2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

MotionGS-SLAM: Event-Modulated Gaussian Splatting for Motion-Blur Robust SLAM

Zhiqiang HU, Shouren HUANG and Masatoshi ISHIKAWA

Abstract— **Current Vision-based SLAM systems fail catastrophically when motion blur corrupts the visual input, as they attempt the ill-posed inverse problem of recovering sharp content from degraded observations. We present MotionGS-SLAM, which fundamentally reimagines motion blur handling through a paradigm shift: rather than removing blur artifacts, we reformulate the challenge as a well-constrained forward problem that generatively models blur formation within the rendering pipeline. By leveraging event cameras' microsecond temporal resolution and immunity to motion blur, we introduce a novel event-modulated Gaussian kernel that dynamically adapts each Gaussian's rasterization based on precise motion cues. Our dual-modulation mechanism transforms 2D Gaussian projections from isotropic dots into anisotropic, motion-aligned elliptical brush strokes (spatial modulation) while adaptively varying exposure integral sampling density based on local velocity (temporal modulation). This physics-based approach enables joint optimization of intra-exposure camera trajectories and 3D scene geometry through blur-aware photometric and event-based constraints. Extensive experiments demonstrate significant improvements over state-of-the-art methods in trajectory accuracy and map quality under severe high-motion conditions.**

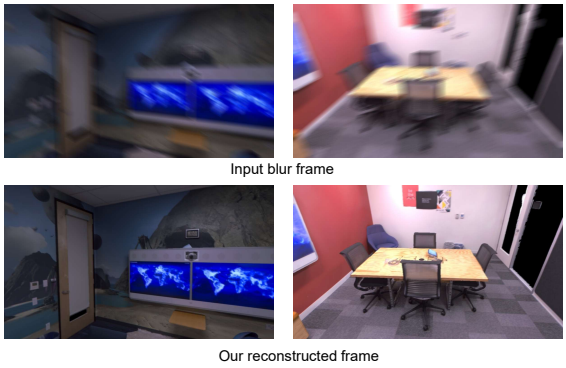

Fig. 1. **MotionGS-SLAM in action.** Existing Vision-based SLAM systems fail in challenging high-motion scenarios due to severe motion blur in the input frames (top row). Our method leverages the co-located event stream to explicitly model the physical process of blur formation. This allows MotionGS-SLAM to robustly track the camera and reconstruct sharp, high-fidelity scenes (bottom row) directly from the degraded imagery, where other methods would fail.

I. Introduction

Simultaneous Localization and Mapping (SLAM) is a cornerstone of robot autonomous systems, empowering robots to navigate and interact with unknown environments [2], [20]. While classical visual SLAM systems excel at localization, they often produce sparse geometric maps. The recent introduction of 3D Gaussian Splatting (3DGS) [10] into the SLAM pipeline has marked a significant leap forward, enabling the creation of dense, photorealistic scene representations in real-time [9], [13], [22]. These GS-SLAM systems hold immense promise for applications like augmented reality and digital twinning.

However, the impressive performance of current GS-SLAM methods is predicated on a critical assumption: the availability of high-quality, sharp input images. This assumption frequently breaks down in real-world robotic applications, particularly during aggressive motion or in low-light environments that necessitate long exposures as shown in Fig. 1. To compensate for dim lighting, cameras rely on long exposure times, which inevitably leads to severe motion blur. This blur corrupts the high-frequency details essential for feature matching, resulting in degraded camera tracking and, consequently, a corrupted 3D Gaussian map filled with artifacts. While some approaches have attempted to tackle this by integrating deblurring modules like [12], they often struggle with severe, persistent blur as they attempt to *invert* a highly degraded signal rather than modeling the underlying physical process.

The fundamental insight of this work is a paradigm shift in how we approach motion blur in visual SLAM. Rather than treating motion blur as an undesirable artifact to be removed through deconvolution, we embrace it as the natural physical consequence of camera motion during exposure and model it generatively. Our key observation is that traditional deblurring methods attempt to solve a severely ill-posed inverse problem recovering sharp content from degraded observations where critical high-frequency information has been irreversibly lost. In contrast, we reformulate this challenge as a well-constrained forward problem: leveraging the high-temporal-resolution motion cues from event cameras, we physically simulate the blur formation process within our differentiable renderer, actively "painting" blur that matches the observed degraded images.

To realize this vision, we introduce MotionGS-SLAM. Our system models the intra-exposure camera trajectory as a continuous path in $\mathbf{SE}(3)$ space. Given the typically brief exposure intervals, we parameterize each trajectory segment using only its boundary poses the camera configurations at the exposure's onset and conclusion. During tracking, we optimize these pose pairs by enforcing photometric consistency between the observed blurry frames and our physically-rendered blur simulations. This joint constraint of

The authors are with the Research Institute for Science & Technology, Tokyo University of Science {zhiqiang.hu, huang, ishikawa}@ishikawa-vision.org

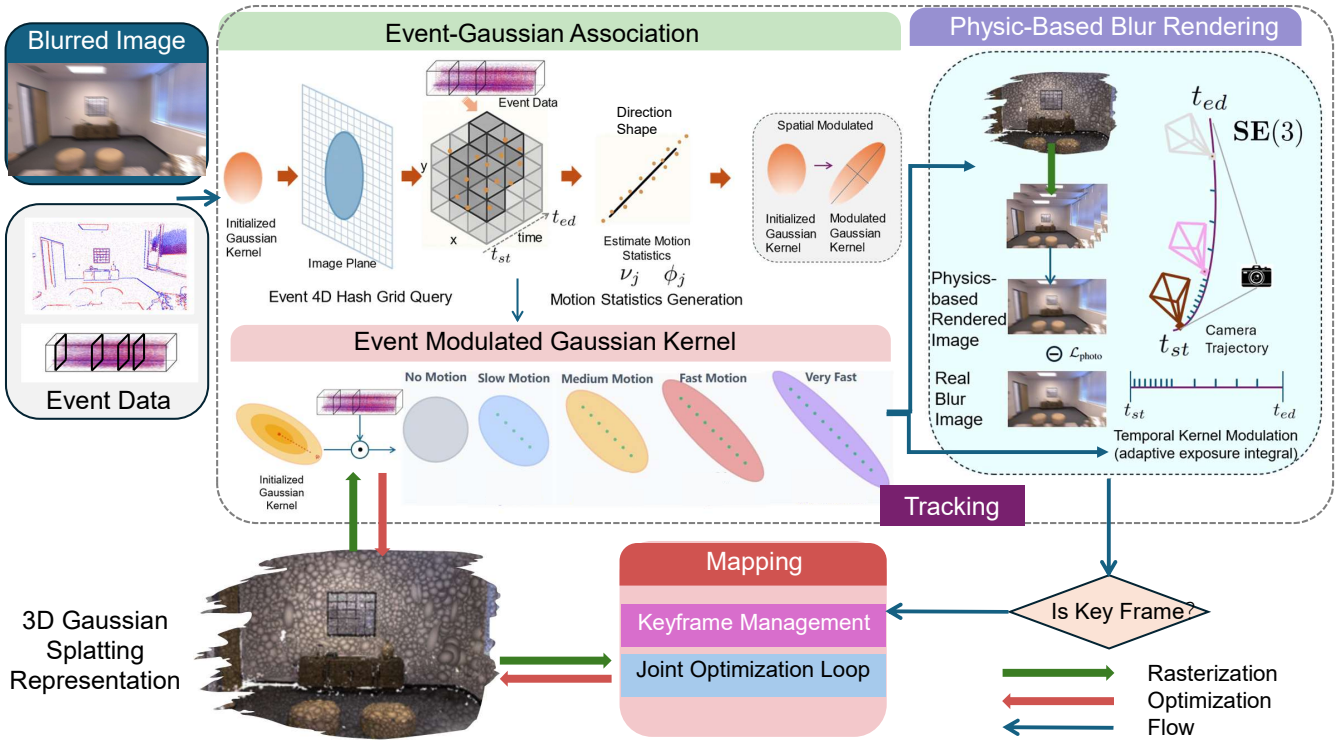

Fig. 2. **Overview of the MotionGS-SLAM pipeline.** Our system takes a blurry RGB image and a stream of events as input. The core of our method is the **Event-Modulated Gaussian Kernel**, which leverages events to physically model motion blur. **(1) Event-Gaussian Association:** For each projected Gaussian, we use a 4D spatio-temporal hash grid to efficiently query associated events. From these, we compute local motion statistics (velocity ν_j, direction ϕ_j). **(2) Kernel Modulation:** These statistics drive our novel dual-modulation mechanism, anisotropically scaling the projected 2D Gaussian's shape (spatial modulation) to replicate the motion streak. **(3) Physics-Based Rendering:** Our renderer uses the modulated kernels and an interpolated camera trajectory (T_{st}, T_{ed}) to synthesize a blurred image that matches the real observation. This entire differentiable process is used in two threads: a real-time **Tracking** front-end that optimizes only the camera trajectory, and a **Mapping** back-end that performs joint optimization over a keyframe window to refine both the 3D Gaussian map and poses.

event-guided motion estimation and blur-aware photometric alignment substantially enhances tracking precision. In the mapping phase, we perform joint optimization over both the trajectories of strategically selected keyframes and the 3D Gaussian scene representation. This optimization minimizes a combined objective of photometric reconstruction error and event-based geometric constraints, resulting in sharp, geometrically accurate maps despite severely degraded input imagery.

The technical cornerstone of our approach is an **event-modulated Gaussian kernel** that dynamically adapts how each 3D Gaussian is rasterized onto the image plane. This adaptation operates through a dual-modulation mechanism: **Spatial Modulation:** Guided by local motion statistics extracted from the event stream, we transform each Gaussian's 2D projection from an isotropic circular "dot" into an anisotropic elliptical "brush stroke." The ellipse's orientation aligns with the detected motion direction while its eccentricity scales with motion magnitude, geometrically replicating the characteristic streak patterns of motion blur. **Temporal Modulation:** We adaptively adjust the temporal sampling density of the exposure integral based on the scene's aggregate motion. Frames with higher average motion velocity, computed across all visible Gaussians, are rendered with a greater number of discrete camera poses.This strategy accurately models the cumulative photometric effect while maintaining computational efficiency.

Our main contributions are summarized as follows:

- We propose **MotionGS-SLAM**, that tackles motion blur from a physics-based perspective. Instead of treating blur as an artifact to be removed, we model its physical formation process within the renderer, guided by an event camera.
- We introduce an **event-modulated Gaussian kernel** featuring a dual-modulation strategy. It dynamically adapts the spatial shape and temporal sampling density of projected Gaussians based on local motion statistics derived efficiently from events via a 4D hash grid.
- We demonstrate that our system, which requires no depth sensor, significantly outperforms state-of-the-art visual and GS-SLAM methods in both tracking accuracy and reconstruction quality on challenging synthetic and real-world sequences with severe motion blur.

II. Related Works

A. Gaussian Splatting based SLAM

The introduction of 3D Gaussian Splatting [10] has recently enabled a new class of SLAM systems that create

dense, photorealistic maps in real time. Pioneering works like GS-SLAM [22], SplaTAM [9], MonoGS [13], Sgs-slam [11], Large spatial model [6] and Compact 3d gaussian splatting [5] successfully integrated 3DGS into a unified tracking and mapping framework, demonstrating remarkable results on high-quality video sequences. These systems established the viability of using an explicit, differentiable representation for both localization and scene reconstruction.

However, the core assumption of sharp, well-exposed images makes these methods brittle. Recognizing this, subsequent works have attempted to address the challenge of motion blur. While approaches like I^2-SLAM [1] model the camera's imaging process, these image-only methods are fundamentally constrained. They attempt to recover lost information from an already corrupted signal, an approach that fails when severe motion makes the blur too ambiguous to reliably invert. Our work differs by leveraging a secondary, high-frequency modality to directly inform the rendering of motion, rather than relying solely on the degraded image.

B. Event-based 3D Gaussian Splatting/NeRF

The asynchronous nature and inherent immunity to motion blur make event cameras particularly advantageous for three-dimensional scene reconstruction. A recent surge of research has focused on combining event streams with NeRF/3DGS. E2NeRF [15] integrates both blurred image supervision and event-based constraints to recover high-fidelity NeRF representations from severely degraded inputs. Building upon this, Ev-DeblurNeRF [3] introduces a learned event-to-intensity mapping that effectively suppresses event noise, leading to enhanced reconstruction fidelity. Meanwhile, the method in [16] demonstrates the feasibility of event-guided implicit neural SLAM by fusing event streams with RGB-D inputs, achieving robust tracking under motion blur and lighting variation through a differentiable CRF rendering technique. Methods like EvaGaussians [23] and Event3DGS [7], [21] have shown that by leveraging event data, it is possible to reconstruct crisp, detailed 3D Gaussian scenes from a set of blurry images. These works successfully demonstrate the power of events for high-fidelity reconstruction. However, these are primarily offline reconstruction pipelines that require pre-computed camera poses (e.g., via SfM), rendering them incapable of online SLAM. Our work bridges this critical gap by integrating our event-guided generative rendering directly into the front-end tracking loop. This enables a complete and robust online GS-SLAM system that succeeds where prior methods fail.

III. METHOD

The core architecture of our method is shown in Fig. 2.

A. Motion Blur Image Formation Model

We represent the scene as a set of 3D Gaussians $\mathcal{G} = \{g_i\}$, where each Gaussian is defined by a mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$, covariance $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$, color $\boldsymbol{c}_i$, and opacity o_i. In a standard static rendering, each 3D Gaussian is projected onto the 2D image plane given a single camera pose $\mathbf{T}$, and the final color $\mathbf{C}(\boldsymbol{u})$ at a pixel $\boldsymbol{u}$ is computed via α-blending.

However, motion blur is a physical process of light integration over a non-zero exposure interval Δt_m, during which the camera is moving. We model the camera's continuous trajectory $\mathbf{T}(t)$ for $t \in [t_{st}, t_{ed}]$ by interpolating a start pose $\mathbf{T}_{st}$ and an end pose $\mathbf{T}_{ed}$. The resulting blurred image $\bar{\mathbf{C}}_m$ is the temporal integral of infinitesimally short exposures along this path. This is approximated by discretely summing N latent sharp images:

$$\bar{\mathbf{C}}_m(\boldsymbol{u}) \approx \frac{1}{N}\sum_{k=1}^{N} \mathbf{C}(\mathbf{T}(t_k), \boldsymbol{u}). \tag{1}$$

where $\mathbf{C}(\mathbf{T}(t_k), \boldsymbol{u})$ is the sharp image rendered at an intermediate pose $\mathbf{T}(t_k)$.

Camera Motion Trajectory Modeling We parameterize the camera motion using two poses: the exposure start $\mathbf{T}_{st} \in \mathbf{SE}(3)$ and end $\mathbf{T}_{ed} \in \mathbf{SE}(3)$. The pose at any time $t \in [0, 1]$ during exposure is:

$$\mathbf{T}(t) = \mathbf{T}_{st} \cdot \exp(t \cdot \log(\mathbf{T}_{st}^{-1} \cdot \mathbf{T}_{ed})), \tag{2}$$

where $\exp$ and $\log$ are the exponential and logarithm maps on $\mathbf{SE}(3)$.

B. Motion Blur Aware Tracker

Real-Time Event-Gaussian Association

To provide each Gaussian with its local motion context, we must efficiently associate it with the relevant subset of events. A brute-force search is computationally infeasible for real-time operation.

4D Hash Grid. We employ a multi-resolution hash grid to index events in a 4D spatio-temporal domain. Each event $e = (x, y, t, p)$ is characterized by its spatial coordinates $(x, y) \in \mathbb{R}^2$, timestamp $t \in \mathbb{R}$, and polarity $p \in \{-1, +1\}$ indicating brightness decrease or increase, respectively.

Query Process. For each visible 3D Gaussian g_j, we project it to the 2D image plane, obtaining its mean $\hat{\boldsymbol{\mu}}_j$ and covariance $\hat{\boldsymbol{\Sigma}}_j$. We then query the hash grid for all events within the 3σ axis-aligned bounding box of the 2D Gaussian and within the camera's exposure window $[t_{st}, t_{ed}]$. We refine this candidate set by retaining only the events $\mathcal{E}_j$ that satisfy the Mahalanobis distance criterion:

$$(\boldsymbol{p} - \hat{\boldsymbol{\mu}}_j)^\top \hat{\boldsymbol{\Sigma}}_j^{-1} (\boldsymbol{p} - \hat{\boldsymbol{\mu}}_j) \leq \tau, \tag{3}$$

where τ is a tunable threshold (typically $\tau \approx 9$, corresponding to the 99% confidence contour of a 2D Gaussian). This hash-based approach reduces the association complexity from $O(N_{\text{gauss}} \times N_{\text{event}})$ to an efficient $O(N_{\text{gauss}} \times K)$, where N_{gauss} is the number of visible Gaussians in the current frame, N_{event} is the number of events in $[t_{st}, t_{ed}]$, and K is the small average events returned per Gaussian by the grid query.

Motion Statistics. For each Gaussian g_j, we estimate its local motion statistics from the associated event set $\mathcal{E}_j$ on the image plane. Specifically, we first compute a polarity-weighted average displacement of events relative to the Gaussian center:

$$\boldsymbol{v}_j = \frac{\sum_{e_i \in \mathcal{E}_j} p_i \left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j\right)}{\sum_{e_i \in \mathcal{E}_j} |p_i| + \varepsilon}, \qquad \nu_j = \|\boldsymbol{v}_j\|, \qquad \phi_j = \operatorname{atan2}(v_y, v_x), \tag{4}$$

where $p_i \in \{-1, +1\}$ is the polarity and $\mathbf{x}_i \in \mathbb{R}^2$ is the 2D spatial coordinate of event e_i, $\hat{\boldsymbol{\mu}}_j$ is the projected 2D mean of Gaussian g_j, and ε is a small constant to avoid division by zero. The resulting vector $\boldsymbol{v}_j$ represents the dominant local motion flow: events with opposite polarities moving in the same direction reinforce each other, while inconsistent noise cancels out. We use its magnitude ν_j as the estimated motion speed and its angle ϕ_j as the motion direction.

Event-Modulated Gaussian Kernel

The computed motion statistics (ϕ_j, ν_j) drive our dual-modulation kernel, which adapts both the spatial and temporal aspects of the rendering process for each Gaussian. Our approach separates the *true 3D geometry* from *motion blur effects* through a two-component covariance design, ensuring the core 3D map remains geometrically accurate while enabling physics-based blur synthesis.

a) Spatial Kernel Modulation.: To model the spatial characteristics of motion blur, we construct a motion-aligned prior covariance from event measurements:

$$\boldsymbol{\Sigma}^{2D}_{j,\text{prior}} = \mathbf{R}(\phi_j)\, \operatorname{diag}\!\left(\sigma_\perp^2,\ \kappa_j^2 \sigma_\perp^2\right) \mathbf{R}(\phi_j)^\top, \qquad \kappa_j = 1 + \beta \nu_j, \tag{5}$$

where $\mathbf{R}(\cdot)$ is the 2D rotation matrix, $\sigma_\perp^2$ is a base variance, and κ_j is a velocity-dependent stretch factor. This formulation creates an anisotropic Gaussian aligned with the motion direction ϕ_j and elongated proportionally to the velocity ν_j, replicating the characteristic streak pattern of motion blur.

We enforce this motion-consistent shape through a regularization loss:

$$\mathcal{L}_{\text{shape}} = \frac{1}{|\mathcal{G}_{\text{vis}}|} \sum_{j \in \mathcal{G}_{\text{vis}}} \left\| \hat{\boldsymbol{\Sigma}}^{2D}_j - \boldsymbol{\Sigma}^{2D}_{j,\text{prior}} \right\|_F^2, \tag{6}$$

where $\hat{\boldsymbol{\Sigma}}^{2D}_j$ is the projected 2D covariance of the optimized 3D Gaussian, and $\mathcal{G}_{\text{vis}}$ is the set of Gaussians visible in the current frame.

To maintain numerical stability while separating geometry from blur, we decompose the 3D covariance as:

$$\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}_{\text{base},j} + \Delta\boldsymbol{\Sigma}_j, \qquad \boldsymbol{\Sigma}_{\text{base},j} = \mathbf{R}(q_j) \operatorname{diag}(e^{\ell_{x,j}}, e^{\ell_{y,j}}, e^{\ell_{z,j}}) \mathbf{R}(q_j)^\top, \tag{7}$$

where $\boldsymbol{\Sigma}_{\text{base},j}$ encodes the static 3D geometry with guaranteed positive-definiteness through exponential parameterization, and $\Delta\boldsymbol{\Sigma}_j$ represents the motion-induced deformation bounded via activation functions. This decomposition ensures that motion blur modeling does not corrupt the underlying geometric representation the optimizer learns $\Delta\boldsymbol{\Sigma}_j$ indirectly through the shape prior (Eq. (6)) while maintaining photometric consistency.

b) Temporal Kernel Modulation (Adaptive Sampling).: Motion blur arises from integrating light over the exposure interval, during which both the camera and scene features move continuously. To accurately model this process, we adaptively sample the camera trajectory based on the aggregate motion characteristics of all visible Gaussians.

For each frame m, we determine the trajectory sampling density by considering the motion statistics across all Gaussians:

$$\bar{\nu}_m = \frac{1}{|\mathcal{G}_{\text{vis}}|} \sum_{j \in \mathcal{G}_{\text{vis}}} \nu_j, \tag{8}$$

where $\mathcal{G}_{\text{vis}}$ denotes the set of visible Gaussians and $\bar{\nu}_m$ represents the scene's average motion magnitude.

This aggregate motion drives our adaptive sampling strategy:

$$N_m = \operatorname{clamp}\left(\operatorname{round}\left(n_0 \cdot (1 + \alpha \cdot \bar{\nu}_m \cdot \Delta t_m)\right), N_{\min}, N_{\max}\right), \tag{9}$$

where n_0 is the base sampling rate, α scales the motion response, Δt_m is the exposure duration, and the result is clamped to $[N_{\min}, N_{\max}]$ for computational efficiency.

As illustrated in the Fig. 2, this mechanism creates a motion-aware sampling of the camera trajectory: rapid scene motion (high $\bar{\nu}_m$) triggers denser sampling to capture the complex blur formation, while slow or static scenes require fewer samples. The camera poses are then uniformly distributed along the SE(3) geodesic between $\mathbf{T}_{st}$ and $\mathbf{T}_{ed}$:

$$t_k = t_{st} + k \cdot \frac{\Delta t_m}{N_m - 1}, \quad k \in \{0, 1, ..., N_m - 1\}, \tag{10}$$

ensuring accurate physical simulation of the exposure process while maintaining computational efficiency.

Blur Aware Tracking Optimization

The tracking front-end is the workhorse of our SLAM system, responsible for estimating the camera's precise motion for each incoming blurry frame. Its goal is to find the most probable continuous trajectory, parameterized by the exposure start and end poses $(\mathbf{T}_{st}, \mathbf{T}_{ed})$. To achieve this, for each incoming blurry frame m, we estimate the exposure endpoints $(\mathbf{T}_{st}, \mathbf{T}_{ed})$ by minimizing

$$\min_{\mathbf{T}_{st}, \mathbf{T}_{ed}} \mathcal{L}_{\text{photo}} + \lambda_{\text{evt}} \mathcal{L}_{\text{LI}}, \tag{11}$$

where λ_{evt} balances the event terms.

1. Photometric Loss. The foundational constraint of our optimization is photometric consistency. This principle dictates that our model encompassing both the 3D scene representation and the estimated camera trajectory must accurately replicate the physical image formation process. Consequently, the blurred image synthesized by our renderer should closely align with the actual image captured by the camera sensor. To enforce this, we synthesize a blurred image $\bar{\mathbf{C}}_m$ using our physics-based renderer, complete with the event-modulated spatial and temporal kernels. We then define the photometric loss as a robust error metric (e.g., Charbonnier loss ρ) between our rendered output and the observation $\mathbf{I}_{\text{obs}}$:

$$\mathcal{L}_{\text{photo}} = \|\bar{\mathbf{C}}_m - \mathbf{I}_{\text{obs}}\|_\rho, \tag{12}$$

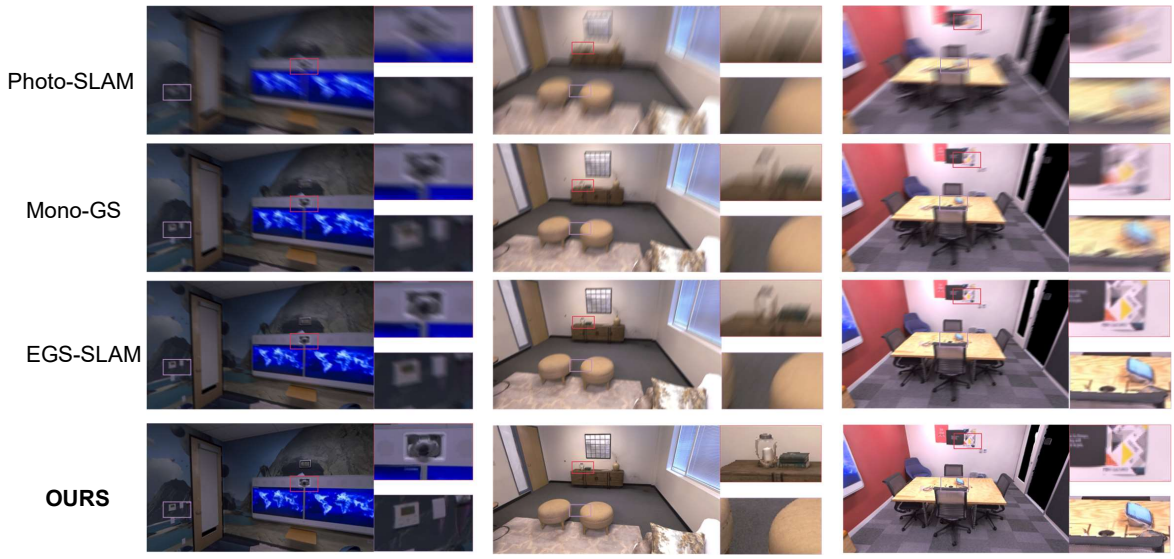

Fig. 3. **Qualitative comparison of reconstructed scenes under severe motion blur conditions.** The top three rows show results from state-of-the-art baselines: Photo-SLAM [8], Mono-GS [13], and EGS-SLAM [4]. Photo-SLAM and Mono-GS struggle significantly, producing highly blurry and distorted reconstructions with numerous artifacts and "ghosting" effects. EGS-SLAM, while benefiting from event data, still exhibits noticeable blur and less accurate details, particularly in fine textures (e.g., the television screen, object edges). In stark contrast, the bottom row showcases results from **MotionGS-SLAM (Ours)**. Our method consistently reconstructs sharp, high-fidelity scenes with clear details and accurate geometry.

where $\bar{\mathbf{C}}_m$ is rendered with per-Gaussian adaptive temporal samples $n_j^{(m)}$ and spatial modulation $\mathbf{\Sigma}_{j,\text{prior}}^{2D}$.

2. Event Alignment Loss. The event camera tells us exactly where and when brightness changed. Our estimated camera motion must be able to reproduce these same brightness changes in the rendered 3D scene. This loss directly aligns the rendered intensity changes with the observed events. Following the event generation model, an event's polarity $p_i \in \{+1, -1\}$ should correspond to the sign of the logarithmic intensity change. We penalize any deviation from this model:

$$\mathcal{L}_{\text{LI}} = \sum_{e_i \in \mathcal{E}} \rho\left(p_i - \frac{\log I(t_i) - \log I(t_i - \delta)}{\theta}\right), \tag{13}$$

where $I(t)$ is the rendered intensity at an event's timestamp t_i, δ is a small temporal offset, and θ is the event camera's contrast threshold. This provides strong geometric constraints, especially in visually ambiguous or low-texture regions.

C. Mapping with Keyframe Management

While the tracker estimates frame-to-frame motion, the mapping back-end ensures long-term map consistency. It operates on a sliding window of keyframes to jointly refine 3D Gaussians and camera poses, maintaining physical agreement between all sensor observations over time.

1) Keyframe Selection: We insert a new keyframe when either: (1) the view overlap with the latest keyframe measured as IoU of visible Gaussians drops below τ_{overlap}, or (2) camera translation exceeds $\tau_{\text{trans}} \cdot \bar{d}$ (where $\bar{d}$ is average scene depth). When the keyframe limit is reached, we remove the keyframe with highest neighbor overlap to preserve view diversity while bounding computation.

2) Bundle Adjustment: The core of our mapping process is a bundle adjustment over the sliding window $\mathcal{W}$, where we jointly optimize the Gaussian map $\mathcal{G}$ and the keyframe poses $\{\mathbf{T}\}$ using a comprehensive, blur-consistent objective:

$$\min_{\mathcal{G},\{\mathbf{T}\}} \sum_{w \in \mathcal{W}} \left[\mathcal{L}_{\text{photo}}^{w} + \lambda_{\text{evt}} \mathcal{L}_{\text{LI}}^{w}\right] + \lambda_{\text{shape}} \mathcal{L}_{\text{shape}}, \tag{14}$$

where $\mathcal{W}$ is the set of keyframes in the current optimization window. This objective has two main parts: **Data Consistency Terms:** The first summation, $\sum_{w \in \mathcal{W}}[\dots]$, ensures that for every keyframe in the window, our optimized map and poses can accurately reproduce the observed blurry images and event streams. This enforces consistency across time.

$\mathcal{L}_{\text{shape}}$: Image-Plane Shape Prior Loss. This loss encourages each 3D Gaussian, after projection onto the image plane, to form a 2D elliptical kernel slightly elongated along the detected motion direction, without altering its underlying 3D geometry. It is defined in Eq. (6): if events detect motion in a certain direction, the projected Gaussian ellipse is encouraged to be gently elongated along that direction; otherwise, it remains isotropic. This enforces a physically meaningful explanation of blur while avoiding contaminating the underlying 3D geometry.

IV. Experiments

In this section, we conduct a series of rigorous experiments to validate the effectiveness of MotionGS-SLAM.

A. Datasets and Metrics

Datasets: We evaluate our method on both synthetic and real-world sequences to ensure a thorough analysis.

TABLE I

RENDERING PERFORMANCE COMPARISON ON *EventReplica*. "–" INDICATES SEQUENCES WHERE REINITIALIZATION DID NOT SUCCEED. BOLDFACE MARKS THE BEST RESULT PER COLUMN.

Method	Metric	room0	room1	room2	office0	office1	office3	office4	Avg.	Rendering FPS
PhotoSLAM [8]	PSNR[dB]↑	17.91	18.96	–	23.15	–	17.04	14.28	–	1058.6
	SSIM↑	0.499	0.588	–	0.640	–	0.596	0.546	–	
	LPIPS↓	0.460	0.455	–	0.417	–	0.391	0.526	–	
MonoGS [13]	PSNR[dB]↑	20.27	21.77	23.85	26.55	26.56	21.21	21.59	23.12	1102.1
	SSIM↑	0.612	0.677	0.748	0.769	0.810	0.729	0.749	0.728	
	LPIPS↓	0.474	0.471	0.355	0.385	0.355	0.287	0.411	0.391	
EGS-SLAM [4]	PSNR[dB]↑	24.06	26.30	27.61	31.72	33.38	26.50	23.79	27.62	1168.3
	SSIM ↑	0.744	0.783	0.838	0.885	0.927	0.846	0.806	0.833	
	LPIPS↓	0.229	0.256	0.172	0.142	0.123	0.113	0.242	0.182	
MotionGS-SLAM (Ours)	PSNR[dB]↑	**24.55**	**27.05**	**28.22**	**32.31**	**33.92**	**26.98**	**24.10**	**28.05**	**1189.7**
	SSIM ↑	**0.756**	**0.796**	**0.848**	**0.893**	**0.933**	**0.855**	**0.814**	**0.842**	
	LPIPS↓	**0.214**	**0.241**	**0.159**	**0.131**	**0.116**	**0.107**	**0.232**	**0.171**	

EventReplica: We created a challenging synthetic dataset by adapting the Replica dataset [18]. For each camera trajectory, we render a high number of intermediate frames and simulate a long exposure by averaging them, which produces significant and physically realistic motion blur. To simulate low-light-induced motion blur conditions, we also inject Poisson-Gaussian noise into the final blurred images. This dataset provides ground truth sharp images for objective evaluation of reconstruction quality. Event streams are synthesized via ESIM [17] using a balanced threshold $\Theta = 0.2$ and monochrome events.

Real Database: To test our system in real-world scenarios, we introduce a real-world dataset of 3 indoor/outdoor scenes recorded with a Color-DAVIS346 [19] camera. This device captures both color frames at 346×260 resolution and color events. We capture 3 challenging scenes using a handheld camera setup under low-light conditions. Each scene contains rich textures and color information. For each scene, we acquire 30s frames across multiple viewpoints with varying degrees of motion blur, along with temporally aligned event streams recorded by the event camera.

Metrics: We use standard metrics for evaluation. For trajectory accuracy, we report the **Absolute Trajectory Error (ATE)** [cm]. For mapping quality on the synthetic dataset, we report **PSNR**, **SSIM**, and **LPIPS** [24].

B. Implementation Details

Our system is implemented in PyTorch and runs on a single NVIDIA RTX 4080 GPU. We build our system upon the MonoGS [13] codebase. For our event-modulated kernel, we set the base temporal samples $n_0 = 9$ and the spatial stretch factor control $\beta = 0.05$. In our loss function, weights are set to $\lambda_{\text{evt}} = 2.0$, $\lambda_{\text{photo}} = 1.0$, and $\lambda_{\text{shape}} = 0.2$. For all comparisons, we ensure that competing methods are configured to their official recommended settings for a fair evaluation. In all tables, **boldface** indicates the best result.

C. Comparison with State-of-the-Art

Evaluation on Blurry-Replica with Event (Synthetic) We compare **MotionGS-SLAM** against representative baselines on the challenging *EventReplica* benchmark under low-light and long-exposure settings: the photometric SLAM **PhotoSLAM** [8], the image-only GS-SLAM **MonoGS** [13], and the event-guided **EGS-SLAM**. We evaluate two aspects: (i) reconstruction fidelity, and (ii) tracking accuracy under severe motion blur.

Reconstruction quality and throughput. Table I reports PSNR/SSIM/LPIPS and rendering FPS. While PhotoSLAM and MonoGS degrade notably with persistent blur, and EGS-SLAM improves upon image-only pipelines by utilizing events, **MotionGS-SLAM** achieves the best fidelity across almost all sequences and the highest rendering FPS. Concretely, *MotionGS-SLAM* improves the *average* PSNR from *23.12 dB* (MonoGS) and *27.62 dB* (EGS-SLAM) to **28.05 dB**; SSIM from *0.728* (MonoGS) and *0.833* (EGS-SLAM) to **0.842**; and LPIPS from *0.391* (MonoGS) and *0.182* (EGS-SLAM) down to **0.171**. On representative sequences, our PSNR is higher (e.g., `office1`: **33.92** vs. 33.38; `office0`: **32.31** vs. 31.72; `room1`: **27.05** vs. 26.30). Throughput-wise, *MotionGS-SLAM* reaches **1189.7** FPS, exceeding EGS-SLAM (1168.3), MonoGS (1102.1), and PhotoSLAM (1058.6), indicating that event-modulated spatial/temporal kernels not only sharpen reconstructions but also preserve rendering speed. **Tracking accuracy.** As shown in Table III, **MotionGS-SLAM** achieves the lowest ATE in all seven sequences. ORB-SLAM2 and PhotoSLAM frequently lose tracking as blurred edges hamper keypoint detection; MonoGS assumes blur-free inputs and suffers on long-exposure frames. EGS-SLAM benefits from events but lacks our exposure-integration modeling and per-Gaussian modulation, leading to consistently higher errors. Overall, *MotionGS-SLAM* reduces the average ATE from *9.22* cm (MonoGS) to **4.44** cm ($\sim$51.9% improvement), and further improves over EGS-SLAM (5.17 cm) by $\sim$14.2%. These numbers exactly correspond to the averages reported in Table III.

Evaluation on Real-World Data We further validate our method on three challenging, low-light handheld scenes. As shown in Table II, the event-based EGS-SLAM is the strongest baseline, significantly outperforming image-only methods.

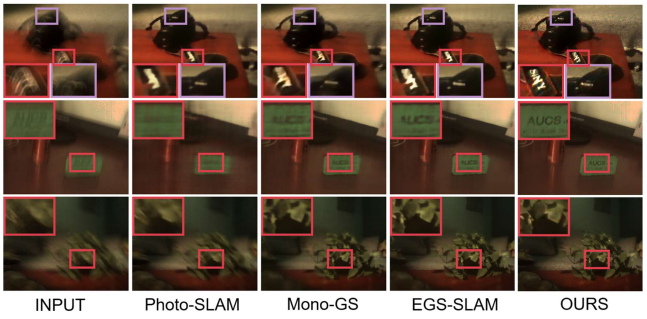

Fig. 4. **Qualitative comparison on our real-world dataset.** We demonstrate our method's performance on challenging, low-light handheld sequences. As shown in the corresponding figure, image-only baselines like Photo-SLAM [8] and Mono-GS [13] produce blurry reconstructions with severe "ghosting" artifacts. While the event-based EGS-SLAM [4] improves upon them, it still suffers from soft details. In contrast, our method is the only one that successfully reconstructs a sharp, geometrically consistent scene, recovering fine details.

TABLE II

TRACKING ACCURACY (ATE IN CM) AND ABLATION STUDY ON OUR REAL-WORLD DATASET.

Method	Scene 1	Scene 2	Scene 3	Avg.
Comparison with State-of-the-Art				
ORB-SLAM3 [2]	8.91	35.42	41.05	28.46
PhotoSLAM [8]	11.24	9.85	28.66	16.58
EGS-SLAM [4]	4.52	5.15	4.81	4.83
Ablation Study				
Baseline (MonoGS) [13]	6.88	7.12	5.95	6.65
+ Events	4.65	5.01	4.42	4.69
+ Spatial Modulation	4.03	4.45	3.88	4.12
Full Model (Ours)	**3.45**	**4.02**	**3.18**	**3.55**

Qualitative Results: Fig.3 and Fig.4 provide a visual comparison of the reconstructed maps. The outputs from MonoGS are visibly blurry and contain significant floating artifacts and "ghosting" effects, consistent with the quantitative results. In contrast, our reconstruction is sharp, detailed, and geometrically coherent, highlighting the power of our physics-based, event-guided rendering approach.

D. Ablation Study

To rigorously validate our design choices, we conduct a detailed ablation study on two representative *EventReplica* sequences. We analyze how each component of our event-guided framework contributes to the final performance. The results are summarized in Table IV and shown in Fig. 5.

Our ablation compares five system variants: **A0 (Baseline):** A standard GS-SLAM system (MonoGS) using only the blurry images, with no event data or explicit blur model. **A1 (Blur Model Only):** Augments the baseline with our physics-based exposure integration renderer, but without any guidance from events. **A2 (Event Assoc.):** Introduces event data and association but does not yet use it to modulate the rendering kernel. Events contribute only through standard losses ($\mathcal{L}_{\text{LI}}$. **A3 (Spatial Modulation Only):** Builds on A2 by adding our spatial kernel modulation (anisotropic deformation via $\mathcal{L}_{\text{shape}}$), but keeps temporal sampling fixed. **A4 (Temporal Modulation Only):** Builds on A2 by adding our adaptive temporal sampling, but keeps the spatial kernel untouched. **A5 (Full / Ours):** The complete MotionGS-SLAM system with both spatial and temporal modulation enabled.

TABLE III

TRACKING RESULTS ATE (CM) ON *EventReplica*. L DENOTES TRACKING FAILURE. BOLDFACE MARKS THE BEST RESULT PER ROW/AVERAGE.

Scenes	ORB-SLAM2 [14]	PhotoSLAM [8]	MonoGS [13]	EGS-SLAM [4]	MotionGS-SLAM
room0	5.57	6.64	12.76	4.85	**3.98**
room1	L	L	8.45	3.55	**2.88**
room2	L	L	3.64	3.25	**2.84**
office0	L	L	7.44	3.75	**3.01**
office1	L	L	7.78	3.66	**3.06**
office3	6.48	6.78	7.91	4.55	**3.92**
office4	L	L	16.55	12.60	**11.37**
Avg.	–	–	9.22	5.17	**4.44**

TABLE IV

ABLATION ON *EventReplica* FOR TWO SCENES. BEST RESULTS IN EACH COLUMN ARE IN **BOLD**.

Variant	room0				office0			
	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓
A0_Baseline	12.76	20.27	0.612	0.474	7.44	26.55	0.769	0.385
A1_Blur_Only	11.80	21.10	0.630	0.455	6.90	27.30	0.780	0.370
A2_Event_Assoc	6.10	22.20	0.700	0.340	4.10	28.50	0.830	0.240
A3_Spatial_Only	5.40	23.60	0.735	0.250	3.50	30.70	0.880	0.160
A4_Temporal_Only	5.10	23.10	0.725	0.265	3.40	30.20	0.870	0.175
A5_Full (Ours)	**3.98**	**24.55**	**0.756**	**0.214**	**3.01**	**32.31**	**0.893**	**0.131**

a) The Necessity of Event Data: The results in Table IV reveal the critical role of event cameras in addressing motion blur. The baseline (A0) fails catastrophically with blur-only inputs, while simply adding our physics-based blur model (A1) provides marginal gains reducing ATE from 12.76cm to 11.80cm on `room0`. The breakthrough comes with event integration (A2): ATE plummets to 6.10cm, a 52% reduction from baseline. This dramatic improvement demonstrates that events provide the missing temporal resolution needed to disambiguate motion during exposure. However, A2's relatively poor perceptual metrics (LPIPS of 0.340 vs. our full model's 0.214) exposes a crucial limitation: standard event losses alone cannot solve the rendering problem. They constrain camera motion but fail to inform how blur should be synthesized.

b) Dissecting the Dual-Modulation Kernel: Comparing variants A2 through A5 validates our core hypothesis: effective blur handling requires modeling both its spatial formation and temporal integration. **Spatial modulation (A3)** transforms the reconstruction quality LPIPS improves from 0.340 to 0.250 on `room0`, the single largest perceptual gain across all components. This confirms our insight that event-guided anisotropic deformation correctly models the physical "painting" of motion streaks. **Temporal modulation (A4)** contributes differently but equally importantly: it ensures photometric accuracy by adapting the exposure integral's sampling density to local motion, improving PSNR while

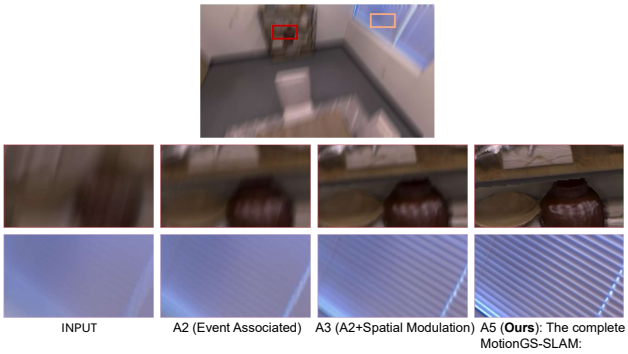

Fig. 5. **Qualitative Ablation Study on EventReplica Dataset.** Top row shows severely blurred input. Red and orange boxes highlight reconstructed details for: **A2:** Event data with standard losses. **A3:** A2 + spatial kernel modulation. **A5 (Ours):** Complete MotionGS-SLAM. Clear progression from blurry A2 to sharp A3, culminating in high-fidelity reconstruction with our full model.

maintaining computational efficiency. The **full model (A5)** demonstrates clear synergy: combining both modulations yields the best performance across all metrics (ATE: 3.98cm, PSNR: 24.55dB, LPIPS: 0.214), surpassing either component alone. This validates our paradigm shift rather than attempting to invert blur, we must model both *how blur spatially manifests* and *how it temporally accumulates* to achieve robust SLAM under severe motion.

V. Summary

We introduce **MotionGS-SLAM**, a robust SLAM system that excels in high-motion scenarios by fundamentally shifting the approach to motion blur. Instead of treating blur as an artifact to be corrected, we reformulate it as a physical process to be generatively modeled within the rendering pipeline. Our central contribution is a novel **event-modulated Gaussian kernel**, which leverages high-frequency event data to guide this physics-based synthesis. By transforming blur from a challenge into a supervisory signal, our method achieves state-of-the-art tracking accuracy and sharp scene reconstruction directly from severely degraded images.

References

[1] Gwangtak Bae, Changwoon Choi, Hyeongjun Heo, Sang Min Kim, and Young Min Kim. I 2-slam: Inverting imaging process for robust photorealistic dense slam. In *European Conference on Computer Vision*, pages 72–89. Springer, 2024.

[2] Carlos Campos, Richard Elvira, Juan J G Rodríguez, et al. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.*, 37(6):1874–1890, 2021.

[3] Marco Cannici and Davide Scaramuzza. Mitigating motion blur in neural radiance fields with events and frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9286–9296, 2024.

[4] Siyu Chen, Shenghai Yuan, Thien-Minh Nguyen, Zhuyu Huang, Chenyang Shi, Jin Jing, and Lihua Xie. Egs-slam: Rgb-d gaussian splatting slam with events. *IEEE Robotics and Automation Letters*, 2025.

[5] Tianchen Deng, Yaohui Chen, Leyan Zhang, Jianfei Yang, Shenghai Yuan, Jiuming Liu, Danwei Wang, Hesheng Wang, and Weidong Chen. Compact 3d gaussian splatting for dense visual slam. *arXiv preprint arXiv:2403.11247*, 2024.

[6] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. *Advances in neural information processing systems*, 37:40212–40229, 2024.

[7] Hanqian Han, Jianing Li, Henglu Wei, and Xiangyang Ji. Event-3dgs: Event-based 3d reconstruction using 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:128139–128159, 2024.

[8] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21584–21593, 2024.

[9] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024.

[10] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[11] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pages 163–179. Springer, 2024.

[12] Li Ma, Xiaoyu Li, Jing Liao, et al. Deblur-nerf: Neural radiance fields from blurry images. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 12861–12870, 2022.

[13] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024.

[14] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.*, 33(5):1255–1262, 2017.

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

[16] Delin Qu, Chi Yan, Dong Wang, Jige Yin, et al. Implicit event-rgbd neural slam. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 19584–19594, 2024.

[17] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018.

[18] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[19] Gemma Taverni, Diederik Paul Moeys, Chenghan Li, Celso Cavaco, Vasyl Motsnyi, David San Segundo Bello, and Tobi Delbruck. Front and back illuminated dynamic and active pixel vision sensors comparison. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(5):677–681, 2018.

[20] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[21] Tianyi Xiong, Jiayi Wu, Botao He, Cornelia Fermuller, Yiannis Aloimonos, Heng Huang, and Christopher A Metzler. Event3dgs: Event-based 3d gaussian splatting for high-speed robot egomotion. *arXiv preprint arXiv:2406.02972*, 2024.

[22] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024.

[23] Wangbo Yu, Chaoran Feng, Jianing Li, Jiye Tang, Jiashu Yang, Zhenyu Tang, Meng Cao, Xu Jia, Yuchao Yang, Li Yuan, et al. Evagaussians: Event stream assisted gaussian splatting from blurry images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24780–24790, 2025.

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