a residual block (resblock) defined as:

$$O^l = \text{Conv}^l(\text{Swish}(\text{GN}^l(O^{l-1}))) + O^{l-1}, \quad (9)$$

where  $O^l$  is the output representation at layer  $l$ ,  $\text{GN}$  signifies the group normalization operation, and  $\text{Swish}$  [40] represents the activation function.



However, directly employing  $GN$  in SNNs may result in performance degradation due to distribution mismatch [53]. Consequently, we substitute the  $GN$  in the U-Net architecture with batch normalization, which is a more SNN-compatible normalization technique [14, 61]. This modification allows the model to better capture spatial features. The residual block in our Spiking U-Net can be formulated as follows:

$$O^l = BN^l(Conv^l(S^{l-1})) + S^{l-1}, \quad (10)$$

$$S^l = SpikeNeuron(O^l), \quad (11)$$

$$O^{l+1} = BN^{l+1}(Conv^{l+1}(S^l)) + S^l, \quad (12)$$

$$S^{l+1} = SpikeNeuron(O^{l+1}), \quad (13)$$

where  $S^l$  is the output spikes at layer  $l$ ,  $BN$  denotes the batch normalization operation and  $SpikeNeuron$  means the spiking activation function in Eq. (3).

The Spiking U-Net receives an input of a 2D image batch  $I_s \in \mathbb{R}^{B \times C \times H \times W}$ , with  $B, C, H$ , and  $W$  standing for batch size, channel, height, and width, respectively. Initially, the image is replicated  $T$  times, resulting in a sequence of images  $I \in \mathbb{R}^{T \times B \times C \times H \times W}$ , a necessary operation for the SNN to incorporate temporal dimension information. However, the 2D convolution and BN cannot directly process the added  $T$  dimension. To circumvent this, we fuse the  $T$  and  $B$  dimensions, represented mathematically as  $I_{fused} \in \mathbb{R}^{TB \times C \times H \times W}$ , which allows the network to concurrently analyze spatial and temporal features.

## 4.2. Pre-spike Residual Learning

In this section, we further explore the structure of the Spiking U-Net. Although the above design can fully apply the U-Net into SNN, it could cause the output range of the residual block to overflow. This is due to the fact that the previous shallow network output  $S^{l-1}$  and the residual mapping representation  $S^l$  are both spike series ( $\{0, 1\}$ ), thus their summation  $O^l$  would result in a value domain of  $\{0, 1, 2\}$ , where  $\{2\}$  is a pathological case without any biological plausibility. This could lead to a larger range of spike signals when the layers become deeper [60], incurring higher energy consumption.

Inspired by [35, 58], we for the first time apply pre-spike residual learning with the structure of *Activation-Conv-BatchNorm* in our Spiking U-Net, so as to overcome the problem of gradient explosion/vanishing and performance degradation in convolution-based SNNs. Through the pre-spike blocks, the residuals, and outputs are summed by floating point addition operation, ensuring that the representation is accurate before entering the next spiking neuron while avoiding the pathological condition mentioned above. The whole pre-spike residual learning process inside a res-

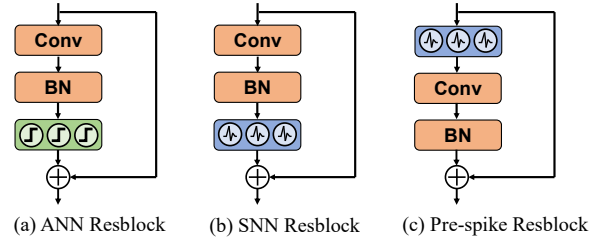


Figure 3. **Comparisons of the residual structures and Pre-spike structure.** Standard SNN resblock (b) entirely inherits from ANN structure (a). In contrast, pre-spike resblock activates first.

block can be formulated as below:

$$S^l = SpikeNeuron(O^{l-1}), \quad (14)$$

$$O^l = BN^l(Conv^l(S^l)) + O^{l-1}, \quad (15)$$

$$S^{l+1} = SpikeNeuron(O^l), \quad (16)$$

$$O^{l+1} = BN^{l+1}(Conv^{l+1}(S^{l+1})) + O^l. \quad (17)$$

Through the pre-spike residual mechanism, the output of the residual block can be summed by two floating points  $BN^l(Conv^l(S^l))$ ,  $O^{l-1}$  at the same scale and then enter the spiking neuron at the beginning of the next block, which guarantees that the energy consumption is still very low. We illustrate the diagram of different resblocks in Fig. 3. Experiments to verify the superiority of the pre-spike structure can be found in Sec. 5.5.

## 4.3. Threshold Guidance in SDDPM

Recall that sampling can be achieved by substituting the score  $\nabla_{x_t} \log q(x_t)$  with either the score network  $s_\theta(x_t, t)$  or the scaled noise network  $-\frac{\epsilon_\theta(x_t, t)}{\sigma(t)}$  while discretizing the reverse SDE as presented in Eq. (6). Because of the inaccuracy of the network estimates, we have the fact that  $s_\theta(x_t, t) \approx -\frac{\epsilon_\theta(x_t, t)}{\sigma(t)} \neq \nabla_{x_t} \log q(x_t)$  in most cases. Therefore, in order to sample better results, we can discretize the following rectified reverse SDE [24]:

$$dx = [f(t)x_t - g^2(t)[s_\theta + c_\theta](x_t, t)] dt + g(t)d\bar{\omega}, \quad (18)$$

where  $s_\theta(x_t, t)$  represents the score network or scaled noise network, while  $c_\theta(x_t, t) = \nabla_{x_t} \log \frac{q(x_t)}{p_\theta(x_t, t)}$  denotes the rectified term for the original reverse stochastic differential equation (SDE) with the estimation errors of neural network. The omission of the rectified term  $c_\theta(x_t, t)$  reduces discretization errors and improves sampling performance. However, the practical calculation of  $c_\theta(x_t, t)$  presents challenges due to the intractability of  $q(x_t)$  and  $p_\theta(x_t, t)$ .

In light of the existence of estimation errors, the formulation in Eq. (18) motivates us to explore if we can improve the sampling performance without additional training by computing  $c_\theta(x_t, t)$ . Although a direct computation of this

Dataset	Model	Method	#Param (M)	Time Steps	IS $\uparrow$	FID $\downarrow$
MNIST*	DDPM [18]	ANN	64.47	/	-	28.70
	FSVAE [22]	SNN	3.87	16	6.209	97.06
	SGAD [16]	SNN	-	16	-	69.64
	Spiking-Diffusion [33]	SNN	-	16	-	37.50
	<b>SDDPM</b>	SNN	63.61	4	-	<b>29.48</b>
Fashion MNIST*	DDPM [18]	ANN	64.47	/	-	20.24
	FSVAE [22]	SNN	3.87	16	4.551	90.12
	SGAD [16]	SNN	-	16	-	165.42
	Spiking-Diffusion [33]	SNN	-	16	-	91.98
	<b>SDDPM</b>	SNN	63.61	4	-	<b>21.38</b>
CelebA*	DDPM [18]	ANN	64.47	/	-	20.34
	FSVAE [22]	SNN	6.37	16	3.697	101.60
	SGAD [16]	SNN	-	16	-	151.36
	<b>SDDPM</b>	SNN	63.61	4	-	<b>25.09</b>
LSUN bedroom*	DDPM [18]	ANN	64.47	/	-	29.48
	<b>SDDPM</b>	SNN	63.61	4	-	47.64
CIFAR-10	DDPM [18]	ANN	64.47	/	8.380	19.04
	DDPM <sub>ema</sub> [18]	ANN	64.47	/	8.846	13.38
	FSVAE [22]	SNN	3.87	16	2.945	175.50
	SGAD [16]	SNN	-	16	-	181.50
	Spiking-Diffusion [33]	SNN	-	16	-	120.50
	<b>SDDPM</b>	SNN	63.61	4	7.440	19.73
	<b>SDDPM</b>	SNN	63.61	8	7.584	17.27
	<b>SDDPM (TG)</b>	SNN	63.61	4	7.482	19.20
	<b>SDDPM (TG)</b>	SNN	63.61	8	<b>7.655</b>	<b>16.89</b>

Table 1. **Results for different dataset.** In all datasets, SDDPM (Ours) outperforms all SNN-based baselines and even some ANN models in terms of sample quality, which is mainly measured by FID $\downarrow$  and IS $\uparrow$ . Results of  $\nabla$  and  $\natural$  are taken from [22] and [16], respectively. *ema* indicates the utilization of EMA training method [51]. For fair comparisons, we re-evaluate the results of DDPM [18] using the same U-Net architecture as SDDPM. We employ the symbol ‘/’ to represent ‘None’ since ANN does not have the concept of time step. \* denotes that only FID is used for measurement since these data distributions are far from ImageNet, making Inception Score less meaningful.

term is infeasible, we can seek suitable approximations to enhance the effectiveness of our sampling process. Meanwhile, a crucial parameter in the SNN is the spike threshold  $V_{th}$ , which influences the SNN’s output. We put forward a threshold guidance (TG) by adjusting the threshold by:

$$\begin{aligned}
& s_{\theta}(x_t, t, V'_{th}) \\
& \approx s_{\theta}(x_t, t, V_{th}^0) + \frac{ds_{\theta}(x_t, t, V_{th})}{dV_{th}} dV_{th} + O(dV_{th}) \\
& \approx s_{\theta}|_{V_{th}^0} + s'_{\theta}|_{V_{th}^0} dV_{th} + O(dV_{th}) \\
& \approx s_{\theta}(x_t, t) + c_{\theta}(x_t, t), \tag{19}
\end{aligned}$$

which means that we can adjust the threshold in SNNs to estimate the rectified term  $c_{\theta}(x_t, t)$ .  $V_{th}^0$  represents the threshold utilized during the training stage, while  $V'_{th}$  denotes the adjusted threshold employed during the inference stage in Eq. (19). The first equation is derived through Taylor expansion. Eq. (19) indicates that adjusting the threshold can

enhance the final sampling outcomes when the derivative term is correlated with the rectified term. Moreover, modifying the threshold allows for the manipulation of both the overall quality and diversity of the generated images, particularly in scenarios where image generation is not highly accurate. A lower threshold encourages the occurrence of more spikes. Experiments show that TG can improve sample quality without extra training. We label cases with decreasing thresholds as inhibitory guidance and the opposite as excitatory guidance.

## 5. Experiment

### 5.1. Experiment Settings

**Datasets and Baselines** To demonstrate the effectiveness and efficiency of the proposed algorithm, we conduct experiments on  $32 \times 32$  MNIST [29],  $32 \times 32$  Fashion-MNIST [54],  $32 \times 32$  CIFAR-10 [28],  $64 \times 64$  CelebA [34]

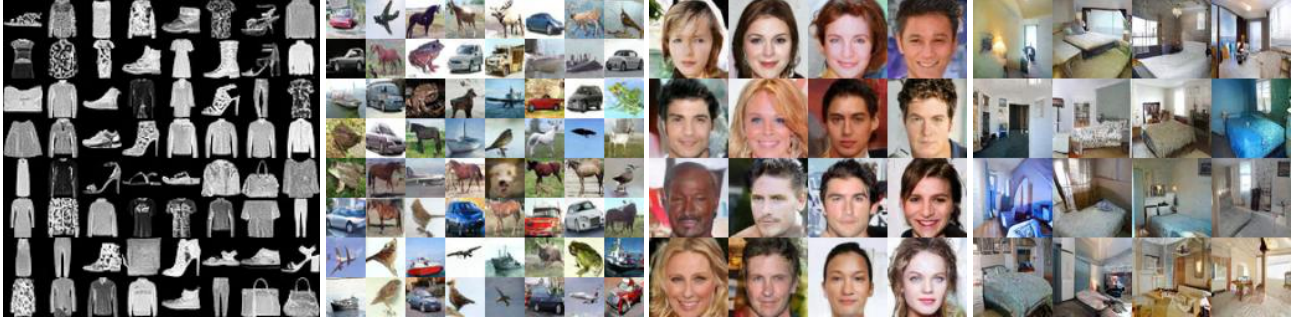


Figure 4. Unconditional image generation results on Fashion-MNIST, CIFAR-10, CelebA and LSUN bedroom by using SDDPM.

and  $64 \times 64$  LSUN bedroom [56]. We use existing spiking generative models FSVAE [22], SGAD [16] and Spiking-Diffusion [33] as our baselines. We also compare our results with ANN baselines.

**Evaluation Metrics.** The qualitative results are compared according to Fréchet Inception Distance (FID [17], lower is better) and Inception Score (IS [44], higher is better). IS evaluates the quality of synthetic images by maximizing the average entropy of Inception V3 model’s probability distribution. FID computes the KL divergence between the assumed Gaussian latent spaces of real and generated images. Both metrics are calculated using 50,000 generated images.

**Implementation Details.** Our Spiking U-Net inherits the standard U-Net [42] architecture and no attention blocks are used. For the hyper-parameter settings, we set the decay rate  $e^{\frac{1}{\tau}}$  in Eq. (2) as 1.0 and the spiking threshold  $\vartheta_{th}$  as 1.0. The SNN simulation time step is  $4/8$ . The learning rate is set as  $1e-5$  with batch size 128 and we train the model without exponential moving average (EMA [51]). U-Net also does not employ attention blocks, and its training process is consistent with Spiking U-Net. More details of the model and the implementation codes can be found in the Supplementary Material.

## 5.2. Comparisons with the state-of-the-art

In Tab. 1, we present a comparative analysis of our Spiking Denoising Diffusion Probabilistic Models (SD-DPM) with state-of-the-art generative models in unconditional generations. To ensure a comprehensive comparison, we also include results derived from ANNs as benchmarks. Fig. 4 provides a visual representation of the qualitative results obtained. Our results demonstrate that *SDDPM outperforms SNN baselines across all datasets by a significant margin*, even with smaller spiking simulation steps ( $4/8$ ). In particular, on the CelebA dataset, SDDPM has  $4 \times$  and  $6 \times$  FID improvement in comparison to FSVAE and SGAD, respectively. Both of these competing models require 16 time steps. On the CIFAR-10 dataset, the enhancement factor is even more substantial, with SDDPM achieving  $11 \times$ ,  $12 \times$  and  $7 \times$  improvements over FSVAE, SGAD and Spiking-

Diffusion, respectively. Moreover, the quality of generated samples escalates with an increase in the number of time steps. In specific, our SDDPM attains a level of sample quality that is comparable to the ANN benchmarks with the same U-Net architecture. In certain instances, such as an FID score comparison of 17.27 (SDDPM) against 19.04 (DDPM), *the SDDPM even outperforms the ANN models*. This outcome highlights the superior expressive capability of SNNs employed in our model. *SDDPM also demonstrates the generative ability on large-scale datasets*. On the LSUN bedroom dataset, which contains more than 3 million images, SDDPM demonstrates commendable qualitative results as depicted in Fig. 4.

## 5.3. Effectiveness of Threshold Guidance

In Sec. 4.3, we propose a training-free method: Threshold Guidance (TG), which could improve the quality of the generated images by simply changing the threshold of the spiking neurons slightly during inference. As illustrated in Tab. 2, inhibitory guidance helps to further improve the quality of the generated images on two metrics: the Fréchet Inception Distance drops from 19.73 to 19.20 upon decreasing the threshold by 0.3% and the IS score increased from 7.44 to 7.55 upon reducing the threshold by 0.2%. On the other hand, the excitatory guidance also improves sampling quality in some conditions. Those quantitative results suggest that threshold guidance can provide an effective boost to model performance after the model has been trained, while not costing additional training resources.

## 5.4. Evaluation of the Computational Cost

To further emphasize the low-energy nature of our SD-DPM, we perform a comparative analysis of the FID and energy consumption between the proposed SDDPM and its corresponding ANN model. As shown in Tab. 3, when the (spiking) time step is set at 4, the SDDPM presents significantly lower energy consumption, amounting to merely 37.5% of that exhibited by the ANN model. Moreover, the FID of SDDPM also improved by 0.47, indicating that our model can effectively minimize energy consumption while

Method	Threshold	FID↓	IS↑
Baseline	1.000	19.73	7.44
Inhibitory Guidance	0.999	<b>19.25</b>	<b>7.48</b>
	0.998	19.38	<b>7.55</b>
	0.997	<b>19.20</b>	7.47
Excitatory Guidance	1.001	20.00	7.47
	1.002	19.98	<b>7.48</b>
	1.003	20.04	7.46

Table 2. **Results on CIFAR-10 by different threshold guidances.** The top-1 and top-2 results are colored in red and blue, respectively. The findings indicate that TG can further enhance the FID score by adjusting the spike threshold.

Models	DDPM-ANN	SDDPM-4T	SDDPM-8T
FID↓	19.04	19.20	<b>16.89</b>
Energy (mJ)↓	29.23	<b>10.97</b>	22.96

Table 3. **Comparisons of energy and FID of SNN and ANN models.** In comparison to ANN, SNN models exhibit reduced energy consumption while attaining superior FID outcomes.

maintaining competitive performance. As time steps grow from 4 to 8, we witness a corresponding decline in the FID at the cost of elevated energy consumption. This observation points to a trade-off between FID improvement and the associated energy expenses as time steps increase.

### 5.5. Ablation Study

**Impact of different residual learnings.** To showcase the superiority of the pre-spike learning approach we utilize, we compare the FID score with that of the traditional spiking residual block on CIFAR-10 dataset. The results of our study, presented in Tab. 4, reveal that our pre-spike-based model outperforms its traditional counterpart in terms of FID score, thus demonstrating the supremacy of the pre-spike learning approach within the context of SNN.

**Effectiveness of TG on different time steps.** Another critical aspect of our study concerns the examination of our TG strategy’s effectiveness with varying spiking time steps. As demonstrated in Tab. 5, our observations confirm a correlation between an increasing number of time steps and an improvement in the performance of SDDPM. Moreover, the implementation of the TG strategy further amplifies this improvement. For instance, with the application of TG, the FID score improves from 19.73 to 19.20, indicating a relative enhancement of approximately 2.69%. This improvement suggests that the TG strategy is a significant contributing factor to the overall performance of our SNN model. It is also worth noting that there is an additional enhancement of the FID performance by further refining the TG strategy and increasing the time steps.

Method	IS↑	FID↓
SNN Resblock	6.25	48.69
Pre-Spike Resblock	<b>7.44</b>	<b>19.73</b>

Table 4. **Ablation study on spiking resblock structures.** We evaluate the performances of two SNN residual methods on the CIFAR-10 dataset. The results demonstrate the superiority of the pre-spike residual method.

Method	Time Steps	TG	FID↓	$\Delta$ (%)
SDDPM	4		19.73	+0.00
	4	✓	<b>19.20 (-0.53)</b>	+2.69
	8		17.27	+0.00
	8	✓	<b>16.89 (-0.38)</b>	+2.20

Table 5. **Ablation study on proposed TG and time step.** The experiments are conducted on SDDPM with 1k denoising steps.  $\Delta$  represents the improvement of FID. The performance of SDDPM is enhanced by both TG and the increasing time steps.

## 6. Discussion

SDDPM presents a promising opportunity for developing SNN-based generative models, owing to its high-quality generation capabilities. Nonetheless, one limitation of our study is that we have not examined higher-resolution datasets (*e.g.*, ImageNet [8], LSUN [56]). Additionally, employing alternative diffusion solvers, such as DDIM [47], and Analytic-DPM [2], merits consideration in an effort to decrease the number of sampling steps. In future research, we plan to explore SNN generative models with more neural states and investigate further applications of SDDPM in the generation domain, attempt to combine it with quantization methods for improving model performance and explore the use of distillation learning in terms of sampling methods.

## 7. Conclusion

In this work, we propose a new class of SNN-based diffusion models named Spiking Denoising Diffusion Probabilistic Models (SDDPM) that combine the energy efficiency of SNNs with superior generative performance. As a pioneering endeavor employing SNNs on diffusion models, SDDPM provides remarkable advances in generative performances, significantly surpassing existing SNN benchmarks with mere 4 time steps. Moreover, we introduce a purely Spiking U-Net architecture, designed to maximize the inherent energy efficiency of SNNs. The architecture demonstrates the feasibility of matching the performance of its ANN counterpart while simultaneously offering energy savings of up to 62.5%. Further, we propose an innovative threshold-guidance strategy to further enhance performance without training. This research signifies a vital step forward in the field of SNN generation, paving the way for future exploration and development in this area.



## References

- [1] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE TCAD*, 34(10):1537–1557, 2015. 1
- [2] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022. 2, 8
- [3] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, pages 22669–22679, 2023. 2
- [4] Fan Bao, Min Zhao, Zhongkai Hao, Peiyao Li, Chongxuan Li, and Jun Zhu. Equivariant energy-guided sde for inverse molecular design. *arXiv preprint arXiv:2209.15408*, 2022. 4
- [5] Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. *arXiv preprint arXiv:2303.04347*, 2023. 2
- [6] Anthony N Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95:1–19, 2006. 3
- [7] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018. 1
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 8
- [9] Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv preprint arXiv:2103.00476*, 2021. 2
- [10] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022. 1
- [11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 2, 4
- [12] Jianhao Ding, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. *arXiv preprint arXiv:2105.11654*, 2021. 2
- [13] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *NeurIPS*, 35:30150–30166, 2022. 2
- [14] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *NeurIPS*, 34:21056–21069, 2021. 5
- [15] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *ICCV*, pages 2661–2671, 2021. 4
- [16] Linghao Feng, Dongcheng Zhao, and Yi Zeng. Sgad: Spiking generative adversarial network with attention scoring decoding. *arXiv preprint arXiv:2305.10246*, 2023. 3, 6, 7
- [17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 7
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2, 4, 6
- [19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4
- [20] Eric Hunsberger and Chris Eliasmith. Spiking deep networks with lif neurons. *arXiv preprint arXiv:1510.08829*, 2015. 3
- [21] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021. 2
- [22] Hiromichi Kamata, Yusuke Mukuta, and Tatsuya Harada. Fully spiking variational autoencoder. In *AAAI*, volume 36, pages 7059–7067, 2022. 2, 3, 6, 7
- [23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022. 2, 4
- [24] Dongjun Kim, Yeongmin Kim, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv preprint arXiv:2211.17091*, 2022. 4, 5
- [25] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *NeurIPS*, 34:21696–21707, 2021. 2
- [26] Paul Kirkland, Gaetano Di Caterina, John Soraghan, and George Matich. Spikeseg: Spiking segmentation via stdp saliency mapping. In *IJCNN*, pages 1–8. IEEE, 2020. 1
- [27] Vineet Kotariya and Udayan Ganguly. Spiking-gan: A spiking generative adversarial network using time-to-first-spike coding. In *IJCNN*, pages 1–7. IEEE, 2022. 1, 3
- [28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6
- [29] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. att labs, 2010. 6
- [30] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in Neuroscience*, page 119, 2020. 2
- [31] Siqi Li, Yutong Feng, Yipeng Li, Yu Jiang, Changqing Zou, and Yue Gao. Event stream super-resolution via spatiotemporal constraint learning. In *ICCV*, pages 4480–4489, 2021. 1
- [32] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *ICML*, pages 6316–6325. PMLR, 2021. 2

- [33] Mingxuan Liu, Rui Wen, and Hong Chen. Spiking-diffusion: Vector quantized discrete diffusion model with spiking neural networks. *arXiv preprint arXiv:2308.10187*, 2023. 3, 6, 7
- [34] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 6
- [35] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*, pages 722–737, 2018. 5
- [36] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022. 2
- [37] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. 2
- [38] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171. PMLR, 2021. 2
- [39] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 2
- [40] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 4
- [41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2
- [42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 2015. 7
- [43] Bleema Rosenfeld, Osvaldo Simeone, and Bipin Rajendran. Spiking generative adversarial networks with a neural network discriminator: Local training, bayesian models, and continual meta-learning. *IEEE Transactions on Computers*, 71(11):2778–2791, 2022. 3, 4
- [44] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29, 2016. 7
- [45] Alex Shonenkov, Misha Konstantinov, Daria Bakshandeva, Christoph Schuhmann, Ksenia Ivanova, and Nadiia Klokova, 2023. <https://github.com/deep-floyd/IF>. 4
- [46] Nicolas Skatchkovsky, Osvaldo Simeone, and Hyeryung Jang. Learning to time-decode in spiking neural networks through the information bottleneck. *arXiv preprint arXiv:2106.01177*, 2021. 2, 4
- [47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 8
- [48] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019. 2
- [49] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2, 4
- [50] Kenneth Stewart, Andreea Danielescu, Timothy Shea, and Emre Neftci. Encoding event-based data with a hybrid snn guided variational auto-encoder in neuromorphic hardware. In *NICE*, pages 88–97, 2022. 2, 4
- [51] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 30, 2017. 6, 7
- [52] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. 4
- [53] Ziqing Wang, Yuetong Fang, Jiahang Cao, Qiang Zhang, Zhongrui Wang, and Renjing Xu. Masked spiking transformer. In *ICCV*, pages 1761–1771, 2023. 1, 2, 5
- [54] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6
- [55] Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Yisen Wang, and Zhouchen Lin. Training feedback spiking neural networks by implicit differentiation on the equilibrium state. *NeurIPS*, 34:14516–14528, 2021. 2
- [56] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 7, 8
- [57] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *CVPR*, pages 8801–8810, 2022. 1
- [58] Yichi Zhang, Zhiru Zhang, and Lukasz Lew. Pokebnn: A binary pursuit of lightweight accuracy. In *CVPR*, pages 12475–12485, 2022. 5
- [59] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *arXiv preprint arXiv:2207.06635*, 2022. 4
- [60] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Han Zhang, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. Spiking-former: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*, 2023. 5
- [61] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*, 2022. 1, 5