Spike4DGS: Towards High-Speed Dynamic Scene Recontruction with 4D Gaussian Splatting via a Spike Camera Array

Qinghong Ye[1,3,*], **Yiqian Chang**[2,3,*], **Jianing Li**[4], **Haoran Xu**[3,5], **Xuan Wang**[2], **Wei Zhang**[3,†], **Yonghong Tian**[1,3,4,†], **Peixi Peng**[1,3,†]
[1]Shenzhen Graduate School of Peking University
[2] Shenzhen Campus of Harbin Institute of Technology
[3]Peng Cheng Laboratory
[4] School of Computer Science, Peking University
[5] Shenzhen Campus of Sun Yat-sen University

Abstract

Spike camera with high temporal resolution offers a new perspective on high-speed dynamic scene rendering. Most existing rendering methods rely on Neural Radiance Fields (NeRF) or 3D Gaussian Splatting (3DGS) for static scenes using a monocular spike camera. However, these methods struggle with dynamic motion, while a single camera suffers from limited spatial coverage, making it challenging to reconstruct fine details in high-speed scenes. To address these problems, we propose Spike4DGS, the first high-speed dynamic scene rendering framework with 4D Gaussian Splatting using spike camera arrays. Technically, we first build a multi-view spike camera array to validate our solution, then establish both synthetic and real-world multi-view spike-based reconstruction datasets. Then, we design a multi-view spike-based dense initialization module that obtains dense point clouds and camera poses from continuous spike streams. Finally, we propose a spike-pixel synergy constraint supervision to optimize Spike4DGS, incorporating both rendered image quality loss and dynamic spatiotemporal spike loss. The results show that our Spike4DGS outperforms state-of-the-art methods in terms of novel view rendering quality on both synthetic and real-world datasets. More details are available at the **project page**.

1 Introduction

Novel view synthesis [16] is a cornerstone of many cutting-edge applications, enabling the creation of precise novel views from ideal image sequences. However, conventional cameras struggle in high-speed motion scenarios, where rapid movement causes motion blur and significantly degrades reconstruction quality [39]. While some approaches [30, 44] attempt to improve reconstruction from motion-blurred images, they remain fundamentally limited by the sampling rates of RGB cameras. As a result, the use of new vision sensors for high-quality rendering in high-speed scenes has garnered increasing attention.

Spike cameras [9, 20, 48] with high temporal resolution offer a new perspective on high-speed scene rendering. Unlike conventional cameras, they asynchronously encode absolute light intensity into continuous spike streams at rates of up to 20k Hz. This unique property makes them particularly

*Qinghong Ye and Yiqian Chang contributed equally to the paper.
†Peixi Peng,Yonghong Tian({pxpeng,yhtian}@pku.edu.cn), and Wei Zhang(zhangwei1213052@126.com) are corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

effective for preserving high-speed scene textures with greater detail. Existing studies [64, 65, 24, 67, 66, 62, 60, 58, 57, 46, 59, 47] have demonstrated their capability for fine-grained 2D reconstruction. The advantages of spike cameras also indicate their immense potential for advancing 3D scene reconstruction and novel view synthesis.

Extensive works [15, 61] have explored neuromorphic cameras for rendering in high-speed scenarios. Some studies [19, 38, 21, 28, 3, 31, 52, 36] utilize Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS) for scene representation and novel view synthesis using event cameras. However, event cameras capture only light changes rather than absolute brightness, making them challenging for fine-grained 3D reconstruction. Other efforts [69, 27, 18, 51, 54] have explored 3DGS and NeRF with alternative spike cameras to overcome these limitations and enhance rendering quality. For example, pioneering works with SpikeNeRF [69] and Spike-NeRF [18] have demonstrated the feasibility of using spike streams to reconstruct 3D scenes. Nevertheless, NeRF-based methods suffer from time-consuming training and inference processes, and their implicit representations limit scene editing capabilities. In contrast, Gaussian Splatting offers a compelling alternative, providing high accuracy and fast inference speeds. For instance, SpikeGS [51, 54], SpikeNVS [7], and USP-Gaussian [5] achieve impressive results in photorealistic 3D reconstruction using spike cameras. However, these 3DGS-based methods may face difficulties in handling high-speed dynamic scenes. Additionally, relying solely on monocular cameras may present 3D reconstruction challenges in areas with weak textures or when using static cameras. In fact, camera arrays could enhance texture information through multi-view perspectives, enabling dynamic scene rendering even without camera motion.

To address the aforementioned challenges, we propose Spike4DGS, the first high-speed dynamic scene rendering framework utilizing 4D Gaussian Splatting with spike camera arrays. We aim at overcoming the following challenges: (i) *Camera setup and dataset* – How could we establish a multi-view spike camera array and build high-quality spike-based reconstruction datasets? (ii) *Effective model* – How could we design an efficient dynamic scene rendering model that directly processes multi-view spike streams with 4D Gaussian splatting(4DGS)?

To be specific, we first build a multi-view spike camera array and then establish both synthetic and real-world spike-based reconstruction datasets. Then, we design a novel multi-view spike-based dense initialization module to generate dense point clouds and camera poses from continuous spike streams. Finally, we propose a pixel-spike synergy supervision strategy to optimize Spike4DGS, which incorporates both reconstructed image quality loss and dynamic spatiotemporal spike loss. Experimental results show that our Spike4DGS outperforms state-of-the-art methods in terms of rendering quality on both synthetic and real-world datasets. We further verify that spike cameras achieve higher rendering quality than event cameras and RGB cameras in high-speed scenes. Meanwhile, rendering quality improves as the number of spike cameras increases.

The main contributions of this work are summarized as:

- We introduce Spike4DGS, the first framework that combines spike camera arrays with 4DGS, enabling novel view synthesis in high-speed dynamic scenarios.
- We present a Spike-Pixel Synergy Supervision strategy to optimize the parameters of our Spike4DGS for enhanced rendering quality.
- We build a spike camera array, along with a highly realistic synthetic and real-world datasets that contain multi-view spike streams. We believe two standardized datasets open up opportunities for research in this novel problem.

2 Related Work

2.1 NVS for Dynamic Scenes

Novel View Synthesis (NVS) tasks aim to generate unknown views of an object or scene from a set of images of known views. Representative papers include Neural Radiance Fields (NeRF) [32] and 3D Gaussian Splatting (3DGS) [23]. Recently, a large number of static 3DGS-based techniques [4, 22, 53] have been proposed due to their high quality and real-time rendering without using neural networks like NeRF. However, the assumption of static scenes prevents application to real-world scenarios with moving objects. Therefore, several works extend the 3DGS to dynamic scenes[45, 50, 29, 26, 2]. These methods are usually divided into two main lines. For instance, De3DGS [50] and 4DGS [45]

model spatial-temporal deformation with an implicit deformation field as the first line. On the other hand, the second line is based on the idea that scenes' motion could be encoded into the 3D Gaussian representation straightly, such as STG [26] and D3DGS [29], which represents changes in 3D Gaussian over time through a temporal opacity and a polynomial function for each Gaussian. The above approaches could perform well on synthetic datasets and simple real-world datasets. However, when there are some high-speed objects in the scene, traditional dynamic reconstruction methods using RGB image data may suffer motion blur since most standard RGB cameras have limited frame rates. This motivates us to use the neuromorphic cameras to avoid this problem.

2.2 Neuromorphic Cameras on 3DGS

There are two types of bio-inspired sensors: event cameras [15] and spike cameras [9]. Event cameras are based on the temporal contrast sampling method and generate events asynchronously when pixel brightness changes exceed a threshold. Early event-based reconstruction approaches [28, 3, 31, 38, 21, 42] have been proposed to derive neural radiance fields directly from event streams. E2nerf [36] and evagaussian [52] achieved sharp reconstruction from blurry images. E-4DGS [13] achieved high-fidelity dynamic reconstruction from the multi-view event cameras. GS2E [25] has introduced an effective event stream generator by gaussian splatting. Another type of neuromorphic camera is called the spike camera. Spike cameras record the absolute light intensity at a fairly high frame rate and provide a more explicit input format for detailed reconstruction. Some nerf-based methods [69, 27, 18] have verified the feasibility of reconstruction with spike, but they suffer suboptimal training and rendering speeds due to the complex spike simulation network. Some 3DGS-based spike reconstruction methods have emerged to optimize this defect. Yu's SpikeGS[51] reconstructed view synthesis results from a continuous spike stream captured by a moving spike camera. In a harder setting, Zhang's SpikeGS[54] reconstructed scenes via a single spike stream with monocular high-speed camera motion. However, there is no established spike-based method for addressing the challenge of rendering high-speed dynamic scenes using multi-view spike streams. On this basis, our work aims to overcome the limitations and construct a spike-based 3D Gaussian Splatting model for high-speed dynamic scenes via a spike camera array.

3 Methodology

3.1 Preliminaries

Spike Camera. Spike camera is a bio-inspired sensor which records and converts the absolute light intensity at a fairly high frame rate (up to 20kHz) into accumulated voltage through photoreceptors [64, 8]. If the accumulated voltage V reaches the scheduling threshold Θ, a spike will be triggered and V is reset to zero, mathematically formulated as follows:

$$V\left(t\right) = \int_{t_s}^{t} \sigma \cdot L(t)dt \bmod \Theta, \tag{1}$$

where $L(t)$ represents the instant light intensity at time t, t_s is the moment when the previous spike was emitted, and σ is the constant photoelectric conversion coefficient.

DUSt3R Initialization. DUSt3R [41] is a dense initialization method used for 3D reconstruction. Compared with COLMAP Initialization [40], DUSt3R provides more accurate point clouds under low-quality image input with less time, which is more suitable for high-speed scenes. Specifically, given a pair of images (I_1, I_2), DUSt3R utilizes a ViT for the encoder and decoder [11] and a DPT head [37] for estimating point clouds $\mathcal{PC}$:

$$\mathcal{PC} = \mathrm{DPT}(\mathrm{ViT}(I_1, I_2)). \tag{2}$$

Although dUST3r and its improved versions [49, 55] could generate dense point clouds based on multi-view images, however, the quality of point clouds depends on the quality of input images.

4D Gaussian Splatting. 4D Gaussian Splatting (4DGS) [45] is used for rendering dynamic scenes. It proposes a network that learns the Gaussian deformation field to predict the deformation of each 3D Gaussian. For input 3D Gaussian $\mathcal{G}$ and time t, a spatial-temporal structure encoder $\mathcal{H}$ and a multi-head Gaussian deformation decoder $\mathcal{D}$ are used for calculating the deformations $\Delta\mathcal{G}$:

$$\Delta\mathcal{G} = \mathcal{D}(\mathcal{H}(\mathcal{G}, t)). \tag{3}$$

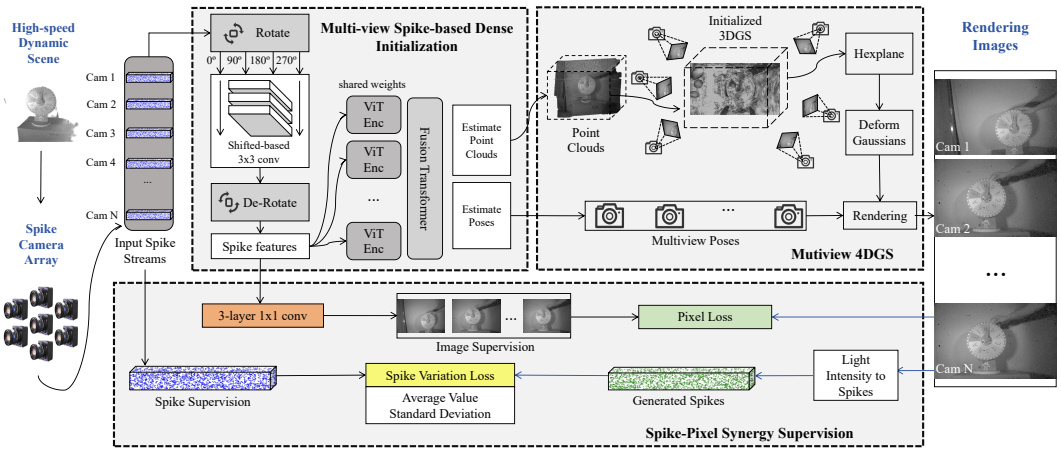

Figure 1: The Framework of our Spike4DGS, we establish the connection between the real-world spike streams and the dynamic scene rendering images. The input multi-view spike streams are sent to a **Multi-view Spike-based Dense Initialization** to estimate point cloud and camera poses. Based on the initialization, a 4DGS with **Spike-Pixel Synergy Supervision** consisting of a Pixel Loss and a Spike Variation Loss is utilized for rendering. Our Spike4DGS could reconstruct high-speed dynamic scenes with delicate motion and texture details.

3.2 Problem Fomulation

We aim to reconstruct high-speed dynamic scenes with delicate motion and texture details. To reach this goal, we build a multi-view spike camera array to capture the high-speed motion and propose a novel method called Spike4DGS for rendering based on continuous spike streams. The problem could be denoted as:

$$\textbf{Spike4DGS} : \{(S_1, S_2, ..., S_{N-1}), V_N, t\} \rightarrow \hat{I}_t \tag{4}$$

where $S_1, S_2, ..., S_{N-1}$ are the spike streams captured from N-1 views of spike cameras for training, $\hat{I}_t$ is the rendering image of novel view V_N (the view of N-th spike camera) at time t.

To solve this problem, Spike4DGS first builds an end-to-end Multi-view Spike-based Dense Initialization (MSDI) method to estimate the dense point cloud and camera poses from input spike streams, as detailed in Sec. 3.3. Then, the initial point cloud and camera poses are sent to a 4D Gaussian Splatting to generate novel view rendering results. To get more delicate motion and texture details, Spike-Pixel Synergy Supervision (SPSS) is proposed for the constraint of 4D Gaussian Splatting in Sec. 3.4. The total framework of Spike4DGS is shown in Fig. 1.

3.3 Multi-view Spike-based Dense Initialization

Previous rendering tasks based on spike cameras, such as SpikeNeRF [69], employ a two-step initialization method. They first use spike-to-image methods such as TFI [64], TFP [64] and spk2img [6] to get images, and then utilize COLMAP [40] or DUSt3R [41] on these images for the scene initialization. However, this two-step method is complex and requires high-quality spike-to-image results. When dealing with high-speed dynamic scenes, the lack of texture details in converted images may influence the final point cloud estimation. In contrast, we propose an end-to-end framework, called Multi-view Spike-based Dense Initialization (MSDI), which consists of a spike feature extractor, a point cloud and pose estimator, and an image generator. Given a series of spike streams from N spike cameras, MSDI aims to estimate 3D point clouds, camera poses, and their corresponding images:

$$\textbf{MSDI} : (S_1, ..., S_N) \rightarrow \{\mathcal{PC}, ([R|T]_1, ..., [R|T]_N), (I_1, ..., I_N)\}. \tag{5}$$

where $\mathcal{PC}$ is the estimated point cloud, $[R|T]_1, ..., [R|T]_N$ are the camera rotation and translation parameters of N spike cameras, $I_1, ..., I_N$ are the images generated from input spike streams. As

an end-to-end network, MSDI is fine-tuned together with pre-trained weights on multi-view spike streams captured from the Carla [10] simulator.

Spike Feature Extractor. Firstly, we build a feature extractor for the multi-view spike streams. Assuming Γ_t is a time interval around frame time t. For an input spike stream $S_i(\Gamma)$ which lasts for a time interval Γ_t and is captured from the i-th camera view, MSDI rotates S_{Γ_t} four times to obtain a complete receptive field in four directions and concentrate them together:

$$R(S_i(\Gamma_t)) = \mathrm{CAT}\{\mathrm{Rot}(S_i(\Gamma_t), \theta)\} | \theta \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}, \tag{6}$$

where CAT and Rot mean concatenation and rotation operations respectively. Then, we utilize a shift-based 3×3 convolution layer from [54] to extract features:

$$\hat{f}_i(\Gamma_t) = \mathcal{M}(R(S_i(\Gamma_t))), \tag{7}$$

where $\mathcal{M}$ is the shift-based convolution layer, $\hat{f}_{\Gamma_t}$ is the extracted spike features for input spike streams $S_i(\Gamma_t)$. Replacing the input $S_i(\Gamma_t)$ to N views of spike streams, we can get a series of extracted features $(\hat{f}_1(\Gamma_t), ..., \hat{f}_N(\Gamma_t))$.

Point Cloud and Pose Estimator. To estimate the point cloud and camera poses, the problem could be formulated as:

$$\mathrm{Estimator} : (\hat{f}_1(\Gamma_t), ..., \hat{f}_N(\Gamma_t)) \rightarrow \mathcal{PC}, ([R|T]_1, ..., [R|T]_N). \tag{8}$$

To solve the problem, MSDI builds an estimator consisting of N ViT encoders [11] (equals to the number of views) and a fusion transformer. For N ViT encoders, each encoder ViT_i handles a camera view with shared weights:

$$\hat{F}_i(\Gamma_t) = \mathrm{ViT}_i(\hat{f}_i(\Gamma_t)), i \in (1, ..., N). \tag{9}$$

For the fusion transformer $\mathrm{FusionTF}$, it is a 24-layer transformer which is the same as [49]:

$$G_1(\Gamma_t), G_2(\Gamma_t), ..., G_N(\Gamma_t) = \mathrm{FusionTF}\left(\hat{F}_1(\Gamma_t), \hat{F}_2(\Gamma_t), ..., \hat{F}_N(\Gamma_t)\right), \tag{10}$$

where $G_i(\Gamma_t)$ is the temporal feature of i-th camera view. This operation generates temporal features with global contextual understanding from all views.

Then, a DPT [37] decoding head is utilized for decoding the temporal features into a point cloud $\mathcal{PC}$ and its corresponding confidence map $\Sigma_{\mathcal{PC}}$:

$$(\mathcal{PC}, \Sigma_{\mathcal{PC}}) = \mathrm{DPT}(G_1(\Gamma_t), G_2(\Gamma_t), ..., G_N(\Gamma_t)). \tag{11}$$

Finally, we utilize the global point cloud $\mathcal{PC}$ to estimate camera rotations and translations $([R|T]_1, ..., [R|T]_N)$ via RANSAC-PnP [14, 1].

Image Generator. In addition, MSDI also generates discrete images from continuous spike streams. For the time interval Γ_t around frame time t, the image at time t in i-th camera view could be generated from:

$$\bar{I}_i(t) = \mathrm{BCONV}(\hat{f}_i(\Gamma_t)), i \in (1, ..., N), \tag{12}$$

where $\bar{I}_i(t)$ is the generated image, BCONV is a network consisting of three 1×1 convolutions followed from BSN [6].

Compared with the previous two-step initialization methods like TFI [64]+COLMAP [40], our end-to-end MSDI method could avoid the errors of the final estimations which occur from the low-quality images generated in the intermediate steps.

3.4 4DGS with Spike-Pixel Synergy Supervision

After MSDI, we could get the initial point cloud and camera poses. Then the initial 3D Gaussian $\mathcal{G}$ could be obtained from them. Thus, we could generate the deformed Gaussians from 4DGS [45] and render an image $\hat{I}(t)$ at i-th view p_i:

$$\hat{I}_i(t) = \mathrm{Render}(\mathrm{4DGS}(\mathcal{G}, t), p_i). \tag{13}$$

According to 4DGS, the rendered image loss could be formulated as follows to offer our Spike4DGS pixel supervision:

$$\mathcal{L}_t^{\text{pixel}} = ||\hat{I}_i(t) - \bar{I}_i(t)||_1, \tag{14}$$

where $\hat{I}_i(t)$ is the rendered image at the time t, $\bar{I}_i(t)$ is the generated images from the above MSDI. However, this pixel loss concentrates only on image similarity but ignores the texture and motion details, which are contained in spike streams. To take advantage of the spike characteristics, we propose a spike variation loss which first translates the rendered images into spike streams and then compares the variation between generated and real spike streams.

Translate from Rendered Image to Spike Stream. Let us denote the intensity values of the pixel (x, y) in rendered images at time t as $\hat{I}_i(x, y, t)$. After getting the real light of the scene, we convert the scene light intensity into spike streams using an Integrate-and-Fire (IF) [17, 12] mechanism. Following intensity translation method [68], we could establish the following relationship:

$$\hat{S}_i\left(x, y, t\right) = \text{IF}(\hat{I}_i\left(x, y, t\right) \cdot n(x, y)), \tag{15}$$

where $n(x, y)$ is the deviation matrix corresponding to the response nonuniformity noise which could be obtained by capturing a uniform light scene and recording the intensity. $\text{IF}(\cdot)$ is a IF neuron, and $\hat{S}_i\left(x, y, t\right)$ is the predicted spike values of the pixel (x, y) at time t.

Spike Variation Supervision. Different from static scenes, objects in high-speed moving scenes are constantly changing, thus the segment of spikes $S_i(\Gamma_t)$ in a time interval Γ_t around t is a continuously changing sequence. For high-speed moving objects, naive L1 or L2 loss treats each frame separately, without considering how these objects move over time. In contrast, we propose spike variation supervision,in order to concentrate on the dynamic changes of objects. Since high-speed motions are continuous, spike streams in a short time interval around time t could be generated from a single spike at time t. Thus, we simply design a one-layer MLP network ϕ_s to map a single spike $\hat{S}_i\left(t\right)$ to a spike sequence $\hat{S}_i(\Gamma_t)$ in a time interval Γ_t around t:

$$\hat{S}_i(\Gamma_t) = \phi_s(\hat{S}_i\left(t\right)), \tag{16}$$

where the shape of $\hat{S}_i\left(t\right)$ is $(H, W, 1)$ and the shape of $\hat{S}_i(\Gamma_t)$ is (H, W, Γ_t). H and W are the height and width of the spikes. Then, we calculate the average value and the standard deviation of this spike sequence $\hat{S}_i(\Gamma_t)$ and compare them with ground truth. Therefore, the Spike Variation Loss could be presented as:

$$\mathcal{L}_t^{\text{spikeV}} = ||\text{avg}(\hat{\text{S}}_\text{i}(\Gamma_\text{t})) - \text{avg}(\text{S}_\text{i}^\text{gt}(\Gamma_\text{t}))||_1 + ||\text{std}(\hat{\text{S}}_\text{i}(\Gamma_\text{t})) - \text{std}(\text{S}_\text{i}^\text{gt}(\Gamma_\text{t}))||_1. \tag{17}$$

Combining this regularization with the original Pixel Loss, we get the final synergy training loss for our Spike4DGS:

$$\mathcal{L}_t^{\text{total}} = \mathcal{L}_t^{\text{spikeV}} + \mathcal{L}_t^{\text{pixel}}. \tag{18}$$

4 Experiment

4.1 Datasets

To verify the validity of our proposed Spike4DGS, we create two datasets. The first dataset is a real-world object dataset collected by the aforementioned spike camera array. The second dataset is a high-speed synthetic outdoor dataset generated by the CARLA simulator [10]. All our experiments are conducted on these two self-made datasets. We achieve the best results on both two datasets, which demonstrates the superiority of our Spike4DGS.

Real-world Object Dataset. As shown in Fig. 2, the spike array consists of 9 spike cameras that are capable of capturing spike streams with a spatial resolution of 250 × 400 and a temporal resolution of 20k Hz. During the data collection process, synchronized recordings were made from all 9 cameras, ensuring that motion was consistently represented across different

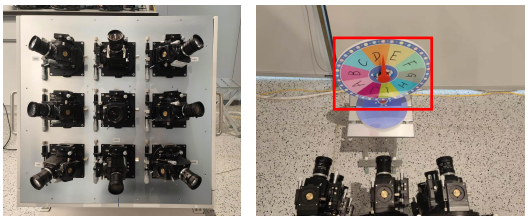

Figure 2: Our spike camera array.

Table 1: Quantitative evaluation on synthetic outdoor dataset.Unit: PSNR-dB ↑ , SSIM ↑ , LPIPS ↓.

Method	Jaywalk			Bicycle			Motor			Car			Van		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
TFI[64]+D3DGS[29]	20.65	75.7	0.384	21.73	77.7	0.385	19.98	76.2	0.377	20.09	76.2	0.387	19.24	75.3	0.402
TFP [67]+D3DGS[29]	20.88	76.5	0.357	21.78	78.9	0.346	19.85	76.0	0.372	20.07	76.6	0.361	19.20	76.4	0.389
Spk2img[57]+D3DGS[29]	20.94	76.6	0.304	21.68	77.5	0.379	20.12	76.5	0.380	20.15	76.3	0.389	18.93	75.9	0.392
TFI[64]+STG[26]	24.48	84.2	0.224	24.52	84.1	0.221	24.47	84.3	0.227	23.50	84.0	0.223	26.55	88.7	0.201
TFP [67]+STG[26]	24.45	84.0	0.220	24.50	84.2	0.225	24.53	84.1	0.222	24.60	84.3	0.226	26.48	89.0	0.205
Spk2img[57]+STG[26]	24.72	84.6	0.221	24.75	84.5	0.220	24.70	84.7	0.223	23.73	84.8	0.219	25.94	88.4	0.202
TFI[64]+4DGS[45]	26.88	87.7	0.213	27.02	89.7	0.202	25.52	90.5	0.198	25.38	86.6	0.219	25.21	86.1	0.214
TFP [67]+4DGS[45]	27.96	91.2	0.192	27.15	90.4	0.196	26.57	88.3	0.206	25.01	86.5	0.213	25.27	87.0	0.220
Spk2img[57]+4DGS[45]	27.04	90.9	0.208	26.17	88.6	0.219	26.35	88.9	0.218	24.75	87.8	0.229	24.87	87.9	0.227
Dy-SpikeGS[51]	23.04	0.852	0.243	22.17	0.846	0.249	23.35	0.857	0.238	22.75	0.849	0.244	22.87	0.851	0.242
Dy-SpikeNeRF[69]	17.04	0.784	0.372	16.17	0.773	0.384	16.35	0.776	0.379	14.75	0.759	0.395	15.87	0.768	0.388
Our Spike4DGS	**28.29**	**93.1**	**0.189**	**27.69**	**91.3**	**0.185**	**27.92**	**91.6**	**0.193**	**27.74**	**91.2**	**0.199**	**27.13**	**90.1**	**0.197**

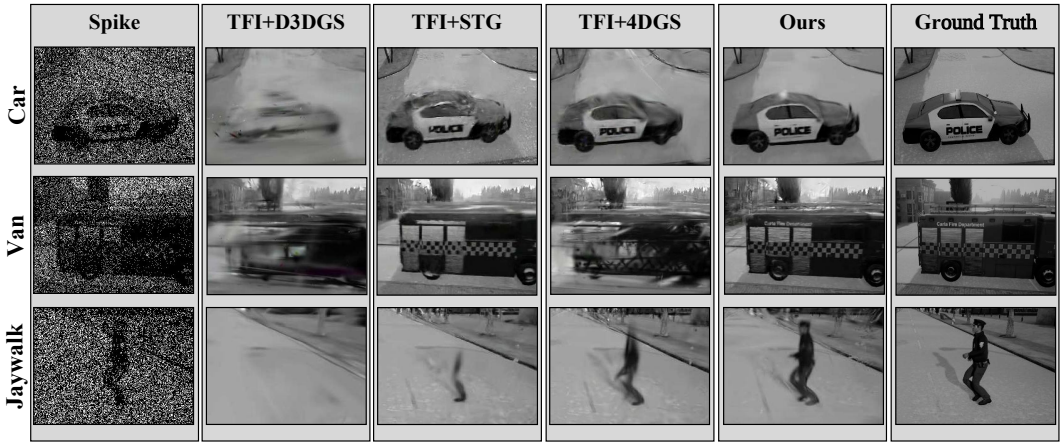

Figure 3: Quantitative comparison with other methods on the dataset on our synthetic outdoor dataset. We mainly compare our method with some SOTA approaches. In contrast, our method delivers both superior outlines and clear details.

views. When capturing spike streams, we fix the position of our spike camera array and place the high-speed dynamic objects in front of the cameras.Then we record 9 spike streams of approximately 0.5 seconds for each real-world scene with high-speed objects. Firstly, we choose high-speed dynamic objects such as “the collapse of bricks" (Bricks) and “the spin of a turntable"(Turntable) and put them before our spike camera array.Secondly, we minimize noise by providing the spike camera with ideal light intensity and obtaining multiple ideal spike streams. These spike streams could be converted to images by our MSDI module in the training process, and we use them as image supervision.

Synthetic Outdoor Dataset. To quantitatively analyze our superiority, we create a synthetic outdoor dataset using the Carla [10] simulator, which includes scenarios like Jaywalk, Bicycle, Van, Car, and Motor. These scenarios feature objects of varying sizes and speeds, with objects appearing from the left side of the image and moving to the right. An array of 9 camera sets with an overhead view was set up to capture different views of the objects. Each view setting consists of a spike camera at 20k FPS for training and an RGB camera at 1k FPS for evaluation.

4.2 Experimental Setup

Competitors. Due to the relative lack of methods for dynamic novel view synthesis based on spike cameras, we completed the comparison using some two-stage rendering approaches. First, we choose some direct spike-to-image approaches: TFI[64], TFP[64], and spk2img[57], and then combine them with previous multiview dynamic NVS methods[45, 26, 29] .For a more comprehensive experiment, we also manually integrated a deformation network from 4DGS into SpikeGS and SpikeNeRF (denoted as Dy-SpikeGS and Dy-SpikeNeRF) as comparison. Those comparisons are initialized by

Table 2: Quantitative evaluation on real-world object dataset. Unit: Brisque ↓, NIQE ↓, MetaIQA ↑.

Method	Bricks			Chips			Turntable			Bird		
	Brisque	NIQE	MetaIQA	Brisque	NIQE	MetaIQA	Brisque	NIQE	MetaIQA	Brisque	NIQE	MetaIQA
TFI[64]+D3DGS[29]	57.35	16.71	0.114	61.32	16.43	0.124	56.53	15.34	0.117	57.87	14.87	0.112
TFP [67]+D3DGS[29]	56.87	15.37	0.127	60.15	15.01	0.131	55.57	14.93	0.123	56.37	14.53	0.126
Spk2img[57]+D3DGS[29]	58.78	15.98	0.133	59.27	15.04	0.135	56.94	15.87	0.129	57.14	14.65	0.137
TFI[64]+STG [26]	45.33	13.88	0.143	43.77	13.93	0.149	36.45	10.53	0.150	33.32	12.64	0.143
TFP [67]+STG [26]	44.83	13.21	0.148	44.46	11.08	0.150	37.94	10.87	0.161	34.09	9.98	0.155
Spk2img[57]+STG [26]	45.61	13.63	0.147	44.67	12.08	0.152	38.01	10.92	0.157	34.02	9.27	0.149
TFI[64]+4DGS [45]	39.56	10.23	0.165	45.15	10.01	0.162	37.57	9.93	0.170	33.37	9.53	0.163
TFP [67]+4DGS [45]	40.58	10.42	0.164	44.46	11.08	0.150	37.94	10.87	0.161	34.09	9.98	0.155
Spk2img[57]+4DGS [45]	40.53	10.55	0.167	44.67	12.08	0.152	38.01	10.92	0.157	34.02	9.27	0.149
Dy-SpikeGS[51]	48.73	14.23	0.136	49.15	13.84	0.139	47.92	13.57	0.133	46.88	13.22	0.137
Dy-SpikeNeRF[69]	61.84	17.25	0.116	62.17	16.93	0.113	60.58	16.71	0.118	59.74	16.35	0.115
Our Spike4DGS	**34.29**	**9.95**	**0.176**	**33.52**	**8.03**	**0.183**	**26.86**	**9.86**	**0.179**	**23.86**	**8.36**	**0.180**

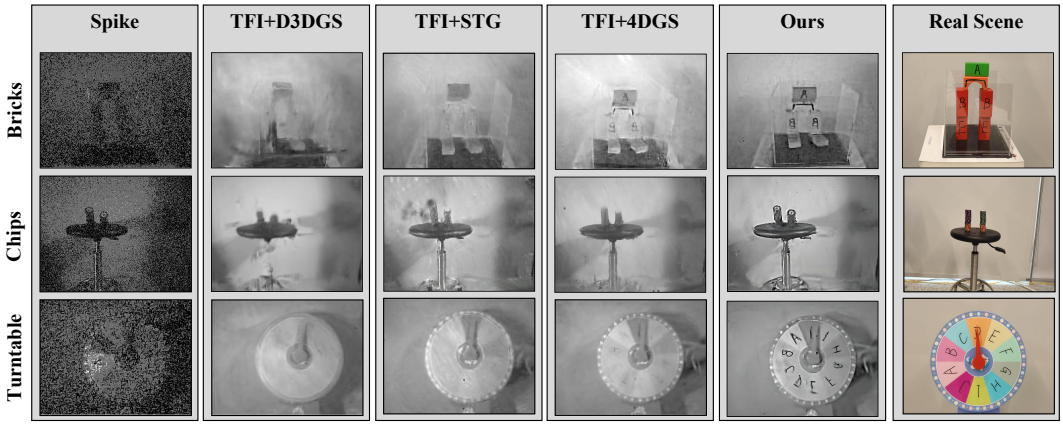

Figure 4: Qualitative comparison on real-world object datasets with SOTA rendering methods. We present the RGB photos of real scenes in the last column due to the lack of ground truth.

DUSt3R [41] to obtain point clouds. In our framework, the training data are multiple spike streams for both synthetic and real datasets.

Implementation Details. Firstly, we use the MSDI in Sec. 3.3 to estimate initialize point clouds, camera poses, and generate images. We fine-tune this end-to-end framework together with pre-trained weights using multi-view spike streams captured from dynamic scenes in the Carla simulator. The fine-tuning adopts L1 loss supervised by Carla images and confidence-aware pointmap regression loss in DUSt3R [41]. We utilize Adam optimizer with a learning rate of 0.0001 to effectively minimize both losses and ensure stable convergence during training. Secondly, for 4DGS with Spike-Pixel Synergy supervision in Sec. 3.4, the learning rate is the same as 4DGS [45]. At each optimization iteration, we randomly sample a batch of views from the same time t. The total experiments are conducted on a single NVIDIA GTX 4090 with PyTorch and the optimization for a single scene typically takes about 20 minutes to converge and 40 FPS when rendering. For the metrics of the synthetic outdoor dataset, we employ three widely-used image quality assessment metrics, PSNR [45], SSIM [43] and LPIPS [56]. For the real dataset, which lacks corresponding ground truth images, we employ NIQE [35], BRISQUE [33, 34] and MetaIQA [63] as no-reference image quality evaluation metrics, the same as state-of-the-art works [69, 51].

4.3 Performance Comparison

4.3.1 Synthetic Outdoor Data Experiments

Quantitative Performance. We present a detailed comparison of our method against some two-stage rendering approaches, such as TFI [64]+STG [26], TFI[64]+D3DGS[29], TFI[64]+4DGS [45], on our synthetic outdoor dataset. As demonstrated in Table 1, our method outperforms the SOTA two-stage rendering methods across all five distinct scenes. Note that our Spike4DGS improves more on the higher-speed scenes like “Car", which proves the effect in high-speed scenarios.

Qualitative Performance. We also present the qualitative results in Fig. 3, where: (I) TFI [64]+D3DGS [29] almost failed to reconstruct the outlines and details of each scene. (II) TFI [64]+4DGS [45] could only reconstruct the outlines with texture details missing. (iii) In contrast, our method demonstrates superior performance, generating clearer and more detailed novel views, especially in challenging high-speed scenes like “Van".

4.3.2 Real-world Object Data Experiments

Quantitative Performance. As shown in Table 2, our method achieves superior performance compared to those SOTA rendering approaches. Our method achieves an average improvement in BRISQUE [33, 34], NIQE [35] and MetaIQA [63] **by 15.4%, 4.6% and 6.7%** respectively, indicating higher-quality novel view rendering.

Qualitative Performance. In qualitative experiments, we focus on the generated texture details. Fig. 4 presents that our method could generate high-quality texture details. For example, in the third line of Fig. 4, our rendering could **generate clear alphabets "A, B, C…"** on the Turntable dataset while others could not.

4.3.3 Analysis

Performance Analysis. The quantitative and qualitative results above show that our Spike4DGS could significantly surpass the SOTA approaches. Moreover, on the synthetic outdoor dataset, our method proves the ability to render the outdoor high-speed scenes. While on the real-world object dataset, our method could also reconstruct indoor high-speed objects. This highlights the robustness and comprehensive improvements of our Spike4DGS on different high-speed dynamic scenes.

4.4 Ablation Study

Contribution of Each Component. We conduct an ablation study to assess the contribution of each component in our Spike4DGS. As shown in Table 3, we combine 4DGS with some initialization frameworks. The results demonstrate that our MSDI initialization achieves the best performance, and the SPSS module consistently improves reconstruction quality under different initialization frameworks. Results in the last row demonstrate that the full model (4DGS+MSDI+SPSS) achieves the best performance, validating the effectiveness of our design for spike-based 3D reconstruction.

Table 3: The contribution of each component.

Methods	PSNR↑	SSIM↑	LPIPS↓
TFI [64]+COLMAP [40]	21.08	84.9	0.253
TFI [64]+DUSt3R [41]	25.66	89.3	0.205
MSDI	26.66	90.3	0.198
TFI [64]+COLMAP [40]+SPSS	22.76	85.2	0.233
TFI [64]+DUSt3R [41]+SPSS	26.34	90.2	0.201
MSDI+SPSS(Full model)	**27.74**	**91.2**	**0.190**

Table 4: Ablation on supervision strategy.

Supervision Strategy	PSNR↑	SSIM↑	LPIPS↓
PixelLoss [45]	21.22	85.0	0.251
SpikeLoss [69]	13.79	76.3	0.408
PixelLoss [45]+SpikeLoss [69]	26.69	89.5	0.203
SPSS (Ours)	**27.74**	**91.2**	**0.190**

Ablation on Supervision. To evaluate the effect of our supervision strategy, we compare our SPSS in Sec. 3.4 with pixel loss in vanilla 4DGS [45], spike loss in SpikeNerf [69], and the combination of the above. The quantitative results of novel view synthesis are listed in Table 4. Our SPSS, consisting of pixel loss and spike variation loss, obtains the highest PSNR performance.

Ablation on View Numbers. In this part, we investigate the relationship between view numbers and performance. As shown in Fig. 5, we present both quantitative and qualitative comparisons. For the quantitative experiments, we use Brisque [33, 34] and NIQE [35] as the evaluation metric. Our Spike4DGS achieves the best performance

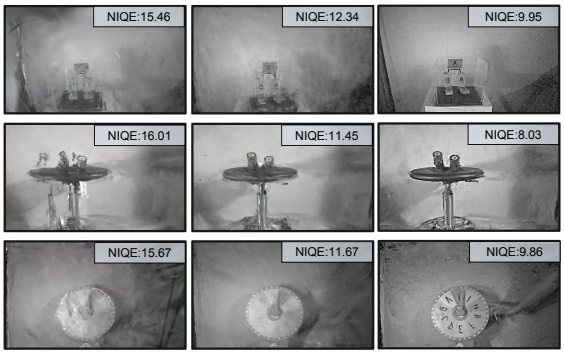

Figure 5: Quantitative and qualitative ablation of view numbers on real-world object datasets.

Table 5: Comparison of different methods under varying numbers of views. (Brisque/NIQE)

Methods	3 Views	4 Views	5 Views	6 Views	7 Views	8 Views
TFI+D3DGS [29]	83.4/23.87	73.6/21.67	65.2/20.81	62.9/18.60	59.8/16.20	56.53/15.34
TFI+STG [26]	71.0/22.80	61.2/17.50	57.0/16.70	52.1/14.60	42.0/13.10	36.45/10.53
TFI+4DGS [45]	70.3/22.70	60.8/18.40	56.4/16.50	51.8/14.40	43.6/13.90	37.57/9.93
Ours	**63.43/18.54**	**53.62/15.67**	**45.23/13.81**	**32.91/11.67**	**29.84/10.20**	**26.86/9.86**

in both visual quality and quantitative score. The quantitative comparisons across different methods are summarized in Table 5. It proves that more view numbers will lead to higher rendering quality. This means that our method could be extended to more views in the future.

The effect of MSDI's Generator. To validate the effectiveness of our MSDI module for spike2image initialization, we conducted quantitative comparisons with representative spike-to-image methods, including TFI, TFP, and spk2img. On the synthetic dataset where ground truth images are available, we evaluated the spike-to-img quality using average PSNR and SSIM. The results are summarized in Tab 6. Note the reconstruction images in test have the same view with the training images, hence the improvements are more clearly than NVS.

Table 6: Quantitative comparison on average quality.

Metric	TFI [3]	TFP [3]	TVS [4]	Spike2img [5]	MSDI's images
Avg PSNR↑	24.51	25.37	22.43	30.76	**34.90**
Avg SSIM↑	0.850	0.852	0.821	0.904	**0.961**

The effect of Whole MSDI. This part provides a more detailed evaluation of the module MSDI. Specifically, we feed the MSDI outputs (i.e., reconstructed images, point clouds, and poses) into the standard 4DGS procedure, respectively, to validate the contribution of MSDI. In addition, the traditional spike-to-image methods (e.g., TFI) and MSDI+SPSS are also listed for comparison. The experiments are conducted on the synthetic dataset, and the average performances are reported in the following Tab. 7. It indicates MSDI's contribution positively.

Table 7: Focus Comparison of MSDI without SPSS.

Method	Avg PSNR↑	Avg SSIM↑	Avg LPIPS↓
TFI + COLMAP + 4DGS	21.08	0.849	0.253
TFI + DUST3R + 4DGS	25.66	0.893	0.205
MSDI IMAGE + COLMAP + 4DGS	22.43	0.854	0.233
MSDI IMAGE + DUST3R + 4DGS	25.76	0.898	0.203
MSDI + 4DGS	26.66	0.903	0.198
MSDI + SPSS (Ours)	**27.74**	**0.912**	**0.190**

5 Conclusions

This paper introduces Spike4DGS, a novel framework that seamlessly integrates multiple spike streams captured by a spike array into 4DGS training, effectively addressing the challenges of reconstructing high-speed dynamic scenes. Spike4DGS designs a novel Multi-view Spike-based Dense Initialization module to obtain dense point clouds from continuous spike streams and a Spike-Pixel Synergy Supervision strategy to optimize the parameters for enhanced rendering quality. We contribute two novel datasets and conduct comprehensive evaluations. The results on the datasets demonstrate that our Spike4DGS surpasses previous SOTA dynamic reconstruction approaches in high-speed dynamic scenes, with almost no sacrifice in training cost and rendering FPS.

Limitation. While our spike camera array is functional, the approach has limitations which we will solve in our future works. The images rendered by our Spike4DGS are grayscale due to the lack of RGB information. Additionally, the spike array is not portable, which limits its use in mobile or field-based applications.

Acknowledgements The study was funded by the National Natural Science Foundation of China under contracts Shenzhen Science and Technology Program(KQTD20240729102051063), No. 62422602, No. 62425101, No. 62332002, No. 62372010, No. 62027804, No. 62088102, No. 62206281, and the major key project of the Peng Cheng Laboratory (PCL2021A13 and PCL2024AS204). Computing support was provided by Pengcheng Cloudbrain.

References

[1] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 30(9/10):1333–1341, 2001.

[2] Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. *arXiv preprint arXiv:2404.03613*, 2024.

[3] Anish Bhattacharya, Ratnesh Madaan, Fernando Cladera, Sai Vemprala, Rogerio Bonatti, Kostas Daniilidis, Ashish Kapoor, Vijay Kumar, Nikolai Matni, and Jayesh K Gupta. Evdnerf: Reconstructing event data with dynamic neural radiance fields. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5846–5855, 2024.

[4] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19457–19467, 2024.

[5] Kang Chen, Jiyuan Zhang, Zecheng Hao, Yajing Zheng, Tiejun Huang, and Zhaofei Yu. Usp-gaussian: Unifying spike-based image reconstruction, pose correction and gaussian splatting. *arXiv*, 2024.

[6] Shiyan Chen, Chaoteng Duan, Zhaofei Yu, Ruiqin Xiong, and Tiejun Huang. Self-supervised mutual learning for dynamic scene reconstruction of spiking camera. In *IJCAI*, pages 2859–2866, 2022.

[7] Gaole Dai, Zhenyu Wang, Qinwen Xu, Ming Lu, Wen Chen, Boxin Shi, Shanghang Zhang, and Tiejun Huang. Spikenvs: Enhancing novel view synthesis from blurry images via spike camera. *arXiv*, 2024.

[8] Siwei Dong, Lin Zhu, Daoyuan Xu, Yonghong Tian, and Tiejun Huang. An efficient coding method for spike camera using inter-spike intervals. *arXiv preprint arXiv:1912.09669*, 2019.

[9] Siwei Dong, Tiejun Huang, and Yonghong Tian. Spike camera and its coding methods. *arXiv*, 2021.

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. In *Advances in Neural Information Processing Systems*, pages 21056–21069. Curran Associates, Inc., 2021.

[13] Chaoran Feng, Zhenyu Tang, Wangbo Yu, Yatian Pang, Yian Zhao, Jianbin Zhao, Li Yuan, and Yonghong Tian. E-4dgs: High-fidelity dynamic reconstruction from the multi-view event cameras. In *ACM MM 2025*, 2025.

[14] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[15] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

[16] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022.

[17] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[18] Yijia Guo, Yuanxi Bai, Liwen Hu, Mianzhi Liu, Ziyi Guo, Lei Ma, and Tiejun Huang. Spike-nerf: Neural radiance field based on spike camera. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024.

[19] Haiqian Han, Jianing Li, Henglu Wei, and Xiangyang Ji. Event-3dgs: Event-based 3d reconstruction using 3d gaussian splatting. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 128139–128159, 2025.

[20] Tiejun Huang, Yajing Zheng, Zhaofei Yu, Rui Chen, Yuan Li, Ruiqin Xiong, Lei Ma, Junwei Zhao, Siwei Dong, Lin Zhu, et al. 1000× faster camera and machine vision with ordinary devices. *Engineering*, 25: 110–119, 2023.

[21] Inwoo Hwang, Junho Kim, and Young Min Kim. Ev-nerf: Event based neural radiance field. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 837–847, 2023.

[22] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024.

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[24] Jianing Li, Xiao Wang, Lin Zhu, Jia Li, Tiejun Huang, and Yonghong Tian. Retinomorphic object detection in asynchronous visual streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1332–1340, 2022.

[25] Yuchen Li*, Chaoran Feng*, Zhenyu Tang, Kaiyuan Deng, Wangbo Yu, Yonghong Tian, and Li Yuan. Gs2e: Gaussian splatting is an effective data generator for event stream generation. *arXiv preprint arXiv:2505.15287*, 2025.

[26] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024.

[27] Zhanfeng Liao, Yan Liu, Qian Zheng, and Gang Pan. Spiking nerf: Representing the real-world geometry by a discontinuous representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13790–13798, 2024.

[28] Weng Fei Low and Gim Hee Lee. Robust e-nerf: Nerf from sparse & noisy events under non-uniform motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18335–18346, 2023.

[29] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024.

[30] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. Deblur-nerf: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12861–12870, 2022.

[31] Qi Ma, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Deformable neural radiance fields using rgb and event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3590–3600, 2023.

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65 (1):99–106, 2021.

[33] Anish Mittal, Anush K Moorthy, and Alan C Bovik. Blind/referenceless image spatial quality evaluator. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, pages 723–727. IEEE, 2011.

[34] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.

[35] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

[36] Yunshan Qi, Lin Zhu, Yu Zhang, and Jia Li. E2nerf: Event enhanced neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13254–13264, 2023.

[37] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[38] Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. Eventnerf: Neural radiance fields from a single colour event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4992–5002, 2023.

[39] Ryusuke Sagawa, Ryo Furukawa, and Hiroshi Kawasaki. Dense 3d reconstruction from high frame-rate video using a static grid pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9): 1733–1747, 2014.

[40] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[41] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[42] Shuaixian Wang, Haoran Xu, Yaokun Li, Jiwei Chen, and Guang Tan. Ie-nerf: Exploring transient mask inpainting to enhance neural radiance fields in the wild. *Neurocomputing*, 618:129112, 2025.

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[44] Yuchen Weng, Zhengwen Shen, Ruofan Chen, Qi Wang, and Jun Wang. Eadeblur-gs: Event assisted 3d deblur reconstruction with gaussian splatting. *arXiv*, 2024.

[45] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024.

[46] Xijie Xiang, Lin Zhu, Jianing Li, Yixuan Wang, Tiejun Huang, and Yonghong Tian. Learning super-resolution reconstruction for high temporal resolution spike stream. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(1):16–29, 2021.

[47] Jiangtao Xu, Jiawei Zou, Zhiyuan Gao, and Jianguo Ma. Analysis of input-dependent noise in self-timed reset dynamic vision sensor and its impact on data quality. *IEEE Sensors Journal*, 19(15):6240–6250, 2019.

[48] Jiangtao Xu, Liang Xu, Zhiyuan Gao, Peng Lin, and Kaiming Nie. A denoising method based on pulse interval compensation for high-speed spike-based image sensor. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(8):2966–2980, 2020.

[49] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. *arXiv preprint arXiv:2501.13928*, 2025.

[50] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024.

[51] Jinze Yu, Xin Peng, Zhengda Lu, Laurent Kneip, and Yiqun Wang. Spikegs: Learning 3d gaussian fields from continuous spike stream. In *Proceedings of the Asian Conference on Computer Vision*, pages 4280–4298, 2024.

[52] Wangbo Yu, Chaoran Feng, Jiye Tang, Xu Jia, Li Yuan, and Yonghong Tian. Evagaussians: Event stream assisted gaussian splatting from blurry images. *arXiv preprint arXiv:2405.20224*, 2024.

[53] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19447–19456, 2024.

[54] Jiyuan Zhang, Kang Chen, Shiyan Chen, Yajing Zheng, Tiejun Huang, and Zhaofei Yu. Spikegs: 3d gaussian splatting from spike streams with high-speed camera motion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9194–9203, 2024.

[55] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

[56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[57] Jing Zhao, Ruiqin Xiong, Hangfan Liu, Jian Zhang, and Tiejun Huang. Spk2imgnet: Learning to reconstruct dynamic scene from continuous spike stream. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11996–12005, 2021.

[58] Jing Zhao, Ruiqin Xiong, Jian Zhang, Rui Zhao, Hangfan Liu, and Tiejun Huang. Learning to super-resolve dynamic scenes for neuromorphic spike camera. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3579–3587, 2023.

[59] Rui Zhao, Ruiqin Xiong, Jian Zhang, Zhaofei Yu, Shuyuan Zhu, Lei Ma, and Tiejun Huang. Spike camera image reconstruction using deep spiking neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(6):5207–5212, 2023.

[60] Rui Zhao, Ruiqin Xiong, Jing Zhao, Jian Zhang, Xiaopeng Fan, Zhaofei Yu, and Tiejun Huang. Boosting spike camera image reconstruction from a perspective of dealing with spike fluctuations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24955–24965, 2024.

[61] Xu Zheng, Yexin Liu, Yunfan Lu, Tongyan Hua, Tianbo Pan, Weiming Zhang, Dacheng Tao, and Lin Wang. Deep learning for event-based vision: A comprehensive survey and benchmarks. *arXiv*, 2023.

[62] Yajing Zheng, Lingxiao Zheng, Zhaofei Yu, Boxin Shi, Yonghong Tian, and Tiejun Huang. High-speed image reconstruction through short-term plasticity for spiking cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6358–6367, 2021.

[63] Hancheng Zhu, Leida Li, Jinjian Wu, Weisheng Dong, and Guangming Shi. Metaiqa: Deep meta-learning for no-reference image quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14143–14152, 2020.

[64] Lin Zhu, Siwei Dong, Tiejun Huang, and Yonghong Tian. A retina-inspired sampling method for visual texture reconstruction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1432–1437. IEEE, 2019.

[65] Lin Zhu, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. Retina-like visual image reconstruction via spiking neural model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1438–1446, 2020.

[66] Lin Zhu, Jianing Li, Xiao Wang, Tiejun Huang, and Yonghong Tian. Neuspike-net: High speed video reconstruction via bio-inspired neuromorphic cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2400–2409, 2021.

[67] Lin Zhu, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. Ultra-high temporal resolution visual reconstruction from a fovea-like spike camera via spiking neuron model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1233–1249, 2022.

[68] Lin Zhu, Yunlong Zheng, Mengyue Geng, Lizhi Wang, and Hua Huang. Recurrent spike-based image restoration under general illumination. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8251–8260, 2023.

[69] Lin Zhu, Kangmin Jia, Yifan Zhao, Yunshan Qi, Lizhi Wang, and Hua Huang. Spikenerf: Learning neural radiance fields from continuous spike stream. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6285–6295, 2024.

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 claim the contributions of this work at the end of the Introduction, and elaborate on how we achieve them in the Methodology.

   Guidelines:

   - The answer NA 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 NA 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: See supplementary materials.

   Guidelines:

   - The answer NA 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 present the assumption and its proof in our Methodology section, along with clear statements and proper references.

Guidelines:

- The answer NA 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 detailed model designs and experimental settings in the main text.

   Guidelines:

   - The answer NA 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: The dataset and code used in this work will be released shortly after simple organization.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The detailed experimental settings are presented in Section 4.2 and the supplementary materials.

   Guidelines:

   - The answer NA 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: [Yes]

   Justification: We use statistical significance in the evaluation metrics.

   Guidelines:

   - The answer NA 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 computational resources are described in Section 4.2.

   Guidelines:

   - The answer NA 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: There are no ethical concerns involved in this work.

   Guidelines:

   - The answer NA 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: [Yes]

    Justification: See supplementary materials.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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 creators or original owners of assets (e.g., code, data, and models) used in the paper are properly credited, and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

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

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.