IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

EiGS: Event-Informed 3D Deblur Reconstruction with Gaussian Splatting

Yuchen Weng[1], Nuo Li[1], Peng Yu[1], Qi Wang[1], Yongqiang Qi[2], Shaoze You[1], and Jun Wang[1]

***Abstract*—Neural Radiance Fields (NeRF) have significantly advanced photorealistic novel view synthesis. Recently, 3D Gaussian Splatting has emerged as a promising technique with faster training and rendering speeds. However, both methods rely heavily on clear images and precise camera poses, limiting performance under motion blur. To address this, we introduce Event-Informed 3D Deblur Reconstruction with Gaussian Splatting(EiGS), a novel approach leveraging event camera data to enhance 3D Gaussian Splatting, improving sharpness and clarity in scenes affected by motion blur. Our method employs an Adaptive Deviation Estimator to learn Gaussian center shifts as the inverse of complex camera jitter, enabling simulation of motion blur during training. A motion consistency loss ensures global coherence in Gaussian displacements, while Blurriness and Event Integration Losses guide the model toward precise 3D representations. Extensive experiments demonstrate superior sharpness and real-time rendering capabilities compared to existing methods, with ablation studies validating the effectiveness of our components in robust, high-quality reconstruction for complex static scenes.**

***Index Terms*—Sensor Fusion; Mapping; Deep Learning Methods**

I. Introduction

RECONSTRUCTING 3D scenes from images is a fundamental challenge in computer vision and graphics, with Neural Radiance Fields (NeRF) [1] setting a benchmark for photorealistic novel view synthesis. NeRF models scenes as continuous volumetric functions, enabling high-quality rendering from 2D inputs. However, its reliance on clear images and precise camera poses makes it highly susceptible to motion blur, a pervasive issue in capturing intricate 3D environments with unstable or rapidly moving cameras. Motion blur, caused by camera shake or rapid object motion, degrades image quality and disrupts pose estimation, leading to inaccurate and distorted 3D representations. As illustrated in Fig. 1, a blurry input image of a dynamic scene—such as a LEGO bulldozer model captured from multiple angles—is processed by a 3D Gaussian Splatting (3DGS) method.

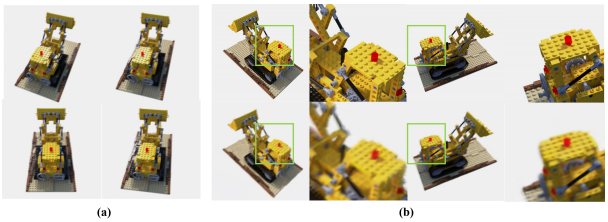

Fig. 1: Impact of motion blur on 3D Gaussian Splatting reconstruction. (a) A set of blurry inputs. (b) Resulting synthetic novel views with loss of fine details compared to the clear ground truth images above.

To mitigate motion blur in 3D reconstruction, various methods leveraging standalone RGB images have been proposed within NeRF and 3DGS frameworks. NeRF-based approaches, such as Deblur-NeRF [2], PDRF [3], and ExBluRF [4], optimize for sharp radiance fields or refine blurry inputs, achieving high-quality results but at the cost of intensive computation, often requiring hours or days. In contrast, 3D Gaussian Splatting (3DGS) [5] represents scenes with Gaussian ellipsoids, offering comparable rendering quality with faster training and real-time capabilities. However, methods like Deblur-GS [6] and neural network-based approaches [7] struggle with complex camera motion, either failing to capture intricate patterns or introducing instability by jointly modeling positional and shape changes. These limitations highlight the need for a solution that combines efficiency with accurate motion handling for scenes with motion blur.

Event cameras mitigate these challenges, offering high temporal resolution and dynamic range, which excels at capturing rapid motion. Over the past few years, extensive research has explored integrating event cameras into 3D reconstruction frameworks to leverage their high temporal resolution for motion sensing. NeRF-based methods, such as EventNeRF [8] and Qi et al.'s approach [9], utilize event streams to improve visual fidelity in blurry scenes but remain constrained by NeRF's inherent limitations, suffering from slow training and rendering speeds, which hinder real-time applications. In contrast, 3DGS-based methods, including EvaGaussian [10], E2GS, EventSplat [11], and Event3DGS [12], [13], achieve faster training and rendering, delivering high-fidelity novel views. However, these approaches often struggle with complex camera motion or scenes with intricate geometries due to reliance on interpolated poses or insufficient focus on precise

Manuscript received July 15, 2025; Revised November 2, 2025; Accepted November 26, 2025.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported in part by the Key Research and Development Task Plan Project of Xinjiang Uygur Autonomous Region (Department Office Joint Project) under Grant 2023B01006-1, in part by the National Natural Science Foundation of China under Grant 52574209 and 52504178.

[1]Y. Weng, N. Li, P. Yu, Q. Wang, S. You, and J. Wang are with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China.tb23810008a41@cumt.edu.cn (Corresponding author: Jun Wang.)

[2]Y. Qi is with the School of Mathematics, China University of Mining and Technology, Xuzhou, China.qiyongqiang3@163.com

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

deblurring.

To overcome these limitations, we propose **Event-Informed 3D Deblur Reconstruction with Gaussian Splatting(EiGS)**, a novel framework that integrates event camera data with 3DGS to achieve robust, real-time 3D reconstruction from inputs with severe motion blur, such as city roads captured from a moving vehicle, ensuring broad applicability. By combining the inherent efficiency of 3DGS with the high temporal resolution of event streams, EiGS compensates for camera motion during exposure, producing sharp and coherent 3D scene representations. Unlike prior 3DGS-based methods that jointly estimate positional and shape changes, our approach focuses solely on Gaussian positional displacements, avoiding the instability introduced by rotation or scale modeling. We introduce the Adaptive Deviation Estimator (ADE) network, which leverages event data to effectively model motion blur dynamics through Gaussian offsets. Additionally, our motion consistency loss enforces coherent displacements across the entire scene, ensuring stability and fidelity even in complex geometric structures.

Our contributions are summarized as follows:

- We propose EiGS, a framework that combines event camera data with 3DGS to enable high-fidelity 3D reconstruction of scenes affected by severe motion blur.
- We introduce a novel motion consistency loss to enforce coherent Gaussian displacements across the entire scene, significantly improving reconstruction stability.
- We develop an Adaptive Deviation Estimator (ADE) network to robustly model camera shake during exposure, focusing on positional dynamics for enhanced precision.

II. Related work

A. Deblurring 3D Reconstruction with NeRF

Neural Radiance Fields (NeRF) [1] have transformed 3D reconstruction by representing scenes as continuous volumetric functions, excelling in photorealistic rendering. However, NeRF assumes static scenes with precise poses, making it sensitive to motion blur. NeRF-based approaches, such as Deblur-NeRF [2], PDRF [3], and ExBluRF [4], integrate blur modeling or refine inputs but suffer from prohibitively high computational costs. Despite their success, these methods inherit NeRF's computational complexity, requiring extensive training and rendering times that hinder real-time use. Event-augmented NeRF variants leverage event cameras to enhance reconstruction. EventNeRF [8] uses event streams for colored 3D outputs, and E^2NeRF [9] combines blurry frames with event data via the Event Double Integral (EDI) [14] and COLMAP [15] poses. Robust e-NeRF [16] tackles sparse and noisy events under non-uniform motion, employing a realistic event generation model with threshold-normalized difference and gradient losses for improved robustness. Ev-NeRF [17] uses multi-view consistency as a self-supervision signal to filter noise in low-light or extreme-motion data, producing integrated neural volumes over 2–4 seconds. E-NeRF [18] estimates NeRF from fast-moving event cameras, recovering volumetric representations under severe blur. Yet, these methods retain NeRF's efficiency drawbacks and struggle with real-time applicability, issues that EiGS addresses with its efficient 3DGS framework and Adaptive Deviation Estimator (ADE) network.

B. Deblurring 3D Reconstruction with Gaussian Splatting

3D Gaussian Splatting (3DGS) [5] has emerged as a powerful alternative to NeRF, yet it struggles with motion-blurred imagery. Initial solutions like Deblur-GS [19] use pose interpolation, which is often insufficient for complex, non-linear camera motion. Other methods, such as Deblurring 3D Gaussian Splatting [7], learn per-Gaussian shape and position changes, but this joint optimization can introduce instability and degrade performance, a finding consistent with our own observations.

Recent works have shifted towards modeling a single, global camera motion for each frame. A pioneering approach in this area is BAD-Gaussians [20], which integrates a bundle adjustment framework to optimize a continuous camera trajectory represented by a cubic B-spline. Similarly, Robust Gaussian Splatting [21] enhances robustness by treating motion blur as a probabilistic distribution over camera poses, unifying pose refinement and deblurring. While effective, these methods are based purely on RGB data.

The integration of event cameras offers a promising direction. For instance, EvaGaussians [10] and E2GS [22] use event-processed images to improve initialization and rendering. A direct competitor, EBAD-Gaussian [23], advances this by using the event stream to directly supervise the optimization of the global camera trajectory within the bundle adjustment framework. Concurrently, physics-informed methods [24], [25] focus on forward simulation, which is distinct from our inverse problem of deblurring.

In contrast, our EiGS carves out a unique position. Instead of optimizing a single global camera trajectory like BAD-Gaussian [20] and EBAD-Gaussian [23], we learn a per-Gaussian displacement field using our Adaptive Deviation Estimator (ADE). This local modeling offers greater flexibility. To ensure physical plausibility, we introduce the Motion Consistency Loss, which constrains these local displacements to a best-fitting global rigid motion. This acts as a lightweight physical prior, distinguishing our work from complex forward simulators like PhysGaussian [24]. Furthermore, our unique use of event data for both robust initialization and high-frequency motion supervision allows our framework to achieve a compelling balance of flexibility, stability, and performance.

III. Method

As depicted in Fig. 2, our proposed EiGS framework reconstructs 3D scenes from a blurry RGB image and its corresponding event stream. The framework is organized into two main stages: In the Initialization Stage, we leverage the event stream as a crucial aid for robust initialization. The events are first processed by the Event Double Integral (EDI) technique to generate latent sharp images. This step is vital, as it allows COLMAP to successfully estimate camera poses and create an initial sparse point cloud from images that would otherwise be too blurry for standard Structure-from-Motion(SfM) methods. This provides a solid foundation of

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

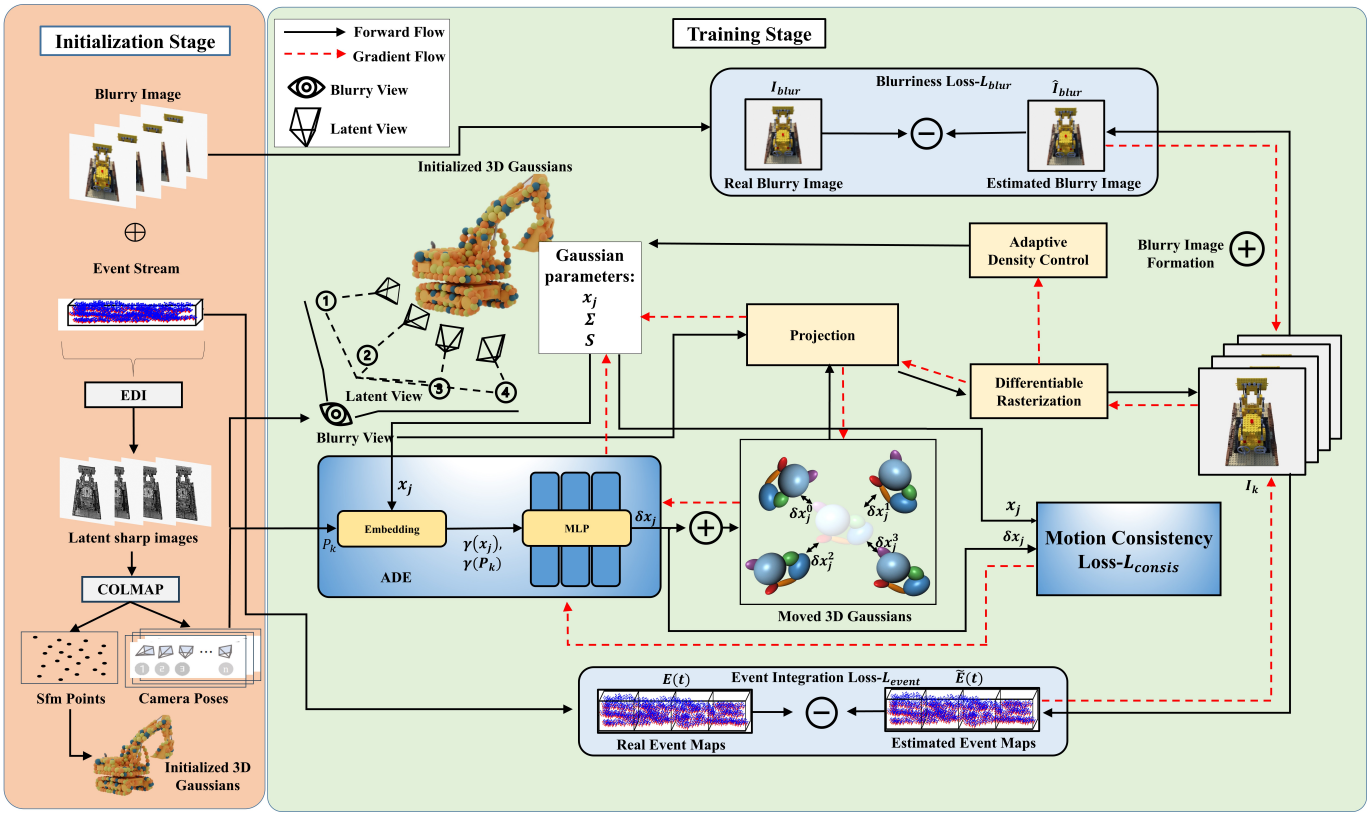

Fig. 2: Overview of our EiGS framework for 3D reconstruction from a blurry image and an event stream. The framework is divided into two main parts: an Initialization Stage and a Training Stage. In the Initialization Stage, we use the blurry image and event data to robustly initialize a set of 3D Gaussians via EDI and COLMAP. In the Training Stage, the Adaptive Deviation Estimator (ADE) predicts per-Gaussian displacements to model motion blur. The final 3D scene is optimized by minimizing three losses: a Blurriness Loss between the rendered and real blurry images, a Motion Consistency Loss to enforce a global rigid motion, and an Event Integration Loss that uses event data to refine the motion estimation.

3D Gaussians for the next stage. The Training Stage then optimizes these Gaussians. The initial Gaussian parameters are passed to the Adaptive Deviation Estimator (ADE), which predicts a positional displacement $\delta \mathbf{x}_j$ for each Gaussian to model the camera's motion during exposure. To ensure the predicted motion is physically plausible and consistent, we introduce two constraints. First, the Motion Consistency Loss ($\mathcal{L}_{consis}$) enforces that all individual displacements adhere to a single global rigid motion model. Second, the Event Integration Loss ($\mathcal{L}_{event}$), computed from the event stream, provides direct supervisory signals. As shown by the dashed red lines, the gradients from this loss refine the ADE's predictions, guiding it to better capture complex camera shake. Finally, to optimize the scene's appearance, the moved Gaussians are rendered into multiple sharp views (I_k). These views are averaged to form the estimated blurry image ($\hat{I}_{blur}$), which is compared against the real blurry image using the Blurriness Loss ($\mathcal{L}_{blur}$). The gradients from all three losses are back-propagated to jointly optimize the ADE network and the Gaussian parameters, resulting in a sharp, high-fidelity 3D reconstruction.

A. Adaptive Deviation Estimator

Traditional multi-frame rendering methods simulate blur by averaging frames, but their simple interpolation (e.g., linear) fails to capture the complex, non-linear motion distributions of real-world camera jitter. To overcome this limitation, we propose the Adaptive Deviation Estimator network, a learning-based approach designed to capture intricate motion dynamics for intermediate frames. The ADE network predicts per-Gaussian positional deviations to model the inverse of camera jitter during exposure, representing complex, non-linear motions across the scene. This per-Gaussian approach is essential, as assigning a uniform offset to all Gaussians would fail to capture intricate blur patterns caused by varying depths and motions; instead, ADE learns individualized shifts guided by event streams for precise simulation. Instead, ADE learns individualized shifts guided by the high temporal resolution of event streams, enabling it to resolve high-frequency motion patterns that traditional RGB-based methods cannot. For instance, in cases of abrupt camera shake, event-driven temporal cues significantly enhance ADE's ability to simulate plausible camera trajectories.

While the general concept of using a neural network to predict displacements is shared with methods like Deblurring-GS [7], our approach is fundamentally different in its design and inputs. First and foremost, ADE is specifically engineered to leverage event camera data, a crucial distinction from RGB-only methods like Deblurring-GS. This allows us to

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

model motion with much higher temporal fidelity. Second, our network utilizes a richer input representation. Instead of relying on a single main camera pose, the ADE module is conditioned on the full set of latent poses $\{\mathbf{P}_k\}_{k=0}^{N}$ estimated by EDI and COLMAP, providing a more comprehensive initial guess of the camera's trajectory. Third, we deliberately focus on positional deviations and exclude shape deformations (e.g., rotation and scaling). This is a strategic choice to enhance stability, not a simplification.

The ADE network, implemented as a three-layer Multi-Layer Perceptron (MLP) $\mathcal{F}_\theta$, takes as input COLMAP-estimated camera poses $\{\mathbf{P}_k\}_{k=0}^{N}$ and initial Gaussian centers x_j, both encoded via a frequency encoding function γ. For each Gaussian, ADE predicts N positional deviations:

$$\{\delta x_j\}_{j=1}^{N_G} = \mathcal{F}_\theta\big(\gamma(x_j), \gamma(\{\mathbf{P}_k\}_{k=0}^{N})\big) \tag{1}$$

where N_G is the total number of Gaussians, and δx_j denotes the deviation for the j-th Gaussian across N latent poses. These deviations adjust the Gaussian positions for each intermediate frame as $\hat{x}_j^{(k)} = x_j + \lambda_p \delta x_j^{(k)}$, producing N sets of moved Gaussians. These are rendered into $N+1$ sharp images $\{I_k\}_{k=0}^{N}$ by using original 3DGS's differential rasterization, including a reference image from the original Gaussian positions ($k=0$). The rendered images are averaged to simulate the blurry image:

$$\hat{I}_{\text{blur}} = \frac{1}{N+1}\sum_{k=0}^{N} I_k, \quad I_k = \text{Render}(\{G(\hat{x}_j^{(k)})\}_{j=1}^{N_G}) \tag{2}$$

In summary, the ADE module leverages the full set of original Gaussian positions and all hidden perspective poses predicted by EDI and COLMAP as inputs, enabling it to predict individualized positional offsets for each Gaussian. These offsets, appended to the original positions, yield multiple shifted Gaussian sets that represent the inverse of the camera's jitter during exposure, ensuring higher accuracy thanks to the initialized hidden poses. Furthermore, photometric differences between consecutively rendered predicted images are used to generate predicted event integral maps, supervised by real event streams, which refines ADE's focus on motion information and justifies omitting Gaussian shape parameters as inputs or attempting shape predictions. Ultimately, as only the original Gaussian parameters are optimized during training, inference bypasses multi-frame rendering and ADE computations, maintaining the high-speed rendering of 3DGS.

B. Loss Functions

Our optimization combines three loss functions, with the novel motion consistency loss ensuring accurate camera motion parsing.

1) Motion Consistency Loss: While the ADE network's ability to estimate independent displacements for each Gaussian is powerful for capturing complex camera motion, it can also introduce inconsistencies. Without a global constraint, the predicted per-Gaussian displacements $\delta\mathbf{x}_j$ might result in chaotic motion patterns that are physically implausible, thereby undermining the geometric integrity of the reconstructed scene.

To address this, we introduce the Motion Consistency Loss ($\mathcal{L}_{consis}$). This loss enforces a globally consistent motion model, ensuring that the displacements of all Gaussian centers adhere to a single rigid body transformation. This strikes a balance between modeling complex, non-linear motion at a local level and maintaining global geometric stability. The principle of this loss is illustrated in Figure 3, which contrasts the chaotic, independent displacements with the globally rigid motion enforced by our approach.

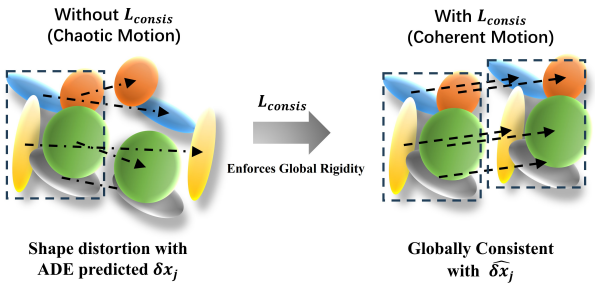

Fig. 3: The effect of the Motion Consistency Loss ($\mathcal{L}_{consis}$). **Left:** Without the loss, per-Gaussian displacements ($\delta\mathbf{x}_j$) predicted by the ADE can result in non-rigid, chaotic motion that distorts the scene's structure. **Right:** By enforcing a single rigid body transformation, $\mathcal{L}_{consis}$ ensures that all displacements ($\widehat{\delta\mathbf{x}}_j$) are globally consistent, preserving the geometric integrity of the object.

We model this global motion as a single rigid body transformation from the Lie algebra se(3), which can be parameterized by a motion vector (or twist) $\xi = [\omega, \nu] \in \mathbb{R}^6$. Here, $\omega = [\omega_x, \omega_y, \omega_z]$ is the angular velocity and $\nu = [\nu_x, \nu_y, \nu_z]$ is the linear velocity. For a given initial Gaussian center $\mathbf{x}_j = [x_j, y_j, z_j]$, the displacement $\widehat{\delta\mathbf{x}}_j$ predicted by this global model is:

$$\widehat{\delta\mathbf{x}}_j = -\omega \times \mathbf{x}_j - \nu \tag{3}$$

The core objective of our loss is to measure how much the ADE's predictions deviate from a single, best-fitting global rigid transformation. This approach explicitly enforces a global alignment across all Gaussians, in contrast to a local smoothness constraint which would only regularize neighboring displacements. To find this best-fitting transformation, we determine the optimal motion vector ξ^* that collectively minimizes the error for the entire set of predicted displacements $\{\delta\mathbf{x}_j\}_{j=1}^{N_G}$. This is achieved by formulating a single linear system $\mathbf{A}\xi = \mathbf{b}$. As shown in our framework diagram, the inputs to build this system are the original Gaussians' positions $\mathbf{x}_j$ and the ADE predicted deviations $\delta\mathbf{x}_j$. Specifically, $\mathbf{b} \in \mathbb{R}^{3N_G}$ is the flattened vector of all predicted displacements $\delta\mathbf{x}_j$. The coefficient matrix $\mathbf{A} \in \mathbb{R}^{3N_G \times 6}$ is constructed from the coordinates of each Gaussian center, with each row corresponding to the expanded form of Equation 3:

$$\begin{aligned} \Delta x_j &= y_j\omega_z - z_j\omega_y - \nu_x \\ \Delta y_j &= z_j\omega_x - x_j\omega_z - \nu_y \\ \Delta z_j &= x_j\omega_y - y_j\omega_x - \nu_z \end{aligned} \tag{4}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

With $\mathbf{A}$ and $\mathbf{b}$ constructed, we solve for the optimal ξ^* using regularized least squares:

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \|\mathbf{A}\xi - \mathbf{b}\|_2^2 + \lambda_{reg}\|\xi\|_2^2 \tag{5}$$

where λ_{reg} is a small regularization weight. We then recompute the globally consistent displacements for all Gaussians using the solved global motion: $\widehat{\delta\mathbf{x}}_j = -\omega^* \times \mathbf{x}_j - \nu^*$.

Finally, the Motion Consistency Loss is defined as the Huber loss between the ADE's predicted displacements and the displacements derived from our best-fitting rigid transform, ensuring robustness against outliers:

$$\mathcal{L}_{consis} = \frac{1}{N_G}\sum_{j=1}^{N_G} L_\delta(\delta\mathbf{x}_j - \widehat{\delta\mathbf{x}}_j) \tag{6}$$

where L_δ is the Huber loss function with a threshold $\delta = 0.1$. This loss effectively guides the ADE to predict displacements that are not only locally accurate but also globally coherent.

2) Blurriness Loss: To model motion blur, we average rendered sharp images:

$$\hat{I}_{\text{blur}} = \frac{1}{b+1}\sum_{i=0}^{b} I_i, \quad I_i = \text{Rasterize}(\{G(\hat{x}_j^{(i)})\}_{j=1}^{N_G}) \tag{7}$$

and compute:

$$L_{\text{blur}} = (1 - \lambda_{\text{D-SSIM}})\|I_{\text{blur}} - \hat{I}_{\text{blur}}\|_1 + \lambda_{\text{D-SSIM}} L_{\text{D-SSIM}} \tag{8}$$

with $\lambda_{\text{D-SSIM}} = 0.2$.

3) Event Integration Loss: We integrate event polarities between frames:

$$\mathbf{E}(t) = \int_{t_0}^{t_0+\delta t} \mathbf{e}(t)\, dt \tag{9}$$

and compare with the estimated map $\tilde{\mathbf{E}}(t)$:

$$L_{\text{event}} = \|\mathbf{E}(t) - \tilde{\mathbf{E}}(t)\|_1 \tag{10}$$

4) Final Loss Function: The total loss is:

$$L = L_{\text{blur}} + \lambda_{\text{event}} L_{\text{event}} + \lambda_{\text{consis}} L_{\text{consis}} \tag{11}$$

with $\lambda_{\text{event}} = 0.005$, $\lambda_{\text{consis}} = 1.0$, enabling robust deblurring via multi-frame rendering.

IV. Experiments

A. Implementation Details

Our code builds upon 3D Gaussian Splatting [5] and Deblurring-GS [7]. We utilize the Adam optimizer, setting the learning rate to 1e-3 for the MLP and 1.6e-3 for the positions of the 3D Gaussians. The Gaussian pruning threshold and densification threshold are set to 1e-2 and 5e-4, respectively. The ADE module's network consists of an input frequency encoding followed by a three-layer MLP for pose correction δx estimation. Each layer contains 256 hidden units, employs ReLU activation for non-linearity, and is initialized using Xavier initialization. The scaling parameters for position corrections (λ_p) are set to 1e-2. For the loss function, the weight of the Event Integration Loss (λ_{event}) is set to 0.005, while the weight of the SSIM loss (λ_{D-SSIM}) is set to 0.2 for the Blurriness Loss and the weight of motion consistency loss is set to 1.0. All experiments were conducted on an NVIDIA RTX 4080 GPU.

B. Comparisons and Results

To evaluate the effectiveness of our proposed EiGS method, we conducted a comprehensive comparison with several deblurring 3D reconstruction approaches, including the original 3DGS [5], Deblurring-GS [7], BAD-Gaussian [20], E2GS [22], EvaGaussian [10], BeSplat [26] and NeRF-based deblurring methods such as Deblur-NeRF [2], EDI-NeRF [14], E^{2}NeRF [9], and the original NeRF [1]. It is important to note that for some NeRF-based methods, such as EDI-NeRF, where it was established as a baseline by training a standard NeRF on images deblurred by the EDI [14] technique. We utilized the E^{2}NeRF dataset, which comprises five real-world scene image sequences and six synthetic scene image sequences, providing a robust testbed for evaluating reconstruction quality under motion blur. All reported results represent the average of the final experimental outcomes across these sequences. For real-world scenes, due to the absence of clear ground truth images, we employed the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE↓), where a lower score indicates better perceived image quality with less distortion. For synthetic scenes, we used several standard metrics for precise quantitative assessment: Peak Signal-to-Noise Ratio (PSNR↑) and Structural Similarity Index (SSIM↑), where higher values indicate better reconstruction quality, and AlexNet-based Learned Perceptual Patch Similarity (LPIPS↓), where lower values indicate better perceptual similarity. Unlike trajectory-based methods , we model blur via per-Gaussian deviations. Given the absence of real-world ground truth poses, we validate motion modeling implicitly through reconstruction fidelity. A crucial setting difference exists on E^{2}NeRF synthetic data: methods like E2GS [22] use ground-truth poses, while our method is tested with poses estimated from blurry images (via EDI+COLMAP). To ensure a fair robustness comparison, we re-ran E2GS using our same estimated poses, noted as 'E2GS(w/ Est. Poses)' in Table I. Furthermore, to validate our design, we trained a variant that jointly optimizes Gaussian shape (scaling and rotation) and position which is presented as 'Deblurring-GS+' in Table I. We observed that simultaneous optimization under severe blur introduces ambiguity and instability. Thus, modeling motion solely via positional deviations proves more robust.

The results, detailed in the accompanying table, highlight EiGS's superior performance in real-world scenes, achieving the best BRISQUE score and significantly outperforming competing methods. In synthetic scenes, while E^{2}NeRF leads with higher metric scores, our method delivers highly competitive results, closely approaching its performance. Notably, EiGS shows a substantial improvement over Deblurring-GS, particularly in capturing high-frequency motion dynamics, which underscores the effectiveness of integrating event data in enhancing reconstruction quality.

Although E²NeRF [9] achieves higher metric scores on synthetic data, our method, based on 3D Gaussian Splatting, offers substantial advantages in computational efficiency. A critical distinction is that while our method uses multi-frame rendering during training, at inference time it only requires a single forward pass of the trained 3D Gaussian model, thus

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: Quantitative analysis. All experiments are tested on real-world and synthetic scenes from the E^{2}NeRF dataset. We use red to mark the best result and blue for the second best. For all metrics except LPIPS and BRISQUE, higher is better.

real-world dataset experiments										
Metrics	**NeRF [1]**	**Deblur-NeRF [2]**	**EDI-NeRF [9]**	**E^{2}NeRF [9]**	**3DGS [5]**	**Deblurring-GS [7]**	**E2GS [22]**	**BeSplat [26]**	**BAD-Gaussian [20]**	**Ours**
Brisque↓	44.66	38.41	31.98	30.26	38.89	27.51	32.50	36.03	32.10	26.08

synthetic dataset experiments											
Metrics	**NeRF [1]**	**Deblur-NeRF [2]**	**EDI-NeRF [9]**	**E^{2}NeRF [9]**	**3DGS [5]**	**Deblurring-GS [7]**	**Deblurring-GS+**	**E2GS(w/ Est. Poses) [22]**	**EvaGaussian [10]**	**BAD-Gaussian [20]**	**Ours**
PSNR↑	22.27	19.93	27.71	29.56	20.27	24.44	22.72	23.47	19.48	20.61	28.21
SSIM↑	0.90	0.86	0.95	0.96	0.86	0.85	0.84	0.84	0.81	0.70	0.95
LPIPS↓	0.15	0.26	0.09	0.07	0.15	0.14	0.15	0.14	0.21	0.12	0.08

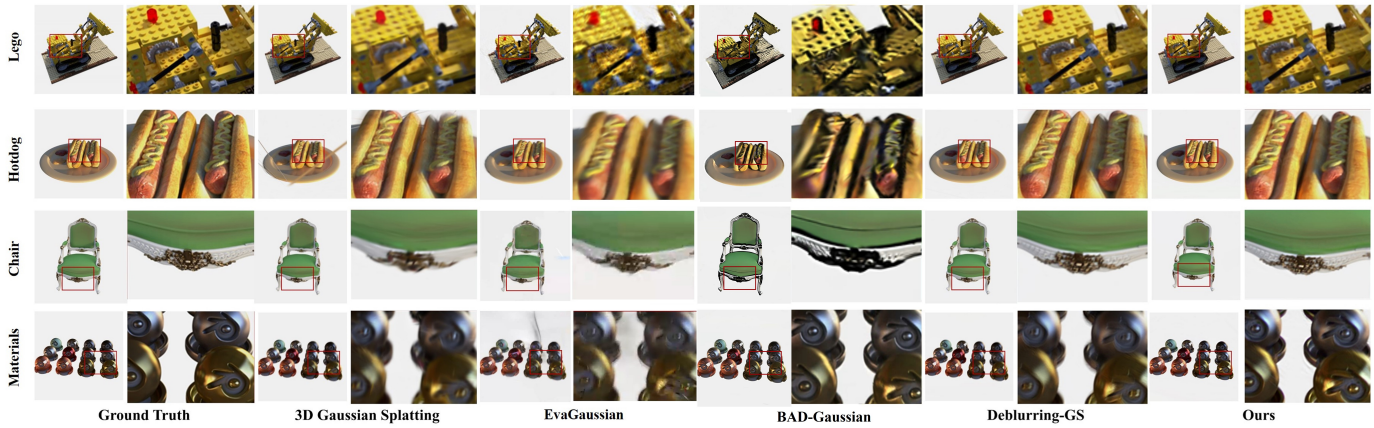

Fig. 4: Qualitative results on E^{2}NeRF synthetic dataset.

maintaining the high-speed rendering capabilities of 3DGS. Quantitatively, our method achieves an average rendering speed of 190 FPS, with an average training time of approximately 3 hours and 25 minutes per scene. In sharp contrast, NeRF-based methods like E²NeRF [9] are computationally intensive, often requiring up to 2 days of training time and achieving rendering speeds as low as 0.04 FPS, making them unsuitable for real-time applications. This efficiency, combined with its superior deblurring performance in real-world scenarios and comparable quality metrics in synthetic scenes, positions EiGS as a compelling and practical solution for high-fidelity 3D reconstruction in complex static scenes.

Qualitative comparisons in Fig. 4 (synthetic) and Fig. 5 (real-world) confirm our method's advantages. For instance, on the synthetic 'materials' scene, our method recovers intricate textures lost by others, and on the real-world 'plant' scene, it renders sharp, well-defined leaf structures, underscoring its robust performance. It is important to note that BAD-Gaussians [20] achieves sharpness via optimized, approximate 'virtual viewpoints.' However, this optimization lacks stability, leading to quality degradation in other views, which ultimately results in inferior BRISQUE scores.

C. Ablation studies

The Contribution of Losses and ADE Module. To evaluate the contribution of each innovative module in our EiGS framework, we conducted ablation experiments on the all six scenes from the E^{2}NeRF synthetic dataset. The results, presented in Table II, report PSNR and SSIM metrics for various configurations, demonstrating the impact of our proposed modules and the use of event stream data. Starting from the 3DGS baseline (20.27 PSNR), configuration (a) introduces our ADE module with $\mathcal{L}_{blur}$, achieving a significant +3.83 dB gain. This demonstrates the significant effectiveness of our per-Gaussian deviation learning for modeling motion. From this new baseline, we can isolate the effects of our two proposed losses. Adding the Event Integration Loss ($\mathcal{L}_{event}$) (configuration (b)) further improves performance to 25.50 PSNR (+1.40 dB). Similarly, adding only the Motion Consistency Loss ($\mathcal{L}_{consis}$) (configuration Ours-) improves performance to 24.52 PSNR (+0.42 dB). The most significant improvement, however, comes from combining both losses in our Ours (full) model, which reaches 28.21 PSNR. This result demonstrates a strong synergistic effect: adding $\mathcal{L}_{consis}$ to the event-guided model (Ours vs. b) yields a +2.71 dB gain, while adding $\mathcal{L}_{event}$ to the consistency-guided model (Ours vs. Ours-) yields an even larger +3.69 dB gain. These findings strongly indicate that both $\mathcal{L}_{event}$ and $\mathcal{L}_{consis}$ are critical and highly complementary. The qualitative results in Fig. 6 reinforce this, showing how our full model (Ours) avoids both the structural distortions seen in the model without $\mathcal{L}_{consis}$ (Ours-) and the blurring of fine details seen in the model without $\mathcal{L}_{event}$ (b).

The Number of Latent Poses. As shown in Table III, 6 latent poses offers the best quality-to-time ratio (28.81 PSNR / 3h28m). Increasing to 8 poses significantly increases training time (over 2 hours) for a marginal gain (+0.12 PSNR). Our main experiments used 4 poses. While our robust ablation reveals 6 poses could offer a modest boost, our 4-pose configuration already demonstrates competitive performance and significant improvement over baselines, thus sufficiently validating our framework.

The Width and Depth of MLP Hidden Layers. We also examined the MLP hidden layer configurations in the ADE network, with results shown in Table IV and Table V. As

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

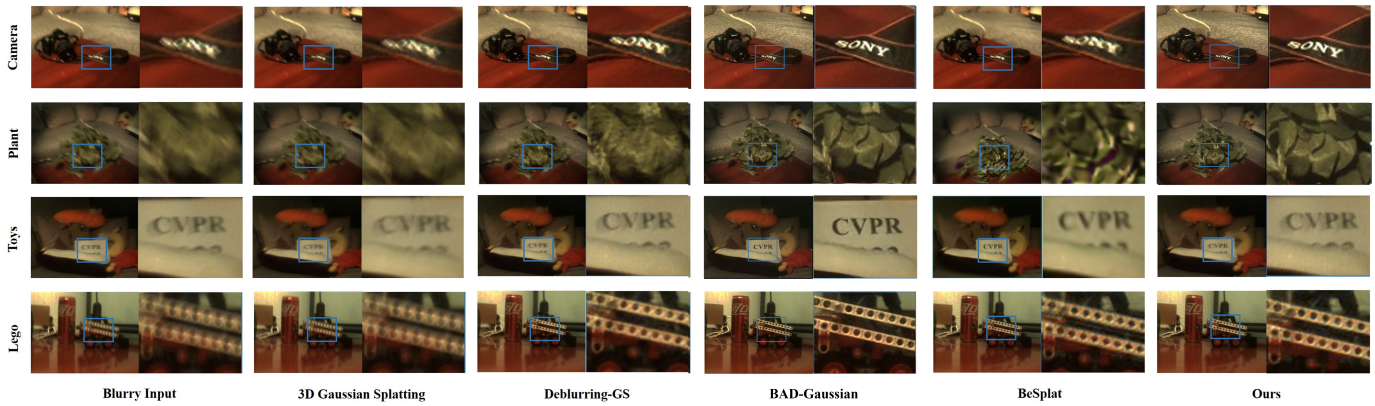

Fig. 5: Qualitative results on E^{2}NeRF real-world dataset.

TABLE II: Ablation Studies to validate the effectiveness of losses and ADE module. "+" represents the configuration with specific loss or module. "-" represents the configuration without specific loss or module.

	Configurations				Metrics	
Methods	L_{blur}	L_{event}	ADE	L_{consis}	PSNR↑	SSIM↑
3DGS	-	-	-	-	20.27	0.8653
(a)	+	-	+	-	24.10	0.8775
(b)	+	+	+	-	25.50	0.9009
Ours-	+	-	+	+	24.52	0.8868
Ours	+	+	+	+	28.21	0.9452

Note: (a) represents the baseline deblurring 3D Gaussian Splatting method. (b) represents the Deblurring-GS augmented with the event-integrated image reconstruction module and Event Integration Loss.

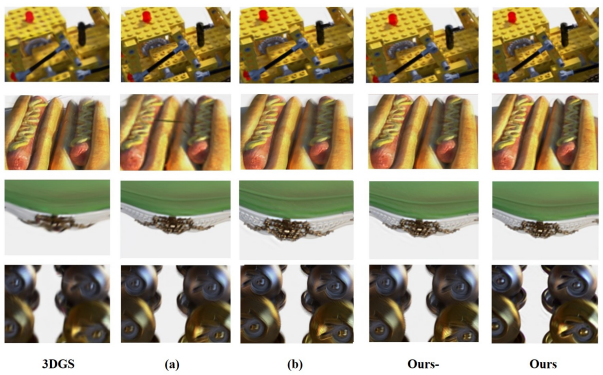

Fig. 6: Qualitative for ablation studies.

TABLE III: Ablation study to explore the effect of different latent poses number.

Latent Poses Number	**4**	**6**	**8**
PSNR↑	28.21	28.81	28.93
SSIM↑	0.9452	0.9529	0.9540
Training time↓	3h25m	3h28m	5h49m

shown in Table IV, performance peaks at 256 hidden units; using more units (512) slightly degrades performance while maintaining a similar training time. As shown in Table V,

TABLE IV: Ablation study to explore the effect of different layer width numbers.

Layer Width	**64**	**128**	**256**	**512**
PSNR↑	26.55	27.02	28.21	28.05
SSIM↑	0.9135	0.9303	0.9452	0.9473
Training time↓	3h3m	3h16m	3h25m	3h25m

TABLE V: Ablation study to explore the effect of different MLP network depths.

Layer Depth	**2**	**3**	**4**
PSNR↑	27.12	28.21	28.18
SSIM↑	0.9389	0.9452	0.9536
Training time↓	3h37m	3h25m	3h26m

using 3 layers (28.21 PSNR) provides a significant boost over 2 layers (27.12 PSNR) with a slight *decrease* in training time. Increasing to 4 layers offers no significant benefit. Therefore, we adopted a three-layer MLP with 256 hidden units per layer as the optimal configuration that clearly balances reconstruction quality and training efficiency.

V. Conclusion

In this paper, we present EiGS, a novel framework that integrates event camera data with 3D Gaussian Splatting to achieve high-fidelity 3D reconstruction under severe motion blur. By introducing the Adaptive Deviation Estimator (ADE) network, we enable precise estimation of latent poses during exposure through Gaussian positional deviations, leveraging the high temporal resolution of event streams. Our proposed Motion Consistency Loss further enhances reconstruction stability by enforcing coherent Gaussian displacements across complex static scenes, while the Event Integration Loss and Blurriness Loss refine the deblurring process. Extensive experiments on the E^{2}NeRF dataset demonstrate that EiGS outperforms existing methods, such as Deblurring-GS [7] and E^{2}NeRF [9], in real-world scenarios with superior BRISQUE scores, and achieves competitive results in synthetic scenes with real-time rendering capabilities. Ablation studies validate the effectiveness of each component, confirming the critical

role of event data in capturing high-frequency motion dynamics.

VI. Limitations and future work

While our proposed method, EiGS, demonstrates significant advancements in event-informed 3D deblur reconstruction, we acknowledge several limitations that also open promising avenues for future research. The primary limitation lies in the motion model underpinning our Motion Consistency Loss, which assumes a single, global rigid body transformation. While our Adaptive Deviation Estimator (ADE) effectively models the complex non-linear dynamics of camera jitter, the consistency loss restricts the scene to a rigid assumption. Consequently, our framework is specifically designed for static scenes and is less suited for dynamic environments containing multiple independently moving objects, as a single global motion constraint cannot simultaneously account for divergent object trajectories. Furthermore, our framework operates under an idealized sensor model, without explicitly accounting for real-world sensor artifacts like signal noise or temporal randomness in the event streams. Future work could address these challenges by developing more sophisticated non-rigid or per-object motion models to disentangle camera and object motion, as well as by integrating realistic sensor noise models into the training pipeline to enhance robustness on real-world data. We believe that tackling these aspects will further advance the state-of-the-art in high-fidelity 3D reconstruction.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] L. Ma, X. Li, J. Liao, Q. Zhang, X. Wang, J. Wang, and P. V. Sander, "Deblur-nerf: Neural radiance fields from blurry images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 861–12 870.

[3] C. Peng and R. Chellappa, "Pdrf: progressively deblurring radiance field for fast scene reconstruction from blurry images," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 2029–2037.

[4] D. Lee, J. Oh, J. Rim, S. Cho, and K. M. Lee, "Exblurf: Efficient radiance fields for extreme motion blurred images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 639–17 648.

[5] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

[6] J. Oh, J. Chung, D. Lee, and K. M. Lee, "Deblurgs: Gaussian splatting for camera motion blur," *arXiv preprint arXiv:2404.11358*, 2024.

[7] B. Lee, H. Lee, X. Sun, U. Ali, and E. Park, "Deblurring 3d gaussian splatting," *arXiv preprint arXiv:2401.00834*, 2024.

[8] V. Rudnev, M. Elgharib, C. Theobalt, and V. Golyanik, "Eventnerf: Neural radiance fields from a single colour event camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4992–5002.

[9] Y. Qi, L. Zhu, Y. Zhang, and J. Li, "E^2nerf: Event enhanced neural radiance fields from blurry images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 254–13 264.

[10] W. Yu, C. Feng, J. Tang, X. Jia, L. Yuan, and Y. Tian, "Evagaussians: Event stream assisted gaussian splatting from blurry images," *arXiv preprint arXiv:2405.20224*, 2024.

[11] T. Yura, A. Mirzaei, and I. Gilitschenski, "Eventsplat: 3d gaussian splatting from moving event cameras for real-time rendering," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 26 876–26 886.

[12] T. Xiong, J. Wu, B. He, C. Fermuller, Y. Aloimonos, H. Huang, and C. A. Metzler, "Event3dgs: Event-based 3d gaussian splatting for high-speed robot egomotion," *arXiv preprint arXiv:2406.02972*, 2024.

[13] H. Han, J. Li, H. Wei, and X. Ji, "Event-3dgs: Event-based 3d reconstruction using 3d gaussian splatting," *Advances in Neural Information Processing Systems*, vol. 37, pp. 128 139–128 159, 2024.

[14] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai, "Bringing a blurry frame alive at high frame-rate with an event camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6820–6829.

[15] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] W. F. Low and G. H. Lee, "Robust e-nerf: Nerf from sparse & noisy events under non-uniform motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 335–18 346.

[17] I. Hwang, J. Kim, and Y. M. Kim, "Ev-nerf: Event based neural radiance field," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 837–847.

[18] W. F. Low and G. H. Lee, "Robust e-nerf: Nerf from sparse & noisy events under non-uniform motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 18 335–18 346.

[19] W. Chen and L. Liu, "Deblur-gs: 3d gaussian splatting from camera motion blurred images," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 7, no. 1, May 2024. [Online]. Available: https://doi.org/10.1145/3651301

[20] L. Zhao, P. Wang, and P. Liu, "Bad-gaussians: Bundle adjusted deblur gaussian splatting," in *Computer Vision – ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds. Cham: Springer Nature Switzerland, 2025, pp. 233–250.

[21] F. Darmon, L. Porzi, S. Rota-Bulò, and P. Kontschieder, "Robust gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2404.04211

[22] H. Deguchi, M. Masuda, T. Nakabayashi, and H. Saito, "E2gs: Event enhanced gaussian splatting," in *2024 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2024, pp. 1676–1682.

[23] Y. Deng, Y. Wang, R. Xiao, C. Tang, J. Zhou, J. Fan, D. Xiong, J. Lv, and H. Tang, "Ebad-gaussian: Event-driven bundle adjusted deblur gaussian splatting," 2025. [Online]. Available: https://arxiv.org/abs/2504.10012

[24] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang, "Physgaussian: Physics-integrated 3d gaussians for generative dynamics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 4389–4398.

[25] J. Li, Z. Song, S. Zhou, and B. Yang, "Freegave: 3d physics learning from dynamic videos by gaussian velocity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 12 433–12 443.

[26] G. R. Matta, T. Reddypalli, and K. Mitra, "Besplat: Gaussian splatting from a single blurry image and event stream," in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV) Workshops*, February 2025, pp. 917–927.