Spik-NeRF v2: Pushing the Limit of Spiking Neural Radiance Fields with ±I-LIF

Chen Cheng[1,*], **Qinlong Lan**[2,*], **Gang Wan**[2], **Hao Guo**[1,†], **Lei Liu**[2], **Zhanji Wei**[2], **Yitian Wu**[2], **Yufei Guo**[3,†]
[1]Taiyuan University of Technology
[2]Space Engineering University
[3]Intelligent Science & Technology Academy of CASIC

Abstract

Spiking Neural Networks (SNNs) offer a promising energy-efficient alternative to Artificial Neural Networks (ANNs) through event-driven, multiplication-free computation. However, when applied to downstream tasks such as Neural Radiance Fields (NeRF), SNNs suffer from significant information loss, resulting in a noticeable performance gap. In this paper, we present *Spik-NeRF v2* to advance SNN-based neural rendering. We propose the ±I-LIF spiking neuron, which extends I-LIF to support signed integer values during training, thereby mitigating information loss. During inference, it is converted to ternary spikes $\{-1, 0, 1\}$, preserving event-driven properties with addition-only operations. Furthermore, we introduce a re-parameterization technique that transforms a trained I-LIF-based Spiking NeRF with t timesteps into an equivalent ±I-LIF-based model with $t/2$ timesteps. This enables faster inference while preserving rendering quality, overcoming the suboptimal performance of directly trained ±I-LIF models with reduced timesteps. Extensive experiments on synthetic and realistic datasets demonstrate that *Spik-NeRF v2* surpasses existing SNN-based NeRF methods and achieves rendering quality comparable to ANN-based approaches.

1 Introduction

Spiking Neural Networks (SNNs) Rathi et al. (2020); Carnevale & Hines (2006); Fang et al. (2021); Ghosh-Dastidar & Adeli (2009); Rathi & Roy (2020); Guo et al. (2025, 2024c) have gained considerable attention for their potential in energy-efficient computing, thanks to their event-driven nature and the absence of multiplication operations. These attributes make SNNs particularly suitable for power-constrained applications, such as mobile devices, robotics, and edge computing. Recent advances have demonstrated that SNNs can achieve comparable accuracy to Artificial Neural Networks (ANNs) on upstream tasks like classification. However, due to the inherent information loss in SNNs, a noticeable performance gap persists when applying them to downstream tasks, including object detection Qu et al. (2023), image de-occlusion Guo et al. (2024b), sequential learning Ponghiran & Roy (2022), human activity recognition Li et al. (2022), pose tracking Zou et al. (2023), 3D recognition Ren et al. (2023), and autonomous driving Viale et al. (2022).

Neural Radiance Fields (NeRF) Mildenhall et al. (2021) have revolutionized novel view synthesis by representing 3D scenes as continuous neural fields. Nevertheless, the high computational cost of ANNs employed in NeRF limits their deployment in resource-constrained environments. Therefore, integrating SNNs into NeRF holds substantial promise for enabling energy-efficient neural rendering. Prior works, such as Spiking NeRF Liao et al. (2024), Spiking Nerfacto Gu et al. (2024), Spiking-

*Equal Contribution. E-mail: chengchen@tyut.edu.cn, whulanql@whu.edu.cn.
†Corresponding Authors. E-mail: guohao@tyut.edu.cn, yfguo@pku.edu.cn.

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

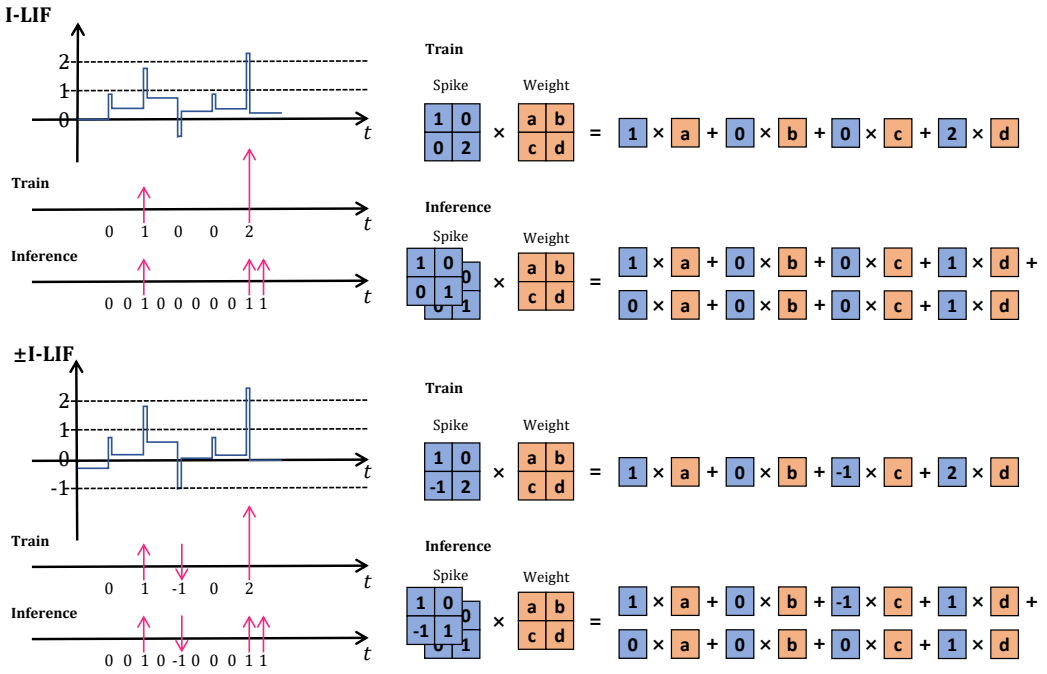


Figure 1: Comparison of the I-LIF and ±I-LIF neurons during training and inference. (Training Phase) The I-LIF neuron outputs non-negative integer spikes within the range $\{0, 1, \ldots, D\}$, while our ±I-LIF neuron extends the output to signed integer spikes $\{-D, -D+1, \ldots, D\}$, capturing both positive and negative activation information to mitigate quantization error. (Inference Phase) To maintain energy efficiency: I-LIF decomposes each integer spike into D binary spike sequences $\{0, 1\}$, with MAC operations simplified to sparse additions; ±I-LIF reorganizes the spike representation into D ternary spike sequences $\{-1, 0, 1\}$. Both neurons preserve SNNs' event-driven advantages, but ±I-LIF achieves a better representational capacity.

Nerf Li et al. (2025), SpikingNerf Yao et al. (2023), and Spik-NeRF Wan et al. (2025), have attempted to bridge SNNs and NeRF. Among these, Spik-NeRF Wan et al. (2025) presents a directly trained fully spiking NeRF that utilizes ternary spiking neurons to reduce information loss and improve performance. Despite this progress, there remains room for further improvement in both rendering quality and computational efficiency.

The I-LIF (Integer Leaky Integrate-and-Fire) neuron Luo et al. (2024) was proposed to reduce quantization error by emitting integer values during training, which are subsequently converted to binary spikes $\{0, 1\}$ during inference. Empirical results demonstrate that the I-LIF neuron achieves excellent performance on classification tasks. Meanwhile, Ternary-LIF Guo et al. (2024a), which employs ternary spikes $\{-1, 0, 1\}$, enhances representational capacity without introducing additional computational overhead compared to binary spikes. Inspired by these two works, we unify them under the common perspective of mitigating information loss and propose the ±I-LIF spiking neuron to further enhance the expressive power of Spik-NeRF. Through carefully designed spike conversion, the ±I-LIF spiking neuron retains the event-driven properties of SNNs and relies solely on addition operations during inference after being converted to ternary spikes $\{-1, 0, 1\}$. Furthermore, we observe that directly training a ±I-LIF-based model with t timesteps yields inferior performance compared to training an I-LIF-based spiking NeRF with $2t$ timesteps. To address this, we propose a novel reparameterization method that converts a pre-trained I-LIF-based spiking NeRF into a ±I-LIF-based counterpart, halving the number of timesteps to improve inference efficiency without sacrificing rendering quality.

In summary, we present *Spik-NeRF v2*, which advances the state of the art in spiking neural rendering through three key contributions:

- We introduce the $\pm$I-LIF spiking neuron for *Spik-NeRF v2*, an enhanced variant of the I-LIF neuron that supports both positive and negative integer values during the training phase. This extension further boosts model performance while preserving the inherent event-driven characteristics of spiking neural networks. Notably, the $\pm$I-LIF neuron relies exclusively on addition operations during inference, as it is converted to a $\{-1, 0, 1\}$ spike neuron through spike conversion.
- We find that training an I-LIF-based Spik-NeRF with t timesteps outperforms training a $\pm$I-LIF-based Spik-NeRF with $t/2$ timesteps. Leveraging this insight, we develop a novel re-parameterization method that transforms a trained I-LIF-based Spik-NeRF with t timesteps into a $\pm$I-LIF-based Spik-NeRF with $t/2$ timesteps, enabling faster inference without sacrificing performance.
- Through these innovations, *Spik-NeRF v2* achieves rendering quality that matches or exceeds that of ANN-based NeRF models, while maintaining the energy efficiency benefits of SNNs. Extensive experiments on synthetic and realistic datasets validate the effectiveness of our approach.

2 Related Work

2.1 Spiking Neural Networks

SNNs are typically trained using three mainstream approaches: spike-timing-dependent plasticity (STDP) Bi & Poo (1998), ANN-to-SNN conversion Hao et al. (2023b,a); Liu et al. (2022); Deng & Gu (2021); Ding et al. (2021); Bu et al. (2022); Han et al. (2020); Li et al. (2021a), and direct training methods Cheng et al. (2020); Neftci et al. (2019); Wu et al. (2019); Rathi & Roy (2020); Li et al. (2021b); Wu et al. (2021b,a). Direct training methods, which leverage surrogate gradients to approximate the non-differentiable spike function, have gained widespread traction owing to their capability of achieving high performance with a small number of timesteps Guo et al. (2022); Li et al. (2021b); Guo et al. (2023). The I-LIF neuron Luo et al. (2024) is a recent direct training-based variant that emits integer values during training to reduce quantization error, which are then converted to binary spikes during inference for computational efficiency. Meanwhile, Ternary-LIF Guo et al. (2024a) adopts ternary spikes $\{-1, 0, 1\}$ to mitigate quantization error. In this paper, we fuse the core ideas of I-LIF and Ternary-LIF to propose the $\pm$I-LIF neuron tailored for spiking NeRF models.

2.2 Spiking Neural Networks for NeRF

Several works have explored SNN-based NeRF models. Spiking NeRF Liao et al. (2024) and Spiking Nerfacto Gu et al. (2024) use hybrid ANN-SNN architectures. Spiking-Nerf Li et al. (2025) employs ANN-to-SNN conversion, requiring many timesteps. SpikingNerf Yao et al. (2023) builds on DVGO Sun et al. (2022) and TensoRF Chen et al. (2022) frameworks, also with high timestep requirements. In contrast, Spik-NeRF Wan et al. (2025) is a directly trained SNN-based NeRF model based on the original NeRF framework, using ternary spike neurons to improve performance. Our work extends Spik-NeRF by integrating $\pm$I-LIF neurons and a re-parameterization technique to further boost performance and efficiency.

3 Preliminary and Methodology

In this section, we first review the NeRF framework and the Spik-NeRF model. Then, we introduce the I-LIF spiking neuron. Finally, we present our proposed $\pm$I-LIF neuron and re-parameterization method.

3.1 Neural Radiance Fields (NeRF)

NeRF Mildenhall et al. (2021) represents a 3D scene as a continuous 5D neural radiance field, mapping spatial coordinates $\mathbf{p} = (x, y, z) \in \mathbb{R}^3$ and viewing directions $\mathbf{v} = (\theta, \phi) \in \mathbb{S}^2$ to volume density $\sigma \in \mathbb{R}^+$ and view-dependent RGB color $\mathbf{c} = (r, g, b) \in [0, 1]^3$. This is implemented using

two multi-layer perceptrons (MLPs):

$$\text{Geometry network:} \quad F_\theta : \mathbf{p} \mapsto (\mathbf{e}, \sigma), \tag{1}$$

$$\text{Appearance network:} \quad G_\gamma : (\mathbf{e}, \mathbf{v}) \mapsto \mathbf{c}, \tag{2}$$

where θ and γ are network parameters, and $\mathbf{e} \in \mathbb{R}^N$ is an intermediate feature embedding. Each MLP layer typically uses ReLU activations:

$$\mathbf{h} = \text{ReLU}(\text{Linear}(\mathbf{a})), \tag{3}$$

where $\mathbf{a}$ is the input activation.

Rendering is performed via volume rendering. For a camera ray $\mathbf{r}(g) = \mathbf{o} + g\mathbf{d}$ with near and far bounds $[g_n, g_f]$, K points are sampled along the ray. The pixel color is computed as:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{K} G_i \alpha_i \mathbf{c}_i, \tag{4}$$

where $G_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$ is the transmittance, $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ is the alpha value, and $\delta_i = g_{i+1} - g_i$ is the distance between samples. The model is optimized using a photometric loss between rendered and ground-truth pixel colors.

3.2 Spik-NeRF

Spik-NeRF Wan et al. (2025) is a spiking neural network adaptation of NeRF, where the MLP layers are replaced with spiking neurons. The Leaky Integrate-and-Fire (LIF) neuron model is used, with soft reset dynamics defined in discrete time as:

$$\mathbf{v}^t = \mathbf{u}^{t-1} + \frac{1}{\tau}(\mathbf{W}\mathbf{s}^{t-1} - \mathbf{u}^{t-1}), \tag{5}$$

$$\mathbf{s}^t = \Theta(\mathbf{v}^t - V_{\text{th}}), \tag{6}$$

$$\mathbf{u}^t = \mathbf{v}^t \odot (1 - \mathbf{s}^t), \tag{7}$$

where $\mathbf{u}^t$ is the membrane potential, $\mathbf{s}^t$ is the spike output, $\Theta(\cdot)$ is the Heaviside step function, τ is the time constant, V_{th} is the firing threshold, and u_{reset} is the reset potential.

To address information loss in binary spike activations, Spik-NeRF introduced ternary spike neurons that output values in $\{0, 1, 2\}$ during training, and use a re-parameterization technique to shift to $\{-1, 0, 1\}$ during inference, preserving multiplication-free operations.

3.3 I-LIF Spiking Neuron

The I-LIF (Integer Leaky Integrate-and-Fire) neuron Luo et al. (2024) is designed to reduce quantization error by emitting integer values during training. As shown in Fig. 1, the spike generation is modified to:

$$\mathbf{s}^t = \text{Clip}\left(\text{round}(\mathbf{v}^t), 0, D\right), \tag{8}$$

where D is the maximum integer output. We denote the I-LIF configuration with a maximum integer output value of D as I-LIF^D.

Training. Eq. 8 is non-differentiable. A rectangular window was adopt as the surrogate gradient, retaining gradients only for neurons with outputs in $[0, D]$.

Inference. To preserve energy efficiency, integer spikes are converted to binary spikes by extending T timesteps to $T \times D$ virtual steps as shown in Fig. 1. The integer spike $\mathbf{s}^t$ is decomposed into a binary sequence $\{\mathbf{s}^{t,d}\}_{d=1}^{D}$ satisfying $\sum_{d=1}^{D} \mathbf{s}^{t,d} = \mathbf{s}^t$. The input to the next layer becomes:

$$\mathbf{x}^t = \mathbf{W} \sum_{d=1}^{D} \mathbf{s}^{t,d} = \sum_{d=1}^{D} \left(\mathbf{W}\mathbf{s}^{t,d}\right). \tag{9}$$

Since $\mathbf{s}^{t,d} \in 0, 1$, all MACs are reduced to sparse ACs, ensuring spike-driven inference.

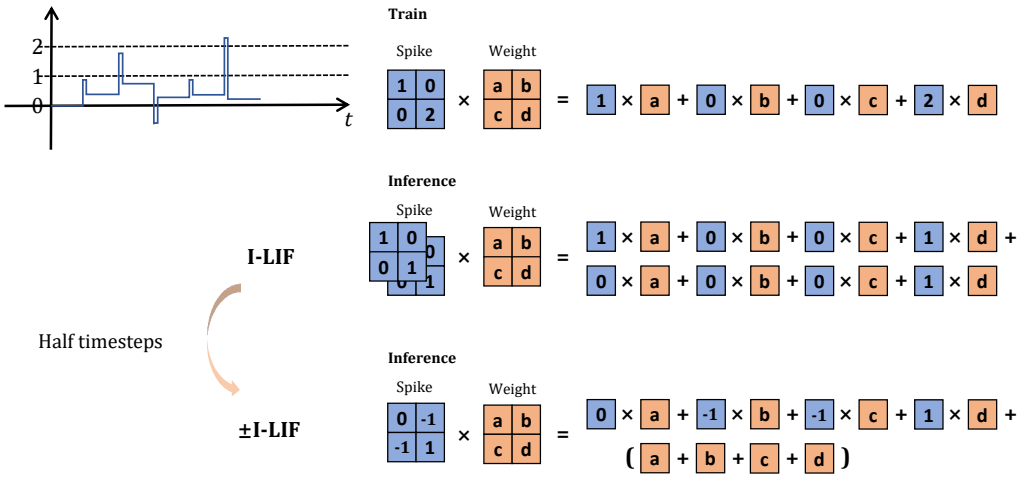


Figure 2: Illustration of the re-parameterization process for converting I-LIF^{2D} to $\pm$I-LIF^{D} during inference. The pre-trained I-LIF^{2D} model operates with $2D$ virtual timesteps of binary spikes $\{0, 1\}$ for inference, which incurs relatively high latency. Our proposed re-parameterization pipeline realizes an equivalent transformation to the $\pm$I-LIF^{D} model through three keys: (1) reorganizing pairs of binary spikes from I-LIF^{2D} into ternary spikes $\{-1, 0, 1\}$ for $\pm$I-LIF^{D}; (2) directly inheriting the weight matrix of I-LIF^{2D} without complex restructuring; (3) compensating the bias vector to offset the signed spike offset. Critically, this transformation halves the inference timesteps from $2D$ (I-LIF^{2D}) to D ($\pm$I-LIF^{D}) while preserving the equivalence of the next layer's input, achieving faster inference speed (lower latency) without sacrificing model performance.

3.4 Spik-NeRF v2 with $\pm$I-LIF

3.4.1 $\pm$I-LIF Neuron

The $\pm$I-LIF (Signed Integer Leaky Integrate-and-Fire) neuron is designed to further reduce quantization error by emitting signed integer values during training, inspired by the I-LIF neuron and Ternary-LIF neuron.

As shown in Fig. 1, the spike generation is modified to:

$$\mathbf{s}^t = \text{Clip}\left(\text{round}(\mathbf{v}^t), -D, D\right), \tag{10}$$

where D is the maximum absolute integer output, and the output range is $\{-D, -D+1, \ldots, 0, \ldots, D-1, D\}$. This extension of output range enhances the network's representational capacity compared to the I-LIF neuron (limited to non-negative integers).

Training. Eq. 10 exhibits non-differentiable behavior, which hinders standard backpropagation. To address this issue, a rectangular window function is employed as the surrogate gradient too here, restricting gradient propagation exclusively to neurons whose output values fall within the interval $[-D, D]$.

Inference. For the sake of maintaining energy-efficient operation, signed integer spikes are transformed into ternary spikes through the expansion of T original timesteps into $T \times D$ virtual timesteps, as illustrated in Fig. 2. Each signed integer spike $\mathbf{s}^t$ is decomposed into a ternary spike sequence $\{\mathbf{s}^{t,d}\}_{d=1}^{D}$ that satisfies the constraint $\sum_{d=1}^{D} \mathbf{s}^{t,d} = \mathbf{s}^t$, with $\mathbf{s}^{t,d} \in \{-1, 0, 1\}$ serving to encode the polarity information of the original integer spike. Accordingly, the input fed into the subsequent layer is formulated as:

$$\mathbf{x}^t = \mathbf{W} \sum_{d=1}^{D} \mathbf{s}^{t,d} = \sum_{d=1}^{D} \left(\mathbf{W}\mathbf{s}^{t,d}\right). \tag{11}$$

Given that $\mathbf{s}^{t,d}$ takes values only in $\{-1, 0, 1\}$, all conventional multiply-accumulate operations (MACs) are simplified to sparse addition and subtraction operations, thus ensuring both spike-driven processing and superior energy efficiency during inference. Here, $\pm$I-LIF^{D} refers to the $\pm$I-LIF variant parameterized by a maximum absolute integer output D.

3.4.2 $\pm$I-LIF^D Neuron vs. I-LIF^{2D} Neuron

Notably, the $\pm$I-LIF^D and I-LIF^{2D} neurons share an identical number of quantized integer levels (i.e., $2D+1$ levels: $\{-D, \ldots, D\}$ for $\pm$I-LIF^D and $\{0, \ldots, 2D\}$ for I-LIF^{2D}). To investigate their performance discrepancy under comparable quantization granularity, we conduct comparative experiments on complex synthetic scenes (Fern, Lego, and Drums), with additional baseline comparisons to vanilla LIF neurons.

Experimental Setup. For fair comparison, all $\pm$I-LIF and I-LIF variants are trained with a single timestep ($T = 1$), following the implementation protocol in prior work Luo et al. (2024). During inference: - $\pm$I-LIF^D decomposes into D virtual timesteps of ternary spikes $\{-1, 0, 1\}$ (total inference timesteps = D); - I-LIF^K decomposes into K virtual timesteps of binary spikes $\{0, 1\}$ (total inference timesteps = K).

We specifically compare $\pm$I-LIF^2 (inference timesteps = 2) against I-LIF^4 (inference timesteps = 4) to evaluate $\pm$I-LIF^D vs. I-LIF^{2D}, and $\pm$I-LIF^2 against I-LIF^2 (inference timesteps = 2) to validate the superiority of our proposed neuron over the original I-LIF. Vanilla LIF with 2 and 4 timesteps are included as baselines.

Table 1: Per-scene quantitative comparison between $\pm$I-LIF^D and I-LIF^{2D} neurons. Higher PSNR/SSIM and lower LPIPS indicate better performance.

Metric	Method	Fern	Lego	Drums
PSNR↑	LIF (2 timesteps)	25.08	29.33	24.51
	LIF (4 timesteps)	25.59	30.35	25.07
	I-LIF^2	25.43	30.37	24.96
	$\pm$I-LIF^2	25.62	30.83	25.31
	I-LIF^4	25.82	31.11	25.35
SSIM↑	LIF (2 timesteps)	0.769	0.940	0.907
	LIF (4 timesteps)	0.795	0.951	0.918
	I-LIF^2	0.788	0.951	0.916
	$\pm$I-LIF^2	0.800	0.958	0.924
	I-LIF^4	0.809	0.960	0.925
LPIPS↓	LIF (2 timesteps)	0.207	0.039	0.086
	LIF (4 timesteps)	0.173	0.029	0.071
	I-LIF^2	0.185	0.029	0.071
	$\pm$I-LIF^2	0.160	0.023	0.061
	I-LIF^4	0.150	0.022	0.061

Empirical results in Table 1 yield three key insights:

First, I-LIF^4 (i.e., I-LIF^{2D} with $D = 2$) achieves marginally better performance than $\pm$I-LIF^2 across all metrics and scenes. This indicates that when granted twice the inference timesteps (4 vs. 2), I-LIF^{2D} can outperform $\pm$I-LIF^D—a trade-off between timestep count and representational efficiency.

Second, $\pm$I-LIF^2 consistently outperforms I-LIF^2 under the same training/inference timestep constraints (1 training timestep, 2 inference timesteps). For example, on the Fern scene, $\pm$I-LIF^2 improves PSNR by 0.19 dB, SSIM by 0.012, and LPIPS by 0.025 compared to I-LIF^2. This validates that our proposed $\pm$I-LIF neuron is a superior alternative to the original I-LIF neuron when timesteps are limited.

Third, both $\pm$I-LIF^2 and I-LIF^2—which translate to 2 inference timesteps of ternary LIF and binary LIF, respectively—outperform the vanilla LIF with 2 timesteps and even 4 timesteps. This further confirms that under identical inference timestep limits, $\pm$I-LIF stands as the optimal choice among the compared neuron architectures, balancing performance and energy efficiency.

Revisiting the first key observation, we note that while I-LIF^{2D} achieves marginally better performance than $\pm$I-LIF^D, it incurs twice the inference latency. A natural question thus arises: can we train the model with I-LIF^{2D} and then equivalently transform it into $\pm$I-LIF^D during inference, thereby halving the inference latency? To address this question, we innovatively propose a dedicated transformation method that converts a pre-trained I-LIF^{2D} model into its $\pm$I-LIF^D counterpart, the details of which are elaborated in the following section.

3.4.3 Re-parameter I-LIF^{2D} Neuron to $\pm$I-LIF^D Neuron

To resolve the latency-performance trade-off between I-LIF^{2D} and $\pm$I-LIF^D, we propose a mathematically rigorous re-parameterization method, as shown in Fig. 2. The core insight is to preserve the equivalence of the next layer's input (i.e., weight-spike matrix multiplication result) by aligning the signed spike offset with corresponding weight/bias adjustments, enabling a direct transformation from pre-trained I-LIF^{2D} to $\pm$I-LIF^D with halved inference timesteps.

Equivalence Criterion The transformation must guarantee that the input to the subsequent layer remains identical before and after conversion. Formally, let:

- For I-LIF^{2D}: - Output spike vector $\mathbf{s}_{\text{I-LIF}} \in \{0, 1, \ldots, 2D\}^N$ (N = number of neurons), decomposed into $2D$ binary spike matrices $\{\mathbf{S}^k_{\text{I-LIF}}\}_{k=1}^{2D}$ ($\mathbf{S}^k_{\text{I-LIF}} \in \{0,1\}^N$) satisfying $\sum_{k=1}^{2D} \mathbf{S}^k_{\text{I-LIF}} = \mathbf{s}_{\text{I-LIF}}$. - Next layer input: $\mathbf{X} = \mathbf{W}_{\text{I-LIF}} \cdot \sum_{k=1}^{2D} \mathbf{S}^k_{\text{I-LIF}} + \mathbf{b}_{\text{I-LIF}}$ (where $\mathbf{W}_{\text{I-LIF}} \in \mathbb{R}^{M\times N}$ is weight matrix, $\mathbf{b}_{\text{I-LIF}} \in \mathbb{R}^{M\times 1}$ is bias vector, M = number of next-layer neurons).

- For $\pm$I-LIF^{D}: - Output spike vector $\mathbf{s}_{\pm\text{I-LIF}} \in \{-D, \ldots, 0, \ldots, D\}^N$, with the key numerical mapping:

$$\mathbf{s}_{\pm\text{I-LIF}} = \mathbf{s}_{\text{I-LIF}} - D \cdot \mathbf{1}_{N\times 1} \tag{12}$$

(D is scalar, $\mathbf{1}_{N\times 1}$ is all-ones vector for dimension matching). - Decomposed into D ternary spike matrices $\{\mathbf{S}^d_{\pm\text{I-LIF}}\}_{d=1}^{D}$ ($\mathbf{S}^d_{\pm\text{I-LIF}} \in \{-1,0,1\}^N$) satisfying $\sum_{d=1}^{D} \mathbf{S}^d_{\pm\text{I-LIF}} = \mathbf{s}_{\pm\text{I-LIF}}$. - Next layer input: $\mathbf{X} = \mathbf{W}_{\pm\text{I-LIF}} \cdot \sum_{d=1}^{D} \mathbf{S}^d_{\pm\text{I-LIF}} + \mathbf{b}_{\pm\text{I-LIF}}$.

The equivalence requires $\mathbf{W}_{\pm\text{I-LIF}} \cdot \sum_{d=1}^{D} \mathbf{S}^d_{\pm\text{I-LIF}} + \mathbf{b}_{\pm\text{I-LIF}} = \mathbf{W}_{\text{I-LIF}} \cdot \sum_{k=1}^{2D} \mathbf{S}^k_{\text{I-LIF}} + \mathbf{b}_{\text{I-LIF}}$ for all possible spike outputs.

Transformation Pipeline The re-parameterization consists of three mutually dependent steps, strictly derived from Eq. 12 and matrix multiplication equivalence:

1. **Binary-to-Ternary Spike Reorganization**
   Reorganize the $2D$ binary spikes of I-LIF^{2D} into D ternary spikes of $\pm$I-LIF^{D}, ensuring the offset constraint in Eq. 12 is satisfied. For each virtual timestep $d \in \{1, 2, \ldots, D\}$, pair the $(2d-1)$-th and $2d$-th binary spike matrices of I-LIF^{2D} (denoted $\mathbf{S}^{2d-1}_{\text{I-LIF}}$ and $\mathbf{S}^{2d}_{\text{I-LIF}}$) to form a ternary spike matrix:

$$\mathbf{S}^d_{\pm\text{I-LIF}} = \mathbf{S}^{2d-1}_{\text{I-LIF}} + \mathbf{S}^{2d}_{\text{I-LIF}} - \mathbf{1}_N \tag{13}$$

   This formula guarantees the required spike sum equivalence:

$$\begin{aligned}
\sum_{d=1}^{D} \mathbf{S}^d_{\pm\text{I-LIF}} &= \sum_{d=1}^{D} \left(\mathbf{S}^{2d-1}_{\text{I-LIF}} + \mathbf{S}^{2d}_{\text{I-LIF}} - \mathbf{1}_N\right) \\
&= \sum_{k=1}^{2D} \mathbf{S}^k_{\text{I-LIF}} - D \cdot \mathbf{1}_N \\
&= \mathbf{s}_{\text{I-LIF}} - D \cdot \mathbf{1}_N \\
&= \mathbf{s}_{\pm\text{I-LIF}}
\end{aligned}$$

   Notably, $\mathbf{S}^d_{\pm\text{I-LIF}} \in \{-1, 0, 1\}$ holds because $\mathbf{S}^{2d-1}_{\text{I-LIF}}$ and $\mathbf{S}^{2d}_{\text{I-LIF}}$ are binary (their sum is $\{0, 1, 2\}$; subtracting 1 gives $\{-1, 0, 1\}$), satisfying the ternary spike constraint of $\pm$I-LIF^{D}.
2. **Weight Re-parameterization**
   To preserve the linear transformation of weight-spike multiplication, the weight matrix of $\pm$I-LIF^{D} must be identical to that of I-LIF^{2D}. This is derived by substituting Eq. 13 into the next layer input:

$$\mathbf{W}_{\pm\text{I-LIF}} \cdot \sum_{d=1}^{D} \mathbf{S}^d_{\pm\text{I-LIF}} = \mathbf{W}_{\pm\text{I-LIF}} \cdot \left(\sum_{k=1}^{2D} \mathbf{S}^k_{\text{I-LIF}} - D \cdot \mathbf{1}_N\right) =$$
$$\mathbf{W}_{\pm\text{I-LIF}} \cdot \sum_{k=1}^{2D} \mathbf{S}^k_{\text{I-LIF}} - D \cdot \mathbf{W}_{\pm\text{I-LIF}} \cdot \mathbf{1}_N$$

   For this to match $\mathbf{W}_{\text{I-LIF}} \cdot \sum_{k=1}^{2D} \mathbf{S}^k_{\text{I-LIF}}$, the coefficient of the spike sum must be identical:

$$\mathbf{W}_{\pm\text{I-LIF}} = \mathbf{W}_{\text{I-LIF}} \tag{14}$$

   This eliminates the need for complex weight rearrangement and ensures direct weight inheritance from the pre-trained I-LIF^{2D} model.
3. **Bias Compensation**
   The offset term $-D \cdot \mathbf{W}_{\pm\text{I-LIF}} \cdot \mathbf{1}_{N\times B}$ introduced by the signed spike mapping must be canceled by

adjusting the bias vector. Substituting Eqs. 13 and 14 into the equivalence criterion:

$$\mathbf{W}_{\text{I-LIF}} \cdot \sum_{k=1}^{2D} \mathbf{S}_{\text{I-LIF}}^{k} - D \cdot \mathbf{W}_{\text{I-LIF}} \cdot \mathbf{1}_N + \mathbf{b}_{\pm\text{I-LIF}} = \mathbf{W}_{\text{I-LIF}} \cdot \sum_{k=1}^{2D} \mathbf{S}_{\text{I-LIF}}^{k} + \mathbf{b}_{\text{I-LIF}}$$

Canceling the common spike sum term and rearranging gives the bias compensation formula:

$$\mathbf{b}_{\pm\text{I-LIF}} = \mathbf{b}_{\text{I-LIF}} + D \cdot \mathbf{W}_{\text{I-LIF}} \cdot \mathbf{1}_N \tag{15}$$

Here, $\mathbf{W}_{\text{I-LIF}} \cdot \mathbf{1}_{N\times 1}$ computes the sum of each row in $\mathbf{W}_{\text{I-LIF}}$ (sum of weights connecting each next-layer neuron to all current-layer neurons), and scaling by D offsets the signed spike's linear transformation.

Validity Verification Substituting Eqs. 13, 14, and 15 into the $\pm$I-LIF^D input formula confirms full equivalence:

$$\begin{aligned}
\mathbf{X}_{\pm\text{I-LIF}} &= \mathbf{W}_{\text{I-LIF}} \cdot \left(\sum_{k=1}^{2D} \mathbf{S}_{\text{I-LIF}}^{k} - D \cdot \mathbf{1}_N \right) + \mathbf{b}_{\text{I-LIF}} + D \cdot \mathbf{W}_{\text{I-LIF}} \cdot \mathbf{1}_N \\
&= \mathbf{W}_{\text{I-LIF}} \cdot \sum_{k=1}^{2D} \mathbf{S}_{\text{I-LIF}}^{k} - D \cdot \mathbf{W}_{\text{I-LIF}} \cdot \mathbf{1}_N + \mathbf{b}_{\text{I-LIF}} + D \cdot \mathbf{W}_{\text{I-LIF}} \cdot \mathbf{1}_N \\
&= \mathbf{W}_{\text{I-LIF}} \cdot \sum_{k=1}^{2D} \mathbf{S}_{\text{I-LIF}}^{k} + \mathbf{b}_{\text{I-LIF}} = \mathbf{X}_{\text{I-LIF}}
\end{aligned}$$

4 Experiments

We evaluate *Spik-NeRF v2* on both synthetic and realistic datasets adopted from the original NeRF framework Mildenhall et al. (2021). Specifically, the synthetic dataset consists of 8 representative scenes (e.g., Chair, Drums, Lego), where each scene is equipped with 100 training views and 200 test views at a resolution of 400×400. The realistic dataset also contains 8 scenes (e.g., Room, Fern, T-Rex), with each scene providing 20–60 images resized to 400×400 for consistency. For training configuration, we strictly follow the protocol of Spik-NeRF Wan et al. (2025): the Adam optimizer is employed with a total of 300k training iterations, a batch size of 1024 rays, and an initial learning rate of 5×10^{-4} with exponential decay over training. In terms of volumetric sampling, we sample 64 points for the coarse network and 128 points for the fine network, consistent with standard NeRF-based implementations.

For comparative analysis, we select a comprehensive set of baseline models, including: (1) the ANN-based NeRF Mildenhall et al. (2021) as the performance upper bound; (2) Spiking-NeRF Li et al. (2025), an ANN-SNN hybrid model constructed upon the same NeRF framework as our method; (3) Spik-NeRF Wan et al. (2025), a state-of-the-art directly trained SNN-based NeRF model; and (4) a binary spike-based NeRF, denoted as Spiking NeRF for short.

Consistent with the experimental setup of Spik-NeRF Wan et al. (2025), the timestep number is set to $T = 2$ for these methods. To maintain the same total inference timesteps across comparable models, we configure the *Spik-NeRF v2* with a training timestep and a maximum absolute integer output of $D = 2$. Furthermore, to investigate how the performance of *Spik-NeRF v2* approaches that of the ANN-based NeRF with increased timesteps, we additionally report the performance of *Spik-NeRF v2* with $D = 4$, equivalent to $T = 4$.

For quantitative performance evaluation, we adopt three standard metrics widely used in NeRF-related works Mildenhall et al. (2021); Wan et al. (2025): peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM), where higher values indicate better rendering quality, and learned perceptual image patch similarity (LPIPS) Zhang et al. (2018), where lower values correspond to smaller perceptual differences from ground-truth images. Notably, Spiking-NeRF Li et al. (2025) only provides PSNR results in its original publication. Thus, we present the complete SSIM and LPIPS metrics for ANN-based NeRF, Spiking NeRF, Spik-NeRF, and *Spik-NeRF v2* for thorough comparison.

Table 2: Per-scene quantitative results on the synthetic dataset.

Metric	Method	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
PSNR↑	ANN-based NeRF Mildenhall et al. (2021)	34.15	25.64	29.15	36.85	31.48	29.34	33.12	29.42	31.15
	Spiking-NeRF Li et al. (2025)	12.24	11.14	13.98	14.85	9.81	10.36	9.81	13.22	11.93
	Spiking NeRF	31.98	24.51	24.00	34.55	29.33	27.75	31.65	27.85	28.95
	Spik-NeRF Wan et al. (2025)	33.40	25.21	26.05	35.99	30.82	28.86	32.78	28.75	30.23
	Spik-NeRF v2	**33.71**	**25.35**	**27.60**	**36.40**	**31.11**	**29.07**	**33.04**	**29.12**	**30.68**
	Spik-NeRF v2*	**34.06**	**25.57**	**27.72**	**36.69**	**31.42**	**29.41**	**33.35**	**29.33**	**30.95**
SSIM↑	ANN-based NeRF Mildenhall et al. (2021)	0.979	0.929	0.966	0.979	0.965	0.958	0.978	0.874	0.953
	Spiking NeRF	0.963	0.907	0.891	0.969	0.940	0.937	0.969	0.838	0.927
	Spik-NeRF Wan et al. (2025)	0.973	0.921	0.929	0.978	0.957	0.950	0.975	0.856	0.942
	Spik-NeRF v2	**0.975**	**0.925**	**0.949**	**0.976**	**0.960**	**0.953**	**0.977**	**0.866**	**0.948**
	Spik-NeRF v2*	**0.977**	**0.928**	**0.950**	**0.979**	**0.964**	**0.957**	**0.979**	**0.872**	**0.951**
LPIPS↓	ANN-based NeRF Mildenhall et al. (2021)	0.014	0.053	0.022	0.015	0.020	0.024	0.023	0.086	0.032
	Spiking NeRF	0.034	0.086	0.119	0.032	0.039	0.047	0.043	0.126	0.066
	Spik-NeRF Wan et al. (2025)	0.021	0.065	0.063	0.021	0.027	0.033	0.028	0.105	0.045
	Spik-NeRF v2	**0.017**	**0.061**	**0.037**	**0.018**	**0.022**	**0.031**	**0.026**	**0.095**	**0.038**
	Spik-NeRF v2*	**0.016**	**0.056**	**0.036**	**0.016**	**0.020**	**0.026**	**0.023**	**0.088**	**0.035**

Table 3: Per-scene quantitative results on the realistic dataset.

Metric	Method	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns	Avg.
PSNR↑	ANN-based NeRF Mildenhall et al. (2021)	31.38	26.25	21.98	31.35	21.20	27.51	27.27	28.10	26.88
	Spiking-NeRF Li et al. (2025)	17.07	15.91	9.68	14.51	9.12	10.35	15.05	13.05	13.09
	Spiking NeRF	30.12	25.08	20.75	30.07	20.51	26.19	25.26	26.34	25.54
	Spik-NeRF Wan et al. (2025)	30.90	25.70	21.46	30.79	20.94	26.94	26.15	27.16	26.26
	Spik-NeRF v2	**31.07**	**25.82**	**21.57**	**30.91**	**21.05**	**27.10**	**26.43**	**27.37**	**26.42**
	Spik-NeRF v2*	**31.11**	**25.99**	**21.75**	**31.02**	**21.12**	**27.29**	**26.76**	**27.60**	**26.58**
SSIM↑	ANN-based NeRF Mildenhall et al. (2021)	0.931	0.836	0.790	0.896	0.734	0.853	0.896	0.877	0.851
	Spiking NeRF	0.904	0.769	0.691	0.837	0.644	0.782	0.825	0.800	0.781
	Spik-NeRF Wan et al. (2025)	0.920	0.802	0.744	0.869	0.688	0.820	0.862	0.835	0.817
	Spik-NeRF v2	**0.923**	**0.809**	**0.753**	**0.874**	**0.698**	**0.828**	**0.870**	**0.846**	**0.825**
	Spik-NeRF v2*	**0.925**	**0.819**	**0.770**	**0.883**	**0.714**	**0.839**	**0.881**	**0.857**	**0.836**
LPIPS↓	ANN-based NeRF Mildenhall et al. (2021)	0.049	0.101	0.119	0.059	0.122	0.075	0.062	0.078	0.083
	Spiking NeRF	0.098	0.207	0.203	0.130	0.223	0.134	0.135	0.166	0.162
	Spik-NeRF Wan et al. (2025)	0.066	0.164	0.157	0.092	0.170	0.101	0.095	0.123	0.121
	Spik-NeRF v2	**0.062**	**0.150**	**0.150**	**0.084**	**0.157**	**0.093**	**0.086**	**0.111**	**0.111**
	Spik-NeRF v2*	**0.058**	**0.133**	**0.135**	**0.073**	**0.141**	**0.083**	**0.076**	**0.098**	**0.099**

Quantitative results on the synthetic dataset are presented in Table 2. *Spik-NeRF v2* achieves an average PSNR of 30.68, outperforming Spik-NeRF (30.23) and closely matching ANN-based NeRF (31.15). Doubling timesteps (*Spik-NeRF v2**) further narrows the gap to 30.95.

Table 3 shows results on the realistic dataset. *Spik-NeRF v2* achieves PSNR/SSIM/LPIPS of 26.42/0.825/0.111, with *Spik-NeRF v2** reaching 26.58/0.836/0.099—approaching ANN-based NeRF (26.88/0.851/0.083). Notable per-scene improvements are observed on Fern, Flower, and T-Rex. Due to page constraints, visual rendering results are relegated to the appendix.

5 Conclusion

We have presented *Spik-NeRF v2*, which advances SNN-based neural rendering through the ±I-LIF neuron and a novel re-parameterization technique. *Spik-NeRF v2* achieves rendering quality that matches or surpasses ANN-based NeRF models, with only 2-4 timesteps and energy-efficient operation. This work pushes the limit of spiking neural radiance fields and opens new avenues for energy-efficient 3D reconstruction and novel view synthesis.

Limitations. Since the energy benefit of *Spik-NeRF v2* comes from replacing multiply-accumulate (MAC) operations with sparse accumulations (AC) in spiking neurons, it is most pronounced when neural-network computation dominates the pipeline, as in MLP-based NeRF Mildenhall et al. (2021). For methods where neural-network inference accounts for only a small fraction of the total workload, such as hash-grid-based Instant-NGP Müller et al. (2022), the energy advantage is less significant.

Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2025YFG0100400) and the National Natural Science Foundation of China (No. 62606352).

References

Bi, G.-q. and Poo, M.-m. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24): 10464–10472, 1998.

Bu, T., Ding, J., Yu, Z., and Huang, T. Optimized potential initialization for low-latency spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11–20, 2022.

Carnevale, N. T. and Hines, M. L. *The NEURON book*. Cambridge University Press, 2006.

Chen, A., Xu, Z., Geiger, A., Yu, J., and Su, H. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pp. 333–350. Springer, 2022.

Cheng, X., Hao, Y., Xu, J., and Xu, B. Lisnn: Improving spiking neural networks with lateral interactions for robust object recognition. In *IJCAI*, pp. 1519–1525, 2020.

Deng, S. and Gu, S. Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv preprint arXiv:2103.00476*, 2021.

Ding, J., Yu, Z., Tian, Y., and Huang, T. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. *arXiv preprint arXiv:2105.11654*, 2021.

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2661–2671, 2021.

Ghosh-Dastidar, S. and Adeli, H. Spiking neural networks. *International journal of neural systems*, 19(04):295–308, 2009.

Gu, Y., Wang, Z., Ye, D., and Xu, R. Sharpening your density fields: Spiking neuron aided fast geometry learning. *arXiv preprint arXiv:2412.09881*, 2024.

Guo, Y., Zhang, L., Chen, Y., Tong, X., Liu, X., Wang, Y., Huang, X., and Ma, Z. 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*, pp. 52–68. Springer, 2022.

Guo, Y., Zhang, Y., Chen, Y., Peng, W., Liu, X., Zhang, L., Huang, X., and Ma, Z. Membrane potential batch normalization for spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19420–19430, October 2023.

Guo, Y., Chen, Y., Liu, X., Peng, W., Zhang, Y., Huang, X., and Ma, Z. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12244–12252, 2024a.

Guo, Y., Peng, W., Chen, Y., Zhou, J., and Ma, Z. Improved event-based image de-occlusion. *IEEE Signal Processing Letters*, 2024b.

Guo, Y., Peng, W., Liu, X., Chen, Y., Zhang, Y., Tong, X., Jie, Z., and Ma, Z. Enof-snn: Training accurate spiking neural networks via enhancing the output feature. *Advances in Neural Information Processing Systems*, 37:51708–51726, 2024c.

Guo, Y., Zhang, Y., Jie, Z., Liu, X., Tong, X., Chen, Y., Peng, W., and Ma, Z. Reverb-snn: Reversing bit of the weight and activation for spiking neural networks. *arXiv preprint arXiv:2506.07720*, 2025.

Han, B., Srinivasan, G., and Roy, K. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13558–13567, 2020.

Hao, Z., Bu, T., Ding, J., Huang, T., and Yu, Z. Reducing ann-snn conversion error through residual membrane potential. *arXiv preprint arXiv:2302.02091*, 2023a.

Hao, Z., Ding, J., Bu, T., Huang, T., and Yu, Z. Bridging the gap between anns and snns by calibrating offset spikes. *arXiv preprint arXiv:2302.10685*, 2023b.

Li, Y., Deng, S., Dong, X., Gong, R., and Gu, S. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *International Conference on Machine Learning*, pp. 6316–6325. PMLR, 2021a.

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

Li, Y., Yin, R., Park, H., Kim, Y., and Panda, P. Wearable-based human activity recognition with spatio-temporal spiking neural networks. *arXiv preprint arXiv:2212.02233*, 2022.

Li, Z., Ma, Y., Zhou, J., and Zhou, P. Spiking-nerf: Spiking neural network for energy-efficient neural rendering. *ACM Journal on Emerging Technologies in Computing Systems*, 20(3):1–23, 2025.

Liao, Z., Liu, Y., Zheng, Q., and Pan, G. Spiking nerf: Representing the real-world geometry by a discontinuous representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13790–13798, 2024.

Liu, F., Zhao, W., Chen, Y., Wang, Z., and Jiang, L. Spikeconverter: An efficient conversion framework zipping the gap between artificial neural networks and spiking neural networks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 1692–1701. AAAI Press, 2022. URL `https://ojs.aaai.org/index.php/AAAI/article/view/20061`.

Luo, X., Yao, M., Chou, Y., Xu, B., and Li, G. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, pp. 253–272. Springer, 2024.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65 (1):99–106, 2021.

Müller, T., Evans, A., Schied, C., and Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Neftci, E. O., Mostafa, H., and Zenke, F. 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.

Ponghiran, W. and Roy, K. Spiking neural networks with improved inherent recurrence dynamics for sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8001–8008, 2022.

Qu, J., Gao, Z., Zhang, T., Lu, Y., Tang, H., and Qiao, H. Spiking neural network for ultra-low-latency and high-accurate object detection, 2023.

Rathi, N. and Roy, K. Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks. *arXiv preprint arXiv:2008.03658*, 2020.

Rathi, N., Srinivasan, G., Panda, P., and Roy, K. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. *arXiv preprint arXiv:2005.01807*, 2020.

Ren, D., Ma, Z., Chen, Y., Peng, W., Liu, X., Zhang, Y., and Guo, Y. Spiking pointnet: Spiking neural networks for point clouds. *arXiv preprint arXiv:2310.06232*, 2023.

Sun, C., Sun, M., and Chen, H.-T. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5459–5469, 2022.

Viale, A., Marchisio, A., Martina, M., Masera, G., and Shafique, M. Lanesnns: Spiking neural networks for lane detection on the loihi neuromorphic processor. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 79–86. IEEE, 2022.

Wan, G., Lan, Q., Wang, H., Yitian, W., Li, W., Guo, Y., et al. Spik-nerf: Spiking neural networks for neural radiance fields. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Wu, J., Chua, Y., Zhang, M., Li, G., Li, H., and Tan, K. C. A tandem learning rule for effective training and rapid inference of deep spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021a.

Wu, J., Xu, C., Han, X., Zhou, D., Zhang, M., Li, H., and Tan, K. C. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7824–7840, 2021b.

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., and Shi, L. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1311–1318, 2019.

Yao, X., Hu, Q., Liu, T., Mo, Z., Zhu, Z., Zhuge, Z., and Cheng, J. Spikingnerf: Making bio-inspired neural networks see through the real world. *arXiv preprint arXiv:2309.10987*, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zou, S., Mu, Y., Zuo, X., Wang, S., and Cheng, L. Event-based human pose tracking by spiking spatiotemporal transformer, 2023.

A Visual Results on Synthetic Dataset

Figure 3 visualizes the rendering results on representative synthetic scenes (Lego, Drums, Mic, and Ship) for ANN-based NeRF, Spiking NeRF, Spik-NeRF, and *Spik-NeRF v2*. It is evident that our method recovers intricate geometric details and realistic appearance attributes, which are on par with those generated by ANN-based NeRF.

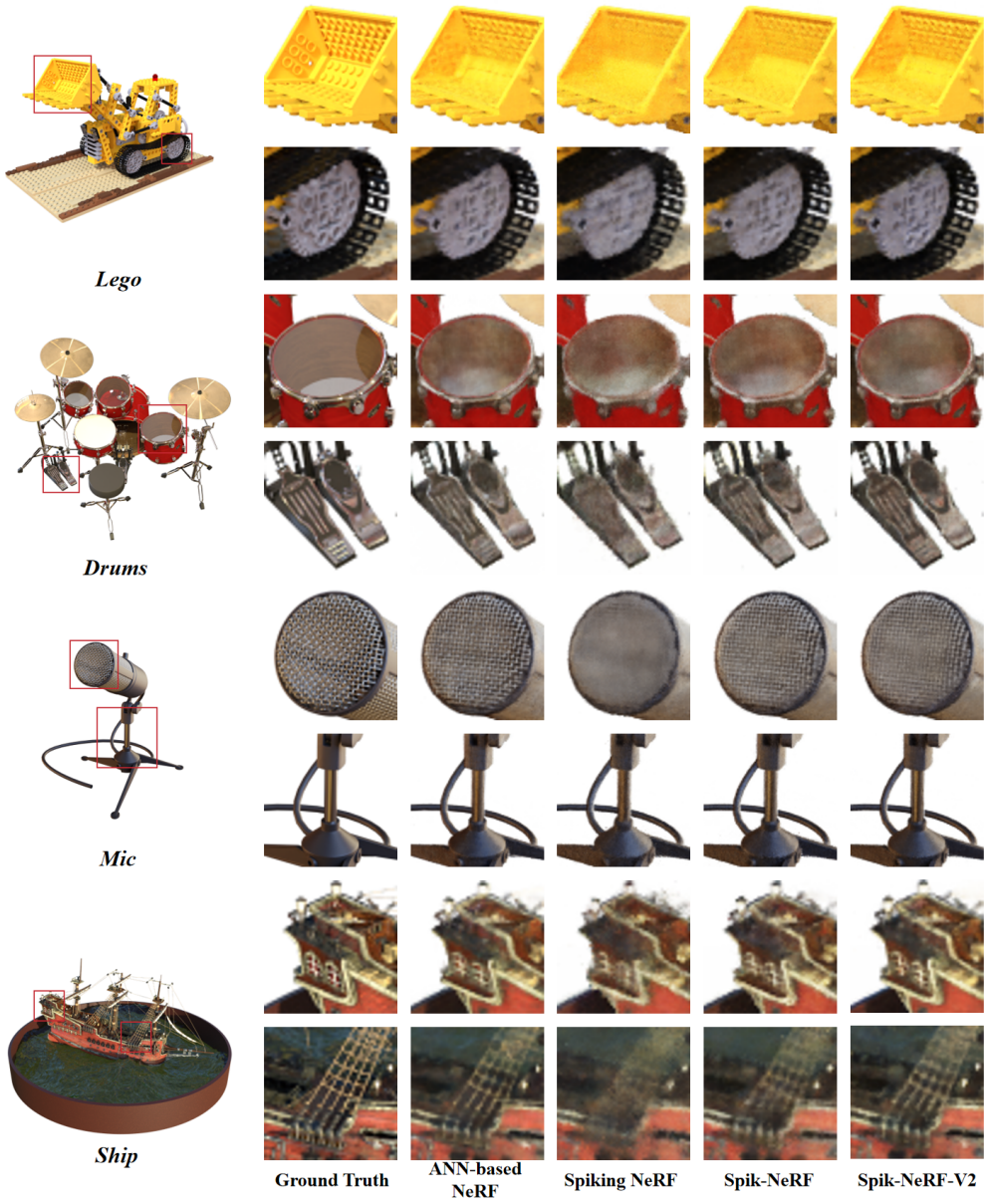


Figure 3: The rendering performance in the synthetic dataset.

B Visual Results on Realistic Dataset

Figure 4 complements the quantitative results in Table 3 with visual rendering performance on the realistic dataset. The visual results validate the effectiveness of our method, demonstrating that *Spik-NeRF v2* produces renderings with finer details and fewer artifacts compared to Spiking NeRF and Spik-NeRF, approaching the quality of ANN-based NeRF.

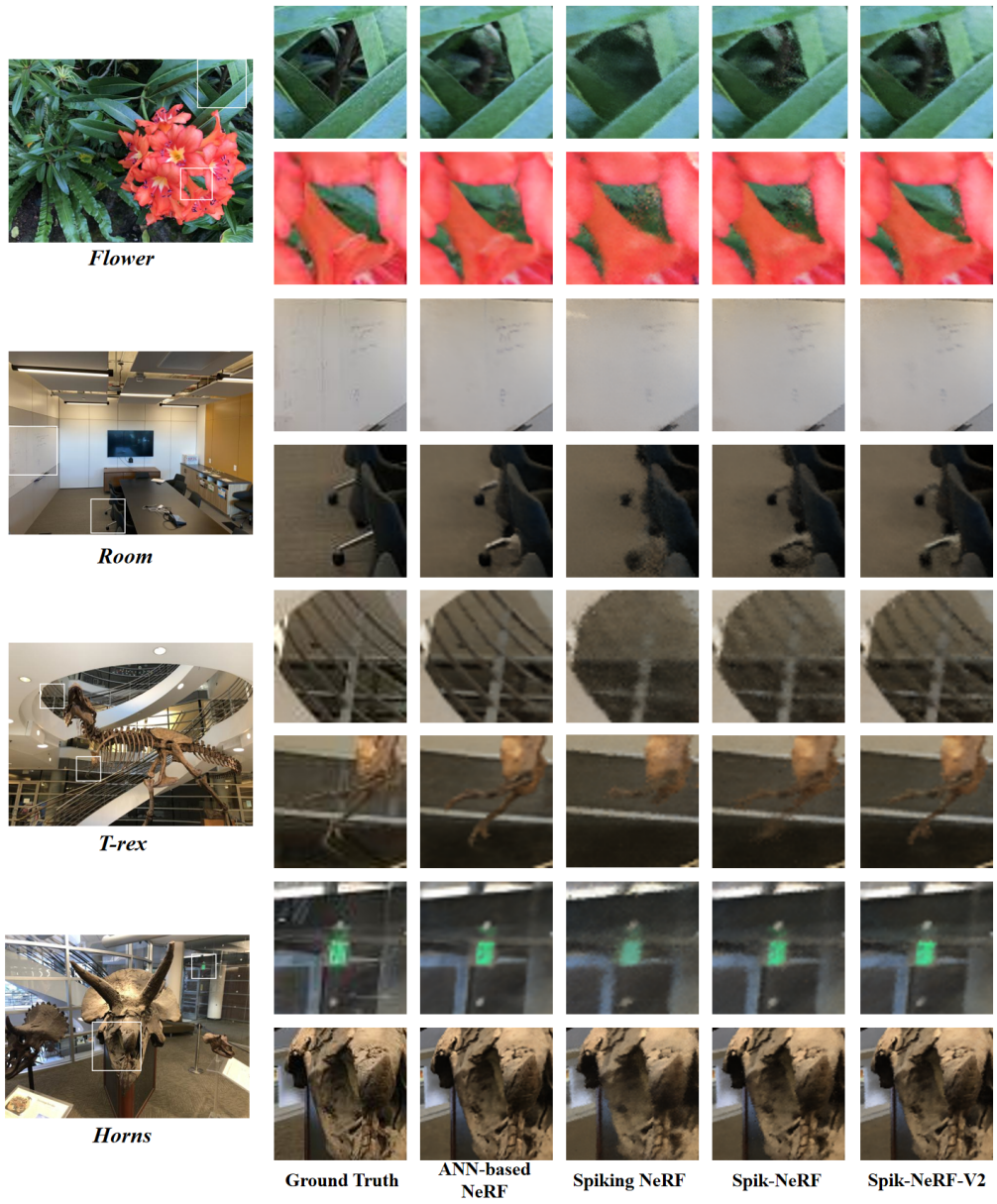


Figure 4: The rendering performance in the realistic dataset.

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes].
   Justification: We clearly state the claims made and the contributions made in both the abstract and introduction.
   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: [N/A].
   Justification: We find no limitation which we feel must be specifically highlighted here.
   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: We provide the full set of assumptions and complete proofs in the Section 3.4.3.
   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: We provide the detail experiment settings in the Section 4.
   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: [No].
   Justification: We will release the code and data upon completion of code organization.
   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: All implementations are described in the experiments section.
   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: Error bars are not reported because it would be too computationally expensive.
   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: The computation resources description is provided in the experiment section.
   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: The research conducted with the NeurIPS Code of Ethics.
   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: [No].
    Justification: There is no societal impact of the work performed.
    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: The paper poses no such risks.
    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: The original paper for datasets we used are all cited.
    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: We adopt public datasets.
    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.