AdderDiff: All-Addition Spiking Diffusion Models with Attention Enhancement

Yuhan Zhang[*], **Yuanpei Chen**[*], **Jie Zhou, Weihang Peng, Xiaode Liu, Yinglei Wang, Yufei Guo** [†]
Intelligent Science & Technology Academy of CASIC

Abstract

Diffusion models have achieved state-of-the-art (SoTA) performance in generative tasks through their iterative denoising mechanism, yet they remain computationally intensive and energy-prohibitive. Spiking Neural Networks (SNNs) represent a promising energy-efficient alternative to traditional Artificial Neural Networks (ANNs). Their binary spike encoding enables the conversion of multiplication operations into addition operations, which is a key attribute for reducing energy consumption. Integrating the strong generative capabilities of diffusion models with the energy efficiency of SNNs thus forms a highly promising research direction. In adherence to the design goal of all-additive computation, this paper proposes a novel spiking diffusion model with attention enhancement. Specifically, we adopt non-negative ternary spiking neurons (NNTSN) to construct the U-Net backbone, which helps mitigate information loss during feature processing. To align NNTSN with the requirement of using only additive operations, we further design two core components: a membrane potential attention mechanism and an all-additive shortcut module. Leveraging a reparameterization technique, the proposed diffusion model is developed to retain only additive operations during inference, ensuring strict compliance with energy-saving design principles. Extensive experiments on relevant benchmarks demonstrate that our method achieves SoTA performance in generative tasks. It also substantially outperforms other SNN-based generative models while using fewer time steps, which validates both its effectiveness and efficiency. Code is available at `https://github.com/hanbaobao950123/AdderDiff`.

1 Introduction

Spiking Neural Networks (SNNs), regarded as a next-generation neural network type, mimic the brain's information processing method to deliver a unique approach for reducing energy consumption. Their core feature is handling information through binary spikes. A neuron produces a spike coded as 1 when its membrane potential crosses a threshold and outputs 0 when the threshold is not met. This information processing method drives SNNs' strong energy efficiency. Unlike traditional Artificial Neural Networks (ANNs), which rely on intricate multiplication operations to model weight-activation interactions, Spiking Neural Networks (SNNs) embrace an **all-addition** computational paradigm—a defining core feature that drastically cuts down computational overhead. Additionally, SNNs work well with neuromorphic hardware via event-driven computing Wei et al. [2024], Ma et al. [2017], Akopyan et al. [2015], Davies et al. [2018], Pei et al. [2019]. In this computing mode, computational components activate only for incoming spikes and stay dormant otherwise, which further cuts energy waste. This energy-saving edge makes SNNs valuable for scenarios with limited resources. They have thus found use in object recognition Li et al. [2021a], Xiao et al. [2021], Guo et al. [2022a,b], object detection Kim et al. [2019], Qu et al. [2023] and pose estimation Zou et al. [2023].

[*]These authors contributed equally.
[†]Corresponding author. Email: yfguo@pku.edu.cn

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

In recent years, as diffusion models have achieved remarkable success in image generation, SNNs have been explored for energy-efficient diffusion frameworks. Cao et al. Cao et al. [2024] proposed a low-energy spiking U-Net with a pre-spike structure and training-free threshold guidance, and later extended SDDPM with a temporal-wise spiking mechanism to improve generation quality Cao et al. [2025]. Yang et al. Yang et al. [2024a] further replaced the U-Net with an SNN-based transformer to cut computation and sampling time while preserving quality. **However, their performance remains unsatisfactory; additionally, these designs inevitably incorporate non-additive computations in certain modules, a violation of SNN's core all-addition paradigm.**

Firstly, we identify that the underperformance of existing spiking diffusion models stems from the information loss inherent in the binary spike-based information transmission of SNNs, and this issue ultimately leads to reduced model accuracy. Notably, this information loss problem becomes more pronounced in the attention mechanisms of diffusion models, as evidenced by the fact that existing spiking diffusion models based on transformers actually exhibit worse performance than those based on U-Nets. The relatively better performance of these U-Net-based spiking diffusion models is primarily attributed to their consistent choice of omitting the use of attention mechanisms. However, this approach is clearly not an elegant solution, and it also runs counter to the current mainstream trend of utilizing attention mechanisms.

To address the information loss issue in spiking diffusion models—particularly within attention mechanisms—we propose a novel ternary-spike {0, 1, 2} neuron to implement an spiking U-Net with attention enhancement. Concurrently, we adapt the attention mechanism using ternary spikes to eliminate operations such as softmax and scaling. However, the use of ternary spikes {0, 1, 2} introduces multiplication operations into the computations of spiking diffusion models. To resolve this issue, we propose a reparameterization technique: during the inference phase, ternary spikes {0, 1, 2} can be converted to ternary spikes {-1, 0, 1} via this technique. This conversion ensures the model retains multiplication-free inference, which is a core advantage of SNNs.

Notably, we cannot directly adopt the ternary spike encoding {-1, 0, 1} employed in Guo et al. [2024]. This is because such a choice leads to unavoidable negative values in the spiking attention weights—arising from the multiplication of ternary spiking queries and keys. Introducing softmax to address this issue would ultimately entail non-additive operations in the spiking diffusion model, thus violating the fundamental all-addition paradigm of SNNs. We will elaborate on this limitation in detail in Section 3.3.

Secondly, to fully uphold the all-addition paradigm in our ternary-spike {0,1,2} neuron-based diffusion model, we further design the membrane potential attention mechanism and all-additive shortcut module. This integrated design not only significantly reduces information loss and enhances the overall performance of spiking diffusion models but also strictly avoids the introduction of non-additive operations. This marks a crucial step forward compared to existing spiking diffusion models, which still incorporate non-additive operations to varying degrees in their architectures. In summary, this paper makes the following four key contributions:

- We propose ternary spikes {0, 1, 2} to address the information loss issue in spiking diffusion models, particularly within attention mechanisms.
- We propose a reparameterization technique for convolutional layers and attention layers, ensuring spiking diffusion models retain only additive operations during inference.
- We conduct a refined design on all modules of the spiking U-Net, including membrane potential attention and all-addition shortcut, enabling the real implementation of spiking diffusion models with only additive operations.
- Our proposed method achieves state-of-the-art performance on relevant benchmarks for spiking diffusion-based image generation with fewer time steps.

2 Related Work

2.1 Spiking Neural Networks

As biologically inspired computational models, Spiking Neural Networks (SNNs) replicate biological brain dynamics more accurately than conventional ANNs. A primary distinction lies in their information transmission: SNNs utilize discrete, time-marked spikes, while ANNs rely on continuous

activation values. The theoretical foundation of SNNs stems from Hodgkin and Huxley's model, which simulates biological neurons' action potentials Hodgkin and Huxley [1952], followed by the Leaky Integrate-and-Fire (LIF) model—a streamlined tool for simulating neuron behaviors Maass [1997]. Over recent years, two prevalent approaches for training SNNs have emerged. The first is ANN-to-SNN conversion, which maps pre-trained ANN parameters to SNN components, with core alignment between ANN activation values and SNN average firing rates Sengupta et al. [2019], Hao et al. [2023], Han and Roy [2020], Li et al. [2021a]. However, bound by rate-coding schemes, it fails to leverage SNNs' unique temporal dynamics (limiting effectiveness on neuromorphic datasets) and typically requires numerous timesteps to match ANN accuracy—increasing energy consumption and conflicting with SNNs' low-power design goal. The second strategy is direct training from scratch, enabling SNNs to be trained via gradient-based optimization (similar to ANNs) and delivering strong performance on large-scale datasets with only a few timesteps Wu et al. [2018], Fang et al. [2021], Li et al. [2021b], Guo et al. [2022c,d], Ren et al. [2023]. Its efficacy in handling temporal data has made it a key focus for practical SNN deployment.

2.2 Diffusion Models

Diffusion models belong to a family of probabilistic generative models. Their core mechanism involves gradually destroying data through noise injection. After this destruction process, they learn to reverse the steps to generate new samples. Current research on diffusion models mainly relies on three dominant formulations. The first is denoising diffusion probabilistic models, also known as DDPMs Ho et al. [2020], Song et al. [2020a], Sohl-Dickstein et al. [2015], Nichol and Dhariwal [2021]. The second is score-based generative models, referred to as SGMs Song and Ermon [2019, 2020]. The third is stochastic differential equations, called Score SDEs Song et al. [2020b, 2021]. Follow-up studies on diffusion models center on enhancing three kinds of diffusion models. The enhancement efforts mainly cover three key directions. The first direction aims to achieve faster and more efficient sampling processes Jolicoeur-Martineau et al. [2021], Bao et al. [2022], Dockhorn et al. [2022], Lu et al. [2022]. The second focuses on improving the accuracy of likelihood Nichol and Dhariwal [2021], Song et al. [2021]. The third direction addresses the handling of data with special structures. Examples of such special structures include permutation invariance Niu et al. [2020], Xu et al. [2022], manifold structures Huang et al. [2022], Yang et al. [2024b]and discrete data Gu et al. [2022], Yang et al. [2025].

2.3 Spiking Neural Networks for Diffusion Models

Previous studies have explored adapting Spiking Neural Networks (SNNs) into diffusion models. Cao et al. Cao et al. [2024] proposed the spiking denoising diffusion probabilistic model (SDDPM), using a pure Spiking U-Net to match ANN performance with only 4 time steps. They further introduced spiking diffusion models (SDMs) Cao et al. [2025], integrating a temporal-wise spiking mechanism to capture richer temporal features (from a bio-plasticity perspective) and pioneering ANN-to-SNN conversion for SNN generation. However, both SDDPM and SDM directly adopt convolution for shortcut connections between different full-precision features, introducing multiplicative operations. Watanabe et al. Watanabe et al. [2023] presented the fully spiking denoising diffusion implicit model (FSDDIM), using synaptic current learning (SCL) to resolve the mismatch between diffusion models' real-valued estimation and SNNs' binary spikes—enabling the entire generative process to run solely on SNNs. Yang et al. Yang et al. [2024a] explored a novel direction by replacing diffusion models' U-Net with a transformer in SNNs, reducing computation and sampling time while improving image quality. Nevertheless, existing spiking diffusion models still face a significant performance gap compared to ANN-based diffusion models in image generation. In this work, we propose a novel fully additive spiking diffusion model integrated with an attention mechanism, aiming to enhance spiking diffusion models' accuracy and narrow the performance disparity with ANN-based counterparts via targeted improvements to spiking neurons and elaborate attention mechanism design.

3 Preliminary and Methodology

3.1 Revisiting DDPMs

Denoising Diffusion Probabilistic Models (DDPMs) Ho et al. [2020] are generative models that learn data distributions via iterative noise injection and denoising. Their operation relies on two Markov chains.

Forward Diffusion Chain. This chain corrupts original image x_0 with Gaussian noise over T steps. The step-wise transition probability is:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\gamma_t}\, x_{t-1}, \gamma_t\, \mathbb{I}\right) \tag{1}$$

where $\gamma_t \in (0,1)$ is a pre-defined noise schedule and $\mathbb{I}$ is the identity matrix. The direct $x_0 \to x_t$ transition is:

$$x_t = \sqrt{\bar{\zeta}_t}\, x_0 + \sqrt{1-\bar{\zeta}_t}\, \eta, \quad \eta \sim \mathcal{N}(0, \mathbb{I}) \tag{2}$$

with $\zeta_t = 1 - \gamma_t$ and $\bar{\zeta}_t = \prod_{s=1}^{t} \zeta_s$.

Reverse Denoising Chain. This chain learns to reconstruct data from noise, modeling the reverse transition as:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right) \tag{3}$$

The mean $\mu_\theta(x_t, t)$ is parameterized as:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\zeta_t}} \left(x_t - \frac{\gamma_t}{\sqrt{1-\bar{\zeta}_t}}\, \eta_\theta(x_t, t) \right) \tag{4}$$

where η_θ is a network predicting the noise component.

The DDPM training objective minimizes a simplified variational lower bound:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{t, x_0, \eta} \left\| \eta - \eta_\theta \left(\sqrt{\bar{\zeta}_t}\, x_0 + \sqrt{1-\bar{\zeta}_t}\, \eta, t \right) \right\|^2 \tag{5}$$

Practically, DDPMs use timestep-conditioned U-Net architectures.

3.2 Vanilla Spiking Diffusion Models with Attention

Spiking neurons are employed to construct vanilla spiking diffusion models with a U-Net architecture. The operational mechanism of spiking neurons involves two sequential information transformation stages. First, the input features are mapped into a continuous membrane potential signal, and second, this analog signal is discretized into binary spike events. These spike events, as the functional output of the neuron, are then transmitted to downstream network modules to initiate and regulate subsequent computational operations. In this paper, we adopt the Leaky Integrate-and-Fire (LIF) model proposed in [Maass, 1997], which is a widely accepted framework for capturing the essential electro physiological characteristics of biological neurons. The entire dynamic process, encompassing the temporal evolution of membrane potential and the generation of spike outputs, can be mathematically formalized through three interrelated equations, as presented below:

$$\mathbf{P}_\tau = \mathbf{H}_{\tau-1} + \mathbf{I}_\tau, \tag{6}$$

$$\mathbf{O}_\tau^B = \begin{cases} 1, & \mathbf{P}_\tau \geq \Theta, \\ 0, & \mathbf{P}_\tau < \Theta. \end{cases}, \tag{7}$$

$$\mathbf{H}_\tau = R \cdot \mathbf{O}_\tau + \gamma \cdot \mathbf{P}_\tau \cdot (1 - \mathbf{O}_\tau), \tag{8}$$

where $\mathbf{I}_\tau$ denotes the spatial input current at time step τ and $\tau \in \{0, 1, 2..., T_s\}$. $\mathbf{P}_\tau$ represents the membrane potential at time step τ integrating the spatial input $\mathbf{I}_\tau$ and the temporal state $\mathbf{H}_{\tau-1}$ from the previous step. $\mathbf{O}_\tau$ stands for the spike output at time step τ. Θ is the spike threshold. R is the reset potential. γ is a decay factor regulating the membrane potential's attenuation rate.

When $\mathbf{P}_\tau$ exceeds the threshold Θ the neuron emits a spike $\mathbf{O}_\tau = 1$ and the internal state $\mathbf{H}_\tau$ resets to R. If $\mathbf{P}_\tau$ does not reach Θ the membrane potential decays toward $\mathbf{H}_{\tau-1}$ at a rate determined by γ.

For notational brevity the LIF neuron layer is denoted as $\mathcal{L}(\cdot)$ which takes the membrane potential tensor $\mathbf{P}$ as input and outputs the spike tensor $\mathbf{O}$.

Diffusion models with a U-Net architecture contain a large number of residual blocks. To meet the design requirements of the ResBlock while preserving the neural activation patterns of the spiking diffusion model, a typical pre-spike unit is employed to process the input features, whose structure is "Activation-Conv-BatchNorm" Cao et al. [2024, 2025] as follows:

$$\mathcal{U}(\mathbf{z};T) = \mathcal{BN}\left(\mathcal{C}onv\left(\mathcal{L}(\mathbf{z};T)\right)\right), \tag{9}$$

where $\mathcal{L}(\cdot;T)$ denotes the LIF spiking activation function with T timesteps. The output membrane potential feature $\mathbf{P}^{l-1}$ at layer $l-1$ is input and processed through a sequence of spiking activation, convolution, and batch normalization:

$$\mathbf{P}^{l} = \mathcal{U}(\mathbf{P}^{l-1};T) + \mathbf{P}^{l-1}. \tag{10}$$

Then $\mathbf{P}^{l}$ undergoes another sequence of spiking activation, convolution, and batch normalization:

$$\mathbf{P}^{l+1} = \mathcal{U}(\mathbf{P}^{l} + \mathbf{t}_{\text{emb}};T) + \mathcal{S}(\mathbf{P}^{l-1}), \tag{11}$$

where $\mathbf{t}_{\text{emb}}$ is the timestep embedding of DDPMs after being projected and broadcasted to match the spatial dimensions, and $\mathcal{S}$ represents residual shortcut projection:

$$\mathcal{S}(z) = \begin{cases} \mathcal{C}onv(\mathbf{z}) & \text{if } C_{\text{in}} \neq C_{\text{out}} \\ \mathcal{I}(\mathbf{z}) & \text{if } C_{\text{in}} = C_{\text{out}} \end{cases}. \tag{12}$$

For $C_{\text{in}} \neq C_{\text{out}}$, a convolution shortcut is applied; otherwise, an identity mapping is used. And $\mathcal{I}(\cdot)$ is the identity operator.

Then if attention is enabled, the tensor is processed through an attention block; otherwise, it remains unchanged:

$$\mathbf{h}_{\text{Res}} = \begin{cases} \mathcal{A}(\mathbf{P}^{l+1}) & \text{if attention is enabled} \\ \mathbf{P}^{l+1} & \text{otherwise} \end{cases} \tag{13}$$

where $\mathcal{A}(\cdot)$ denotes the attention block. In SDDPM Cao et al. [2024] and TSDDPM Cao et al. [2025], attention is not enabled in the ResBlock, and $\mathbf{P}^{l+1}$ is directly output.

Next, this paper explores how to incorporate appropriate attention mechanisms $\mathbf{y}_{\mathcal{A}} = \mathcal{A}(\mathbf{P}^{l+1})$ into U-Net based spiking diffusion models. The Vanilla Spiking Self-Attention (VSSA) mechanism is proposed in Zhou et al. [2022, 2024],is formulated as follows:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathcal{L}\left(\mathcal{U}(\mathbf{P}^{l+1})\right), \tag{14}$$

$$\mathbf{M}_{\text{att}}^{VSSA} = \mathbf{Q}\mathbf{K}^{T}\mathbf{V} \cdot \sigma, \tag{15}$$

$$\mathbf{y}_{\mathcal{A}} = \mathcal{U}\left(\mathbf{M}_{att}\right) + \mathbf{P}^{l+1}. \tag{16}$$

The computation of the attention map is performed using binary spike-form queries and keys, thus replacing traditional matrix multiplication with addition-based operations. To further address the challenges of large values resulting from the matrix multiplication, a scaling factor σ is introduced to regulate the magnitude of the result from the matrix product. However, such a crude design has a drawback: our experiments show that this vanilla attention design not only fails to improve the performance of the spiking diffusion model but also leads to performance degradation. For instance, in our preliminary experiments on the CIFAR-10 dataset, when the timestep is set to 4, the Fréchet Inception Distance (FID) of images generated by the spiking U-Net without attention is 18.017. In contrast, after integrating the aforementioned attention mechanism, the FID of the generated images increases to 19.343. We attribute this phenomenon to the attention degradation caused by the information loss of SNNs, which impairs the effectiveness of the attention mechanism.

3.3 Information Loss in Vanilla Spiking Attention

Compared with attention maps based on ANNs, those in VSSA exhibit a degradation in the ability to represent self-attention, which can be directly quantified through the following relationship:

$$\mathcal{C}(\mathbf{M}_{\text{att}}^{\text{VSSA}}) = \frac{\log_2(C \times H \times W + 1)}{32} \cdot \mathcal{C}(\mathbf{M}_{\text{att}}^{\text{ANN}}). \tag{17}$$

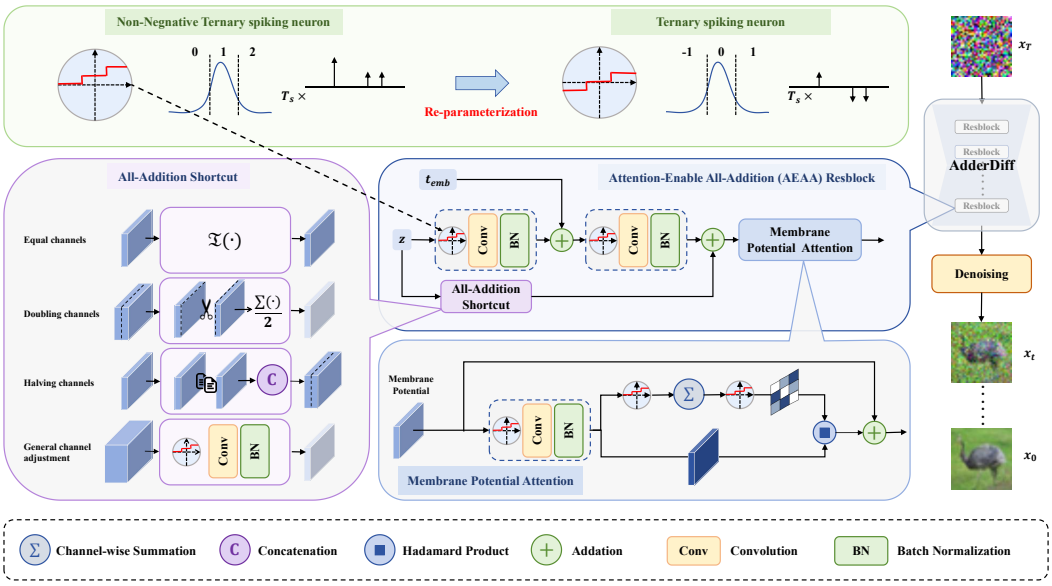

Figure 1: **The overview of the proposed spiking diffusion model AdderDiff.** Our improved spiking U-Net comprises a sequence of attention-enabled all-additive Resblocks, each integrated with membrane potential attention and an all-additive shortcut, both constructed using non-negative ternary spiking neurons (NNTSN).

This equation reveals that the self-attention representational capability of spiking attention maps is fundamentally scaled down relative to ANN-based ones. **In Section A of Appendix, we formalize this relationship through proof and analyze its implications.**

It is worth noting that this information loss in spiking diffusion models is not limited to the attention module, but also widely exists in other convolutional layers. To address this issue, in the subsequent chapters, we will achieve the maximum performance of the network under the computational constraints of SNNs through the improvement of spiking neurons and the elaborate adaptive design of each network module.

3.4 All-Addition Spiking Diffusion Models with Attention Enhancement

Figure 1 illustrates the overall architecture of our improved all-addition spiking diffusion model with attention enhancement.

3.4.1 Non-Negative Ternary Spiking Neuron Model

To address the information loss issue in spiking diffusion models, a kind of ternary spiking neuron proposed in Guo et al. [2024] can achieve more representational capability. Only if the membrane potential $\mathcal{M}_\tau$ is greater than Θ or less than $-\Theta$, the ternary spike neuron will be activated to fire the 1 or -1 spikes as the following formulations.

$$\mathbf{O}_\tau^T = \begin{cases} -1, & \mathbf{V}_\tau < \Theta_1, \\ 0, & \Theta_1 \leq \mathbf{V}_\tau \leq \Theta_2, \\ 1, & \mathbf{V}_\tau > \Theta_2. \end{cases} \tag{18}$$

However, this ternary spiking neuron has a drawback. When constructing the attention mechanism on this basis, the design of VSSA inevitably generates negative attention weights $\mathbf{QK}^T$. Since attention weights inherently denote the importance of different input features and thus must be non-negative, such negative values directly violate the fundamental design rationale of attention mechanisms.

To address this issue, we propose a **non-negative ternary spiking neuron model**, which can be formulated as follows:

$$\mathbf{O}_\tau^{T+} = \begin{cases} 0, & \mathbf{V}_\tau < \Theta_1, \\ 1, & \Theta_1 \leq \mathbf{V}_\tau \leq \Theta_2, \\ 2, & \mathbf{V}_\tau > \Theta_2. \end{cases} \tag{19}$$

$$\mathbf{H}_\tau = R \cdot \chi(\mathbf{P}_\tau \geq \Theta_1) + \gamma \cdot \mathbf{P}_\tau \cdot \chi(\mathbf{P}_\tau < \Theta_1), \tag{20}$$

where $\chi(\cdot)$ is the indicator function, where $\chi(\text{true}) = 1$ and $\chi(\text{false}) = 0$. This non-negative ternary spiking neuron can be regarded as a ternary spiking neuron with an overall 1-unit elevation in its activation amplitude:

$$\mathbf{O}_\tau^{T+} = \mathbf{O}_\tau^{T} + 1. \tag{21}$$

Re-parameterization strategie for convolutional layers. The computing process of convolutional layers in $\mathcal{U}(\cdot)$, where a non-negative ternary spiking neuron's output $\mathbf{O}_\tau^{T+}$ is convolved with a convolution kernel $\mathbf{w}$ connecting to another neuron for information transmission, can be re-parameterized as the addition of the ternary spiking neuron's output convolved with $\mathbf{w}$ and the convolution kernel $\mathbf{w}$ convolved with $\mathbf{1}$, as shown in the following equations:

$$\mathbf{O}_\tau^{T+} * \mathbf{w} = (\mathbf{O}_\tau^{T} + \mathbf{1}) * \mathbf{w} \tag{22}$$

$$= \mathbf{O}_\tau^{T} * \mathbf{w} + \mathbf{1} * \mathbf{w}. \tag{23}$$

Since the vanilla ternary spike neuron has been demonstrated that all the convolutional operations $\mathbf{O}_\tau^T * w$ can be transformed into addition operations Guo et al. [2024]. Thus, according to the equation (23), the non-negative ternary spiking neuron can inherit the addition-only benefits that are inherent to the vanilla ternary spiking neuron.

3.4.2 Membrane Potential Attention Mechanism

To activate the effectiveness of attention in improving the performance of spiking diffusion models, we propose a membrane potential attention mechanism and elaborate on the design of the attention module.

$$\mathbf{P} = \mathcal{U}(\mathbf{P}^{l+1}), \tag{24}$$

$$\mathbf{M}_{MPA} = \mathcal{L}^{T+}\left(\sum_{i=1}^{C} \mathcal{L}^{T+}(\mathbf{P}_i)\right) \odot \mathbf{P}, \tag{25}$$

$$\mathbf{y}_{\mathcal{MPA}} = \mathbf{M}_{MPA} + \mathbf{P}^{l+1}. \tag{26}$$

$\mathbf{P}_i$ denotes the i-th channel of $\mathbf{P}$. $\sum_{i=1}^{C}$ explicitly specifies a summation along the tensor's channel dimension. And $\odot$ denotes the Hadamard product, a core element-wise operation where each element in the output is the product of the elements at the same coordinate in the two input tensors. It is worth noting that the membrane potential attention mechanism also possesses the property of all-additive inference computation. Furthermore, $\mathbf{M}_{MPA}$ is a full-precision attention map, and its information loss is significantly smaller than that of VSSA.

Re-parameterization strategy for membrane potential attention $\mathbf{M}_{MPA}$. The feature of each channel in P undergoes a series of activation, channel-wise summation, and re-activation to generate the ternary importance of the membrane potential feature, which also represents the attention weight, taking the form of $\{0, 1, 2\}$. Let $\mathcal{L}^{T+}\left(\sum_{i=1}^{C} \mathcal{L}^{T+}(\mathbf{P}_i)\right) = \mathbf{O}^{T+}$, we derive

$$\mathbf{M}_{MPA} = \mathbf{O}^{T+} \odot \mathbf{P}, \tag{27}$$

$$= (\mathbf{O}_\tau^{T} + \mathbf{1}) \odot \mathbf{P}, \tag{28}$$

$$= \mathbf{O}_\tau^{T} \odot \mathbf{P} + \mathbf{P}. \tag{29}$$

Similarly, the operation $\mathbf{O}_\tau^T \odot \mathbf{P}$ is an all-additive computation; thus, all calculations within the global attention module are also all-additive.

3.4.3 All-Additive Shortcut Module

The shortcut modules in SDDPM Cao et al. [2024] and TSDDPM Cao et al. [2025] involve convolution computations on full-precision feature maps, which does not meet the expectation of all-additive operations. To address this issue, as shown in Figure 1, we designed an improved all-additive shortcut module that avoids the multiplication operations during the inference process. For an input tensor $\mathbf{x} \in \mathbb{R}^{T_s \times C_{\text{in}} \times H \times W}$, the module outputs $\mathbf{s} \in \mathbb{R}^{T_s \times C_{\text{out}} \times H \times W}$ using channel-adjustment strategies determined by the relationship between C_{in} and C_{out}, as follows:

Equal channel dimensions ($C_{\text{in}} = C_{\text{out}}$). When input and output channels match, an identity mapping preserves the input tensor:

$$\mathbf{s} = \mathcal{I}(\mathbf{x}), \tag{30}$$

where $\mathcal{I}(\cdot)$ denotes the identity operator.

Doubling channels ($C_{\text{out}} = 2C_{\text{in}}$). To double the channel count, the input tensor is concatenated with itself along the channel dimension:

$$\mathbf{s} = \mathbf{x} \oplus \mathbf{x}. \tag{31}$$

Halving channels ($C_{\text{in}} = 2C_{\text{out}}$). To halve the channel count, the input tensor is split into two equal channel subsets, which are then averaged element-wise:

$$\mathbf{s} = \frac{1}{2}\left(\mathbf{x}_{[:,:C_{\text{out}},:,:]} + \mathbf{x}_{[:,C_{\text{out}}:,:,:]}\right), \tag{32}$$

where $\mathbf{x}_{[:,:C_{\text{out}},:,:]}$ and $\mathbf{x}_{[:,C_{\text{out}}:,:,:]}$ represent the first and second half of the input channels, respectively.

General channel adjustment (other $C_{\text{in}} : C_{\text{out}}$ ratios). For arbitrary channel ratios, a sequence of spiking activation, convolution, and batch normalization is used:

$$\mathbf{s} = \mathcal{U}(\mathbf{x}; T), \tag{33}$$

where $\mathcal{U}$ is the pre-spike unit. The training, reparameterization, and inference of our AdderDiff are detailed in Algorithm 1 of appendix.

4 Experiments

To comprehensively validate the effectiveness and computational efficiency of the proposed approach, extensive experimental tests are performed on four widely used benchmark datasets. These datasets include 32×32 MNIST LeCun et al. [2010], 32×32 Fashion-MNIST Xiao et al. [2017], 32×32 CIFAR-10 Krizhevsky et al. [2009], and 64×64 CelebA Liu et al. [2015], covering diverse image types and resolution scales to ensure the generality of the evaluation results.

For the purpose of comparative performance analysis, eight cutting-edge spiking generative models are selected as baseline references. The selected baselines consist of FSVAE Kamata et al. [2022], SGAD Feng et al. [2023], Spiking-Diffusion Liu et al. [2023], FSDDIM Watanabe et al. [2023], SDIT Yang et al. [2024a], SDDPM Cao et al. [2024], SUM Zheng and Shen [2024], and TSDDPM Cao et al. [2025], which represent the latest advances in spiking-based generative modeling. Beyond spiking model comparisons, additional performance benchmarks are established against traditional ANN models, encompassing both conventional ANN architectures and those integrated with attention mechanisms to fully reflect the proposed method's competitiveness across different model paradigms. The quality of synthetic images generated by various models is evaluated quantitatively through two well-recognized metrics in generative modeling research. These metrics are Fréchet Inception Distance (FID Heusel et al. [2017]) and Inception Score (IS Salimans et al. [2016]), each capturing distinct aspects of generation performance. In detail, the IS assesses the quality of synthesized samples by maximizing the average entropy of the probability distribution produced by the Inception V3 network, where elevated scores correspond to superior sample quality with higher clarity and diversity. The FID, by contrast, measures the similarity between real and generated image distributions through computing the Kullback-Leibler (KL) divergence between their latent spaces—spaces that are hypothesized to conform to Gaussian distributions. In this case, smaller FID values indicate that the generated images are more consistent with real data in terms of distribution characteristics and exhibit higher realism.

Table 1: **Results for different dataset.** For a fair comparison, we re-evaluate DDPM Ho et al. [2020] using the same U-Net architecture as our AdderDiff. * indicates that only FID is reported because these datasets differ significantly from ImageNet, rendering the Inception Score less meaningful.

Dataset	Model	Method	#Param (M)	Time Steps $T_s\downarrow$	IS↑	FID↓
MNIST*	DDPM Ho et al. [2020]	ANN	32.70	/	9.420	4.45
	$\text{DDPM}_{\text{attention}}$ Ho et al. [2020]	ANN	34.21	/	2.011	0.98
	FSVAE Kamata et al. [2022]	SNN	3.87	16	6.209	97.06
	SGAD Feng et al. [2023]	SNN	-	16	-	69.64
	Spiking-Diffusion Liu et al. [2023]	SNN	-	16	-	37.50
	FSDDIM Watanabe et al. [2023]	SNN	-	4	-	7.48
	SDIT Yang et al. [2024a]	SNN	11.67	4	2.452	5.54
	SDDPM Cao et al. [2024]	SNN	63.61	4	-	29.48
	SUM Zheng and Shen [2024]	SNN	-	4	-	27.46
	AdderDiff (Ours)	SNN	33.49	**2**	–	**1.19**
Fashion MNIST*	DDPM Ho et al. [2020]	ANN	32.70	/	3.960	2.82
	$\text{DDPM}_{\text{attention}}$ Ho et al. [2020]	ANN	34.21	/	3.967	2.37
	FSVAE Kamata et al. [2022]	SNN	3.87	16	4.551	90.12
	SGAD Feng et al. [2023]	SNN	-	16	-	165.42
	Spiking-Diffusion Liu et al. [2023]	SNN	-	16	-	91.98
	FSDDIM Watanabe et al. [2023]	SNN	-	4	-	9.17
	SDIT Yang et al. [2024a]	SNN	11.67	4	4.549	5.49
	SDDPM Cao et al. [2024]	SNN	63.61	4	-	21.38
	SUM Zheng and Shen [2024]	SNN	-	4	-	21.30
	AdderDiff (Ours)	SNN	33.49	**2**	–	**6.40**
CelebA*	DDPM Ho et al. [2020]	ANN	34.21	/	2.358	7.96
	$\text{DDPM}_{\text{attention}}$ Ho et al. [2020]	ANN	32.70	/	2.469	6.80
	FSVAE Kamata et al. [2022]	SNN	6.37	16	3.697	101.60
	SGAD Feng et al. [2023]	SNN	-	16	-	151.36
	FSDDIM Watanabe et al. [2023]	SNN	-	4	-	36.08
	SDDPM Cao et al. [2024]	SNN	63.61	4	-	25.09
	AdderDiff (Ours)	SNN	33.49	**2**	–	**14.25**
CIFAR-10	DDPM Ho et al. [2020]	ANN	32.70	/	9.420	4.45
	$\text{DDPM}_{\text{attention}}$ Ho et al. [2020]	ANN	34.21	/	9.341	3.31
	FSVAE Kamata et al. [2022]	SNN	3.87	16	2.945	175.50
	SGAD Feng et al. [2023]	SNN	-	16	-	181.50
	Spiking-Diffusion Liu et al. [2023]	SNN	-	16	-	120.50
	FSDDIM Watanabe et al. [2023]	SNN	-	4	-	46.14
	SDIT Yang et al. [2024a]	SNN	11.67	4	4.080	22.17
	SDDPM Cao et al. [2024]	SNN	63.61	4	7.440	19.73
	SDDPM Cao et al. [2024]	SNN	63.61	8	7.584	17.27
	SUM Zheng and Shen [2024]	SNN	-	4	7.860	18.85
	TSDDPM Cao et al. [2025]	SNN	63.61	4	7.654	18.57
	TSDDPM Cao et al. [2025]	SNN	63.61	8	7.814	15.45
	AdderDiff (Ours)	SNN	33.49	**2**	**7.938**	**17.77**
	AdderDiff (Ours)	SNN	33.49	**4**	**8.242**	**15.24**

As the core backbone of the proposed framework, the U-Net structure originally described in Ho et al. [2020] is incorporated for feature extraction and representation learning. Consistent with the parameter setup reported in Ho et al. [2020], the diffusion process is parameterized with a total of $T = 1000$ time steps, a configuration maintained uniformly across all conducted experiments. For the forward diffusion dynamics, the sequence of variance values is defined as fixed constants that exhibit a linear ascending trend, starting from $\beta_1 = 10^{-4}$ and reaching $\beta_T = 0.02$ at the final time step. In terms of model-specific parameters, the decay coefficient γ involved in Eq. (20) is specified as 1.0. Meanwhile, the spiking thresholds $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ are set to constant values of 0.5 and 1.5 in turn. For the spiking neural network (SNN) component, two alternative values (2 and 4) are adopted for the time step to explore potential performance impacts. During model training, an optimization strategy is employed with a batch size of 128, and the training process is initialized with a learning rate of 10^{-5}.

4.1 Performance Comparison

Table 1 presents comparative results of generation quality (IS and FID) across ANN-based and SNN-based generative models on four datasets (MNIST, Fashion-MNIST, CelebA, CIFAR-10). Our

SNN-based model achieves remarkable performance among spiking baselines: it secures the best FID scores (1.19, 6.40, 14.25 on MNIST, Fashion-MNIST, CelebA with 2 time steps, and 15.24 on CIFAR-10 with 4 time steps), outperforming all SNN baselines by a large margin. Notably, our model even approaches or matches some ANN-based DDPM variants on MNIST and Fashion-MNIST while maintaining SNN's inherent energy efficiency. With fewer time steps (2 or 4) and moderate parameter counts (33.4 M), our model demonstrates superior efficiency-quality trade-off, validating its effectiveness in advancing spiking generative modeling across diverse datasets. Qualitative results are shown in Figure 2 in Appendix.

4.2 Ablation Study

We conduct ablation experiments on CIFAR10 with 2 time steps to verify the contributions of the non-negative ternary spiking neuron ({0,1,2}), membrane potential attention (MPA), and all-addition shortcut (AAS) to the generation performance of AdderDiff, as shown in Table 2. Baseline binary spiking neurons ({0,1}) yield an FID of 22.90 and IS of 7.254. Replacing binary spikes with {-1,0,1} brings marginal gain, while non-negative ternary spikes ({0,1,2}) slightly improve the metrics (FID: 22.55, IS: 7.377). We further compare different attention designs: VSSA degrades performance, whereas MPA alone substantially reduces FID to 18.43 with IS=7.931. Using only AAS also achieves moderate improvements (FID: 21.80, IS: 7.926). Notably, combining non-negative ternary spiking neurons, MPA and AAS together achieves the best results, with the lowest FID of 17.77 and the highest IS of 7.938. This confirms that the three components work synergistically and are essential for boosting the generative capability of AdderDiff.

Table 2: **Ablation study.** The performance of AdderDiff is enhanced by the integration of non-negative ternary spiking neuron, MPA and AAS.

Spike	Attention	AAS	FID↓	IS ↑
{0,1}			22.90	7.254
{0,1}	VSSA		24.02	7.063
{0,1}	MPA		21.77	7.919
{-1,0,1}			22.78	7.281
{0,1,2}			22.55	7.377
{0,1,2}	MPA		18.43	7.931
{0,1,2}		✓	21.80	7.926
{0,1,2}	MPA	✓	**17.77**	**7.938**

4.3 Evaluation of the Computational Cost

We calculate the computational energy cost following the method proposed in Rathi and Roy [2021]. Table 3 presents a comparison of generation quality (quantified by FID) and energy consumption across the proposed AdderDiff (2T/4T) on CIFAR10, two TSDDPM variants (4T/8T), and the ANN-based DDPM. In terms of energy efficiency, AdderDiff-2T achieves the lowest consumption (3.66 mJ), which is only 12.8% of DDPM-ANN's 28.61 mJ and significantly lower than TSDDPM-4T (10.97 mJ) and TSDDPM-8T (22.96 mJ). For generation quality, AdderDiff-4T outperforms both TSDDPM variants with the lowest FID (15.24) among spiking models—surpassing TSDDPM-8T (15.45) and TSDDPM-4T (18.57). Overall, AdderDiff achieves an optimal balance between ultra-low energy consumption and superior generation quality among spiking diffusion models, demonstrating its competitiveness for energy-constrained scenarios.

Table 3: Energy and FID Comparison of SNN and ANN-based Diffusion Models.

Models	DDPM-ANN	TSDDPM-4T	TSDDPM-8T	AdderDiff-2T	AdderDiff-4T
FID ↓	3.31	18.57	15.45	17.77	**15.24**
Energy (mJ) ↓	28.61	10.97	22.96	**3.66**	6.88

5 Conclusion

In this paper, we present an all-addition spiking diffusion model with attention enhancement to address the high computational cost of diffusion models while mitigating information loss in spiking neural networks. By introducing non-negative ternary spiking neurons and a reparameterization technique, our model retains only additive operations during inference, ensuring energy efficiency without sacrificing expressive power. The proposed membrane potential attention and all-addition shortcut module further enable the integration of attention mechanisms into spiking U-Nets in a strictly additive manner. Extensive experiments show that our method achieves state-of-the-art performance among spiking diffusion models, using fewer time steps and demonstrating superior efficiency. Our work provides a promising path toward energy-efficient, high-performance generative AI systems.

Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2025YFG0100400).

References

Wenjie Wei, Malu Zhang, Jilin Zhang, Ammar Belatreche, Jibin Wu, Zijing Xu, Xuerui Qiu, Hong Chen, Yang Yang, and Haizhou Li. Event-driven learning for spiking neural networks. *arXiv preprint arXiv:2403.00270*, 2024.

De Ma, Juncheng Shen, Zonghua Gu, Ming Zhang, Xiaolei Zhu, Xiaoqiang Xu, Qi Xu, Yangjing Shen, and Gang Pan. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Journal of Systems Architecture*, 77:43–51, 2017.

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 transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015.

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.

Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572 (7767):106–111, 2019.

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, 2021a.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Yisen Wang, and Zhouchen Lin. Training feedback spiking neural networks by implicit differentiation on the equilibrium state. *Advances in Neural Information Processing Systems*, 34:14516–14528, 2021.

Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 326–335, June 2022a.

Yufei Guo, Liwen Zhang, Yuanpei Chen, Xinyi Tong, Xiaode Liu, YingLei Wang, Xuhui Huang, and Zhe Ma. Real spike: Learning real-valued spikes for spiking neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 52–68. Springer, 2022b.

Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: Spiking neural network for energy-efficient object detection, 2019.

Jinye Qu, Zeyu Gao, Tielin Zhang, Yanfeng Lu, Huajin Tang, and Hong Qiao. Spiking neural network for ultra-low-latency and high-accurate object detection, 2023.

Shihao Zou, Yuxuan Mu, Xinxin Zuo, Sen Wang, and Li Cheng. Event-based human pose tracking by spiking spatiotemporal transformer, 2023.

Jiahang Cao, Ziqing Wang, Hanzhong Guo, Hao Cheng, Qiang Zhang, and Renjing Xu. Spiking denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4912–4921, January 2024.

Jiahang Cao, Hanzhong Guo, Ziqing Wang, Deming Zhou, Hao Cheng, Qiang Zhang, and Renjing Xu. Spiking diffusion models. *IEEE Transactions on Artificial Intelligence*, 6(1):132–143, 2025. doi: 10.1109/TAI.2024.3453229.

Shu Yang, Hanzhi Ma, Chengting Yu, Aili Wang, and Er-Ping Li. Sdit: Spiking diffusion model with transformer. *arXiv preprint arXiv:2402.11588*, 2024a.

Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 12244–12252, 2024.

Allan L Hodgkin and Andrew F Huxley. Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo. *The Journal of physiology*, 116(4):449, 1952.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

Zecheng Hao, Tong Bu, Jianhao Ding, Tiejun Huang, and Zhaofei Yu. Reducing ann-snn conversion error through residual membrane potential. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11–21, 2023.

Bing Han and Kaushik Roy. Deep spiking neural network: Energy efficiency through time based coding. In *European Conference on Computer Vision*, pages 388–404. Springer, 2020.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12:331, 2018.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in Neural Information Processing Systems*, 34:23426–23439, 2021b.

Yufei Guo, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Yinglei Wang, Xuhui Huang, and Zhe Ma. IM-loss: Information maximization loss for spiking neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022c. URL `https://openreview.net/forum?id=Jw34v_84m2b`.

Yufei Guo, Yuanpei Chen, Liwen Zhang, YingLei Wang, Xiaode Liu, Xinyi Tong, Yuanyuan Ou, Xuhui Huang, and Zhe Ma. Reducing information loss for spiking neural networks. In *European Conference on Computer Vision*, pages 36–52. Springer, 2022d.

Dayong Ren, Zhe Ma, Yuanpei Chen, Weihang Peng, Xiaode Liu, Yuhan Zhang, and Yufei Guo. Spiking pointnet: Spiking neural networks for point clouds. *Advances in Neural Information Processing Systems*, 36: 41797–41808, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

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*, 2020b.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

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.

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.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *NeurIPS*, 35: 30150–30166, 2022.

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.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International conference on artificial intelligence and statistics*, pages 4474–4484. PMLR, 2020.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

Ling Yang, Haotian Qian, Zhilong Zhang, Jingwei Liu, and Bin Cui. Structure-guided adversarial training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7256–7266, 2024b.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10696–10706, 2022.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

Ryo Watanabe, Yusuke Mukuta, and Tatsuya Harada. Fully spiking denoising diffusion implicit models. *arXiv preprint arXiv:2312.01742*, 2023.

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.

Zhaokun Zhou, Kaiwei Che, Wei Fang, Keyu Tian, Yuesheng Zhu, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer v2: Join the high accuracy club on imagenet with an snn ticket. *arXiv preprint arXiv:2401.02020*, 2024.

Hiromichi Kamata, Yusuke Mukuta, and Tatsuya Harada. Fully spiking variational autoencoder. In *AAAI*, volume 36, pages 7059–7067, 2022.

Linghao Feng, Dongcheng Zhao, and Yi Zeng. Sgad: Spiking generative adversarial network with attention scoring decoding. *arXiv preprint arXiv:2305.10246*, 2023.

Mingxuan Liu, Rui Wen, and Hong Chen. Spiking-diffusion: Vector quantized discrete diffusion model with spiking neural networks. *arXiv preprint arXiv:2308.10187*, 2023.

Peixiang Zheng and Zhao Shen. Biologically inspired spike-unet for enhanced and efficient image generati. In *2024 21st International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 1–5, 2024. doi: 10.1109/ICCWAMTIP64812.2024.10873764.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. att labs, 2010.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015.

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.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29, 2016.

Nitin Rathi and Kaushik Roy. Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6): 3174–3182, 2021.

A Proof of Information Loss in VSSA

Then, to quantify the information loss, the representational capability $\mathcal{C}(\mathbf{X})$ of a set $\mathbf{X}$ is defined as the maximum information entropy of $\mathbf{X}$, which equals $\log_2 K_{\mathbf{X}}$ when all $K_{\mathbf{X}}$ distinct elements of $\mathbf{X}$ are equally probable:

$$\mathcal{C}(\mathbf{X}) = \log_2 K_{\mathbf{X}}, \tag{34}$$

where $K_{\mathbf{X}}$ denotes the total number of distinct elements in $\mathbf{X}$.

ANN-based attention map $\mathbf{M}_{\text{att}}^{\text{ANN}} \in \mathbb{R}^{C\times H\times W}$ consists of $C \times H \times W$ real-valued units, each encoded with 32 bits. A 32-bit real value supports 2^{32} distinct states, so the total number of distinct activation patterns in $\mathbf{M}_{\text{att}}^{\text{ANN}}$ is:

$$K_{\mathbf{M}_{\text{att}}^{\text{ANN}}} = \left(2^{32}\right)^{C\times H\times W} = 2^{32\cdot C\times H\times W}. \tag{35}$$

Substituting into the capability definition equation (34):

$$\mathcal{C}(\mathbf{M}_{\text{att}}^{\text{ANN}}) = \log_2\left(2^{32\times C\times H\times W}\right) = 32\cdot C\times H\times W. \tag{36}$$

Then, to quantify the representational capability of spiking attention map $\mathbf{M}_{\text{att}}^{\text{VSSA}} \in \mathbb{Z}^{C\times H\times W}$ in equation (15), the following theorem A.1 must be introduced.

Theorem A.1. *Given binary query, key, and value matrices $Q, K, V \in \mathbb{B}^{C\times H\times W}$ with entries in $\{0,1\}$ and a scalar σ, the number of distinct values attainable by any element Z_{ij} in the output matrix $Z = (QK^TV)\cdot\sigma$ is exactly $C \times H \times W + 1$.*

Proof. The computation can be decomposed into two stages: the attention score calculation and the weighted value aggregation.

Stage 1: Attention Scores

Let $A = QK^T$. An arbitrary element A_{im} represents the inner product between the i-th query vector q_i and the m-th key vector k_m:

$$A_{im} = \sum_{k=1}^{C} Q_{ik}K_{mk} \tag{37}$$

Since $Q_{ik}, K_{mk} \in \{0,1\}$, the term $Q_{ik}K_{mk}$ is a binary variable representing the logical AND operation. Consequently, A_{im} is a sum of d binary variables. The range of A_{im} is the integer set:

$$\text{Range}(A_{im}) = \{0, 1, \ldots, d\} \tag{38}$$

Stage 2: Value Aggregation

Let $Y = AV$. The element Y_{ij} is the inner product of the i-th row of A and the j-th column of V:

$$Y_{ij} = \sum_{m=1}^{W\times H} A_{im}V_{mj} \tag{39}$$

Substituting the expansion of A_{im}:

$$Y_{ij} = \sum_{m=1}^{W\times H}\left(\sum_{k=1}^{C} Q_{ik}K_{mk}\right)V_{mj} \tag{40}$$

Rearranging the summation terms:

$$Y_{ij} = \sum_{m=1}^{W\times H}\sum_{k=1}^{C}\underbrace{(Q_{ik}K_{mk}V_{mj})}_{\delta_{mk}} \tag{41}$$

Let $\delta_{mk} = Q_{ik}K_{mk}V_{mj}$. Since the inputs form a closed set under multiplication within $\{0,1\}$, $\delta_{mk} \in \{0,1\}$. The expression for Y_{ij} becomes a summation of $W \times H \times C$ binary indicator variables.

Cardinality Analysis We now determine the bounds and continuity of Y_{ij}.

- **Lower Bound:** If $Q, K, V = \mathbf{0}$, then $Y_{ij} = 0$.
- **Upper Bound:** If $Q, K, V = \mathbf{1}$, then $Y_{ij} = \sum_{m=1}^{M} \sum_{k=1}^{d} 1 = C \times H \times W$.
- **Reachability:** By constructing K and V such that exactly n triplets (i, m, k) satisfy the condition $Q_{ik} = 1, K_{mk} = 1, V_{mj} = 1$, any integer sum is achievable.

Thus, the set of possible values for the unscaled output Y_{ij} is $\mathcal{S}_Y = \{0, 1, \ldots, C \times H \times W\}$. The size of this set is $|\mathcal{S}_Y| = C \times H \times W + 1$.

Finally, applying the scalar s, the set of output values becomes:

$$\mathcal{S}_Z = \{n \cdot s \mid n \in \mathbb{Z}, 0 \leq n \leq C \times H \times W\} \tag{42}$$

Since scaling by a non-zero constant is a bijection, $|\mathcal{S}_Z| = |\mathcal{S}_Y|$. □

According to Theorem A.1, spiking attention map $\mathbf{M}_{\text{att}}^{\text{VSSA}} \in \mathbb{Z}^{C \times H \times W}$ (where $\mathbb{T} = \{0, 1, ..., C \times H \times W\}$) consists of $C \times H \times W$ discrete units. Each unit has $C \times H \times W + 1$ distinct states, so the total number of distinct patterns is:

$$K_{\mathbf{M}_{\text{att}}^{\text{VSSA}}} = (C \times H \times W + 1)^{C \times H \times W} \tag{43}$$

Substituting into the capability definition equation (34):

$$\mathcal{C}(\mathbf{M}_{\text{att}}^{\text{VSSA}}) = \log_2 \left((C \times H \times W + 1)^{C \times H \times W} \right). \tag{44}$$

By taking the ratio of Equation (44) to Equation (36), the following relationship can be derived:

$$\frac{\mathcal{C}(\mathbf{M}_{\text{att}}^{\text{VSSA}})}{\mathcal{C}(\mathbf{M}_{\text{att}}^{\text{ANN}})} = \frac{\log_2(C \times H \times W + 1)}{32}. \tag{45}$$

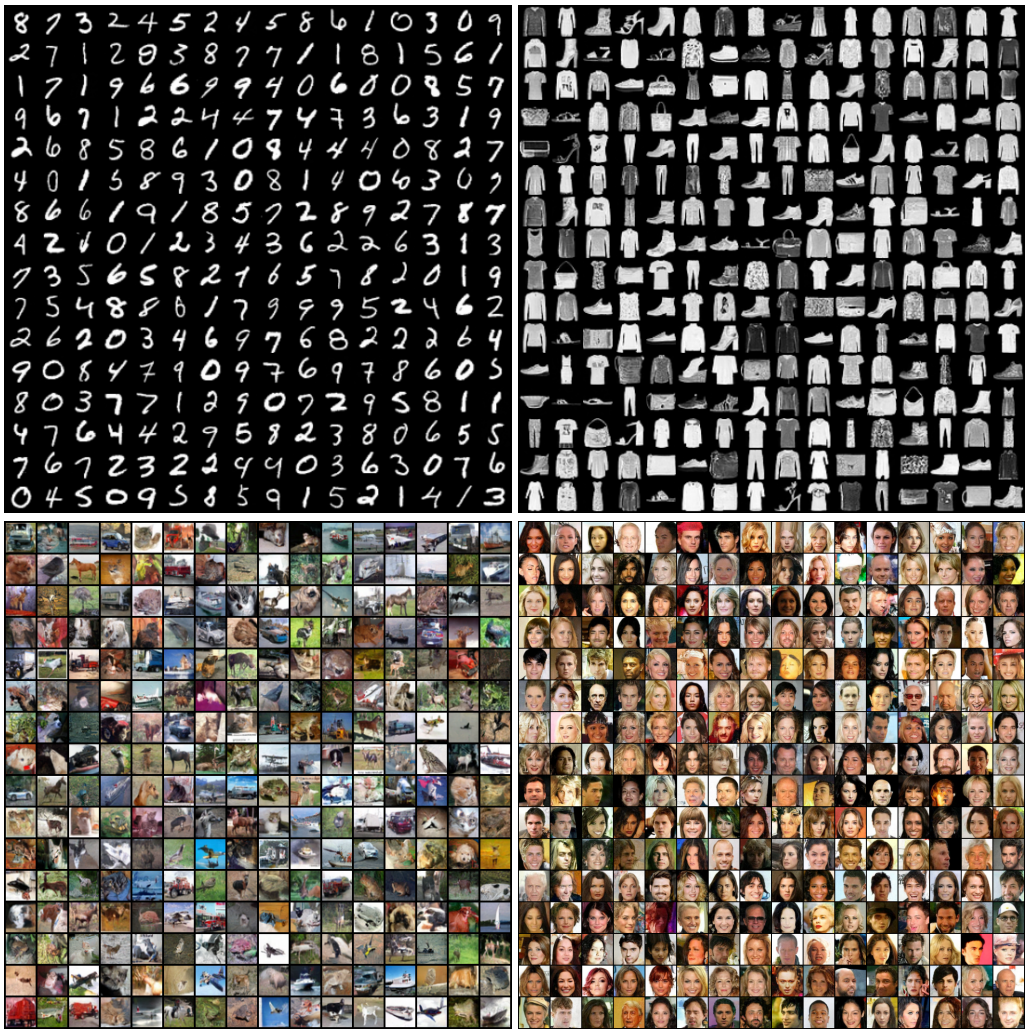

Figure 2: **Unconditional image generation results on MNIST, Fashion-MNIST, CIFAR-10, and CelebA by using AdderDiff.**

Algorithm 1 Training, Reparameterization & Inference of AdderDiff

Stage 1: Training
Input: AdderDiff, training data, iterations I_{train}
Output: Trained weights

for $i = 1$ to I_{train} **do**
  Sample noisy input & time embedding
  Compute membrane potential, generate $O_\tau^{T+} \in \{0, 1, 2\}$ via NNTSN-STE
  Forward pass with MPA and AAS (multiplication allowed)
  Calculate diffusion loss $\mathcal{L}_{\text{diff}}$
  Backpropagate surrogate gradients
  Update weights with Adam
end for

Stage 2: Offline Re-parameterization
Input: Trained AdderDiff, total layers l
Output: Multiplication-free inference model

for $k = 1$ to l **do**
  Fuse BN params into conv kernels
  Spike transform: $O_\tau^T = O_\tau^{T+} - 1$
  Decompose $O_\tau^{T+} \odot P = O_\tau^T \odot P + P$
end for

Stage 3: Fully Additive Inference
Input: Re-parameterized AdderDiff, sampling steps I_{test}
Output: Generated images

for $i = 1$ to I_{test} **do**
  Initialize Gaussian noise
  Iterative denoising with pure-add MPA/AAS
  Convert spikes via fixed offset rule
  Output denoised samples
end for

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: [Yes]

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

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

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

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

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: [Yes]

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

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

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

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

    Guidelines:

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