E-3DGS: Event-Based Novel View Rendering of Large-Scale Scenes Using 3D Gaussian Splatting

Sohaib Zahid[1,2] Viktor Rudnev[1,2] Eddy Ilg[1] Vladislav Golyanik[2]
[1]Saarland University [2]MPI for Informatics, SIC

Abstract

Novel view synthesis techniques predominantly utilize RGB cameras, inheriting their limitations such as the need for sufficient lighting, susceptibility to motion blur, and restricted dynamic range. In contrast, event cameras are significantly more resilient to these limitations but have been less explored in this domain, particularly in large-scale settings. Current methodologies primarily focus on front-facing or object-oriented (360-degree view) scenarios. For the first time, we introduce 3D Gaussians for event-based novel view synthesis. Our method reconstructs large and unbounded scenes with high visual quality. We contribute the first real and synthetic event datasets tailored for this setting. Our method demonstrates superior novel view synthesis and consistently outperforms the baseline EventNeRF by a margin of $11-25\%$ *in PSNR (dB) while being orders of magnitude faster in reconstruction and rendering.*

1. Introduction

Novel view synthesis offers a fundamental approach to visualizing complex scenes by generating new perspectives from existing imagery. This has many potential applications, including virtual reality, movie production and architectural visualization [27]. An emerging alternative to the common RGB sensors are event cameras, which are bio-inspired visual sensors recording events, i.e. asynchronous per-pixel signals of changes in brightness or color intensity.

Event streams have very high temporal resolution and are inherently sparse, as they only happen when changes in the scene are observed. Due to their working principle, event cameras bring several advantages, especially in challenging cases: they excel at handling high-speed motions and have a substantially higher dynamic range of the supported signal measurements than conventional RGB cameras. Moreover, they have lower power consumption and require varied storage volumes for captured data that are often smaller than those required for synchronous RGB cameras [5, 19].

The ability to handle high-speed motions is crucial in static scenes as well, particularly with handheld moving cameras, as it helps avoid the common problem of motion blur. It is, therefore, not surprising that event-based novel view synthesis has gained attention, although color values are not directly observed. Notably, because of the substantial difference between the formats, RGB- and event-based approaches require fundamentally different design choices.

The first solutions to event-based novel view synthesis introduced in the literature demonstrate promising results [12, 25] and outperform non-event-based alternatives for novel view synthesis in many challenging scenarios. Among them, EventNeRF [25] enables novel-view synthesis in the RGB space by assuming events associated with three color channels as inputs. Due to its NeRF-based architecture [17], it can handle single objects with complete observations from roughly equal distances to the camera. It furthermore has limitations in training and rendering speed: the MLP used to represent the scene requires long training time and can only handle very limited scene extents or otherwise rendering quality will deteriorate. Hence, the quality of synthesized novel views will degrade for larger scenes.

We present Event-3DGS (E-3DGS), i.e., a new method for novel-view synthesis from event streams using 3D Gaussians [9] demonstrating fast reconstruction and rendering as well as handling of unbounded scenes. The technical contributions of this paper are as follows:

- With E-3DGS, we introduce the first approach for novel view synthesis from a color event camera that combines 3D Gaussians with event-based supervision.
- We present frustum-based initialization, adaptive event windows, isotropic 3D Gaussian regularization and 3D camera pose refinement, and demonstrate that high-quality results can be obtained.
- Finally, we introduce new synthetic and real event datasets for large scenes to the community to study novel view synthesis in this new problem setting.

Our experiments demonstrate systematically superior results compared to EventNeRF [25] and other baselines. The source code and dataset of E-3DGS are released[1].

[1] 4dqv.mpi-inf.mpg.de/E3DGS/

2. Related Work

2.1. Novel View Synthesis from RGB Inputs

Novel view synthesis of rigid scenes is predominantly handled assuming RGB inputs. A widely used approach to this problem is to learn coordinate-based neural scene representations allowing rendering novel views at test time. Earlier works such as Neural Radiance Fields (NeRF) and its direct follow-ups [17, 27] used implicit neural representations in combination with volume rendering. They are based on expensive-to-optimize Multi-Layer Perceptrons (MLPs) and are slow at training and evaluation while requiring a relatively low amount of storage space once they are trained. Their stochastic ray sampling requires many samples to obtain an accurate scene approximation, and shooting rays through empty space constitutes unnecessary overhead. Most of these approaches focus on single objects or bounded scenes. Recent techniques accelerate neural MLP-based representations or ray sampling [20, 24] or avoid MLPs [2, 4, 26] by using voxel grids. Some techniques [3] support unbounded scenes by employing radial basis functions, thereby overcoming the limitations of voxel-grid-based methods. Several ray tracing-based methods support large-scale scenes and uncontrolled camera trajectories thanks to progressive NeRF optimization [16, 29]. Instant-NGPs [20] are neural feature volumes with a hash grid that can be learned and evaluated quickly at test time. They can also handle multi-scale training scenarios efficiently.

A promising recent development is the shift from ray tracing to rasterization, marked by the introduction of 3D Gaussian Splatting (3DGS) [9]. This approach presents an alternative paradigm for 3D reconstruction and novel view synthesis using differentiable rasterization with 3D Gaussians as geometric primitives. Since GPU technology and algorithmic research have evolved over several decades to provide high performance for rasterization applications, 3DGS trains substantially quicker and provides much higher rendering throughput than NeRF. Moreover, since it explicitly represents the geometry, it can scale easily as the scene size increases with no special handling required for unbounded scenes. Our approach adopts the 3D Gaussian representation and presents its application to the supervision from event streams. It inherits thereby the advantages of event streams and 3DGS for view synthesis.

2.2. Novel View Synthesis from Event Streams

Event-aided sparse odometry and simultaneous localization and mapping approaches are distantly related to our setting, as they do not allow photo-realistic and dense rendering of novel views [7, 10, 13, 22].

As previously discussed, event cameras represent an alternative to RGB sensors for dense novel view synthesis, and some initial work was done on learning 3D scene representations from event streams only. EventNeRF [25] is a seminal framework for training MLP-based implicit 3D representations (see Sec. 2.1) using frames of accumulated color events. While it demonstrates impressive results, it is restricted to camera trajectories with uniform motion and the assumption that the background is a constant color (triggering no events). E-NeRF [12] is another work that resembles the training methodology of EventNeRF for single-channel (intensity) event cameras and allows training a colored 3D representation from a combination of blurry RGB images and grayscale events. Robust E-NeRF by Low and Lee [14] is a model aiming to reduce the issues caused by uncontrolled camera motion. They introduce the refractory period to the event generation model, i.e. the time during which a pixel is inactive after an event firing. Supervision happens on the level of individual events, and they reformulate the event loss to handle intra-pixel variances of the contrast threshold optimized during training. All these methods adopt ray tracing and can be primarily applied on 360° object-centric datasets or front-facing trajectories.

Our approach differs from previous event-based methods in that it demonstrates that rasterization can be efficiently combined with event-based supervision instead of ray tracing. The main design choices of our method are tailored to 3D Gaussians. As a result, our method inherits the primary advantages of 3DGS [9], such as fast training and inference. Similar to EventNeRF [25], our method supports color. However, in contrast, it is not limited to single objects and can handle large-scale scenes.

3. Preliminaries

3.1. 3D Gaussians

3D Gaussian Splatting [9] is a high-quality and efficient scene representation. The Gaussians are defined by a 3D covariance matrix $\boldsymbol{\Sigma}_i$ centered around a point $\boldsymbol{\mu}_i$:

$$G_i(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right), \quad (1)$$

and their overlay models the geometry at scene location $\boldsymbol{x}$. Each Gaussian is additionally associated with an opacity o_i and spherical harmonics that model view-dependent color. For rendering purposes, the means $\boldsymbol{\mu}_i$ and covariance matrices $\boldsymbol{\Sigma}_i$ are transformed into image coordinates. The projected matrix $\boldsymbol{\Sigma}_i'$ can be obtained by applying the viewing transformation $\boldsymbol{W}$ and the Jacobian $\boldsymbol{J}$ of the affine approximation of the projective transformation:

$$\boldsymbol{\Sigma}_i' = \boldsymbol{J}\boldsymbol{W}\boldsymbol{\Sigma}_i\boldsymbol{W}^T\boldsymbol{J}^T. \quad (2)$$

The third row and column of $\boldsymbol{\Sigma}_i'$ are dropped to obtain a 2D matrix. Using Equation 1, one can then evaluate the different Gaussians i that overlap with an image pixel x and

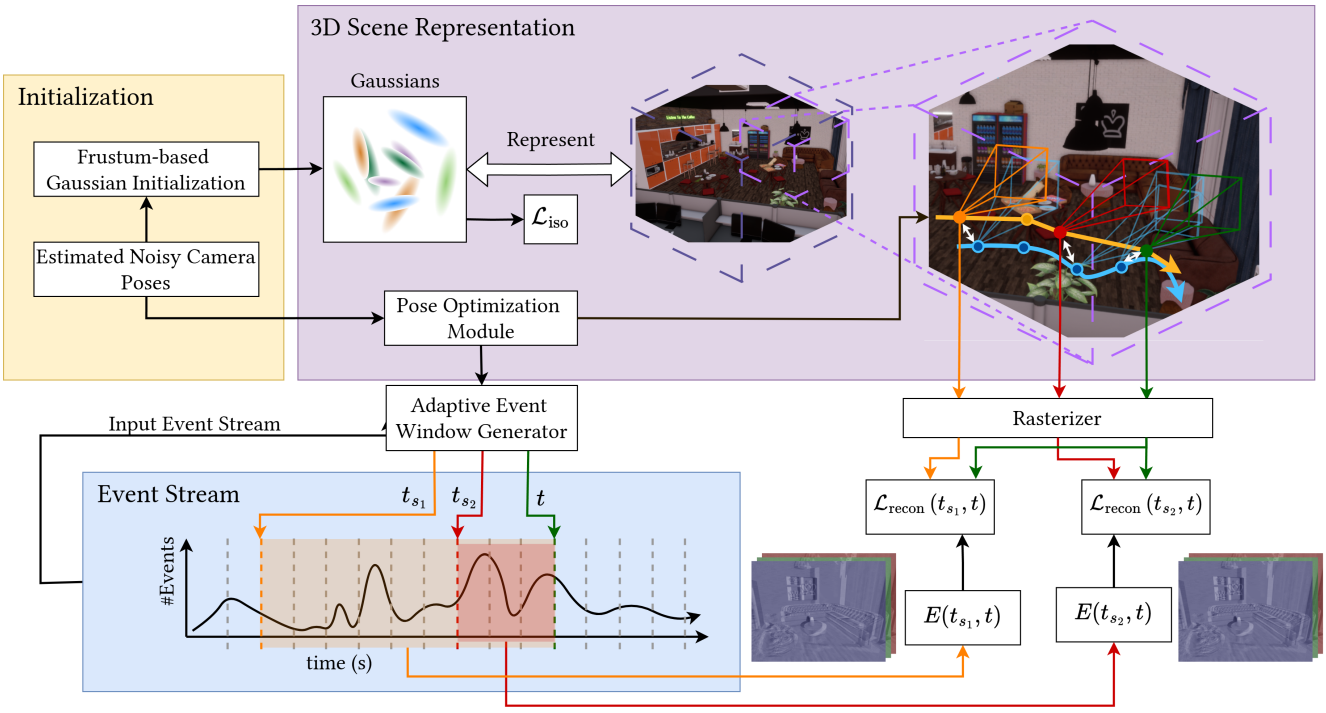

Figure 1. **Overview of our E-3DGS Method.** We use 3D Gaussians [9] as the scene representation and assume that initial noisy camera poses are available. We randomly initialize the scene with our frustum-based initialization (Sec. 4.2) and then optimize the Gaussians and the camera poses jointly (Sec. 4.5). To obtain a high-quality reconstruction of both, low-frequency structure and high-frequency detail, we propose a strategy using a large event window from t_{s_1} to t and a small one from t_{s_2} to t (Sec. 4.3). We then define the loss $\mathcal{L}_{\text{recon}}$ (Sec. 4.6) between renderings from our model at the current time t (indicated green) and previous times t_{s_1} (indicated orange) and t_{s_2} (indicated red), and the accumulated incoming events $E(t_{s_1}, t)$ and $E(t_{s_2}, t)$. We regularize the 3D Gaussians with the loss $\mathcal{L}_{\text{iso}}$ (Sec. 4.4).

obtain alpha values as $\alpha_{i,x} = o_i G'_i(\boldsymbol{x})$. The Gaussians are then sorted according to their depth, and alpha blending for every pixel is performed by combining the view-dependent colors c_i using the following equation:

$$C_x = \sum_{i=1}^{N} T_{i,x} \alpha_{i,x} c_i \,, \tag{3}$$

where $T_{i,x} = \prod_{k=1}^{i-1}(1-\alpha_{k,x})$ represents the transmittance.

3.2. Event Formation Model

Event cameras generate a continuous stream of events denoted as $e = (\boldsymbol{x}, p, \tau)$, where $\boldsymbol{x}$ are the pixel coordinates at which an event is triggered at time τ, and $p \in \{-1, +1\}$ signifies the polarity of the event, indicating an increase or decrease in the logarithmic intensity by the predefined contrast threshold Δ. Thus, the relationship between the triggered event and the logarithmic image intensity reads:

$$L_{\boldsymbol{x}}(\tau) - L_{\boldsymbol{x}}(\tau^{\text{prev}}) = p\Delta, \tag{4}$$

where τ^{prev} is the time when the previous event for the pixel was triggered. This concept can then be generalized to apply for an accumulation of events within a time interval $(\tau_1, \tau_2) | \tau_1 < \tau_2$ for a pixel location $\boldsymbol{x}$ as follows:

$$L_{\boldsymbol{x}}(\tau_2) - L_{\boldsymbol{x}}(\tau_1) = \sum_{\tau_1 < \tau_t \leq \tau_2} p_t \Delta \stackrel{\text{def}}{=} E_{\boldsymbol{x}}(t_1, t_2) \,, \tag{5}$$

where t_1, t_2 index the sequence of events closest to τ_1, τ_2.

4. The E-3DGS Method

Our aim is to learn a 3D representation of a static scene using only a color event stream, where each pixel observes changes in brightness corresponding to one of the red, green, or blue channels according to a Bayer pattern, with known camera intrinsics $K_t \in \mathbb{R}^{3\times3}$, and noisy initial poses $P_t \in \mathbb{R}^{3\times4}$, at reasonably high-frequency time steps indexed by t. Following 3DGS [9], we represent our scene by anisotropic 3D Gaussians. Our methodology comprises a technique to initialize Gaussians in the absence of a Structure from Motion (SfM) point cloud, adaptive event frame supervision of 3DGS, and a pose refinement module. An overview of our method is provided in Fig. 1.

Our E-3DGS method is not restricted to scenes of a certain size and can handle unbounded environments. It

does not rely on any assumptions regarding the background color, type of camera motion, or speed. Thus, it ensures robust performance across a wide range of scenarios.

4.1. Event Stream Supervision

There are two main categories of approaches to learning 3D scene representations from event streams. Some apply the loss to single events [14] based on Eq. (4). Others use the sum of events $E_{\boldsymbol{x}}(t_1, t_2)$ from Eq. (5). We choose the second approach, as rasterization in 3DGS is well suited to efficiently render entire images rather than individual pixels.

To optimize our Gaussian scene representation using event data, we can make a logical equivalence between the observed event stream and the scene renderings. To do so, we replace the true logarithmic intensities $L_{\boldsymbol{x}}$ in Eq. (5) with the rendered logarithmic intensities $\hat{L}_{\boldsymbol{x}}$ from our scene, and the times τ with the camera poses P_t that were used to render the scene at the respective time steps. Following the approach used in [25], the log difference is then point-wise multiplied with a Bayer filter F to obtain the respective color channel. We can finally calculate the error between the logarithmic change from our model and the actual change observed from the event stream, and define the following per-pixel loss:

$$\mathcal{L}_{\boldsymbol{x}}(t_1, t_2) = \left\| F \odot (\hat{L}_{\boldsymbol{x}}(P_{t_2}) - \hat{L}_{\boldsymbol{x}}(P_{t_1})) - F \odot E_{\boldsymbol{x}}(t_1, t_2) \right\|_1, \tag{6}$$

where "$\odot$" denotes pixelwise multiplication.

4.2. Frustum-Based Initialization

In the original 3DGS [9], the Gaussians are initialized using a point cloud obtained from applying SfM on the input images. The authors also experimented with initializing the Gaussians at random locations within a cube. While this worked for them with a slight performance drop, it requires an assumption about the extent of the scene.

Applying SfM directly to event streams is more challenging than RGB inputs [10] and exploring this aspect is not the primary focus of this paper. In the absence of an SfM point cloud, we use the randomly initialized Gaussians and extend this approach to unbounded scenes. To this end, we initialize a specified number of Gaussians (on the order of 10^4) in the frustum of each camera. This gives two benefits: 1) All the initialized Gaussians are within the observable area, and 2) We only need one loose assumption about the scene, which is the maximum depth z_{far}.

4.3. Adaptive Event Window

Rudnev et al. [25] demonstrated in EventNeRF that using a fixed event window duration results in suboptimal reconstruction. They find that larger windows are essential for capturing low-frequency color and structure, and smaller ones are essential for optimization of finer high-frequency details. While they randomly sampled the event window duration, a drawback is that it does not consider the camera speed and event rate, thus the sampled windows may contain too many or too few events. As our dataset features variable camera speeds, we improve upon this by sampling the number of events rather than the window duration. To achieve this, for each time step we randomly sample a target number of events from within the range $[N_{\min}, N_{\max}]$. Given a time step t, we search for a previous time step t_s such that the number of events in the event frame $E(t_s, t)$ is approximately equal to the desired number.

When determining $N_{\max}$, we find that for values where details and low-frequency structure are optimal, 3DGS tends to get unstable and sometimes prunes away Gaussians in homogeneous areas. While this can be mitigated by choosing a much larger $N_{\max}$, this again deteriorates the details. Therefore, we propose a strategy to incorporate both, small and large windows. For each t, we choose two earlier time steps t_{s_1} and t_{s_2}. The ranges for sampling the event counts for both are empirically chosen to be $[\frac{N_{\max}}{10}, N_{\max}]$ and $[\frac{N_{max}}{300}, \frac{N_{\max}}{30}]$. We then render frames from our model at times t, t_{s_1} and t_{s_2}, and use two concurrent losses for the event windows $E_{\boldsymbol{x}}(t_{s_1}, t)$ and $E_{\boldsymbol{x}}(t_{s_2}, t)$.

4.4. As-Isotropic-As-Possible Regularization

In 3DGS, Gaussians are unconstrained in the direction perpendicular to the image plane. This lack of constraint can result in elongated and overfitted Gaussians. And while they may appear correct from the training views, they introduce significant artifacts when rendered from novel views by manifesting as floaters and distortions of object surfaces. We also observe that the lack of multi-view consistency and tendency to overfit destabilize the pose refinement.

To mitigate these issues, we draw inspiration from Gaussian Splatting SLAM [15] and SplaTAM [8], and apply isotropic regularization:

$$\mathcal{L}_{\text{iso}} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \left\| S_g - \bar{S}_g \right\|_1, \tag{7}$$

where $\mathcal{G}$ is the set of Gaussians visible in the image. Eq. (7) imposes a soft constraint on the Gaussians to be as isotropic as possible. We find that it helps to improve pose refinement, minimizes floaters and enhances generalizability.

4.5. Pose Refinement

To obtain the most accurate results, we allow the poses to be refined during optimization by modeling the refined pose as $P'_t = P^e_t P_t$, where P^e_t is an error correction transform. Instead of directly optimizing P^e_t as a 3×3 matrix, following Hempel et al. [6] we represent it as $[r_1\ r_2\ T]$, where r_1 and r_2 represent two rotation vectors of the rotation matrix $R = [r_1\ r_2\ r_3]$, while T is the translation. We can then

obtain the P_t^e matrix from the representation using Gram-Schmidt orthogonalization (see details in Supplement II), hence ensuring that during optimization, our error correction transform always represents a valid transformation matrix. P_t^e is initialized to be the identity transform. Since the loss function from Eq. (6) depends on the camera pose as well, it allows us to use the same loss to backpropagate and obtain gradients for pose refinement.

As our goal is to refine the estimated noisy poses rather than perform SLAM, this training signal is sufficient for our needs. Moreover, we observe that poses tend to diverge with 3DGS due to the periodic opacity reset. To combat this, we impose a soft constraint with an additional pose regularization, that encourages the matrices P_t^e to stay close to the identity matrix I:

$$\mathcal{L}_{\text{pose}} = \|P^e_{t_{s_1}} - I\|_2 + \|P^e_{t_{s_2}} - I\|_2 + \|P^e_t - I\|_2\,, \quad (8)$$

with all terms weighted equally.

4.6. Optimization

Eq. (6) defines the reconstruction loss per pixel for a single event frame. However, naively averaging these per-pixel losses over whole images leads to problems. For small event windows, most pixels have no events, which are not very informative but will then make up the majority of the loss. To address this, we compute separate averages of the losses for pixels with events $\mathcal{X}_{\text{evs}}$ and pixels without events $\mathcal{X}_{\text{noevs}}$. These averages are then scaled by the hyperparameter $\alpha = 0.3$ to obtain the complete weighted reconstruction loss:

$$\begin{aligned}\mathcal{L}_{\text{recon}}\left(t_s, t\right) = \frac{\alpha}{|\mathcal{X}_{\text{noevs}}|} \cdot \left(\sum_{\boldsymbol{x}\in\mathcal{X}_{\text{noevs}}} \mathcal{L}_{\boldsymbol{x}}\left(t_s, t\right)\right) + \\ + \frac{1-\alpha}{|\mathcal{X}_{\text{evs}}|} \cdot \left(\sum_{\boldsymbol{x}\in\mathcal{X}_{\text{evs}}} \mathcal{L}_{\boldsymbol{x}}\left(t_s, t\right)\right).\end{aligned} \quad (9)$$

To obtain the final loss, we take a weighted sum of the reconstruction losses for the two event windows from Sec. 4.3 along with the isotropic and pose regularization:

$$\begin{aligned}\mathcal{L} = \lambda_1\mathcal{L}_{\text{recon}}\left(t_{s_1}, t\right) + \lambda_2\mathcal{L}_{\text{recon}}\left(t_{s_2}, t\right) \\ + \lambda_{\text{iso}}\mathcal{L}_{\text{iso}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}}\end{aligned}, \quad (10)$$

where λ_1, λ_2 and λ_{iso} are hyper-parameters. In our experiments, we use $\lambda_1 = \lambda_2 = 0.65$, and λ_{iso} is set to 10 initially and reduced to 1 after 10^4 iterations.

5. Experimental Evaluation

5.1. Implementation details

We provide the full implementation details in the supplemental material. Running our method on a scene takes one to two hours (depending on the scene size) with a single NVIDIA GeForce RTX 3090.

(a) (b)

Figure 2. Two different views of the scene with inanimate objects assembled in the multi-view studio of MPI for Informatics.

5.2. Datasets

We next describe the new event datasets we provide to analyze large-scale scenes, along with the existing datasets that we use in the experiments.

E-3DGS-Real. Our real dataset was captured within a studio environment. The scene consists of a diverse set of objects, as shown in Fig. 2. We used a DAVIS346C color event camera to capture our scene with a resolution of 346×260. The contrast threshold settings were kept at their default values, which are symmetric. We capture multiple clips of the scene, each roughly $60-120$s long with varying motion characteristics and levels of scene coverage. The captured data consists of the event stream and RGB images at 2.5 frames per second. The studio is equipped with 115 traditional cameras distributed uniformly across the walls and capturing 4K footage at 50 FPS. Similar to the approach of Millerdurai et al. and annotation of the EE3D-R (Real) dataset [18], we use these cameras to estimate and track the camera pose by detecting a checkerboard mounted to the event camera rig, providing tracking data at a frequency of up to 50 Hz. Note that in some timestamps the checkerboard is not detected due to occlusions and thus the 50 Hz is only the best case. The data from the external cameras is relevant for camera pose estimation, but cannot be used as ground truth because of the significantly different perspectives from the training views.

E-3DGS-Synthetic. For creating the synthetic dataset, we choose three scenes of UnrealEgo [1]. We rendered 60s clips of each scene at 1000 FPS. The scenes contain large-scale environments and exhibit various types of surfaces, including reflections. We noticed that a few of the small highly reflective objects (e.g., metallic rods) cause unnatural aliasing in the renders, so we changed them to use diffuse materials. The event generation model from Sec. 4 was used to simulate event data from these high-fidelity frames. While we had access to pose data 1000 Hz, we downsampled it to 50 Hz to simulate a real-world setting in which the poses are estimated from externally captured RGB images.

E-3DGS-Synthetic-Hard. This dataset is designed specifically to highlight and rigorously evaluate the key contribu-

tions of our method during the ablation study. To assess the significance of our pose refinement module—which cannot be quantitatively evaluated on the E-3DGS-Real dataset—we introduce artificial noise into the E-3DGS-Synthetic dataset, which is carefully matched to the one observed in real data (see Supplement III for details). This allows us to assess the performance of our pose refinement module effectively. In addition to introducing noise, we also address the issue of camera speed variation. While the camera speed in the E-3DGS-Synthetic dataset generally stays within a narrow range, this does not fully test the capabilities of our adaptive event windows. To create a more challenging scenario, we varied the camera speed sinusoidally, with a ratio between its maximum and minimum speed of 100. This modification enables a more comprehensive evaluation of our adaptive event windows.

TUM-VIE. This dataset consists of recordings from a Prophesee Gen4 sensor [11]. RGB views from an externally calibrated camera are also provided. The camera extrinsics are tracked at 120 Hz. Two of the recordings have been used in Robust E-NeRF [14]; we train our method on these recordings, namely `mocap-1d-trans` and `mocap-desk2` to compare with Robust E-NeRF. However, as also argued in Low and Lee [14], these recordings are not well suited for novel view synthesis since the captures are predominantly front-facing, with some small displacements either in circles or from side to side.

EventNeRF Datasets. EventNeRF [25] provides 360° object-centric event data, which we use to show that our method also outperforms previous methods on object-centric data. To be consistent with the original work, we evaluate our method on poses that are a part of the training trajectory instead of novel views, for our evaluation metrics to be comparable to theirs. We train our method on the synthetic sequences to perform the quantitative comparison. In these experiments, the background color is set to $159/255$, following the original paper [25].

5.3. Evaluation Metrics

For E-3DGS-Real dataset, the RGB frames are of too low quality to be used for evaluation purposes, and, therefore, we only perform qualitative comparisons. With TUM-VIE, as suggested in Robust E-NeRF [14], it is not trivial to do the tone mapping correctly. Therefore, we do quantitative evaluation only with the synthetic datasets. For the evaluation on synthetic data, keeping in line with the previous literature, we adopt the following evaluation metrics:

- Peak Signal-to-Noise Ratio (PSNR);
- Learned Perceptual Image Patch Similarity (LPIPS) [31];
- Structural Similarity Index Measure (SSIM).

5.3.1 Color Correction

As our method only learns logarithmic differences rather than absolute color intensities, there is an ambiguity in the reconstructed color balance and illumination of the scene. Hence, color needs to be adjusted, as otherwise, the evaluation metrics will be less meaningful. We correct predicted images using the following equation:

$$L'_c = L' + (\mathbb{E}[L] - \mathbb{E}[L']) \,, \tag{11}$$

where L'_c is the color corrected logarithmic image and "$\mathbb{E}[\cdot]$" is the expectation operator. Eq. (11) is applied separately to each color channel, which effectively aligns the per-channel logarithmic means of the predicted images with the ground-truth ones. Since in the synthetic setting, we already know the exact contrast threshold, there is no need for correcting the scale of the image as done in some previous works [14, 25]. Since we lack reference images for the real dataset, neither evaluation nor color correction is applicable to it. However, some minor color and contrast adjustments are manually made for better visualization.

5.4. Comparisons to Related Methods

RGB-Based Methods. We train Deblur-GS [28] on blurry RGB images from our E-3DGS-Real dataset to establish a reference using RGB inputs. We also convert the event stream to images using E2VID [23] and apply 3DGS (referred to as "E2VID + 3DGS"). This method is evaluated on all E-3DGS datasets. To train both methods, we interpolate the camera poses at discrete time steps provided by the external tracking system, which is necessary because the pose timestamps do not align with the frame timestamps. We use Spherical Linear Interpolation (SLERP) for the rotations and Linear Interpolation (LERP) for the translations to obtain the camera poses for the images.

Event-Based Methods. For comparison with event-based methods, we train EventNeRF [25] on all E-3DGS datasets. To adapt it for our datasets, we normalize the camera poses within a unit sphere and following NeRF++ [30] added a background network to model areas outside the sphere, as the scene extent is unknown. Furthermore, the maximum event window length is increased by the factor of 10 to aid convergence (up to one second). We do not train our method on the synthetic dataset provided by Robust E-NeRF [14], as it is designed for extremely long refractory periods that are not observed in other datasets. However, we compare their method to ours on two sequences from TUM-VIE in Fig. 3, namely `mocap-1d-trans` and `mocap-desk2`.

5.4.1 Observations

The results of all evaluations are reported in Tables 1–2 and Figs. 3–6. As visible, our method consistently outperforms

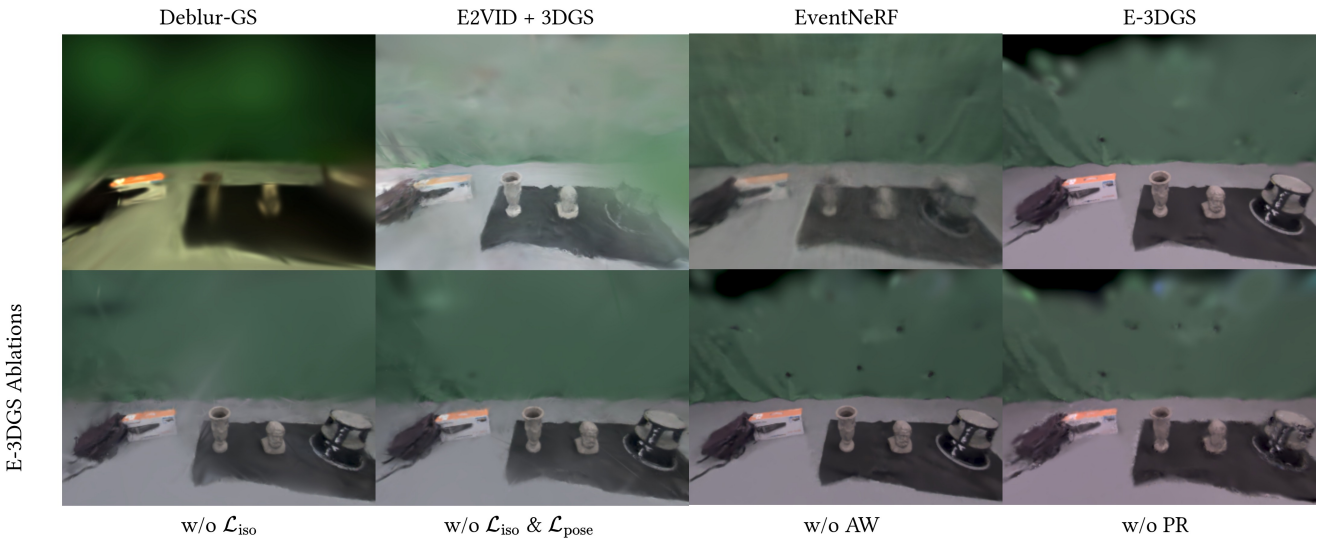

Figure 3. Comparison of E-3DGS against the baselines and ablation study on the E-3DGS-Real dataset. Deblur-GS, E2VID + 3DGS and EventNeRF suffer from various issues including blurring, floaters, and noise. In contrast, our method delivers clear details, such as the intricate structure of the sculpture's face.

Method	Company			ScienceLab			Subway			Average		
	↑PSNR	↓LPIPS	↑SSIM	↑PSNR	↓LPIPS	↑SSIM	↑PSNR	↓LPIPS	↑SSIM	↑PSNR	↓LPIPS	↑SSIM
EventNeRF [25]	19.59	0.41	0.65	17.22	0.46	0.60	18.71	0.34	0.67	16.80	0.50	0.61
E2VID [23] + 3DGS [9]	9.79	0.37	0.48	11.86	0.38	0.54	9.79	0.40	0.43	10.48	0.38	0.49
E-3DGS (ours)	**20.78**	**0.29**	**0.72**	**18.41**	**0.28**	**0.73**	**19.92**	**0.20**	**0.74**	**19.70**	**0.26**	**0.73**

Table 1. Comparison of several methods on the E-3DGS-Synthetic dataset: We outperform the baselines by a large margin in all cases. Furthermore, E2VID + 3DGS shows lower PSNR but achieves better LPIPS than EventNeRF due to E2VID's frame reconstruction, which has poor color consistency but an adequate level of edge details (see Fig. 6). Green and yellow are the best and the second-best, respectively.

Scene	EventNeRF [25]			E-3DGS (ours)		
	↑PSNR	↓LPIPS	↑SSIM	↑PSNR	↓LPIPS	↑SSIM
Chair	**30.62**	0.05	0.94	30.42	**0.03**	**0.95**
Drums	27.43	0.07	0.91	**31.07**	**0.03**	**0.95**
Ficus	31.94	0.05	0.94	**34.08**	**0.02**	**0.96**
Hotdog	30.26	0.04	0.94	**30.79**	**0.03**	**0.96**
Lego	25.84	0.13	0.89	**30.74**	**0.04**	**0.94**
Materials	24.10	0.07	0.94	**33.73**	**0.02**	**0.97**
Mic	31.78	0.03	0.96	**35.87**	**0.02**	**0.98**
Average	28.85	0.06	0.93	**32.39**	**0.03**	**0.96**

Table 2. Comparisons on the synthetic EventNeRF dataset. Our method demonstrates significant improvements over EventNeRF across all evaluation metrics.

the baselines both on synthetic and real data. In the EventNeRF object-centric datasets, our method shows clear superiority across almost all evaluation metrics. The only exception is a marginally lower PSNR score on the "Chair" scene, as detailed in Table 2. The general performance advantage is further backed by the qualitative results in Fig. 4, where our method produces more accurate reconstructions.

Similarly, on the E-3DGS-Synthetic dataset, E-3DGS significantly surpasses both EventNeRF and E2VID+3DGS by a wide margin; see Table 1. The qualitative results on the E-3DGS-Real dataset, highlighted in Fig. 3, further demonstrate our method's superior performance: Deblur-GS struggles with excessive blur; EventNeRF suffers from noise due to ray sampling and memory constraints, and E2VID+3DGS exhibits noisy Gaussians and floaters.

While Robust E-NeRF achieves higher local contrast, it struggles with global brightness consistency due to single-event training; see Fig. 5. Our E-3DGS maintains consistent brightness across the scene, with only a slight reduction in local contrast. Note that we can observe some holes and floaters near the outer peripheries in Figs. 3 and 5. These effects are due to out-of-bound areas at the edges of the observations that occur as a result of the undistortion of the event stream.

5.5. Ablation Studies

To evaluate the effects of individual contributions, we do extensive qualitative and quantitative ablation studies. We primarily train different variants of our method on the E-3DGS-Real and E-3DGS-Synthetic-Hard datasets, focusing on the effects of four key components: $\mathcal{L}_{iso}$, $\mathcal{L}_{pose}$, **P**ose **R**efinement (PR), and the **A**daptive Event **W**indow (AW).

For the ablation experiments without adaptive window,

Components				Company			ScienceLab			Subway			Average		
$\mathcal{L}_{\text{iso}}$	$\mathcal{L}_{\text{pose}}$	PR	AW	↑ PSNR	↓ LPIPS	↑ SSIM	↑ PSNR	↓ LPIPS	↑ SSIM	↑ PSNR	↓ LPIPS	↑ SSIM	↑ PSNR	↓ LPIPS	↑ SSIM
✓	✓	✓	✓	20.742	**0.404**	**0.661**	**18.823**	**0.414**	**0.677**	18.923	**0.436**	**0.619**	**19.496**	**0.418**	**0.652**
	✓	✓	✓	20.519	0.434	0.631	18.099	0.454	0.631	19.401	0.475	0.601	19.340	0.454	0.621
✓	✓		✓	20.229	0.539	0.606	17.646	0.587	0.601	18.746	0.620	0.569	18.874	0.582	0.592
✓	✓	✓		20.667	0.427	0.642	18.354	0.440	0.657	18.742	0.440	0.606	19.254	0.436	0.635
	✓	✓		**20.845**	0.441	0.623	17.792	0.472	0.616	**19.475**	0.469	0.600	19.371	0.460	0.613
		✓	✓	19.834	0.537	0.583	17.317	0.577	0.571	18.111	0.605	0.532	18.421	0.573	0.562

Table 3. Ablation study on the E-3DGS-Synthetic-Hard dataset. The overall tendency is that the performance declines when one of the components is removed, confirming their contribution to the overall performance. Notably, E-3DGS without AW consistently ranks second, while omitting L_{iso} often results in third place or close. (PR: **P**ose **R**efinement, AW: **A**daptive Event **W**indow). Green, yellow, and orange indicate the best, second-best, and third-best results, respectively.

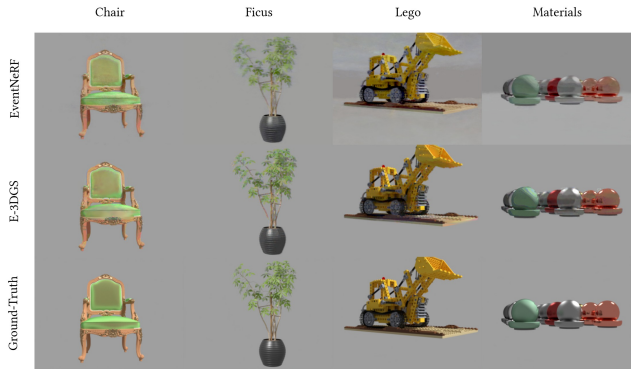

Figure 4. Comparison of E-3DGS vs. EventNeRF on the synthetic EventNeRF dataset. EventNeRF struggles with noise in the Drums sequence, blurriness in Ficus, and background artifacts in Lego and Materials sequences, while E-3DGS handles these issues well.

Figure 5. Comparison of E-3DGS vs. Robust E-NeRF on the TUM-VIE dataset. While Robust E-NeRF achieves higher local contrast, it suffers from globally inconsistent brightness. E-3DGS produces consistent brightness across the scene, albeit with some detail loss (e.g., in the table texture of the mocap-desk2 sequence).

we use a maximum time interval T_{max} instead of maximum events N_{max} to sample the event windows. The value of T_{max} is computed from N_{max}, such that the average event window size remains approximately similar.

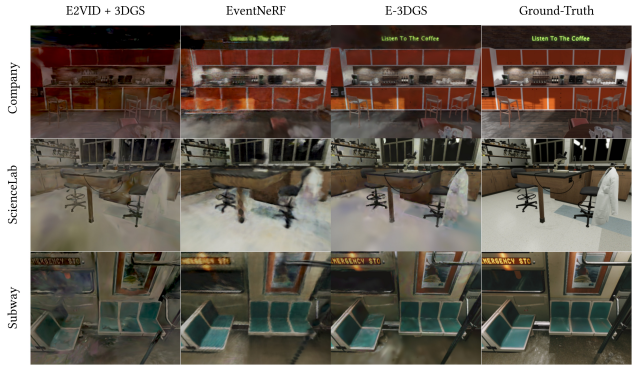

Figure 6. Comparison of E-3DGS vs. baselines on the E-3DGS-Synthetic dataset. E2VID + 3DGS struggles with poor color reconstruction but captures edges and structure reasonably well. EventNeRF suffers from noise and a lack of sharpness. In contrast, our method delivers clear details and accurate colors, with only minor issues in certain areas (such as the coat on a chair in the ScienceLab sequence. Best viewed with zoom.

The results are reported in Table 3 and Figs. 3 and 6. Removing L_{iso} results in a noticeable performance drop, but removing L_{iso} and L_{pose} jointly leads to a much more significant decline. This is likely because L_{pose} prevents pose divergence in unstable conditions, while removal of L_{iso} causes instability due to overfitting. Similar effects could occur when combining L_{iso} with pose refinement.

6. Conclusion

We show that E-3DGS effectively combines the strengths of 3D Gaussian splatting and event-based supervision for 3D reconstruction and novel view synthesis of large-scale scenes. It significantly outperforms the baselines quantitatively and qualitatively, while being orders of magnitude faster. One aspect beyond the scope of this paper is lifting the requirement for camera pose initialization through an external process. We believe this work paves the way for robust and scalable large-scale scene reconstruction utilizing the advantages of event cameras to capture details in challenging conditions, such as low light and fast motion.

References

[1] Hiroyasu Akada, Jian Wang, Soshi Shimada, Masaki Takahashi, Christian Theobalt, and Vladislav Golyanik. Unrealego: A new dataset for robust egocentric 3d human motion capture. In *ECCV*, pages 1–17. Springer, 2022. 5

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 2

[3] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *ICCV*, pages 4182–4194, 2023. 2

[4] Fridovich-Keil and Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[5] Guillermo Gallego et al. Event-based vision: A survey. *IEEE TPAMI*, 44(01):154–180, 2022. 1

[6] Thorsten Hempel, Ahmed A. Abdelrahman, and Ayoub Al-Hamadi. Toward robust and unconstrained full range of rotation head pose estimation. *IEEE TIP*, 33:2377–2387, 2024. 4, 10

[7] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza. Event-aided direct sparse odometry. In *CVPR*, 2022. 2

[8] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *CVPR*, 2024. 4, 11

[9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4), 2023. 1, 2, 3, 4, 7, 11, 12

[10] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *ECCV*, 2016. 2, 4

[11] Simon Klenk, Jason Chui, Nikolaus Demmel, and Daniel Cremers. Tum-vie: The tum stereo visual-inertial event dataset. In *IEEE/RSJ IROS*, pages 8601–8608. IEEE, 2021. 6

[12] Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers. E-nerf: Neural radiance fields from a moving event camera. *IEEE RA-L*, 8(3):1587–1594, 2023. 1, 2

[13] Simone Klenk, Marvin Motzet, Lukas Koestler, and Daniel Cremers. Deep event visual odometry. In *3DV*, 2024. 2

[14] Weng Fei Low and Gim Hee Lee. Robust e-nerf: Nerf from sparse & noisy events under non-uniform motion. In *ICCV*, pages 18335–18346, 2023. 2, 4, 6

[15] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian Splatting SLAM. In *CVPR*, 2024. 4, 11

[16] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H. Kim, and Johannes Kopf. Progressively optimized local radiance fields for robust view synthesis. In *CVPR*, 2023. 2

[17] 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. 1, 2

[18] Christen Millerdurai, Hiroyasu Akada, Jian Wang, Diogo Luvizon, Christian Theobalt, and Vladislav Golyanik. Eventego3d: 3d human motion capture from egocentric event streams. In *CVPR*, 2024. 5

[19] Christen Millerdurai, Diogo Luvizon, Viktor Rudnev, André Jonas, Jiayi Wang, Christian Theobalt, and Vladislav Golyanik. 3d pose estimation of two interacting hands from a monocular event camera. In *3DV*, 2024. 1

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 41(4):102:1–102:15, 2022. 2

[21] Grigorios A Pavliotis. Stochastic processes and applications. *Texts in applied mathematics*, 60, 2014. 11

[22] Henri Rebecq, Timo Horstschaefer, Guillermo Gallego, and Davide Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE RA-L*, 2:593–600, 2017. 2

[23] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE TPAMI*, 2019. 6, 7, 12

[24] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *ICCV*, 2021. 2

[25] Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. Eventnerf: Neural radiance fields from a single colour event camera. In *CVPR*, 2023. 1, 2, 4, 6, 7, 11, 12

[26] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *CVPR*, pages 5449–5459, 2022. 2

[27] Ayush Tewari et al. Advances in Neural Rendering. *Eurographics*, 2022. 1, 2

[28] Chen Wenbo and Liu Ligang. Deblur-gs: 3d gaussian splatting from camera motion blurred images. *I3D*, 7(1), 2024. 6, 12

[29] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *ECCV*, pages 106–122, 2022. 2

[30] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 6

[31] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6