EventGS: Event-based 3D Gaussian Splatting SLAM with Diffusion Model

Author Names Omitted for Anonymous Review.

***Abstract*— 3D Gaussian Splatting (3DGS) has recently advanced dense SLAM with frame cameras, yet frame cameras degrade severely under high dynamic range (HDR) conditions and fast motions. Event cameras offer high temporal resolution and wide dynamic range, making them promising for robust perception in such extreme conditions; however, an effective event-only 3DGS-SLAM solution is still absent. To bridge this gap, we present EventGS, the first near real-time 3DGS SLAM system that performs pose tracking and incremental 3DGS reconstruction directly from monocular event streams. EventGS improves observability by deriving two complementary observations from events: diffusion-restored intensity images that provide stable grayscale appearance for rendering supervision, and event-accumulated change images that provide log-intensity-change observations to enforce change-domain temporal consistency. Moreover, we propose an on-demand progressive refinement strategy that leverages diffusion priors to repair under-constrained regions caused by large viewpoint changes, thereby enhancing tracking robustness under aggressive motions. Experiments on both real-world and synthetic datasets demonstrate that EventGS achieves accurate pose estimation and high-quality renderable mapping using only event-stream input.**

I. Introduction

Dense simultaneous localization and mapping (SLAM) is essential for robotics applications such as navigation, detection, augmented reality, and manipulation. Recently, 3D Gaussian Splatting (3DGS) [1] has been rapidly adopted in SLAM, enabling high-fidelity, differentiable, and efficiently renderable maps [2]. However, RGB-camera-based 3DGS-SLAM often degrades under fast ego-motion and high dynamic range (HDR) scenes, where frames suffer from motion blur and over-/under-exposure. These effects break photometric consistency and destabilize feature matching, leading to degraded tracking and mapping. Developing sensing and learning solutions that remain reliable under such extreme conditions is therefore still an open problem.

Event cameras asynchronously report per-pixel brightness changes with microsecond latency and HDR. Compared with frames, event streams are naturally resistant to motion blur and can robustly capture edge and motion cues in challenging lighting, making them promising for odometry and reconstruction [3], [4].

While event-based odometry has achieved strong pose accuracy [3], most methods do not provide high-quality, directly renderable dense maps. Existing event-based 3D reconstruction is often offline [4], [5], or relies on additional sensors [6]. Moreover, many approaches rely mainly on change-domain supervision [7], [8], whose constraints weaken in low-texture regions, low event-rate settings, or noisy streams, causing unstable optimization. Furthermore, these factors weaken the rendering supervision required by 3DGS-SLAM, making stable deployment of event cameras in 3DGS-SLAM considerably more challenging. Meanwhile, existing 3DGS-SLAM systems are prone to local holes and accumulated artifacts under large viewpoint changes, which weakens subsequent rendering supervision and can further lead to tracking failures. As a result, an online event-only 3DGS-SLAM system that is both stable and produces high-quality renderable maps remains underexplored.

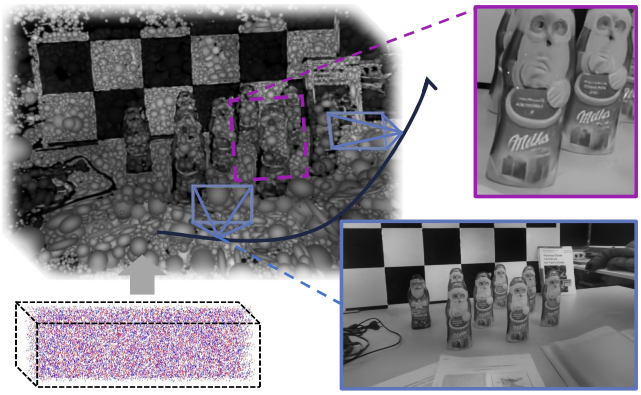

Fig. 1: An example of **EventGS** mapping, where EventGS takes event streams as input and achieves high-quality Gaussian reconstruction.

To bridge this gap, we propose **EventGS**, which, to the best of our knowledge, is the first near real-time 3DGS-SLAM system using only a monocular event camera. EventGS follows a tracking–mapping framework that jointly optimizes camera poses and a 3DGS map within a sliding window, and improves stability via complementary supervision in the appearance and change domains. In the appearance domain, a fine-tuned diffusion model restores grayscale intensity from events and feeds it into a rendering loss for dense photometric constraints. In the change domain, we accumulate events into an event-accumulated change image and enforce temporal consistency via a change-consistency loss. Furthermore, to address tracking degradation caused by large motions, we propose an on-demand progressive refinement strategy that leverages a diffusion prior to locally repair and denoise novel-view observations, thereby improving observation quality and enhancing the robustness of subsequent tracking. Experiments on multiple real and simulated datasets show that EventGS matches or outperforms state-of-the-art methods in both trajectory accuracy and map quality, with ablations validating each module.

In summary, our contributions are as follows:

- We propose **EventGS**, the first near real-time 3D Gaus-

Manuscript 2349 submitted to 2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 5, 2026.

sian Splatting SLAM system using only a monocular event camera.
- We propose a dual-domain supervision scheme that combines rendering supervision on diffusion-restored intensity and temporal consistency on event-accumulated change images, and further introduce a progressive refinement strategy for robust event-only 3DGS-SLAM.
- We conduct comprehensive evaluations and ablation studies on multiple real and simulated datasets, showing that our method matches or outperforms prior approaches in both trajectory accuracy and map quality.

II. Related Work

A. 3DGS SLAM

3DGS [1] has recently become a practical SLAM representation that jointly models geometry and appearance [9]. In monocular settings, MonoGS [2] achieves online SLAM using 3DGS as the sole scene representation. SplaTAM [10] achieves accurate tracking and high-fidelity reconstruction via online optimization and submap mechanisms. Splat-SLAM [11] introduces a deformable 3D Gaussian map that adapts online to loop closure and global bundle adjustment. Nevertheless, most 3DGS-SLAM pipelines still rely on frame-based RGB/RGB-D observations and are vulnerable to blur and exposure failures under fast motion and HDR. In contrast, event-based odometry achieves accurate motion estimation under fast motion and HDR [3], [12], [13], but typically estimates only trajectories and does not maintain high-quality, directly renderable dense maps. Existing attempts such as EGS-SLAM [6] typically fuse events with RGB-D, and an event-only online 3DGS-SLAM system remains largely unexplored.

B. Event-based 3DGS Reconstruction

Compared with event-based NeRF approaches [7], [14], [15], event-based 3DGS typically offers more efficient rendering and more direct optimization of an explicit scene representation. Event-3DGS [4] optimizes Gaussians directly in the event domain and leverages photovoltage estimation to improve robustness to noise. EvGGS [5] proposes a co-learning framework that jointly learns intensity, depth, and Gaussian representation, enabling Gaussian reconstruction from pure event inputs. EventSplat [8] initializes with an event-to-video prior and improves poses via trajectory interpolation to improve rendering quality. IncEventGS [16] adopts an incremental reconstruction framework, but its per-iteration optimization cost hinders real-time operation. Moreover, many event-stream reconstruction methods rely primarily a on change-domain supervision pathway [7], [8], [16], which can become unreliable under low-texture scenes, low event rates, or high noise. Although Event-3DGS [4] reconstructs grayscale frames from event streams for rendering supervision, the limited reconstruction quality still makes it difficult to handle these challenging scenarios. Overall, existing event-based reconstruction is still largely offline, and achieving robust online SLAM with stable closed-loop operation remains challenging.

C. Diffusion Priors for Event Vision and 3DGS

Diffusion models, as strong generative priors, have been used for intensity restoration in event vision. HDRev-Diff [17] conditions a diffusion model on events and an accompanying LDR frame for HDR reconstruction. In 3DGS, diffusion priors are often applied to repair novel-view renderings and feed the improvements back to the 3D representation: Difix3D+ [18] uses a single-step diffusion enhancer to remove rendering artifacts, while GSFix3D [19] performs diffusion-guided view repair and distills the refinements into Gaussian primitives. Moreover, DiET-GS [20] combines event streams with diffusion priors for 3DGS motion deblurring to improve color and fine details. Despite these advances, providing geometrically consistent and reliable restoration supervision remains challenging in pure event-stream settings without known poses.

III. Method

As illustrated in Fig. 2, EventGS consists of three modules: Event Observation Construction (§III-B), Online Pose Tracking (§III-C) and Incremental Gaussian Mapping (§III-D). Taking a continuous event stream as input, the system runs online to jointly estimate the camera trajectory and incrementally maintain a 3DGS map.

A. 3D Gaussian Scene Reconstruction

We represent the scene using 3DGS. Using anisotropic 3D Gaussian primitives, 3DGS performs differentiable splatting to the image plane, which supports gradient-based refinement of camera poses and Gaussian parameters. Each Gaussian is parameterized by its mean μ, covariance Σ, appearance c, and opacity α. Given a camera pose $T \in SE(3)$ and intrinsics K, a 3D Gaussian is transformed to the camera frame and projected to a 2D elliptical Gaussian on the image plane. Using the Jacobian J of the perspective projection, the 2D covariance is approximated as

$$\Sigma^{2D} = J R_T \Sigma R_T^\top J^\top, \tag{1}$$

where R_T is the rotation component of T. For any pixel x, we compute the pixel-wise effective opacity $\alpha(x)$ from the projected 2D Gaussian, and composite all Gaussians front-to-back using alpha blending. The rendered intensity is

$$I^r(x) = \sum_j c_j \, \alpha_j(x) \prod_{k<j} \big(1 - \alpha_k(x)\big). \tag{2}$$

B. Event Observation Construction

Relying only on log-intensity-change supervision can be under-constrained in low-texture, low event-rate, or large viewpoint-change cases. To improve robustness, we construct two complementary observations from events: diffusion-restored intensity for appearance-domain rendering supervision, and event-accumulated change image for change-domain temporal constraints.

Manuscript 2349 submitted to 2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 5, 2026.

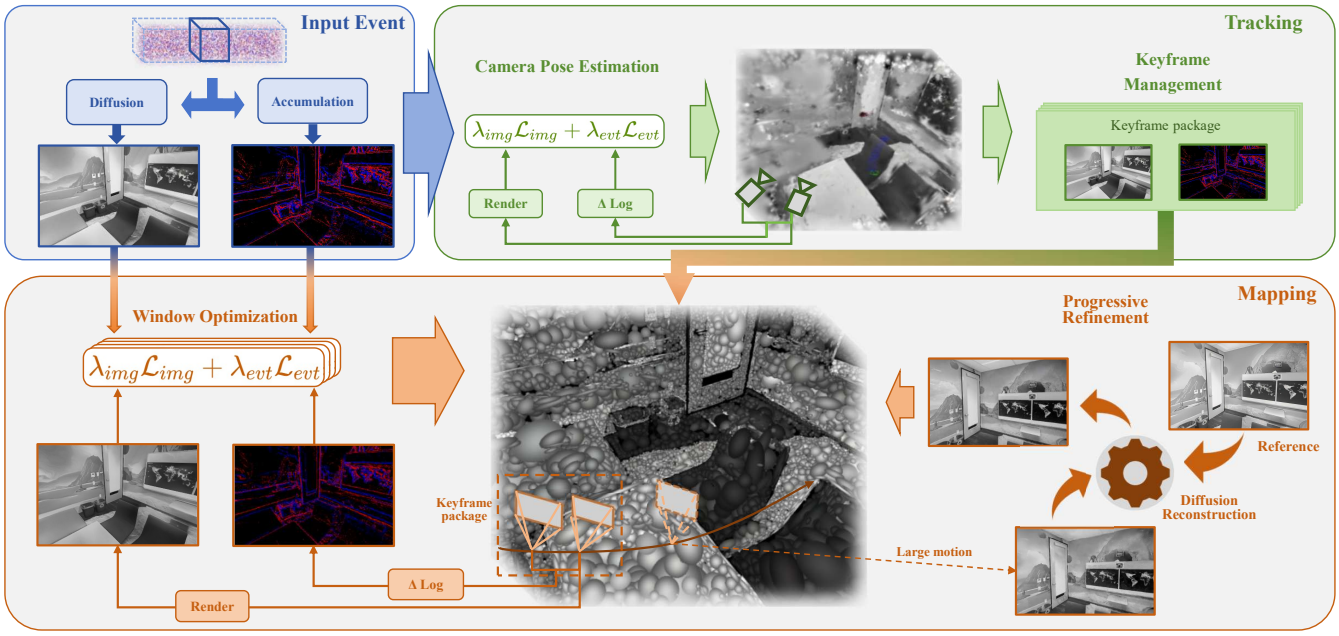

Fig. 2: **System overview.** Given a monocular event stream, EventGS constructs two complementary observations: diffusion-restored intensity for appearance-domain rendering supervision and event-accumulated change image for change-domain temporal constraints. A tracking thread estimates the current pose on the 3DGS map using both losses, while a mapping thread jointly optimizes the 3DGS map and keyframe poses within a sliding window. Progressive refinement is triggered under large motion, restoring a coarse novel-view rendering using diffusion priors and feeding the refined observation back to mapping to improve robustness and mitigate cascading failures.

1) Diffusion-based Intensity Restoration: Event cameras output asynchronous and sparse brightness-change events, where each event is denoted as $e_i = (\mathbf{x}_i, t_i, p_i)$ with pixel location $\mathbf{x}_i = (u_i, v_i)$, timestamp t_i, and polarity $p_i \in \{+1, -1\}$. Directly reconstructing intensity images from event streams often suffers from noise, discontinuities, and artifacts; therefore, we introduce a diffusion-based intensity restoration method. Given the event stream within a time window $\mathcal{E}_t$, we first obtain a coarse grayscale initialization using an event-to-image network [21] $I_t^{deg} = f_{\text{init}}(\mathcal{E}_t)$. While I_t^{deg} captures basic structure and edges, it often contains blurring and artifacts, and is thus treated as a degraded input. We then build a one-step diffusion restoration network f_θ to restore it with optional reference views I_t^{ref}:

$$\hat{I}_t = f_\theta(I_t^{deg}, I_t^{\text{ref}}). \tag{3}$$

Our diffusion restorer serves two purposes: (i) restoring the event-initialized intensity image, and (ii) repairing the coarse novel-view rendering in progressive refinement. In both cases, the corresponding image is treated as a degraded input, and we use reference views only in progressive refinement. As shown in Fig. 3, we use a latent diffusion model with SD-Turbo [22]-style one-step denoising for efficiency, conditioned on event-initialized degraded intensity and a reference view during training. We concatenate the target and reference views along the view dimension and encode them with a VAE encoder $E(\cdot)$ to obtain a latent representation

$$z = E\big([I_t^{deg}, I_t^{\text{ref}}]\big) \in \mathbb{R}^{V\times C\times H\times W}, \tag{4}$$

where V is the number of input views (one target plus references), and C, H, W denote the latent channels and spatial resolution. A U-Net denoiser then performs one-step restoration in the latent space. Inspired by [18], [23], we incorporate reference mixing in self-attention to explicitly fuse cross-view information. Finally, the restored latent of the target view is decoded by a VAE decoder $D(\cdot)$ to obtain $\hat{I}_t$ as the appearance observation.

We observe that the degraded inputs resemble intermediate diffusion states at a low noise level. We thus fix a low-noise diffusion timestep τ for one-step restoration during training and inference, aligning the inputs with the pre-trained diffusion prior at that noise level and balancing artifact removal and content preservation. The training loss:

$$\mathcal{L}_{\text{diff}} = \mathcal{L}_{\text{Recon}} + \mathcal{L}_{\text{LPIPS}} + \lambda_{\text{Gram}}\mathcal{L}_{\text{Gram}}, \tag{5}$$

where $\mathcal{L}_{\text{Recon}} = \|\hat{I}_t - I_t^{\text{gt}}\|_2^2$, $\mathcal{L}_{\text{LPIPS}}$ is the perceptual distance, and $\mathcal{L}_{\text{Gram}}$ is a Gram-style loss on VGG features. We initialize our model with the pretrained weights of DIFIX [18] and fine-tune on event dataset, preserving diffusion priors while better suppressing event-specific artifacts and noise. Compared with frame-based cameras that degrade under HDR and fast motion, event-driven intensity restoration leverages the high temporal resolution and wide dynamic range of events to provide more reliable grayscale observations.

2) Event Accumulation for Log-Intensity Change: Under the standard model, an event is triggered when the log-intensity $L(\mathbf{x}, t) = \log I(\mathbf{x}, t)$ changes sufficiently:

$$L(\mathbf{x}_i, t_i) - L(\mathbf{x}_i, t_i^-) = p_i C, \tag{6}$$

where t_i^- is the previous event time at the same pixel and $C > 0$ is the contrast threshold. This implies that events provide discrete integral measurements of log-intensity changes.

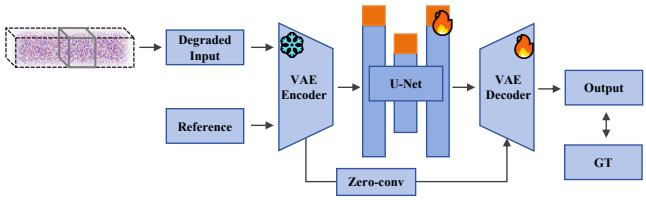

Fig. 3: Training architecture of the diffusion restoration model.

We therefore accumulate events within a short window $[t_a, t_b]$ to form a change-domain observation. Given $\mathcal{E}_{a:b} = \{(\mathbf{x}_i, t_i, p_i) \mid t_a < t_i \le t_b\}$, the event-accumulated change image is

$$\Delta L_{a:b}(\mathbf{x}) = \sum_{(\mathbf{x}_i, t_i, p_i) \in \mathcal{E}_{a:b}} p_i \; \mathbf{1}[\mathbf{x}_i = \mathbf{x}]. \tag{7}$$

Assuming a constant threshold, it is approximately proportional to the true log-intensity change, with a bounded residual due to thresholding and window endpoints:

$$L(\mathbf{x}, t_b) - L(\mathbf{x}, t_a) \approx C \cdot \Delta L_{a:b}(\mathbf{x}). \tag{8}$$

In practice, to avoid imposing incorrect zero-change constraints on pixels with no events, we apply the loss only where event support is sufficient, using a validity mask

$$M_{a:b}(\mathbf{x}) = \mathbf{1}\big[N_{a:b}(\mathbf{x}) \ge \tau_{mask}\big], \tag{9}$$

where $N_{a:b}(\mathbf{x})$ denotes the number of events triggered at pixel $\mathbf{x}$ within the time window $[t_a, t_b]$, τ_{mask} is a minimum event-count threshold. This event accumulation method provides a stable temporal consistency constraint which is jointly enforced with appearance-domain rendering supervision, improving observability and optimization stability.

C. Online Pose Tracking

Given the current event window $[t_a, t_b]$ obtained by slicing the event stream into chunks of N events, we estimate the camera pose $T_{t_b} \in SE(3)$ on a fixed Gaussian map $\mathcal{G}$. In the event observation construction module, we obtain a diffusion-restored intensity observation $\hat{I}_{t_b}$ and an event-accumulated change image $\Delta L_{a:b}$ along with its mask $M_{a:b}$. We then render grayscale intensity images $I^r_{t_b} = \mathcal{R}(\mathcal{G}, T_{t_b})$ and $I^r_{t_a} = \mathcal{R}(\mathcal{G}, T_{t_a})$, where T_{t_a} is taken from the previous window-end estimate and kept fixed during the current pose update. The rendered log-intensity change is

$$\Delta \log I^r_{a:b}(\mathbf{x}) = \log\big(I^r_{t_b}(\mathbf{x}) + \delta\big) - \log\big(I^r_{t_a}(\mathbf{x}) + \delta\big), \tag{10}$$

where δ is a small constant for numerical stability. We use an L_1 loss for appearance-domain rendering supervision:

$$\mathcal{L}_{img} = \sum_{\mathbf{x}} \Big| I^r_{t_b}(\mathbf{x}) - \hat{I}_{t_b}(\mathbf{x}) \Big|. \tag{11}$$

For the change domain, we enforce a squared L_2 log-intensity-change loss:

$$\mathcal{L}_{evt} = \sum_{\mathbf{x}} M_{a:b}(\mathbf{x}) \, \| \Delta \log I^r_{a:b}(\mathbf{x}) - \kappa \Delta L_{a:b}(\mathbf{x}) \|_2^2, \tag{12}$$

where κ is a hyper-parameter that scales the event-accumulated change to match the magnitude of log-intensity change. Tracking updates T_{t_b} by minimizing the joint objective:

$$\min_{T_{t_b} \in SE(3)} \; \lambda_{img} \mathcal{L}_{img} + \lambda_{evt} \mathcal{L}_{evt}. \tag{13}$$

Keyframe selection is based on the co-visibility of the visible Gaussian set and the viewpoint change. We obtain the current visible set V_{t_b} and compute the co-visibility IoU with the visible set V_k of the latest keyframe k:

$$\mathrm{covis}(t_b, k) = \frac{|V_{t_b} \cap V_k|}{|V_{t_b} \cup V_k|}. \tag{14}$$

When $\mathrm{covis}(t_b, k)$ falls below a threshold or the relative pose change exceeds a threshold, we add the current frame as a keyframe. We then encapsulate the observations and state within its associated event window $[t_a, t_b]$ into a *keyframe package* and push it into the mapping sliding window. Specifically, the k-th keyframe package is defined as

$$\mathcal{K}_k \triangleq \Big\{ \hat{I}_k, \; \Delta L_k, \; M_k, \; [t^k_a, t^k_b], \; T^a_k, \; T^b_k \Big\} \tag{15}$$

with $T^a_k \triangleq T_{t^k_a}$, $T^b_k \triangleq T_{t^k_b}$, $\Delta L_k \triangleq \Delta L_{t^k_a : t^k_b}$, $M_k \triangleq M_{t^k_a : t^k_b}$.

D. Incremental Gaussian Mapping

1) Mapping: During mapping, we jointly optimize the Gaussian map $\mathcal{G}$ and the start/end poses (T^a_k, T^b_k) for all keyframe packages $k \in \mathcal{W}$, while periodically inserting and pruning Gaussians to keep the map bounded. Given the current estimates of $\mathcal{G}$ and (T^a_k, T^b_k), we render the grayscale intensity images $I^r_{t^k_a} = \mathcal{R}(\mathcal{G}, T^a_k)$ and $I^r_{t^k_b} = \mathcal{R}(\mathcal{G}, T^b_k)$, and compute the rendered log-intensity change $\Delta \log I^{r,k}_{a:b}$. Analogous to (§III-C), we construct a rendering loss $\mathcal{L}^k_{\text{img}}$ (11) between $I^r_{t^k_b}$ and $\hat{I}_k$, and a log-intensity loss $\mathcal{L}^k_{\text{evt}}$ (12) between $\Delta \log I^{r,k}_{a:b}$ and ΔL_k. We then minimize the weighted sum over $\mathcal{W}$ to update $\mathcal{G}$ and $\{T^a_k, T^b_k\}$:

$$\min_{\mathcal{G}, \{T^a_k, T^b_k\}} \sum_{k \in \mathcal{W}} \Big(\lambda_{img} \mathcal{L}^k_{img} + \lambda_{evt} \mathcal{L}^k_{evt} \Big). \tag{16}$$

In appearance-domain, $\mathcal{L}_{img}$ benefits fine details and rendering-consistent optimization, especially when event constraints become weak in low-texture or low event-rate settings. In change-domain, $\mathcal{L}_{evt}$ is more robust under fast motion and HDR and anchors the optimization to sensor evidence, reducing drift introduced by restored intensities.

2) Progressive Refinement: Under large motion or low observation density, the current Gaussian map often exhibits local holes and accumulated artifacts when rendered from the next viewpoint, which can lead to tracking failures. To mitigate this, we introduce an on-demand progressive refinement strategy. It consists of pose extrapolation, coarse novel-view rendering, diffusion-based refinement, and feeding the restored observation back into mapping.

Let $T_{t-1}, T_t \in SE(3)$ be consecutive poses and $\Delta T_t = T^{-1}_{t-1} T_t$ the relative motion. Define $\xi_t = \log(\Delta T_t) \in \mathfrak{se}(3)$. When tracking detects a large pose increment, i.e., a weighted magnitude $\|\xi_t\| > \tau_\xi$, progressive refinement is

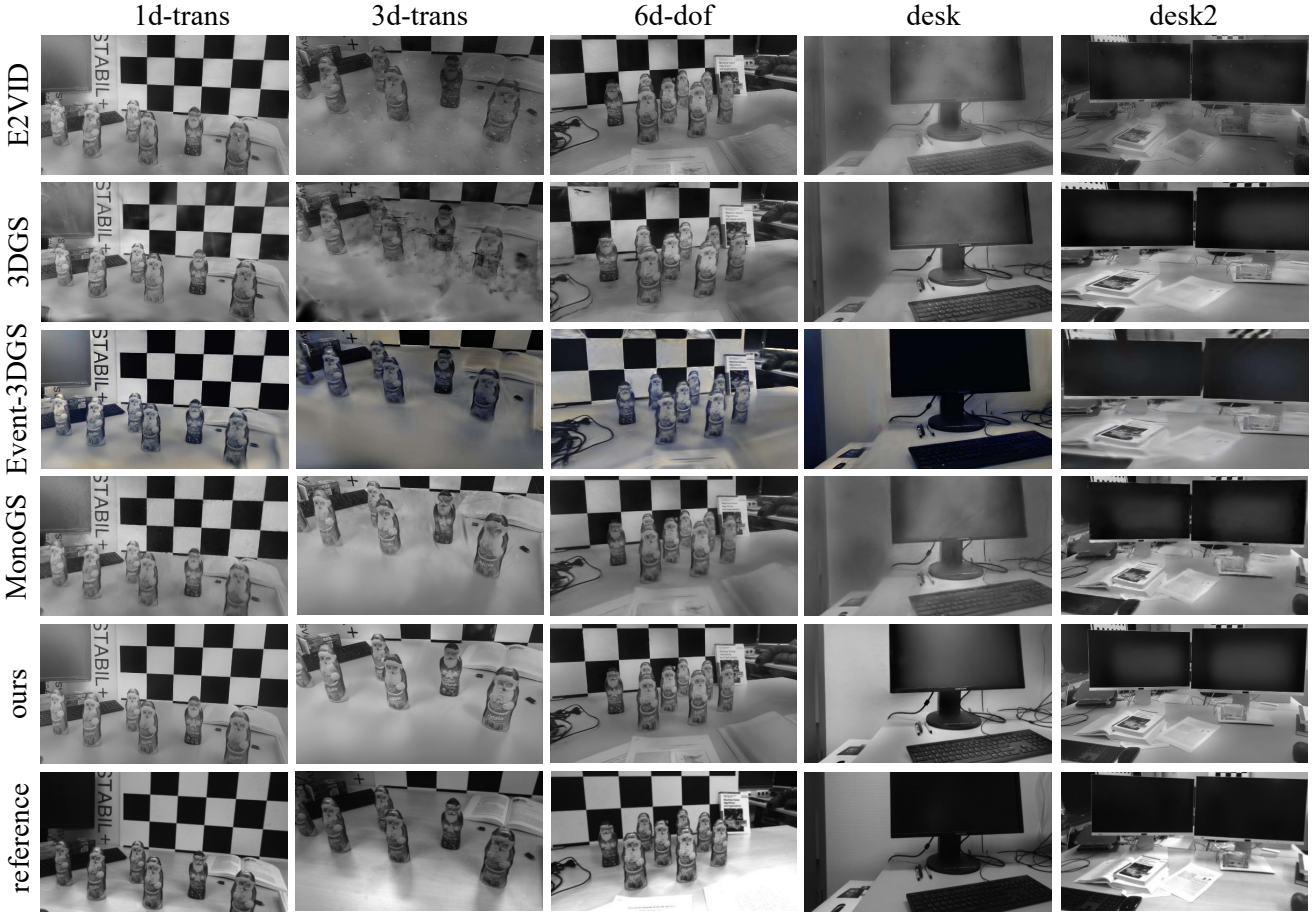

Fig. 4: **Rendering results** on the TUM-VIE [24] dataset. The experimental results demonstrate that our method produces images of higher quality compared to other approaches.

activated. Under a constant-velocity assumption, we extrapolate

$$\tilde{T}_{t+1} = T_t \exp(\xi_t). \tag{17}$$

We render a coarse novel view $\tilde{I}^r_{t+1} = \mathcal{R}(\mathcal{G}, \tilde{T}_{t+1})$, which may contain artifacts. We use the latest appearance observation $\hat{I}_t$ as a reference view I^{ref} and restore the coarse rendering with the conditional diffusion network in §III-B.1:

$$\hat{I}^{prog}_{t+1} = f_\theta(\tilde{I}^r_{t+1}, I^{ref}). \tag{18}$$

The restoration module follows a conditional restoration paradigm: the appearance observation is always used as the reference view to anchor structure and appearance, while the view to be restored provides the viewpoint-specific content scaffold. This strong conditioning substantially reduces degrees of freedom, making the output closer to real observations rather than free-form generation. Finally, we temporarily incorporate $\hat{I}^{prog}_{t+1}$ as an additional intensity rendering supervision signal and iteratively update the map while keeping the window and extrapolated poses fixed:

$$\min_{\mathcal{G}} \sum_{k \in W} \left(\lambda_{img} \mathcal{L}^k_{img}\right) + \lambda_{prog} \mathcal{L}^{prog}_{img}. \tag{19}$$

This refinement repairs unreliable novel-view renderings and distills the information back into the 3DGS map, improving subsequent tracking. Progressive refinement is triggered only in challenging cases, otherwise, the system performs incremental mapping by jointly optimizing $\mathcal{L}_{img}$ and $\mathcal{L}_{evt}$ within the window, keeping the overall computation bounded.

IV. Experiments

In this section, we evaluate our SLAM framework on a variety of real-world and synthetic datasets, and compare our method against existing event-camera-based approaches.

A. Experimental Setup

1) Datasets: We evaluate on both real and synthetic datasets. For real data, we use five sequences from TUM-VIE [24], which provide event streams and grayscale frames under diverse scenes and motions. For synthetic data, we use five event sequences synthesized from Replica [16], [25], as well as synthetic datasets from Event-3DGS [4]. Synthetic data offers controllable trajectories, viewpoints, and ground truth for quantitative evaluation. Since TUM-VIE lacks strictly aligned ground-truth novel views and geometry, we report rendering and geometry metrics on Replica and provide qualitative results on TUM-VIE to reflect real noise and dynamic range.

2) Baselines: We evaluate the method from two aspects. For trajectory estimation, we compare against advanced event-based odometry methods ESVO2 [3], DPVO [26],

TABLE I: Rendering performance on the Replica-based synthetic sequences. Bold numbers indicate the best results, while underlined numbers denote the second-best results.

Method	Office0			Office2			Office3			Room0			Room2		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS [1]	18.91	0.31	0.68	14.03	0.57	0.48	13.25	0.47	0.53	14.45	0.44	0.52	15.74	0.51	0.55
EvGGS [5]	18.51	0.37	0.59	10.95	0.27	0.69	13.13	0.29	0.66	15.16	0.37	0.62	15.85	0.34	0.61
E-NeRF [14]	18.91	0.51	0.57	13.05	0.65	0.44	14.01	0.62	0.48	13.99	0.58	0.51	15.56	0.47	0.58
EventNeRF [7]	18.90	0.43	0.62	15.18	0.66	0.45	16.77	0.73	0.33	17.29	0.62	0.39	16.02	0.54	0.64
Robust e-NeRF [15]	18.93	0.52	0.56	16.81	0.81	0.25	19.22	**0.84**	0.18	17.26	**0.84**	**0.18**	16.43	0.50	0.52
MonoGS	19.35	0.68	0.43	20.03	0.61	0.46	18.28	0.66	0.31	18.14	0.53	0.53	18.15	0.62	0.52
Event-3DGS [4]	20.08	0.71	**0.11**	21.86	0.69	**0.16**	20.23	0.71	**0.11**	19.14	0.63	0.20	20.15	0.67	**0.13**
Ours	**22.43**	**0.75**	0.35	**22.21**	**0.83**	0.23	**21.57**	0.76	0.36	**21.71**	0.70	0.43	**20.60**	**0.71**	0.43

TABLE II: Absolute trajectory error (ATE) on the TUM-VIE mocap sequences (cm).

Method	Modality	1d	3d	6d	desk	desk2	Avg.
MonoGS [2]	Mono VO	2.17	3.66	4.34	5.28	7.04	4.50
ORB-SLAM3 [28]	Mono VO	1.52	2.38	3.95	**1.73**	5.02	2.92
DPVO [26]	Mono EO	2.26	8.32	7.42	5.45	3.93	5.48
ES-PTAM [12]	Stereo EO	**1.05**	8.53	10.25	2.50	7.20	5.91
ESVO [27]	Stereo EO	12.54	17.19	13.46	12.92	4.42	12.11
ESVIO_AA [13]	Stereo EIO	3.86	18.90	*Fail*	8.99	9.47	10.31
ESVO2 [3]	Stereo EIO	3.33	7.26	3.21	6.16	4.02	4.80
Ours	Mono EO	1.39	**1.52**	**1.04**	3.18	**3.04**	**2.03**

ES-PTAM [12], ESVO [27], and ESVIO_AA [13], as well as representative RGB-SLAM baselines ORB-SLAM3 [28] and MonoGS [2], where the latter take TUM-VIE grayscale frames as input. For 3DGS reconstruction, we compare with 3DGS-based methods Event-3DGS [4], EvGGS [5], 3DGS [1], and MonoGS [2], and NeRF-based methods E-NeRF [14], EventNeRF [7], and Robust e-NeRF [15]. We adopt 3DGS [1] as the event-stream reconstruction method, and feed it with grayscale images converted by E2VID [21] and camera poses estimated by COLMAP [29].

3) Metrics: We evaluate tracking accuracy using the RMSE of Absolute Trajectory Error (ATE, cm). For map quality, we measure photometric fidelity with standard novel-view rendering metrics (PSNR, SSIM, and LPIPS): metrics are computed every five frames while excluding keyframe viewpoints to avoid bias from training views. We further assess geometric accuracy using 2D Depth L1(cm) and the F-score at a 1 cm threshold.

4) Implementation Details: We run our SLAM system on a desktop equipped with an Intel Core i9-14900KF CPU and an NVIDIA RTX 4090 GPU. We train the diffusion model on the event datasets [30]–[32] by freezing the initializer $f_{\text{init}}(\cdot)$ [21] and constructing triplets of *event stream–reference view–grayscale ground truth*, where the reference view is selected from nearby poses. We initialize from DIFIX pre-trained weights [18], set the diffusion timestep $\tau = 150$ and $\lambda_{\text{Gram}} = 0.5$, freeze the VAE encoder, and fine-tune the VAE decoder with LoRA while training the U-Net backbone and the decoder. In SLAM, we use $\lambda_{\text{img}} = 0.6$, $\lambda_{\text{evt}} = 0.4$, and $\kappa = 0.2$, while during initialization we warm up with $\lambda_{\text{img}} = 1$ and $\lambda_{\text{evt}} = 0$. We use 100 and 200 optimization iterations for tracking and mapping, respectively, and the keyframe sliding-window size is 5.

B. Quantitative Evaluation

1) Tracking Evaluation on TUM-VIE: We evaluate tracking accuracy on five mocap sequences of TUM-VIE, and report the ATE RMSE results in Table II. EventGS achieves the lowest error on three sequences and attains the best average performance of 2.03 cm over all sequences. Compared with ESVO2, EventGS benefits from two complementary losses in the appearance and change domains, which jointly improve noise suppression and observability. For the frame-based SLAM baselines ORB-SLAM3 and MonoGS, we use the grayscale camera images provided by TUM-VIE as input. These grayscale frames are prone to over-/under-exposure and motion blur under HDR lighting and aggressive motions, which degrades tracking stability. In contrast, our diffusion-restored intensity observations preserve clearer structural details across challenging conditions, while the event-accumulated change images provide stable temporal cues. Moreover, the progressive refinement strategy enables EventGS to remain stable under large motions, improving tracking accuracy on mocap-6d and mocap-desk2. In terms of efficiency, the diffusion network takes about 90 ms per frame, and the overall system runs at an average speed of approximately 2.5 FPS on TUM-VIE, which is on par with MonoGS (2.8 FPS), enabling online near real-time operation.

Fig. 5: Under HDR and fast motion conditions, grayscale camera images suffer from saturation and loss of details, whereas our diffusion-restored intensity observations preserve clear structure and stable contrast.

2) Mapping Evaluation on Replica Synthetic Dataset: We evaluate mapping performance on five scenes of a synthetic event dataset rendered from Replica 3D scene models,

TABLE III: Geometry results on Replica dataset.

Method	Metric	Office0	Office2	Office3	Room0	Room2
3DGS	Depth L1↓	2.39	6.95	5.32	3.04	3.27
	F1↑	57.64	36.18	41.59	47.36	52.39
EvGGS	Depth L1↓	2.75	8.07	5.19	2.24	3.53
	F1↑	50.68	30.43	34.45	57.28	51.67
Ours	Depth L1↓	**2.23**	**4.66**	**3.45**	**1.49**	**2.73**
	F1↑	**59.64**	**52.58**	**53.72**	**60.95**	**56.28**

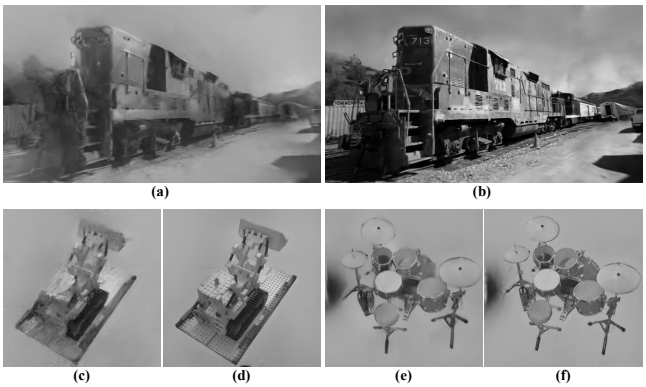

Fig. 6: Qualitative ablation results on the synthetic dataset. Among them, (b), (d), and (f) show the full EventGS results, while (a), (c), and (e) show the results after removing the diffusion-based intensity restoration, the event accumulation, and the progressive refinement, respectively.

reporting novel-view rendering metrics (PSNR, SSIM, and LPIPS) and geometry metrics (Depth L1 and F1). We use E2VID to convert event streams into grayscale images, which are then fed as input to 3DGS and MonoGS. The rendering results in Table I show that our method achieves the best PSNR across all five scenes, with an average of 21.70 dB, yielding clear improvements over existing event-driven 3DGS/NeRF reconstruction baselines. This indicates that the Gaussian map built by EventGS is more accurate in terms of pixel-level reconstruction consistency. Meanwhile, our method also maintains strong SSIM and LPIPS performance, outperforming most baselines. Geometry results are reported in Table III, where our method consistently attains low depth errors and high F1 scores across scenes. For example, on Room0, we achieve a Depth L1 of 1.49 and an F1 of 60.95, demonstrating stable structural completeness and surface coverage. Compared to event-based reconstruction methods that rely on a single loss term, using two complementary losses provides stronger constraints and leads to better reconstruction performance. The diffusion-restored intensity observations provide more reliable appearance constraints for rendering supervision, while the event-driven log-intensity-change loss further supplies temporal constraints for geometric optimization, effectively suppressing drift and maintaining stable geometric consistency.

C. Qualitative Evaluations

We qualitatively compare our method with 3D reconstruction baselines on the TUM-VIE dataset, as shown in Fig. 4. EventGS consistently produces clearer and more coherent novel-view renderings. Compared with the event-stream reconstruction method Event-3DGS and MonoGS with grayscale camera input, EventGS is more robust on sequences such as mocap-3d and mocap-desk, where event noise is higher, observations are sparser, or camera motion is faster. In these challenging cases, EventGS not only suppresses noise and outlier triggers in raw event streams, but also maintains more stable appearance and geometric consistency under weak textures and large viewpoint changes.

TABLE IV: Ablation study on the TUM-VIE mocap sequences [24] to analyze the effectiveness of each component (cm).

Method	mocap-1d	mocap-3d	mocap-6d
w/o Diffusion	5.25	6.72	5.63
w/o Accumulation	3.35	5.19	3.16
w/o Progressive	1.94	2.03	2.37
Ours	**1.39**	**1.52**	**1.04**

TABLE V: Ablation study on the Replica Office0 sequence to analyze the effectiveness of each component.

Method	PSNR↑	SSIM↑	LPIPS↓
w/o Diffusion	18.24	0.56	0.39
w/o Accumulation	19.83	0.68	0.41
w/o Progressive	21.73	0.71	0.39
Ours	**22.43**	**0.75**	**0.35**

In addition, to assess the reliability of appearance observations under degraded conditions, we compare grayscale camera frames with diffusion-restored intensity observations on the running sequence of TUM-VIE (Fig. 5). Under low-light or HDR scenarios with abrupt illumination changes, frame-based grayscale images often suffer from under-/over-exposure; under large motions, they exhibit severe motion blur, which undermines the reliability of appearance supervision. Directly using such grayscale frames for tracking or mapping can therefore lead to increased errors or even failure. In contrast, diffusion-based intensity restoration takes event streams as input and remains robust to extreme exposure variations and high-speed motion, producing grayscale observations with more stable exposure and clearer structure. As shown in Fig. 5, the diffusion-restored intensity images exhibit improved structural sharpness and texture consistency, with markedly reduced artifacts and motion blur. The resulting intensity observations are thus more stable and can be directly used for subsequent rendering optimization, providing critical support for robust pose tracking and high-quality mapping.

D. Ablation Studies

As shown in Table IV and Table V, ablation results demonstrate that each component is crucial for both tracking accuracy on TUM-VIE and mapping quality on Replica. On the three TUM-VIE sequences, removing the diffusion-based intensity restoration significantly increases trajectory error, indicating that relying solely on coarse degraded inputs cannot provide stable and reliable appearance supervision. Removing the event accumulation also leads to a clear degradation in tracking, highlighting the importance of change-domain temporal constraints for improving observability and suppressing drift. Disabling progressive refinement further

reduces overall accuracy; in particular, on the more challenging mocap-6d sequence, the error increases from 1.04 cm to 2.37 cm, validating its effectiveness in mitigating cascading failures under large motions and sparse observations. For mapping, on the Office0 scene of Replica, removing the diffusion-based intensity restoration reduces PSNR from 22.43 to 18.24, with SSIM and LPIPS also deteriorating noticeably. Removing either event accumulation or progressive refinement similarly causes a significant drop in mapping metrics, further confirming that each module contributes to high-quality map reconstruction. Moreover, we compare qualitative differences before and after ablation on a synthetic dataset from [4], as shown in Fig. 6, where the full model exhibits fewer holes and artifacts as well as more stable texture and structural continuity under novel viewpoints, consistent with the quantitative results.

V. Conclusions

We present EventGS, the first near real-time event-only 3DGS-SLAM system. Using only a monocular event camera, EventGS enables robust tracking and incremental dense mapping with 3D Gaussian Splatting. We introduce two complementary constraints—appearance-domain rendering supervision and change-domain consistency constraint—to improve stability under weak-texture and HDR conditions. We further employ an on-demand progressive refinement strategy to enhance robustness under aggressive motions. Experiments on both real-world and synthetic datasets show that EventGS matches or outperforms existing methods in terms of pose accuracy and renderable map quality.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis, *et al.*, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[2] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting slam," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 18 039–18 048.

[3] J. Niu, S. Zhong, X. Lu, S. Shen, G. Gallego, and Y. Zhou, "Esvo2: Direct visual-inertial odometry with stereo event cameras," *IEEE Transactions on Robotics*, 2025.

[4] H. Han, J. Li, H. Wei, and X. Ji, "Event-3dgs: Event-based 3d reconstruction using 3d gaussian splatting," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, pp. 128 139–128 159, 2024.

[5] J. Wang, J. He, Z. Zhang, M. Sun, J. Sun, and R. Xu, "EvGGS: A collaborative learning framework for event-based generalizable Gaussian splatting," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 50 561–50 579.

[6] S. Chen, S. Yuan, T.-M. Nguyen, Z. Huang, C. Shi, J. Jing, and L. Xie, "Egs-slam: Rgb-d gaussian splatting slam with events," *IEEE Robot. Autom. Lett.*, 2025.

[7] V. Rudnev, M. Elgharib, C. Theobalt, and V. Golyanik, "Eventnerf: Neural radiance fields from a single colour event camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 4992–5002.

[8] T. Yura, A. Mirzaei, and I. Gilitschenski, "Eventsplat: 3d gaussian splatting from moving event cameras for real-time rendering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025, pp. 26 876–26 886.

[9] L. Wang, R. Gong, Y. Han, L. Yang, L. Yang, Y. Li, B. Xu, H. Liu, and R. Fu, "Towards next-generation slam: A survey on 3dgs-slam focusing on performance, robustness, and future directions," *arXiv:2602.04251*, 2026.

[10] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat track & map 3d gaussians for dense rgb-d slam," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 21 357–21 366.

[11] E. Sandström, G. Zhang, K. Tateno, M. Oechsle, M. Niemeyer, Y. Zhang, M. Patel, L. Van Gool, M. Oswald, and F. Tombari, "Splat-slam: Globally optimized rgb-only slam with 3d gaussians," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025, pp. 1680–1691.

[12] S. Ghosh, V. Cavinato, and G. Gallego, "ES-PTAM: event-based stereo parallel tracking and mapping," in *Proc. ECCV Workshops (2024)*, 2024, pp. 70–87.

[13] J. Niu, S. Zhong, and Y. Zhou, "Imu-aided event-based stereo visual odometry," *arXiv:2405.04071*, 2024.

[14] S. Klenk, L. Koestler, D. Scaramuzza, and D. Cremers, "E-nerf: Neural radiance fields from a moving event camera," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1587–1594, 2023.

[15] W. F. Low and G. H. Lee, "Robust e-nerf: Nerf from sparse & noisy events under non-uniform motion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 18 335–18 346.

[16] J. Huang, C. Dong, X. Chen, and P. Liu, "Inceventgs: Pose-free gaussian splatting from a single event camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025, pp. 26 933–26 942.

[17] Y. Yang, J. Zhang, Y. Zhang, Y. Wei, D. Zou, J. S. Ren, and B. Shi, "Event-guided hdr reconstruction with diffusion priors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2025, pp. 11 787–11 796.

[18] J. Z. Wu, Y. Zhang, H. Turki, X. Ren, J. Gao, M. Z. Shou, S. Fidler, Z. Gojcic, and H. Ling, "Difix3d+: Improving 3d reconstructions with single-step diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025, pp. 26 024–26 035.

[19] J. Wei, S. Leutenegger, and S. Schaefer, "Gsfix3d: Diffusion-guided repair of novel views in gaussian splatting," *arXiv preprint arXiv:2508.14717*, 2025.

[20] S. Lee and G. H. Lee, "Diet-gs: Diffusion prior and event stream-assisted motion deblurring 3d gaussian splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 739–21 749.

[21] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1964–1980, 2019.

[22] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, "Adversarial diffusion distillation," *arXiv:2311.17042*, 2023.

[23] G. Parmar, T. Park, S. Narasimhan, and J.-Y. Zhu, "One-step image translation with text-to-image models," *arXiv:2403.12036*, 2024.

[24] S. Klenk, J. Chui, N. Demmel, and D. Cremers, "Tum-vie: The tum stereo visual-inertial event dataset," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 8601–8608.

[25] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv:1906.05797*, 2019.

[26] Z. Teed, L. Lipson, and J. Deng, "Deep patch visual odometry," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023, pp. 39 033–39 051.

[27] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, 2021.

[28] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[29] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 4104–4113.

[30] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, 2017.

[31] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 6713–6719.

[32] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2032–2039, 2018.