

1 A Details of Velocity-Controlled Reparameterization

2 In our code, we provide two ways to precisely control speed. These are using continuously defined
3 functions and a discrete speed list.

4 A.1 Continuous speed function

5 A positive, analytic function

$$v : [0, 1] \rightarrow \mathbb{R}_{>0} \quad (\text{dimensionless}),$$

6 sampled at normalised time t , directly prescribes the speed curve. In the released dataset we adopt

$$v(t) = 0.25 \sin(t) + 1.1, \quad t \in [0, 1], \quad (1)$$

7 A.2 Speed list

8 An arbitrary-length float array $\mathbf{r} = \{r_k\}_{k=0}^{L-1}$ ($r_k > 0$) is interpreted as *multipliers* of a base frame
9 rate $f_{\text{base}} = 2400$ fps: the k -th temporal segment $[t_k, t_{k+1}]$ of length $\Delta T = T/L$ is rendered at
10 $f_k = r_k f_{\text{base}}$. To obtain a *continuous* speed curve we blend neighbouring segments with a cubic
11 B-spline¹ in a 2τ -second window centred at each boundary,

$$v(t) = \text{Bspline}(t; \mathbf{r}, \tau), \quad \tau \simeq 0.1 \Delta T.$$

12 A.3 From speed curve to arc-length samples

13 Let M be the desired number of interpolated frames, we sample the chosen speed interface on a
14 uniform grid $t_j = j/(M-1)$:

$$u_j = v(t_j), \quad j = 0, \dots, M-2; \quad (2)$$

$$\Delta s_j = \frac{u_j}{\sum_{k=0}^{M-2} u_k} S; \quad (3)$$

$$s_0 = 0, \quad s_{j+1} = s_j + \Delta s_j. \quad (4)$$

15 Equation (3) rescales the sampled speeds so that $\sum_j \Delta s_j = S$, ensuring the full geometric path is
16 covered.

17 A.4 Evaluating the spline

18 Each interpolated pose $\tilde{P}_j = (\tilde{R}_j, \tilde{\mathbf{T}}_j)$ is obtained by querying the spline at the renormalised
19 arc-length s_j^* :

$$\tilde{P}_j = \mathcal{P}(s_j), \quad j = 0, \dots, M-1.$$

20 Because $s_{j+1} - s_j \propto v(t_j)$, the linear and angular velocities of the discrete trajectory $\{\tilde{P}_j\}$ follow
21 the prescribed speed profile with frame-level accuracy.

22 A.5 Practical remarks

- 23 • **Choice of interface.** The analytic form (1) is convenient for dataset-level consistency; the speed
24 list form offers frame-accurate speed control for bespoke sequences.
- 25 • **Continuity.** Both interfaces yield a C^2 speed curve, hence the final trajectory is at least C^1 ,
26 avoiding jerk during rendering.
- 27 • **Complexity.** The whole pipeline is linear in $N+M$ and is CPU-friendly ($< 0.5 \mu\text{s}$ per interpolated
28 pose).

29 **Summary.** Either a compact analytic law (1) or an arbitrary-length speed list can be mapped, via
30 Eq. (3), to B-spline arc-length samples, providing reliable and precise control over camera velocity
31 for every rendered frame.

¹Order 3 suffices to reach C^2 continuity while keeping local support.

32 B Details of choosing the contrast threshold

33 In our experiments, we found that when we set the contrast threshold $c \leq 0.75$, visible floater
 34 artifacts appeared during the visualization of the event stream. These artifacts occur when the
 35 viewpoint changes and certain Gaussians—originally situated in the background and expected to
 36 be occluded—are mistakenly treated as part of the visible foreground. This misclassification leads
 37 to variations in illumination that induce apparent voltage changes, which the simulator erroneously
 38 interprets as valid event triggers. As a result, the synthesized event stream contains non-physical
 39 textures, manifesting as spurious structures or noise in the visualization. As shown in the figure 1,
 40 once we raise c to 1 or higher, the floater becomes almost invisible.

41 It is worth noting that when the contrast threshold is set too low, according to the research results in
 42 Da4event, it will lead to a loss of dynamic range. Therefore, in this paper, we tend to set a larger
 43 c to solve both problems simultaneously. To ensure that events are not overly sparse and sufficient
 44 information integrity is retained, the GS2E dataset was simulated with the parameter setting $c = 1$.

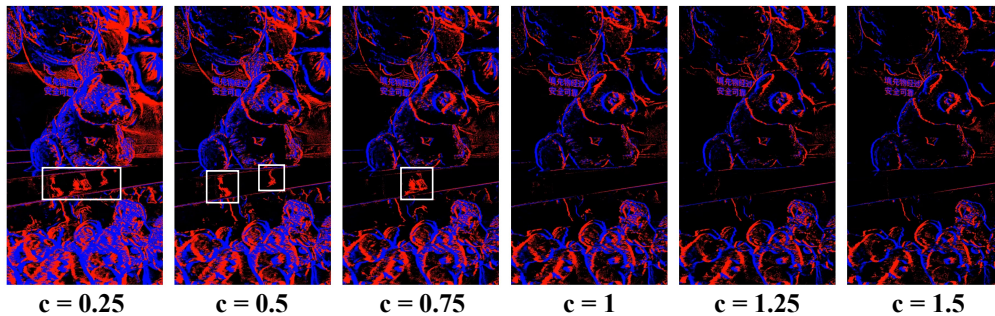


Figure 1: Selecting the same viewpoint and time window(1000 us), visualize events simulated from 3DGS with different contrast threshold(c) values. The results show that when $c \leq 0.75$, error events generated by floater Gaussians can be seen on the integral event diagram, while this phenomenon is greatly alleviated when $c \geq 1$.

45 C Implementation Details

46 For the reconstruction stage, we carefully collect approximately 2k high-quality multi-view im-
 47 ages from 2 public datasets: MVImageNet and DL3DV: we choose and render 1.8k scenes from
 48 MVImageNet and 100 scenes from DL3DV. We employ the official interplementation version of
 49 3DGS in the original setting. For the event generation stage, we utilize the DVS-Voltmeter simulator
 50 to synthesize events from the rendered RGB sequences. We adopt the following sensor-specific
 51 parameters to closely mimic the behavior of real DVS sensors: the ON and OFF contrast thresh-
 52 olds are both set to $\Theta_{\text{ON}} = \Theta_{\text{OFF}} = 1$ as default. The dvs camera parameters are calibrated as
 53 $k_1 = 0.5, k_2 = 1e-3, k_3 = 0.1, k_4 = 0.01, k_5 = 0.1, k_6 = 1e-5$, following the original DVS-
 54 Voltmeter setting. These control the brightness-dependent drift μ and variance σ^2 of the stochastic
 55 process, which determine the polarity distribution and the inverse-Gaussian timestamp sampling for
 56 each event.

57 All events are simulated at 2400 FPS temporal resolution and stored with microsecond timestamps
 58 for high-fidelity spatio-temporal alignment. The overall process are conducted on a workstation
 59 equipped with 8×NVIDIA RTX 3090 GPUs. The selected MVImageNet clip images vary in size,
 60 but most are approximately 1080p in resolution. When training 3DGS on MVImageNet, each
 61 scene takes an average of 16 minutes. For the camera pose upsampling and trajectory control
 62 stage, using an interpolation factor of $\gamma = 5$, the strategy `ada_speed`, and the velocity function
 63 $v(t) = 0.25 \sin(t) + 1.1$, the average runtime per scene is approximately 45 seconds.

64 During event simulation, we adopt the same camera parameter configuration as mentioned previously.
 65 However, the simulation time varies significantly depending on the motion amplitude and speed of
 66 the camera, as well as the scene complexity, making it difficult to estimate a consistent runtime.

67 For the DL3DV dataset, each scene contains 300–400 images. To ensure higher reconstruction and
 68 rendering quality, as well as to generate longer event streams, we do not downsample the input image

69 resolution, nor do we slice the image or event sequences. Using the same hardware configuration
 70 as with MVImageNet, the average per-scene training time is approximately 27 minutes, and the
 71 rendering time is around 41 minutes.

72 **D Existing Event-based 3D Reconstruction Datasets**

73 To contextualize the contribution of GS2E, Table 1 provides a comprehensive comparison of existing
 74 event-based 3D datasets and 3D reconstruction methods. We categorize these into **static scenes** and
 75 **dynamic scenes**, based on whether the underlying geometry remains constant or involves temporal
 76 variation.

77 **Attributes.** Each dataset is evaluated along key axes:

- 78 • **Data Type:** Whether sharp and/or blurry RGB frames are provided. Blurry frames support
 79 deblurring tasks, while sharp ones aid in geometry fidelity.
- 80 • **Scene Num / Scale:** Number of distinct scenes and their spatial scope (object-level vs.
 81 medium/large indoor scenes).
- 82 • **GT Poses:** Availability of ground-truth camera extrinsics.
- 83 • **Speed Profile:** Whether camera motion follows uniform or non-uniform velocity.
- 84 • **Multi-Trajectory:** Whether each scene supports multiple trajectory simulations, enabling
 85 consistent multi-view observations.
- 86 • **Device:** Capture source—real event sensors (e.g., DAVIS346C, DVXplore) or simulated
 87 streams (e.g., ESIM, Vid2E, V2E).
- 88 • **Data Source:** Origin of the base scene data (e.g., NeRF renderings, Blender, Unreal Engine,
 89 or real-world scenes).

90 **Key Findings.** We observe that existing datasets are limited in several aspects:

- 91 • Most datasets focus on small-scale, object-centric scenes with limited spatial or temporal
 92 diversity.
- 93 • Simulators typically use simplified trajectories and fixed contrast thresholds, which constrain
 94 realism.
- 95 • Real event data remains scarce and often lacks consistent trajectory coverage or paired
 96 ground truth.
- 97 • Multi-trajectory support is rare, impeding evaluation under view-consistency and generaliza-
 98 tion settings.

99 **Positioning of GS2E.** Our proposed GS2E benchmark is designed to address these limitations by:

- 100 • Leveraging 3D Gaussian Splatting to reconstruct photorealistic static scenes from sparse
 101 real-world RGB inputs.
- 102 • Generating controllable, dense virtual trajectories with adaptive speed profiles and multiple
 103 interpolated paths per scene.
- 104 • Synthesizing events via a physically-informed simulator that incorporates realistic contrast
 105 threshold modeling.
- 106 • Supporting both object- and scene-level scales with consistent multi-view alignment and
 107 temporal density.

108 By filling the gaps in scale, realism, and trajectory diversity, GS2E enables more robust evaluation of
 109 event-based 3D reconstruction and rendering methods.

E Limitation and Broader impacts

Limitation. While GS2E provides high-fidelity, geometry-consistent event data under a wide range of camera trajectories and motion patterns, it remains fundamentally limited by its reliance on rendered RGB images from 3DGS. Specifically, the current pipeline inherits the photometric constraints of 3D Gaussian Splatting, which may not faithfully replicate extreme illumination conditions such as overexposure or underexposure. As a result, scenes with very low light or high dynamic range may not be accurately modeled in terms of event triggering behavior. Additionally, our framework currently assumes static scenes; dynamic object motion is not yet modeled. In future work, we plan to extend the simulator by incorporating physically-realistic camera models into the 3DGS rendering pipeline, enabling explicit control over exposure, tone mapping, and sensor response curves to better approximate real-world lighting variability.

Broader impacts. This work introduces a scalable, geometry-consistent synthetic dataset for event-based vision research. On the positive side, it lowers the barrier for training high-performance models in domains such as autonomous driving, robotics, and augmented reality, where event-based sensing offers advantages under fast motion or challenging lighting. By providing a flexible, physically-grounded simulation framework, the work supports reproducible and ethical AI development. On the negative side, improved realism in synthetic event data may inadvertently enable misuse such as generating adversarial inputs or synthetic surveillance data. These risks are mitigated by the dataset’s academic licensing and transparency in its construction pipeline. Furthermore, the data generation framework may raise privacy concerns if adapted for real-scene reproduction, which warrants further community discussion and the adoption of usage safeguards.

F License of the used assets

- **3D Gaussian Splatting:** A publicly available method with its dataset released under the CC BY MIT license.
- **MVImgNet:** A publicly available dataset released under the CC BY 4.0 license.
- **DL3DV:** A publicly available dataset released under the CC BY 4.0 license.
- **GS2E:** A publicly available dataset released under the CC BY MIT license.

Method	Column	Type	Color Frame		Scene Num	Scene Scale	GT poses	Speed	Multi-Trajectory	Device	Data Source
			Blurry	Sharp							
Static Scenes											
Event-NeRF ?	CVPR 2023	Synthetic	×	✓	7	object	✓	Uniform	×	Blender+ESIM	NeRF
E2NeRF ?	ICCV 2023	Synthetic	✓	✓	7	object	✓	Uniform	×	Blender+ESIM	NeRF
		Real	✓	×	5	medium&large	×	Uniform	×	DAVIS346C	Author Collection
Robust e-NeRF ?	ICCV 2023	Synthetic	✓	✓	7	object	✓	Non-Uniform	✓	Blender+ESIM	NeRF
Deblur e-NeRF ?	ECCV 2024	Synthetic	×	✓	7	object	✓	Non-Uniform	✓	Blender+ESIM	NeRF
EvaGaussian ?	Arxiv 2024	Synthetic	✓	✓	9	medium&large	✓	Uniform	✓	Blender+ESIM	NeRF, Deblur-NeRF +Author Collection
		Real	✓	×	5	medium&large	×	Uniform	×	DAVIS346C	Author Collection
PAEv3D ?	ICRA 2024	Real	×	✓	6	object	×	Uniform	×	DVXplore event camera	Author Collection
EvDeblurF	CVPR 2024	Synthetic	✓	✓	4	medium	✓	Uniform	×	Blender+ESIM	Deblur-NeRF
		Real	✓	×	5	medium	×	Uniform	×	DAVIS346C	Author Collection
EvGGS ?	ICML 2024	Synthetic	×	×	64	object	✓	Uniform	×	Blender+V2E	Author Collection
IncEventGS ?	CVPR 2025	Synthetic	×	✓	6	large	✓	Uniform	×	Vid2E	UnrealEgo
E-3DGS ?	3DV 2025	Synthetic	×	✓	3	medium	✓	Non-Uniform	×	-	Tanks and Temples
AE-NeRF ?	AAAI 2025	Synthetic	×	✓	3	medium	✓	Non-Uniform	×	-	Author Collection
EF-3DGS ?	Arxiv 2024	Synthetic	×	✓	9	large	✓	Uniform	×	-	Replica
LSE-NeRF ?	Arxiv 2024	Real	✓	✓	10	medium&large	✓	Non-Uniform	-	Prophesee EVK-3 HD + Blackfly S (GigE)	Author Collection
GS2E	Submission	Synthetic	✓	✓	1150	medium&large	✓	Non-Uniform	✓	3DGS + DVS-Volmetre	Author Collection
Dynamic Scenes											
DE-NeRF ?	ICCV 2023	Synthetic	×	✓	3	object	✓	Uniform	×	Blender+ESIM	Author Collection
		Real	×	✓	6	medium&large	✓	Uniform	×	Samsung DVS Gen3, DAVIS 346C	Color Event Camera, HS-ERGB(TimesTen)
EvDNeRF ?	CVPR 2024 Workshop	Synthetic	×	×	3	object	✓	Uniform	✓	Blender+ESIM	Kubric
		Synthetic	×	✓	5	object	✓	Uniform	✓	Blender+ESIM	Author Collection
Dynamic EventNeRF ?	CVPR 2025 Workshop	Real	×	✓	16	medium&large	✓	Uniform	✓	DAVIS346C	Author Collection

Table 1: Comparison of existing event-based 3D reconstruction datasets, categorized by scene type, motion profile, sensor modality, and simulation pipeline.