
Supplementary Material for “Spike4DGS: Towards High-Speed Dynamic Scene Rendering with 4D Gaussian Splatting via a Spike Camera Array”

Anonymous Author(s)

Affiliation

Address

email

A Corrigenda to Table 1

In Table 1 of the main text, there are minor inaccuracies in the last two columns of rows 4 to 6. Specifically, the SSIM values for the TFI + STG, TFP + STG, and spike2img + STG methods should be **88.7, 89.0, and 88.4**, respectively, while the corresponding LPIPS values should be **0.201, 0.205, and 0.202**. Additional quantitative results for more methods are provided in Sec. G.

B Real-world Object Dataset Details

The spike array consists of 9 spike cameras which are capable of capturing spike streams with a spatial resolution of 250×400 and a temporal resolution of 20,000 Hz. The device is shown in Fig. 1. During the data collection process, synchronized recordings were made from all nine cameras, ensuring that motion was consistently represented across different views. Then, we fix the position of our spike camera array and place the high-speed dynamic objects in front of the cameras. The motion process of every high-speed scene lasts about 30 seconds with extremely high speed. We record nine spike streams for every real-world object. We minimize noise by providing the spike camera with ideal light intensity and get multiple ideal spike streams. These spike streams could convert to images by our MSDI module in the training process as the image supervision. Each scene lasts 0.5s with high-speed objects. The recorded spike streams in the real-world object dataset are visually depicted in Fig. 2. An overview of the design purposes of these crafted scenes is provided below:

Device details This device is a spike camera array system within an internal automatic time synchronization mechanism, enabling simultaneous Spike data capture from 9 views. Spike cameras, as event-driven vision sensors, excel at detecting dynamic changes in a scene and recording light intensity variations with microsecond-level temporal resolution. The synchronized multi-view setup allows the device to comprehensively capture dynamic information in three-dimensional space, making it highly effective for applications requiring high-speed motion analysis, 3D reconstruction, and low-latency event detection. By recording only light intensity changes, the system minimizes data redundancy and computational load, while its high dynamic range ensures robust performance in varying lighting

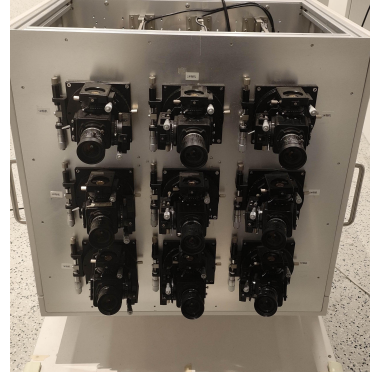


Figure 1: Our experiment Device



Figure 3: The spikes and images of synthetic outdoor dataset.

conditions. This combination of features makes the device a powerful tool for advanced vision-based research and real-time applications.

Real-world Object Dataset

- *Bricks*: A collection of building blocks arranged in a structured formation, subjected to external forces causing them to collapse. This scene is designed to simulate physical interactions and building structural.
- *Chips*: A pile of chips collapsing rapidly under the influence of an external force which is useful for studying structural dynamics and impact analysis in many fields such as physics and engineering.
- *Turntable*: A high-speed rotating turntable with words. The scene is ideal for studying patterns of movement, stability, and the transfer of energy in a controlled environment.
- *Bird*: A toy bird with wings flapping at high speed. This setup vividly simulates the flapping motion of bird flight, making it suitable for research in the principles of flapping flight.

C Synthetic Outdoor Dataset Details

To enable a comprehensive and controlled evaluation, we construct a synthetic dataset using the CARLA simulator, which allows precise manipulation of scene dynamics and sensor configurations. We simulate some representative dynamic outdoor scenes—Jaywalk, Bicycle, Motor, Van, and Car—each designed to contain significant object motion and occlusion. For each scene, data is captured from six different viewpoints to support multiview synthesis. At each viewpoint, we collect three modalities: (1) a low-frame-rate RGB video that exhibits motion blur and artifacts, emulating conventional consumer cameras; (2) a high-frame-rate, sharp RGB video that serves as a proxy for high-speed camera output; (3) a corresponding spike stream simulating the output of a spike camera.

All data is rendered at a fixed resolution of 480×300 pixels to ensure alignment and consistency across modalities. The high-frame-rate videos are used as ground truth for quantitative and qualitative evaluation of novel view synthesis performance, allowing us to assess the fidelity of reconstructed views. Visual examples of both the spike data and corresponding image frames from our synthetic scenes are provided in Fig. 4, illustrating the diversity and complexity of the motion patterns captured in our dataset.

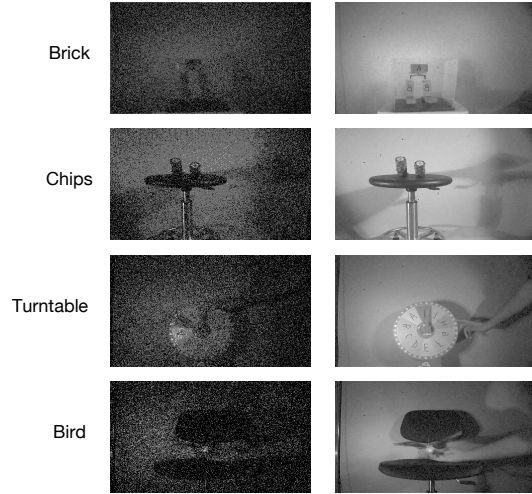


Figure 2: The spikes and images of real dataset.



Figure 4: Qualitative experiment for the best hyperparameter

D Hyperparameter Settings

Our hyperparameters mainly follow the 4DGS settings. The learning rate is set as 1.6×10^{-3} and decays to 1.6×10^{-4} at the end of the training. The basic resolution of our multiresolution HexPlane is set to 256. The Gaussian deformation decoder is a tiny MLP with a learning rate from 1.6×10^{-4} to 1.6×10^{-5} . It is worth mentioning that we explore the impact of the coefficients of pixel loss and spike variation loss in our Spike-pixel Synergy Supervision. The quantitative table is shown in Table. 1 and the qualitative figure is shown in Fig. 1. The results show that when the coefficient of pixel loss and spike variation loss are both 1.0, the Spike-pixel Synergy Supervision works best.

E More Details of Spike Loss

For spike data supervision, we need a spike loss between the simulated spike and the real spike to supervise the scene. Before we establish this loss, we need to first establish their relationship in terms of light intensity. The goal is to estimate the intensity values that correspond to the real light of the scene. Let us denote the intensity values of the pixel (x, y) at time t as $\hat{I}(x, y, t)$. The objective is to estimate $\hat{I}(x, y, t)$ corresponding to the real light of the scene. Assuming that $\hat{I}(x, y, t)$ is clear, to supervise $\hat{I}(x, y, t)$ with the real noisy spike stream, we need to consider multiple noises. Inspired by [11], we can establish the following relationship:

Table 1: Quantitative experiment for hyperparameters.

View Numbers	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
w/o L_{spike}	24.08	0.838	0.231
$L_{image} + 0.2 * L_{spike}$	25.66	0.868	0.221
$L_{image} + 0.5 * L_{spike}$	26.66	0.873	0.210
$L_{image} + 0.8 * L_{spike}$	27.17	0.894	0.194
$L_{image} + L_{spike}$	27.74	0.912	0.190
$L_{image} + 1.2 * L_{spike}$	26.90	0.890	0.202

$$\hat{I}(x, y, t) + I(N, t) = \frac{1}{\frac{Q_r}{L + N_p + N_d} + N_{rnu} + N_q} + N_c \quad (1)$$

Where $I(N, t)$ denotes the intensity changes caused by noise, L represents the scene light intensity, and Q_r is relative quantity matrix of electric charge. N_p , N_d , N_{rnu} , N_q , and N_c represent shot noise, dark current noise, response nonuniformity noise, quantization noise, and truncation noise, respectively. In fact, the deviation matrix $R(x, y)$ corresponding to the response nonuniformity noise can be obtained by capturing a uniform light scene and recording the intensity. So we rewrite the left of Equation.1 as $\hat{I}(x, y, t) \cdot R(x, y)$.

F The details for MSDI

Spike streams could be converted to images by some method, such as TFI and TFP. We could describe them by:

$$\text{TFI: } P_{t_i} = \frac{\Theta}{d_{t_i}}, \text{ TFP: } P_{t_i} = \frac{N_w}{w} * C \quad (2)$$

Those images could produce 3D point cloud by some frameworks like DUST3R. The process is two-step. In contrast, our Multi-view Spike-based Dense Initialization(MSDI) is an end-to-end initialization framework from spike stream to 3d point clouds.

107 **Training for MSDI Image Generator.** MSDI could generate discrete images from continuous spike
 108 streams. For the time interval Γ_t around frame time t , the image at time t in i -th camera view could be
 109 generated as $\bar{I}_i(t)$. The architecture for the process is illustrated in main text. We denote the ground
 110 truth images from Carla as $I_i(t)$. The loss for supervising the image generator could be defined as a
 111 simple L1 Loss:

$$\mathcal{L}_{image} = \|\bar{I}_i(t) - I_i(t)\| \quad (3)$$

112 **Training for MSDI Point Head.** MSDI utilized a ViT[2] encoder to extract the 2D features from
 113 spikes and decodes these features to point cloud and camera poses by a separate DPT-L [5]. The
 114 training objective of MSDI’s point head is inspired by [6], we use a confidence-aware loss to regress
 115 the model:

$$\mathcal{L}_{pts} = \sum_{v,u \in \{1 \dots V\}} C^{v,u} \left\| \frac{1}{z} X^{v,u} - \frac{1}{\bar{z}} \hat{X}^{v,u} \right\| - \beta \log C^{v,u} \quad (4)$$

116 Here, \hat{X} is the ground-truth pointmap. The scale normalization factor $z = \text{norm}(X^{v,u})$ and
 117 $\bar{z} = \text{norm}(\hat{X}^{v,u})$ y represent the average distance of all valid points to the origin.

$$L_{regr}(i, p) = \left\| \frac{1}{z} X_p^{i,j} - \frac{1}{\bar{z}} \hat{X}_p^{i,j} \right\| \quad (5)$$

118 L_{pts} is a confidence-based loss used to handle points with ill-defined depths, such as points cor-
 119 responding to the sky, or to translucent objects. The hyperparameter β governs how confident
 120 the network should be, while z and \bar{z} are normalization factors used for non-metric datasets. The
 121 generated pointmap facilitates the effective representation of 3D shapes in image space and enables
 122 triangulation between rays from various viewpoints, independent of the limitations imposed by depth
 123 estimation quality.

124 G Additional Details of Quantitative Results

125 In this section, we present a comprehensive per-scene analysis of NVS results on the synthetic outdoor
 126 dataset. The detailed quantitative results on six synthetic scenarios are shown in Table.2 and Table. 3.
 127 The adopted metrics include PSNR, SSIM, and LPIPS. As shown in Table. 2, our method achieves the
 128 best PSNR scores across all scenes, highlighting its ability to produce high-quality images. For SSIM,
 129 as presented in Table. 3, our approach significantly outperforms the baselines, achieving the highest
 130 average SSIM score. This demonstrates our method’s robustness in generating images with minimal
 131 perceptual artifacts. which indicates superior perceptual similarity to the ground truth images. These
 132 observations suggest that our method excels in learning a more precise 3D representation of the scene
 133 within the proposed framework.

134 H Supplementary Video

135 We provide a supplementary video to show the video results. For the synthetic outdoor data, we show
 136 the car scene’s results of our baseline TFI+4DGS and our Spike4DGS. For the real-world object data,
 137 we show the comparison of the results in the Turntable and Brick scenes. It is obvious that the results
 138 of our Spike4DGS have less noise, better contrast, and sharper object texture details compared to the
 139 baseline on both synthetic objects and real scenes.

140 I Societal Impact and Limitation.

141 **Societal Impact.** This work presents a spike-based 3D reconstruction framework with significant
 142 potential benefits across several domains. Its high temporal resolution and low energy consumption
 143 offer notable advantages for real-time perception in autonomous driving, intelligent robotics, and
 144 compact wearable devices potentially.

145 **Limitation.** While our spike camera array enables novel view synthesis in high-speed dynamic
 146 scenes, the approach has some limitations. First, the spike cameras offer extremely high frame rates
 147 but suffer from low spatial resolution, which limits the image details. Second, the images rendered by
 148 our Spike4DGS are grayscale due to the lack of RGB information. Additionally, the spike array is
 149 functional but not portable, which limits its use in mobile or field-based applications.

Methods	PSNR \uparrow					
	Jaywalk	Bicycle	Motor	car	Van	Average
TFI [9]+Hexplane [1]	18.35	20.73	18.98	19.02	18.24	19.06
TFP [10]+Hexplane [1]	19.41	19.32	18.87	18.97	18.80	19.07
Spk2img [8]+Hexplane [1]	18.49	19.82	19.02	18.70	17.99	18.80
TFI [9]+D3DGS [4]	20.65	21.73	19.98	20.09	19.24	20.34
TFP [10] [10]+D3DGS [4]	20.88	21.78	19.85	20.07	19.20	20.36
Spk2img [8]+D3DGS [4]	20.94	21.68	20.12	20.15	18.93	20.36
TFI [9]+STG [3]	24.48	22.34	24.52	23.50	26.55	24.28
TFP [10]+STG [3]	24.45	21.78	24.50	24.60	26.48	24.36
Spk2img [8]+STG [3]	24.72	24.75	24.70	23.73	25.94	24.77
TFI [9]+4DGS [7]	26.88	27.02	25.52	25.38	25.21	26.00
TFP [10]+4DGS [7]	27.96	27.15	26.57	25.01	25.27	26.39
Spk2img [8]+4DGS [7]	27.04	26.17	26.35	24.75	24.87	25.84
Ours Spike4DGS	28.29	27.69	27.92	27.74	26.74	27.77

Table 2: The Novel View Synthesis of PSNR in our synthetic outdoor Dataset.

Methods	SSIM \uparrow					
	Jaywalk	Bicycle	Motor	Car	Van	Average
TFI [9]+Hexplane [1]	0.749	0.769	0.750	0.752	0.746	0.753
TFP [10]+Hexplane [1]	0.752	0.763	0.753	0.757	0.748	0.755
Spk2img [8]+Hexplane [1]	0.747	0.760	0.757	0.755	0.744	0.753
TFI [9]+D3DGS [4]	0.757	0.777	0.762	0.762	0.753	0.762
TFP [10]+D3DGS [4]	0.765	0.789	0.760	0.766	0.764	0.767
Spk2img [8]+D3DGS [4]	0.766	0.775	0.765	0.763	0.759	0.766
TFI [9]+STG [3]	0.842	0.841	0.843	0.840	0.887	0.850
TFP [10]+STG [3]	0.840	0.842	0.841	0.843	0.890	0.851
Spk2img [8]+STG [3]	0.846	0.845	0.847	0.848	0.884	0.854
TFI [9]+4DGS [7]	0.877	0.897	0.905	0.866	0.861	0.881
TFP [10]+4DGS [7]	0.912	0.904	0.883	0.865	0.870	0.887
Spk2img [8]+4DGS [7]	0.909	0.886	0.889	0.878	0.879	0.888
Ours Spike4DGS	0.931	0.913	0.916	0.912	0.901	0.915

Table 3: The Novel View Synthesis of SSIM in our synthetic outdoor Dataset.

Methods	LPIPS \downarrow					
	Jaywalk	Bicycle	Motor	Car	Van	Average
TFI [9]+Hexplane [1]	0.389	0.390	0.378	0.391	0.392	0.388
TFP [10]+Hexplane [1]	0.392	0.382	0.401	0.398	0.387	0.392
Spk2img [8]+Hexplane [1]	0.388	0.380	0.378	0.392	0.399	0.387
TFI [9]+D3DGS [4]	0.384	0.385	0.377	0.387	0.402	0.387
TFP [10] [10]+D3DGS [4]	0.357	0.346	0.372	0.361	0.389	0.365
Spk2img [8]+D3DGS [4]	0.304	0.379	0.380	0.389	0.392	0.369
TFI [9]+STG [3]	0.224	0.221	0.227	0.223	0.201	0.219
TFP [10]+STG [3]	0.220	0.225	0.222	0.226	0.205	0.220
Spk2img [8]+STG [3]	0.221	0.220	0.223	0.219	0.202	0.217
TFI [9]+4DGS [7]	0.213	0.202	0.198	0.219	0.214	0.209
TFP [10]+4DGS [7]	0.192	0.196	0.206	0.213	0.220	0.205
Spk2img [8]+4DGS [7]	0.208	0.219	0.218	0.229	0.227	0.220
Ours Spike4DGS	0.189	0.185	0.193	0.199	0.192	0.197

Table 4: The Novel View Synthesis of LPIPS in our synthetic outdoor Dataset.

References

- [1] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.