This paper has been accepted for publication in the IEEE Transactions on Robotics (T-RO), 2026. © IEEE

# Event-Aided Sharp Radiance Field Reconstruction for Fast-Flying Drones

Rong Zou*, Marco Cannici*, and Davide Scaramuzza
Robotics and Perception Group, University of Zurich, Switzerland

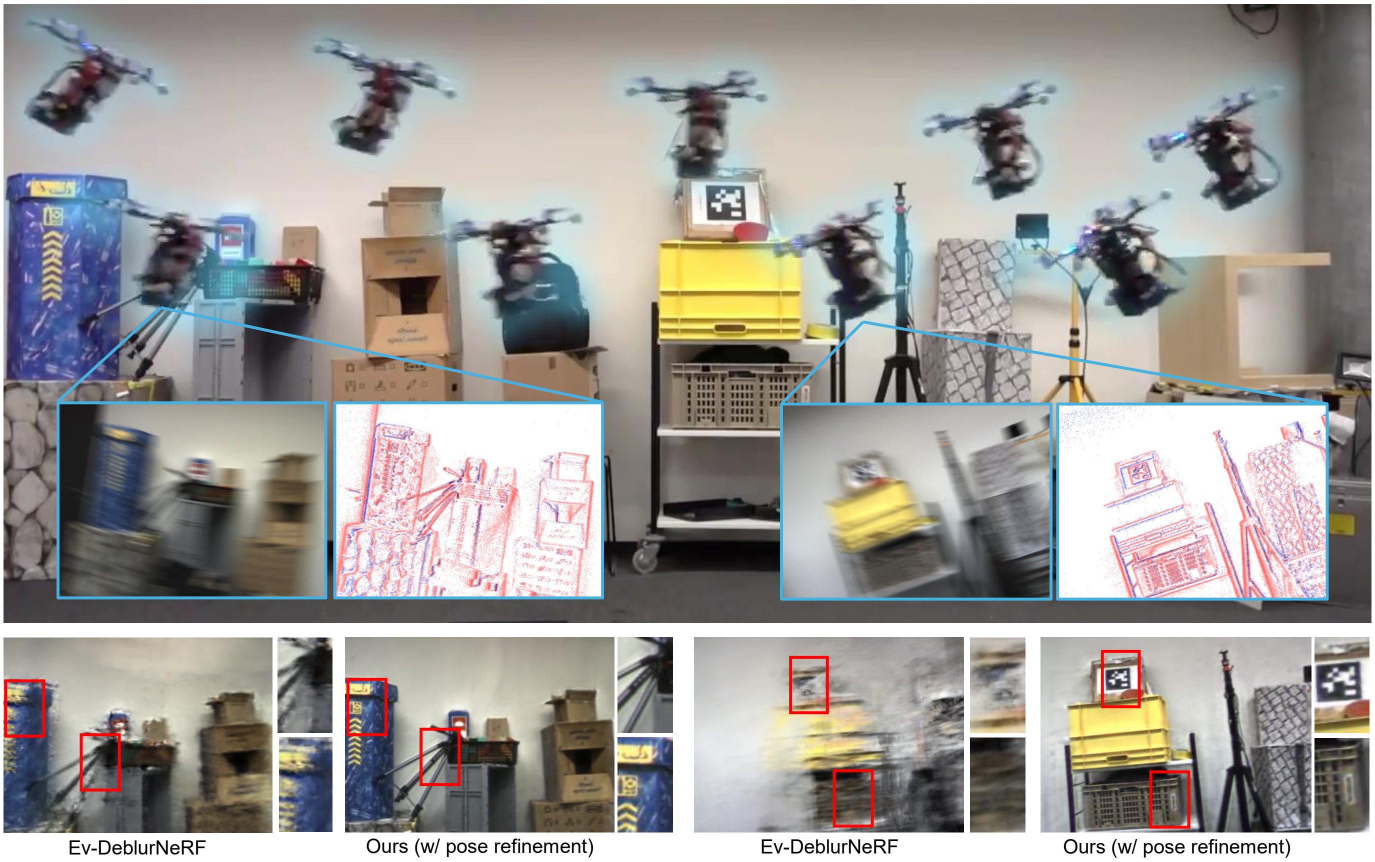


Fig. 1: Our system recovers sharp geometry and texture from footage captured by a drone flying at 2 m/s, learning a neural radiance field directly from motion-blurred images and event data collected during flight (**top**). Without relying on ground-truth poses or external motion capture, it refines the drone's trajectory during training, starting from a rough visual-inertial odometry prior. Once optimized, the model can render photorealistic views from novel perspectives (**bottom**), paving the way for high-speed, vision-based inspection tasks in agile robotics.

***Abstract*—Fast-flying aerial robots promise rapid inspection under limited battery constraints, with direct applications in infrastructure inspection, terrain exploration, and search and rescue. However, high speeds lead to severe motion blur in images and induce significant drift and noise in pose estimates, making dense 3D reconstruction with Neural Radiance Fields (NeRFs) particularly challenging due to their high sensitivity to such degradations. In this work, we present a unified framework that leverages asynchronous event streams alongside motion-blurred frames to reconstruct high-fidelity radiance fields from agile drone flights. By embedding event-image fusion into NeRF optimization and jointly refining event-based visual-inertial odometry priors using both event and frame modalities, our method recovers sharp radiance fields and accurate camera trajectories without ground-truth supervision. We validate our approach on both synthetic data and real-world sequences captured by a fast-flying drone. Despite highly dynamic drone flights, where RGB frames are severely degraded by motion blur and pose priors become unreliable, our method reconstructs high-fidelity radiance fields and preserves fine scene details, delivering a performance gain of over 50% on real-world data compared to state-of-the-art methods.**

## Multimedia Materials

Video of experiments: https://youtu.be/dVaH0VVXhQc
Code: https://github.com/uzh-rpg/event-sharp-nerf-drones

* R. Zou and M. Cannici equally contributed to this work.

This work was supported by the National Centre of Competence in Research (NCCR) Robotics (grant agreement No. 51NF40-185543) through the Swiss National Science Foundation (SNSF), the European Union's Horizon Europe Research and Innovation Programme under grant agreement No. 101120732 (AUTOASSESS), the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT), and the Swiss AI Initiative by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID a03 on Alps.

## I. Introduction

ROBOTIC systems operating in the real world often face a fundamental trade-off between sensing quality and operational efficiency. In particular, fast-flying aerial robots are essential for tasks such as large-scale infrastructure inspection (e.g., powerlines, pipelines, or railways), time-constrained terrain exploration, large-area agricultural and forestry surveys, or search and rescue. In these applications, flying faster enables a larger area to be covered within narrow time windows and under a limited battery budget, which is crucial for maximizing mission efficiency [1]. At low speeds, energy is wasted in maintaining lift without making significant progress, while high-speed flight introduces aerodynamic drag. Striking the right balance can lead to shorter mission durations, fewer battery replacements, and more effective deployment in time-sensitive settings.

However, increasing flight speed makes high-fidelity perception significantly more challenging. Motion blur corrupts visual data, while fast dynamics degrade the accuracy of visual-inertial odometry, limiting the performance of downstream tasks such as 3D reconstruction and scene understanding. These limitations pose a challenge for generating dense, photorealistic 3D models, crucial for map-based planning, semantic reasoning, and safe navigation.

Recent advances in learning-based scene representations, such as Neural Radiance Fields (NeRFs) [2] and 3D Gaussian Splatting [3], have shown great promise for view-consistent reconstruction from sparse camera views. These methods have rapidly found applications in robotics for localization, mapping, and visual simulation. However, they rely on two core assumptions: that input images are free of motion blur, and that accurate camera poses are available. In the context of fast-flying drones, neither of these assumptions holds. As a result, radiance field learning under fast motion remains largely an unsolved problem.

Prior works [4]–[7] have attempted to bridge this gap by leveraging event cameras, which offer high temporal resolution and robustness to motion blur, making them well-suited for high-speed scenarios. These methods typically aim to recover sharp images and accurate poses from events, via model-based [8] or learning-based [9] deblurring approaches. Once deblurred, images are passed to off-the-shelf structure-from-motion tools like COLMAP [10] to recover camera poses for NeRF training. However, under fast motion, these pipelines degrade significantly. Model-based methods often rely on simplified sensor dynamics and can fail under fast motion dynamics or sensor noise. Meanwhile, learning-based approaches may hallucinate inconsistent image content across views, leading to poor pose estimates. Furthermore, by relying solely on reconstructed images for pose estimation, these approaches miss the opportunity to directly incorporate the continuous, high-frequency information available in the event stream.

In this work, we address these limitations by proposing a unified framework for sharp radiance field reconstruction tailored to fast-flying drones operating under noisy pose conditions. Our method integrates motion-blurred images and asynchronous event data using a shared, learnable camera trajectory module, drawing on a continuous-time formulation [11]. We show that this representation enables joint modeling of event-based trajectories and approximation of the motion blur formation process, allowing both events and frames to supervise each other during training. Unlike previous approaches that treat event and image poses separately [12], our formulation refines a single, shared trajectory initialized using event-based visual-inertial odometry. This allows our system to recover accurate scene structure and motion without requiring ground-truth trajectories or slow, offline preprocessing.

We validate our method on synthetic and real-world drone sequences featuring fast flight and challenging motion profiles. To support reproducibility and benchmarking, we introduce the first drone dataset for radiance field reconstruction under fast motion, featuring synchronized motion-blurred RGB images and event data captured with a beamsplitter-based setup. Our results show that the proposed method achieves high-fidelity reconstructions even when initialized with noisy pose priors, significantly outperforming existing NeRF-based deblurring baselines on real drone data.

**Our contributions are:**

- A NeRF-based framework for radiance field reconstruction in high-speed robotic settings, capable of recovering sharp, photorealistic scene representations from motion-blurred images and asynchronous event data. By exploiting the complementary strengths of both modalities, the proposed framework enables high-fidelity mapping even under agile maneuvers. Our method delivers a performance gain of more than 50% on real-world data captured by a fast-flying drone, significantly outperforming state-of-the-art approaches.
- A continuous-time, shared trajectory formulation that unifies pose refinement and motion blur modeling within a single optimization process. Our approach allows mutual supervision between events and frames during training, and improves initial event-based visual-inertial odometry priors, leading to more robust and consistent reconstructions under fast motion.
- A real-world demonstration of our approach on data captured by a custom aerial platform equipped with a beamsplitter-based sensor rig featuring hardware-synchronized RGB and event cameras. The collected data are post-processed offline to demonstrate the method's performance under high-speed flight conditions up to 2 m/s. Accurate ground-truth poses provided by a motion capture system are used for evaluation. We release the dataset and code to support reproducible research in high-speed, event-driven scene reconstruction for robotics.

## II. Related Work

**Learning 3D Reconstruction under Motion Blur.** Neural 3D scene representations, such as Neural Radiance Fields (NeRFs) [2] and 3D Gaussian Splatting (3DGS) [3], have become powerful tools for novel view synthesis and dense scene reconstruction. Their success in recovering geometry and appearance from sparse multi-view imagery has led to growing interest in robotics applications [13]–[15]. While early methods assumed clean, well-calibrated inputs and operated offline, later work introduced faster optimization [16]–[19], support for dynamic scenes [20], [21], and robustness to pose noise [22]–[24].

In line with these efforts, a particularly active area of research focuses on extending these methods to recover accurate 3D geometry from motion-blurred images. Deblur-NeRF [25] addresses this by jointly estimating a latent sharp radiance field and a learned view-dependent blur model. PDRF [26] introduces coarse-to-fine estimation using proxy geometry, while DP-NeRF [27] constrains motion to lie in rigid subspaces. BAD-NeRF [28] takes a more geometric approach, recovering camera motion during exposure via photometric bundle adjustment. This direction has also been explored in the context of 3D Gaussian Splatting. Deblur-GS [29] and BAD-Gaussians [30] adapt NeRF-style joint optimization to 3DGS by modeling image formation under motion blur and refining both camera trajectories and Gaussian parameters. BAGS [31] extends this by introducing a Blur Proposal Network that estimates dense per-pixel blur kernels and a mask identifying sharp regions, enabling robust reconstruction under spatially-varying blur. Gaussian Splatting on the Move [32] further extends this line of work to rolling shutter settings, leveraging visual-inertial odometry within a differentiable 3DGS pipeline. Despite these improvements, these methods remain sensitive to the quality of input poses and struggle when all training views are similarly affected by fast motion. In mobile robotics scenarios—particularly onboard drones—such motion patterns are common and unavoidable. Moreover, these methods rely exclusively on RGB images, which are inherently limited by blur and low temporal resolution.

**Event-Based Neural 3D Reconstruction.** Event cameras offer an alternative sensing modality for robotics applications such as visual odometry, SLAM, and scene reconstruction [33], [34], particularly in fast and challenging environments. Unlike conventional cameras that capture full intensity images at fixed rates, event sensors asynchronously report per-pixel brightness changes at microsecond resolution. This enables low-latency perception, naturally avoids motion blur, and performs reliably under challenging lighting conditions. These advantages have led to growing interest in incorporating events into neural 3D reconstruction frameworks, including both NeRF and Gaussian Splatting approaches, with the potential to enable accurate scene reconstruction under fast motion. Ev-NeRF [35] and EventNeRF [36] demonstrate that events alone can supervise the reconstruction of a static scene, using the event generation model [37] as a supervisory signal. Robust e-NeRF [38] adds supervision through a normalized gradient loss to improve robustness to pose variation. Event-based 3DGS methods have recently emerged as an alternative. EV-GS [39] adopts a fully event-based pipeline, initializing Gaussians from randomly sampled scene points. Event3DGS [40] accumulates events based on scene entropy and uses them to iteratively initialize and refine a 3D Gaussian point cloud. Finally, a different approach is proposed in EvGGS [41], which introduces a generalizable, feedforward pipeline for event-based Gaussian Splatting by combining learned depth estimation, intensity reconstruction, and Gaussian regression for direct geometry prediction from events. Building on these approaches, our method introduces motion-blurred images as an additional source of supervision, allowing for more precise geometry reconstruction and the recovery of colored textures.

**Sharp 3D from Events and Blurred Frames.** Given the success of event cameras in 3D reconstruction under fast motion, recent work has explored combining asynchronous events with motion-blurred RGB frames to enable sharper and more robust neural scene reconstruction. By fusing the high temporal precision of events with the rich appearance information in blurred images, these methods aim to recover accurate geometry and texture even in challenging dynamic conditions. E-NeRF [42] demonstrates that combining events with blurry images can already reduce motion blur in NeRF reconstructions, while E2NeRF [43] takes a step further by explicitly modeling the blur formation process during training, leading to sharper textures and more accurate color reconstruction. Ev-DeblurNeRF [12] further refines this setup by incorporating network enhancements from PDRF [26] and DP-NeRF [27], event-wise supervision, a learnable response function, and model-based priors. Inspired by these NeRF-based approaches, recent methods extend similar ideas to 3D Gaussian Splatting. E2GS [5] adopts a Deblur-GS-style [29] pipeline, using event-based deblurred images and COLMAP [10] poses for Gaussian initialization. In contrast, Event3DGS [40] uses an iterative strategy that first reconstructs geometry using events alone and then refines appearance from blurry images in a second stage, optimizing only color.

Despite these advances, most existing methods rely on pre-deblurred images or external structure-from-motion pipelines such as COLMAP [10], to estimate camera poses. These approaches are not only slow but also brittle under fast motion—event-based deblurring can fail with unmodeled dynamics, and pose estimates may drift due to inconsistent image content. To overcome this, we initialize the trajectory using a real-time event-based visual-inertial odometry system [33], avoiding expensive camera estimation. We then refine this trajectory during training through joint optimization, where both blurry images and events supervise a shared, learnable pose. This enables consistent alignment across modalities and accurate radiance field reconstruction from noisy priors, while fully exploiting the high temporal resolution of events, which is crucial for robust mapping in high-speed robotics scenarios.

## III. Method

We aim to reconstruct a latent, sharp radiance field of a static scene from visual data captured during fast flight. In particular, given input observations $\{\mathcal{I}, \mathcal{E}, \mathcal{M}\}$ corresponding to a sequence of motion-blurred RGB frames $\mathcal{I} = \{\mathbf{C}_i^{\text{blur}}\}_{i=1}^{N_I}$, asynchronous events $\mathcal{E} = \{\mathbf{e}_j\}_{j=1}^{N_E}$, and inertial measurements $\mathcal{M} = \{(\mathbf{a}_k, \boldsymbol{\omega}_k, t_k)\}_{k=1}^{N_M}$ captured from a moving camera, we seek to recover the latent continuous volumetric function

$$\mathcal{F} : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma), \tag{1}$$

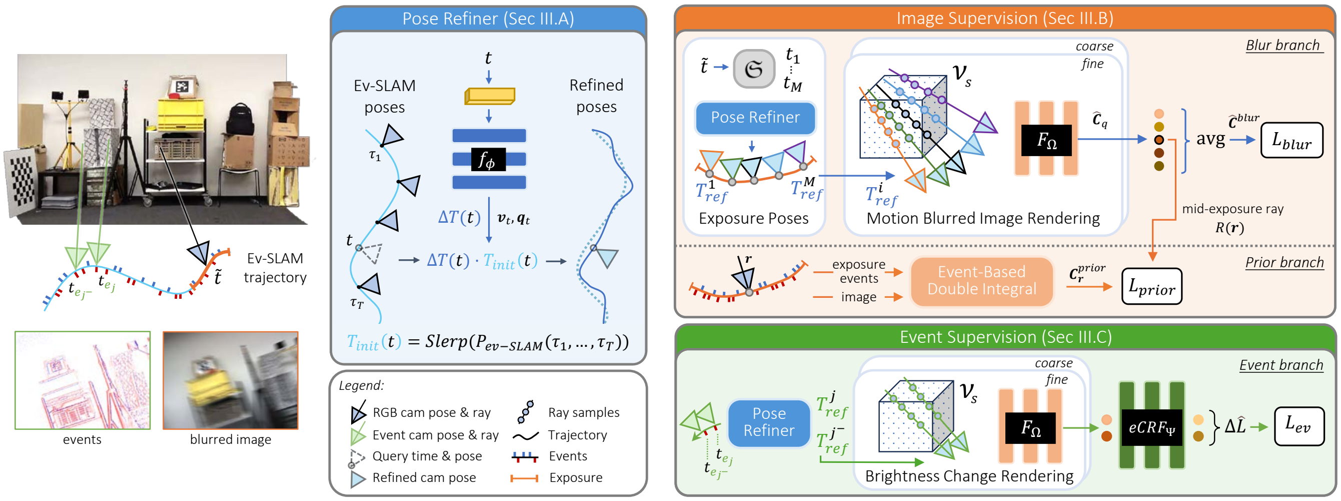


Fig. 2: Overview of our proposed architecture. We reconstruct a sharp radiance field from a set of motion-blurred RGB frames and events. The estimated trajectory from an event-based SLAM system, $T_{init}$, is refined through a learned *Pose Refiner* that takes a query timestamp $t$ as input and predicts residual corrections at arbitrary time resolutions. This refined trajectory is used to supervise the radiance field through three complementary branches: a *blur branch* models image formation across the exposure time; an *event branch* supervises high-temporal-resolution brightness changes via a learned event camera response function; and a *prior branch* introduces model-based deblur constraints. All branches share the same learned camera trajectory, allowing events and images to jointly refine the scene representation and motion estimates.

which maps a 3D location $\mathbf{x} \in \mathbb{R}^3$ and viewing direction $\mathbf{d} \in \mathbb{R}^3$ to its emitted color $\mathbf{c} \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}$. Each input image $\mathbf{C}_i^{\text{blur}} \in \mathbb{R}^{H \times W \times 3}$ captures a motion-blurred RGB frame at timestamp $t_i$, while the event stream provides asynchronous measurements $\mathbf{e}_j = (\mathbf{u}_j, t_j, p_j)$, where $\mathbf{u}_j = (u_j, v_j)$ denotes the pixel location, $t_j$ the timestamp, and $p_j \in \{-1, 1\}$ the polarity of the detected brightness change.

Our method builds upon recent event-aided deblur NeRF models [12], [42], [43], which optimize a NeRF using both image- and event-based supervision. Inspired by Ev-DeblurNeRF [12], we adopt a tri-branch architecture that jointly leverages information from different sensing modalities to guide the learning of a sharp radiance field. The first branch operates on blurred RGB frames and supervises the radiance field via a photometric blur model. A second, event-based branch supervises the radiance field using microsecond-level brightness changes captured by the event sensor, which are interpreted through a learnable camera response function (CRF) that adapts to real sensor characteristics. Finally, the third branch introduces prior knowledge to constrain training further and resolve ambiguities arising from severe blur or sparse event information. An overview of the proposed method is presented in Figure 2.

Unlike previous approaches, our method does not rely on accurate, high-frequency camera poses during image exposure or at event timestamps. Instead, we propose a unified framework that estimates a temporally continuous trajectory by combining event-based odometry with a learned residual correction module based on [11]. This module refines the coarse odometry at arbitrary time resolutions, enabling recovery of sub-millisecond-accurate camera poses for each triggered event as well as the continuous trajectory traced during the exposure of each image. Crucially, in our proposed architecture the image- and event-based branches work together to cooperatively refine both the radiance field and the camera trajectory, enabling sharper reconstructions through improved pose estimation.

In the following sections, we describe each component in detail, starting with the trajectory representation and pose refinement in Sec. III-A, followed by the radiance field supervision via events and blur modeling in Secs. III-B and III-C.

### A. Trajectory Representation and Pose Refinement

Accurate training of event- and image-based NeRF models requires temporally precise camera poses at the exact locations where each observation was captured. This challenge becomes even more pronounced in event-aided deblur settings, where one must estimate not only the camera trajectory during each image exposure, but also the precise pose at which every event was triggered. Prior approaches [12], [42], [43] either assume access to densely sampled ground-truth poses, rarely feasible in real-world scenarios, or rely on two-stage pipelines that deblur frames using event integration [8] and then estimate poses with structure-from-motion tools like COLMAP [10]. These methods not only depend heavily on the quality of the deblurred images, which degrade under fast motion, but also estimate poses only at discrete timestamps, rather than modeling the continuous camera motion. In contrast, we propose a unified, learning-based trajectory model that generalizes to fast, high-dynamic scenarios without requiring accurate intermediate reconstructions or pose annotations.

We begin by computing a coarse estimate of the camera trajectory using an off-the-shelf visual-inertial SLAM or VIO system. In our implementation, we adopt UltimateSLAM [33] for its ability to fuse both image and event features for robust pose tracking. Despite the presence of motion blur in the RGB frames, image gradients still contribute to feature matching, while the high temporal resolution of the event stream ensures resilience to fast motion. This provides discrete pose predictions $\mathcal{P}_{\text{SLAM}} = \{\mathbf{T}_{\text{SLAM}}(\tau)\}_{\tau=1}^{T}$ where each $\mathbf{T}_{\text{SLAM}}(\tau) \in \text{SE}(3)$ represents the estimated camera pose at time $\tau$. In our implementation, UltimateSLAM [33] outputs pose estimates at the RGB camera rate, yielding a relatively sparse set of trajectory samples $\tau \in \{t_1, t_2, \ldots, t_{N_I}\}$, where $N_I$ is the number of captured RGB frames. These sparse pose estimates serve as the initial trajectory prior, which we refine via a continuous-time pose model.

To enable refinement at arbitrary times, we first construct a continuous initial trajectory $\mathbf{T}_{\text{init}}(t) \in \text{SE}(3)$ by interpolating the discrete SLAM pose estimates. Specifically, we apply a spherical linear interpolation (SLERP) that interpolates rotations on the $\text{SO}(3)$ manifold and translations linearly in $\mathbb{R}^3$:

$$\mathbf{T}_{\text{init}}(t) = \text{SLERP}_{\text{SE}(3)}\left(\mathcal{P}_{\text{SLAM}}, t\right), \tag{2}$$

where $t$ denotes any arbitrary timestamp during the camera motion. While this continuous prior provides a smooth estimate of the trajectory, it inevitably suffers from drift and local inaccuracies. To address this, we learn to estimate a residual refinement transformation $\Delta\mathbf{T}(t) \in \text{SE}(3)$ that corrects the initial pose at each timestamp:

$$\mathbf{T}_{\text{ref}}(t) = \Delta\mathbf{T}(t) \cdot \mathbf{T}_{\text{init}}(t). \tag{3}$$

Following [11], we parametrize this refinement as a multi-layer perceptron (MLP) network $\Delta\mathbf{T}_{\text{refine}}(t) = f_\phi(\gamma_{L_t}(t))$:

$$f_\phi : \mathbb{R}^{2L_t} \to \text{SE}(3), \tag{4}$$

where $\gamma_{L_t}(\cdot)$ denotes a frequency-encoded representation of time, following the sinusoidal positional encoding introduced in NeRF [2]. The network $f_\phi$ maps this embedding to the residual pose $\Delta\mathbf{T}_{\text{refine}}(t)$, computed from the predicted translation vector $\mathsf{v}(t) \in \mathbb{R}^3$ and quaternion $\mathsf{q}(t) \in \mathbb{R}^4$.

At any timestamp $t$, the refined camera pose $\mathbf{T}_{\text{ref}}(t)$ defines the origin and orientation of the virtual camera for that instant. The scene is rendered using this pose, and the resulting rendered view is supervised by both event and RGB camera observations (except for the frame-only case in Sec. IV-D where supervision is obtained solely from RGB images). In both cases, the blurry images together with the motion-blur modeling techniques introduced in Sec. III-B are incorporated during training. During each training iteration, the photometric and event-based losses computed on these rendered virtual views are backpropagated through the differentiable rendering process to both the scene representation and the pose refiner. This means that the scene parameters and the camera poses are jointly optimized within the same training step, rather than in separate stages. The initial coarse trajectory is crucial, as it provides a plausible starting point that guides the system toward a consistent structure and appearance. In the next sections, we describe how these refined poses are used in practice to render motion-blurred images and event representations for supervision.

### B. Image Supervision via Motion Blur Rendering

Following previous literature [12], [25]–[27], to enable radiance field learning from motion-blurred RGB frames, we adopt a physically grounded blur formation model that integrates sharp color predictions over the exposure interval. This allows us to supervise the radiance field using blurred observations, without access to ground-truth sharp frames, provided the camera motion during exposure is accurately modeled.

Given a pixel $\mathbf{u}$ in a frame captured at time $t_i$, where $t_i$ denotes the center of the exposure interval for image $i$, we define $\mathbf{r}(\mathbf{u}, t_i)$ as the ray that originates from the camera center of the refined camera pose $\mathbf{T}_{\text{ref}}(t_i)$ and passes through the pixel $\mathbf{u}$. We estimate the color of the blurred pixel $\mathbf{u}$ observed in $\mathbf{C}_i^{\text{blur}}$ by averaging the colors observed along the set of rays $\mathbf{r}(\mathbf{u}, t_i)$ as the camera moves during the exposure interval $[t_i - \tau/2,\ t_i + \tau/2]$. This simulates the physical image formation process, where the observed pixel color corresponds to the integral of the latent sharp radiance accumulated over the exposure duration [25].

To perform this integration, we sample a set of $M$ timestamps $\{t^{(1)}, \ldots, t^{(M)}\} = \mathfrak{S}_\tau(t_i)$ within the exposure interval, centered at the image timestamp $t_i$. For each sampled time $t^{(m)}$, we retrieve the corresponding camera pose from the refined trajectory $\mathbf{T}_{\text{ref}}(t^{(m)})$ and cast a ray $\mathbf{r}(\mathbf{u}, t^{(m)})$ from the resulting viewpoint. In our experiments, we found that implementing $\mathfrak{S}$ as uniform sampling within the exposure interval provides accurate and stable results. However, our formulation is general and could accommodate learned sampling strategies–for example, modeling $\mathfrak{S}(\cdot)$ as an MLP conditioned on $t_i$, similar to $f_\phi$, or adopting adaptive schemes based on event rates, as explored in prior work [4].

To render individual views, we follow standard NeRF volume rendering [2]. Along each ray, we sample a set of 3D points and query the continuous volumetric function $\mathcal{F}$ defined in (1) to obtain colors and densities at each location. In practice, $\mathcal{F}$ is implemented as two MLPs, coarse and fine, denoted by $F_\Omega^c$ and $F_\Omega^f$. Inspired by [12], [26], we augment both networks with explicit feature volumes [17] $\mathcal{V}_s$ and $\mathcal{V}_l$. These volumes are evaluated at the sampled ray positions along each ray to produce features $f^c$, $f^f$, which we concatenate to the inputs of the coarse and fine MLPs, respectively. These features accelerate convergence and enhance rendering quality, as they introduce spatially anchored features that are easier to optimize.

The colors and densities collected along each ray are then aggregated through a volumetric rendering operator [2], which we denote as $\mathcal{R}$. The final blurred pixel color is obtained by averaging these latent sharp values over the sampled timestamps:

$$\hat{\mathbf{C}}^{\text{blur}}(\mathbf{u}, t_i) = g\left(\sum_{m=1}^{M} w_m\, \mathcal{R}\big(\mathbf{r}(\mathbf{u}, t^{(m)})\big)\right), \tag{5}$$

where $w_m$ denotes the blending weight for the $m$-th timestamp and satisfies $\sum_{m=1}^{M} w_m = 1$, and $g(\cdot)$ is a gamma correction function. Similar to [43], we use uniform weights to blend the rendered colors over time.

For brevity, we denote the rendered pixel in Eq. 5 as $\hat{\mathbf{C}}^{\text{blur}}_{\mathbf{r}}$, we finally compare the synthetic and observed blurry pixels over a batch $\mathcal{B}_b$ of pixel-timestamp pairs $(\mathbf{u}_k, t_k)$:

$$\mathcal{L}_{\text{blur}} = \frac{1}{|\mathcal{B}_b|} \sum_{\mathbf{r}_k \in \mathcal{B}_b} \left[ \left\| \hat{\mathbf{C}}^{\text{blur}}_{\mathbf{r}_{k,c}} - \mathbf{C}^{\text{blur}}_{\mathbf{r}_k} \right\|_2^2 + \left\| \hat{\mathbf{C}}^{\text{blur}}_{\mathbf{r}_{k,f}} - \mathbf{C}^{\text{blur}}_{\mathbf{r}_k} \right\|_2^2 \right] \tag{6}$$

where subscripts $c$ and $f$ indicate outputs from the coarse and fine networks, respectively, and $\mathbf{C}^{\text{blur}}_{\mathbf{r}_k}$ the ground-truth blurred color at the pixel corresponding to $\mathbf{r}_k$. Note that, for compactness, we refer to rays $\mathbf{r}_k$ in the summation although the batch is formally defined over pixel-timestamp pairs $(\mathbf{u}_k, t_k)$ from which the corresponding rays are derived (i.e., the rays cast from the refined poses $\mathbf{T}_{\text{ref}}(t_k)$ and passing through $\mathbf{u}_k$).

While the blur rendering in Eq. 6 provides a strong supervisory signal, it still leaves the learning problem underconstrained, especially under high-dynamic motion where motion blur severely degrades textured areas. To further guide the radiance field toward plausible sharp appearances, we follow [12] and introduce an additional supervision signal based on prior deblurring estimates of the training images.

Specifically, for each ray $\mathbf{r}_k \in \mathcal{B}_b$, sampled at mid-exposure $t_k$, we associate a pseudo ground-truth sharp target $\mathbf{C}^{\text{prior}}_{\mathbf{r}_k}$ obtained by deblurring the blurry input using the event-based double integral (EDI) [8]. We then supervise the NeRF-rendered color at the mid-exposure pose to match this deblurred estimate:

$$\mathcal{L}_{\text{prior}} = \frac{1}{|\mathcal{B}_b|} \sum_{\mathbf{r}_k \in \mathcal{B}_b} \left[ \left\| \mathcal{R}_c(\mathbf{r}_k) - \mathbf{C}^{\text{prior}}_{\mathbf{r}_k} \right\|_2^2 + \left\| \mathcal{R}_f(\mathbf{r}_k) - \mathbf{C}^{\text{prior}}_{\mathbf{r}_k} \right\|_2^2 \right], \tag{7}$$

where $\mathcal{R}_c(\mathbf{r}_k)$ and $\mathcal{R}_f(\mathbf{r}_k)$ are the coarse and fine volumetric renderings of the ray $\mathbf{r}_k$ at mid-exposure.

Since EDI is a model-based deblurring process that assumes accurate and noise-free event measurements, its outputs can contain artifacts when the event data is noisy or the scene violates model assumptions. To address this, we employ the prior-based supervision primarily during the early stages of training to guide the radiance field toward a plausible, sharp initialization. As training progresses, we gradually reduce the weight of this loss, allowing the NeRF to refine its reconstruction beyond the model-based estimate, driven by the direct event supervision discussed in the next section.

### C. Event-Based Supervision

While motion-blurred RGB frames provide strong supervision for radiance field optimization, they are inherently limited by their low temporal resolution and the lack of texture under fast motion. To complement the image and prior-based supervision, we leverage the high temporal resolution of events to directly guide both the radiance field and the camera trajectory refinement.

Each event $\mathbf{e}_j = (\mathbf{u}_j, t_j, p_j)$ encodes a brightness change of polarity $p_j$ at pixel $\mathbf{u}_j$ and time $t_j$. Unlike RGB frames, these measurements are significantly less affected by motion blur and can thus provide precise, high-frequency supervision. Following prior works [35], [36], we train the model to predict log-brightness changes that simulate those observed by the event camera during motion. To this end, we synthesize the log-brightness at each event timestamp via volumetric rendering of the radiance field.

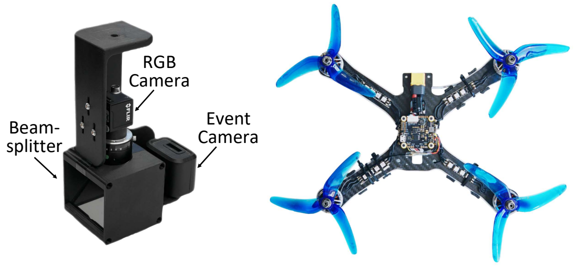


Fig. 3: Data collection setup. A beamsplitter with an RGB and an event camera is attached to a quadrotor platform.

Specifically, for each event $\mathbf{e}_j = (\mathbf{u}_j, t_j, p_j)$, we identify its most recent preceding event at the same pixel, denoted $\mathbf{e}_{j^-} = (\mathbf{u}_j, t_{j^-}, p_{j^-})$ with $t_{j^-} < t_j$. We estimate the camera poses at which these events were triggered using the refined trajectory, $\mathbf{T}_{\text{ref}}(t_j)$ and $\mathbf{T}_{\text{ref}}(t_{j^-})$ (Sec. III-A), and render the corresponding RGB colors $\mathcal{R}(\mathbf{u}_j, t_j)$ and $\mathcal{R}(\mathbf{u}_j, t_{j^-})$ via volumetric rendering at these timestamps.

We estimate the corresponding brightness change by converting the rendered colors to log-brightness values using a learnable event camera response function (eCRF) and computing their difference. In particular, we define the log-brightness perceived by the event camera's pixel $\mathbf{u}_j$ at time $t_j$ as:

$$\Delta \hat{L}_j = \hat{L}_j - \hat{L}_{j^-} \text{ with } \hat{L}_j = \log\left(h(\text{eCRF}_{\Psi}(\mathcal{R}(\mathbf{u}_j, t_j), p_j))\right) \tag{8}$$

where $h(\cdot)$ is a luma conversion function, and $\text{eCRF}_{\Psi}$ is an MLP that models the pixel-wise, polarity-dependent response of the event sensor. This learnable mapping accounts for real-world deviations from the ideal event generation model, such as pixel-level threshold variability and sensor non-linearities.

Finally, we supervise the predicted brightness change to match the expected change recorded by the camera over a batch $\mathcal{B}_e$:

$$\mathcal{L}_{\text{event}} = \frac{1}{|\mathcal{B}_e|} \sum_{\mathbf{e}_j \in \mathcal{B}_e} \left[ \left\| \Delta \hat{L}_{j,c} - \Delta L^{\text{cam}}_j \right\|_2^2 + \left\| \Delta \hat{L}_{j,f} - \Delta L^{\text{cam}}_j \right\|_2^2 \right], \tag{9}$$

where $\Delta L^{\text{cam}}_j = p_j \Theta_{p_j}$ is the expected change recorded by the camera, $\Theta_{p_j}$ is the contrast threshold corresponding to the event polarity (positive or negative), and the subscripts $c$ and $f$ denote quantities computed using the coarse and fine radiance field renderings, respectively.

This event loss not only promotes sharper reconstructions, particularly in regions with limited RGB support, but also directly supervises the refined trajectory at sub-frame resolution. By querying $\mathbf{T}_{\text{ref}}(\cdot)$ at individual event timestamps, the model can propagate gradient signals back to the pose refiner module $f_{\phi}$, contributing to adjusting the trajectory even within frame exposures. This tight coupling of radiance and motion learning enhances overall reconstruction accuracy and consistency, enabling the recovery of sharp geometry even under severe motion blur.

TABLE I: Quantitative comparison on the synthetic Ev-DeblurBlender dataset. Best results are reported in bold.

| Method | Factory | | | Pool | | | Tanabata | | | Trolley | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ |
| DeblurNeRF [25] | 24.52 | 0.25 | 0.79 | 26.02 | 0.34 | 0.69 | 21.38 | 0.28 | 0.71 | 23.58 | 0.22 | 0.79 | 23.87 | 0.27 | 0.74 |
| BAD-NeRF [28] | 21.20 | 0.22 | 0.64 | 27.13 | 0.23 | 0.70 | 20.89 | 0.25 | 0.65 | 22.76 | 0.18 | 0.73 | 22.99 | 0.22 | 0.68 |
| PDRF [26] | 27.34 | 0.17 | 0.87 | 27.46 | 0.32 | 0.72 | 24.27 | 0.20 | 0.81 | 26.09 | 0.15 | 0.86 | 26.29 | 0.21 | 0.81 |
| DP-NeRF [27] | 26.77 | 0.20 | 0.85 | 29.58 | 0.24 | 0.79 | 27.32 | 0.11 | 0.85 | 27.04 | 0.14 | 0.87 | 27.68 | 0.17 | 0.84 |
| MPRNet [44] + NeRF | 19.09 | 0.37 | 0.56 | 25.49 | 0.39 | 0.64 | 17.79 | 0.42 | 0.51 | 19.82 | 0.31 | 0.62 | 20.55 | 0.37 | 0.58 |
| PVDNet [45] + NeRF | 22.50 | 0.29 | 0.71 | 23.89 | 0.43 | 0.52 | 20.26 | 0.33 | 0.64 | 22.49 | 0.25 | 0.74 | 22.28 | 0.32 | 0.65 |
| EFNet [9] + NeRF | 20.91 | 0.32 | 0.63 | 27.03 | 0.31 | 0.73 | 20.68 | 0.31 | 0.64 | 21.69 | 0.25 | 0.69 | 22.58 | 0.30 | 0.67 |
| EFNet* [9] + NeRF | 29.01 | 0.14 | 0.87 | 29.77 | 0.18 | 0.80 | 27.76 | 0.11 | 0.87 | 29.40 | 0.09 | 0.89 | 28.99 | 0.13 | 0.86 |
| ENeRF [42] | 22.46 | 0.19 | 0.79 | 25.51 | 0.28 | 0.72 | 22.97 | 0.16 | 0.83 | 21.07 | 0.20 | 0.80 | 23.00 | 0.21 | 0.79 |
| E$^2$NeRF [43] | 24.90 | 0.17 | 0.78 | 29.57 | 0.18 | 0.78 | 23.06 | 0.19 | 0.74 | 26.49 | 0.10 | 0.85 | 26.00 | 0.16 | 0.78 |
| Ev-DeblurNeRF [12] | 31.79 | 0.06 | 0.93 | **31.51** | 0.14 | 0.84 | 28.67 | 0.08 | 0.90 | 29.72 | 0.07 | 0.92 | 30.42 | 0.08 | 0.90 |
| Ours | **32.37** | **0.06** | **0.94** | 31.13 | **0.14** | **0.84** | **29.06** | **0.07** | **0.90** | **30.04** | **0.06** | **0.92** | **30.65** | **0.08** | **0.90** |

TABLE II: Quantitative comparison on the real-world Ev-DeblurCDAVIS dataset. Best results are reported in bold.

| Method | Batteries | | | Power supplies | | | Lab Equipment | | | Drones | | | Figures | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ |
| DP-NeRF+TensoRF [17] | 26.64 | 0.27 | 0.81 | 25.74 | 0.32 | 0.77 | 27.49 | 0.31 | 0.80 | 26.52 | 0.30 | 0.81 | 27.76 | 0.34 | 0.77 | 26.83 | 0.31 | 0.79 |
| EDI [8] + NeRF | 28.66 | 0.12 | 0.87 | 28.16 | 0.09 | 0.88 | 31.45 | 0.13 | 0.89 | 29.37 | 0.10 | 0.88 | 31.44 | 0.12 | 0.88 | 29.82 | 0.11 | 0.88 |
| E$^2$NeRF [43] | 30.57 | 0.12 | 0.88 | 29.98 | 0.11 | 0.87 | 30.41 | 0.16 | 0.86 | 30.41 | 0.14 | 0.87 | 31.03 | 0.14 | 0.85 | 30.48 | 0.13 | 0.87 |
| Ev-DeblurNeRF [12] | 33.17 | **0.05** | 0.92 | 32.35 | **0.06** | 0.91 | **33.01** | **0.08** | **0.91** | 32.89 | **0.05** | 0.92 | 33.39 | **0.07** | 0.90 | 32.96 | **0.06** | 0.91 |
| Ours | **33.20** | 0.08 | **0.92** | **32.49** | 0.07 | **0.91** | 32.83 | 0.11 | 0.91 | **33.00** | 0.07 | **0.92** | **33.63** | 0.11 | **0.90** | **33.03** | 0.09 | **0.91** |

## IV. Experiments

We validate our method on both synthetic and real-world datasets, comparing it against recent image-based and event-based baselines. We begin with synthetic scenarios using Ev-DeblurBlender [12], which provides ground-truth poses and allows us to ablate components of our network in a controlled setting. We then evaluate performance on real-world data with Ev-DeblurCDAVIS [12], before introducing two new challenging datasets collected under fast motion: Gen3-HandHeld, which tests robustness to varying motion blur, and Gen3-DroneFlight, which represents the most difficult setting with high-speed drone flight data.

### A. Implementation Details

**Training.** We implement our code using PyTorch. We train using a batch size of 1024 rays for the blur loss ($\mathcal{B}_b$) and 2048 rays for the event loss ($\mathcal{B}_e$), and sample 64 coarse and 64 fine points along each ray. The number of motion samples $M$ per exposure is scene-dependent and typically ranges from 7 to 11 based on motion complexity. The pose refiner $f_\phi$ is implemented as an 8-layer MLP with 256-dimensional hidden units and ReLU activations, following [11], while we follow [12] for implementing the fine and coarse MLPs defining the NeRF. We use the Adam optimizer [46] to minimize the total loss $\mathcal{L} = \lambda_b \mathcal{L}_{\text{blur}} + \lambda_e \mathcal{L}_{\text{event}} + \lambda_p \mathcal{L}_{\text{prior}}$ where we set $\lambda_b = 1.0$ and $\lambda_e = 0.1$, while we decay $\lambda_p$ from an initial value of $0.1$ to zero by iteration 20,000 with a cosine scheduling to let the NeRF move beyond the model-based prior. We train for 30,000 iterations with a learning rate exponentially decaying from $5{\cdot}10^{-3}$ to $5{\cdot}10^{-6}$. The total runtime for training a single scene is around 3 hours on one NVIDIA A100 GPU.

### B. Datasets

**Ev-DeblurBlender.** A synthetic dataset introduced in [12], rendered using high-frame-rate Blender simulations and comprising four scenes: *factory*, *pool*, *tanabata*, and *trolley*. Blurry images are generated by integrating high-FPS frames over 40 ms exposures, while events are simulated using ESIM [47] with thresholds $\Theta_{-1} = \Theta_{+1} = 0.2$.

**NoisyPose-EvDeblurBlender.** To evaluate robustness to trajectory errors, we introduce a noisy variant of the Ev-DeblurBlender dataset. Starting from ground-truth poses, we inject synthetic drift by first selecting six uniformly spaced points along each trajectory and sampling at each point a random 6 DoF perturbation. We then linearly interpolate the sampled perturbations to all intermediate poses. The perturbations increase progressively with traveled distance to simulate the behavior of a VIO system affected by drift. We use a global *noise level* $l$ to control the overall magnitude of the perturbations. Letting $\epsilon^{(l)} = (\epsilon_t^{(l)}, \epsilon_R^{(l)})$ denote the accumulated error per meter of traveled distance for noise level $l$, where $\epsilon_t^{(l)}$ is the translation error (in centimeters) and $\epsilon_R^{(l)}$ the rotation error (in degrees), we test 4 levels to quantify performance degradation across methods, defined as follows for noise levels $l = 1$ to $l = 4$: $(2\,\text{cm}, 0.2°)$, $(4\,\text{cm}, 0.4°)$, $(8\,\text{cm}, 0.8°)$, $(12\,\text{cm}, 1.2°)$.

**Ev-DeblurCDAVIS.** A real-world dataset introduced in [12], which includes ground-truth sharp reference images as well as precise poses for training and evaluation. It uses a Color DAVIS346 sensor to record events and RGB frames (346×260 resolution) with a 100 ms exposure time. Ground-truth poses are obtained via a linear slider's motor encoder, enabling evaluation under controlled motion at speeds on the order of 0.1 m/s. We follow the same train-test split as in [12], using 11 to 18 blurry training views and 5 sharp test views.

TABLE III: Quantitative comparison on the NoisyPose-EvDeblurBlender dataset under different pose noise levels. Best results are reported in bold.

| Noise Level | Method | FACTORY | | | POOL | | | TANABATA | | | TROLLEY | | | AVERAGE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ |
| 1 | Ev-DeblurNeRF [12] | 12.67 | 0.40 | 0.19 | 21.07 | 0.38 | 0.36 | 13.40 | 0.39 | 0.23 | 12.15 | 0.52 | 0.21 | 14.82 | 0.42 | 0.25 |
| | Ours (Frame Only) | 28.57 | 0.15 | 0.89 | 29.73 | 0.23 | 0.79 | 25.28 | 0.17 | 0.83 | 27.65 | 0.11 | 0.89 | 27.81 | 0.17 | 0.85 |
| | Ours (Frame + Event) | **31.11** | **0.07** | **0.92** | **30.89** | **0.15** | **0.83** | **28.01** | **0.09** | **0.89** | **29.51** | **0.07** | **0.91** | **29.88** | **0.09** | **0.89** |
| 2 | Ev-DeblurNeRF [12] | 11.30 | 0.55 | 0.12 | 17.66 | 0.57 | 0.18 | 12.73 | 0.64 | 0.17 | 10.61 | 0.62 | 0.12 | 13.08 | 0.60 | 0.15 |
| | Ours (Frame Only) | 28.15 | 0.16 | 0.88 | 29.53 | 0.25 | 0.78 | 25.01 | 0.17 | 0.82 | 27.18 | 0.12 | 0.88 | 27.47 | 0.17 | 0.84 |
| | Ours (Frame + Event) | **31.23** | **0.06** | **0.92** | **30.68** | **0.16** | **0.83** | **27.95** | **0.10** | **0.87** | **29.25** | **0.07** | **0.91** | **29.78** | **0.10** | **0.88** |
| 3 | Ev-DeblurNeRF [12] | 10.33 | 0.63 | 0.09 | 15.80 | 0.71 | 0.13 | 11.29 | 0.71 | 0.11 | 9.13 | 0.66 | 0.07 | 11.64 | 0.68 | 0.10 |
| | Ours (Frame Only) | 27.84 | 0.16 | 0.88 | 29.11 | 0.26 | 0.77 | 21.49 | 0.66 | 0.27 | 25.94 | 0.15 | 0.84 | 26.10 | 0.31 | 0.69 |
| | Ours (Frame + Event) | **31.31** | **0.06** | **0.93** | **30.25** | **0.18** | **0.81** | **27.08** | **0.12** | **0.85** | **27.57** | **0.10** | **0.87** | **29.05** | **0.12** | **0.86** |
| 4 | Ev-DeblurNeRF [12] | 9.99 | 0.67 | 0.08 | 15.81 | 0.72 | 0.12 | 10.81 | 0.75 | 0.08 | 8.80 | 0.71 | 0.05 | 11.35 | 0.71 | 0.08 |
| | Ours (Frame Only) | 18.34 | 0.33 | 0.46 | 20.43 | 0.56 | 0.35 | 20.05 | 0.59 | 0.31 | 18.84 | 0.33 | 0.52 | 19.42 | 0.45 | 0.41 |
| | Ours (Frame + Event) | **30.79** | **0.07** | **0.92** | **25.25** | **0.35** | **0.58** | **23.97** | **0.21** | **0.74** | **25.93** | **0.13** | **0.83** | **26.49** | **0.19** | **0.77** |

**Gen3-HandHeld and Gen3-DroneFlight.** We collect two fast-motion datasets using a beamsplitter setup (Figure 3) consisting of a Prophesee Gen3 event camera (640×480 resolution) and a color FLIR Blackfly S camera, both viewing the same scene through the beamsplitter and hardware-synchronized. The Prophesee Gen3 features an IMU that is synchronized with the event stream and used for pose estimation with UltimateSLAM [33]. We run UltimateSLAM [33] on the full trajectory to obtain an initial pose prior, and then subsample the training images to replicate the setup of Ev-DeblurNeRF [12], using approximately 34 blurred training images and 5 sharp test images per sequence. Gen3-HandHeld features fast motion performed with the rig handheld, moving at variable speeds and with different exposure times for the RGB camera. We use this dataset to evaluate robustness to varying levels of blur severity. Gen3-HandHeld includes three exposure settings (10ms, 30ms and 50ms) and three speed profiles (0.8∼1 m/s, 1.2∼1.4 m/s, 1.4∼1.8 m/s) for a total of 9 sequences. To demonstrate the effectiveness of our approach in real-world scenarios, we record another dataset, named Gen3-DroneFlight, featuring the same beamsplitter setup but mounted on a real quadrotor (shown in Figure 3). We use this setup to perform flight with left-to-right, top-to-down motions in front of target scenes, reaching speeds of up to 2 m/s. We use an external motion capture system to maneuver the drone, as well as to record ground-truth camera poses for evaluation. Crucially, the motion capture poses are never used during training. These poses are used solely for evaluation, as our system only relies on approximate poses from UltimateSLAM [33].

### C. Baselines

We follow the evaluation setup established in previous work [12], ensuring consistent comparisons across both frame-only and event-driven baselines. For NeRF-based frame-only deblurring, we consider DeblurNeRF [25], BAD-NeRF [28], DP-NeRF [27], and PDRF [26]. We also include video deblurring approaches (MPRNet [44], PVDNet [45] and EFNet [9]), followed by scene reconstruction using NeRF [2]. For EFNet [9], we also evaluate a variant, denoted as EFNet*, where the pretrained model is applied to a given scene after finetuning on the other scenes in the dataset. Among event-based methods, we evaluate E-NeRF [42], E$^2$-NeRF [43], and Ev-DeblurNeRF [12]. All baselines are run with official code and default hyperparameters, using the same input data and poses for fairness. Similar to [12], we perform extensive evaluation in simulation and then select the best-performing baselines for a more in-depth analysis on real-world scenarios.

### D. Experimental Validation

**State-of-the-Art Comparison.** We begin our evaluation by benchmarking the proposed method on the Ev-DeblurBlender and Ev-DeblurCDAVIS datasets. Both datasets provide ground-truth camera poses, allowing us to fairly compare against baselines that do not perform pose refinement during training. This setting also serves to validate that our method, under ideal conditions, can accurately infer intermediate poses during exposure and at event timestamps and achieve sharp reconstruction quality on par with the best existing approaches.

Results for the synthetic Ev-DeblurBlender dataset are reported in Table I. Our method consistently outperforms all baselines. It achieves a relative PSNR improvement of 10.7% over the best-performing image-only NeRF-based method and surpasses two-stage approaches that first deblur images and then train a NeRF by 5.7% in PSNR. Compared to event-based NeRF methods such as E-NeRF [42] and E$^2$-NeRF [43], our method provides a 17.9% PSNR gain.

We observe analogous trends on the real-world Ev-DeblurCDAVIS dataset, as shown in Table II. Our method again surpasses frame-only baselines, as well as all event-driven approaches. In particular, we outperform E$^2$-NeRF [43] by 8.4% PSNR and perform on par with Ev-DeblurNeRF [12]. Although our method performs on par with Ev-DeblurNeRF [12], as will be shown in the next section, its advantage becomes more evident as pose noise increases. We provide qualitative comparisons in Figure 5.

**Robustness to Pose Noise.** We next evaluate the robustness of our method to noisy camera trajectories using the NoisyPose-EvDeblurBlender dataset, a modified version of Ev-DeblurBlender where training poses are perturbed by synthetic drift. This perturbation simulates the gradual degradation

TABLE IV: Pose estimation error metrics under different pose noise levels on NoisyPose-EvDeblurBlender. (ATE RMSE [cm] ↓, RPE translation [%] ↓, RPE rotation [deg/m] ↓)

| Noise Level | Method | Factory ATE | Factory $\text{RPE}_{\text{trans}}$ | Factory $\text{RPE}_{\text{rot}}$ | Pool ATE | Pool $\text{RPE}_{\text{trans}}$ | Pool $\text{RPE}_{\text{rot}}$ | Tanabata ATE | Tanabata $\text{RPE}_{\text{trans}}$ | Tanabata $\text{RPE}_{\text{rot}}$ | Trolley ATE | Trolley $\text{RPE}_{\text{trans}}$ | Trolley $\text{RPE}_{\text{rot}}$ | Average ATE | Average $\text{RPE}_{\text{trans}}$ | Average $\text{RPE}_{\text{rot}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Perturbed | 9.96 | 5.41 | 0.23 | 4.73 | 6.49 | 1.62 | 21.55 | 6.61 | 0.19 | 23.19 | 10.08 | 0.32 | 14.86 | 7.15 | 0.59 |
| | Frame Only | 11.13 | 0.83 | 0.12 | 0.99 | 0.76 | 0.26 | 2.57 | 0.74 | 0.18 | 3.11 | 0.98 | 0.14 | 4.45 | 0.83 | 0.18 |
| | Frame + Event | **0.21** | **0.12** | **0.03** | **0.14** | **0.16** | **0.13** | **0.58** | **0.21** | **0.03** | **0.87** | **0.24** | **0.08** | **0.45** | **0.18** | **0.07** |
| 2 | Perturbed | 18.54 | 10.46 | 0.46 | 8.88 | 12.21 | 3.23 | 41.42 | 12.78 | 0.39 | 45.51 | 20.04 | 0.63 | 28.59 | 13.87 | 1.18 |
| | Frame Only | 1.14 | 0.79 | 0.12 | 0.81 | 0.79 | 0.26 | 4.46 | 0.85 | 0.14 | 3.11 | 1.03 | 0.29 | 2.38 | 0.86 | 0.20 |
| | Frame + Event | **0.32** | **0.14** | **0.05** | **0.14** | **0.17** | **0.10** | **1.02** | **0.34** | **0.11** | **1.25** | **0.33** | **0.13** | **0.68** | **0.25** | **0.10** |
| 3 | Perturbed | 32.20 | 19.03 | 0.91 | 15.55 | 21.23 | 6.46 | 74.71 | 23.01 | 0.77 | 83.10 | 36.16 | 1.27 | 51.39 | 24.86 | 2.35 |
| | Frame Only | 6.25 | 0.97 | 0.23 | 4.24 | 1.73 | 0.27 | 10.13 | 3.68 | 1.17 | 4.76 | 1.48 | 0.53 | 6.35 | 1.96 | 0.55 |
| | Frame + Event | **0.52** | **0.26** | **0.10** | **1.36** | **1.10** | **0.10** | **3.02** | **0.79** | **0.27** | **3.23** | **0.87** | **0.42** | **2.03** | **0.75** | **0.22** |
| 4 | Perturbed | 42.69 | 25.63 | 1.37 | 20.43 | 27.58 | 9.70 | 99.50 | 30.31 | 1.16 | 108.12 | 45.55 | 1.90 | 67.69 | 32.27 | 3.53 |
| | Frame Only | 14.26 | 5.80 | 2.21 | 20.11 | 25.52 | 0.90 | 15.52 | 4.83 | 1.40 | 22.98 | 9.40 | 1.40 | 18.22 | 11.39 | 1.48 |
| | Frame + Event | **1.00** | **0.50** | **0.20** | **15.21** | **16.99** | **0.73** | **8.70** | **2.50** | **0.81** | **4.96** | **1.60** | **0.82** | **7.47** | **5.40** | **0.64** |

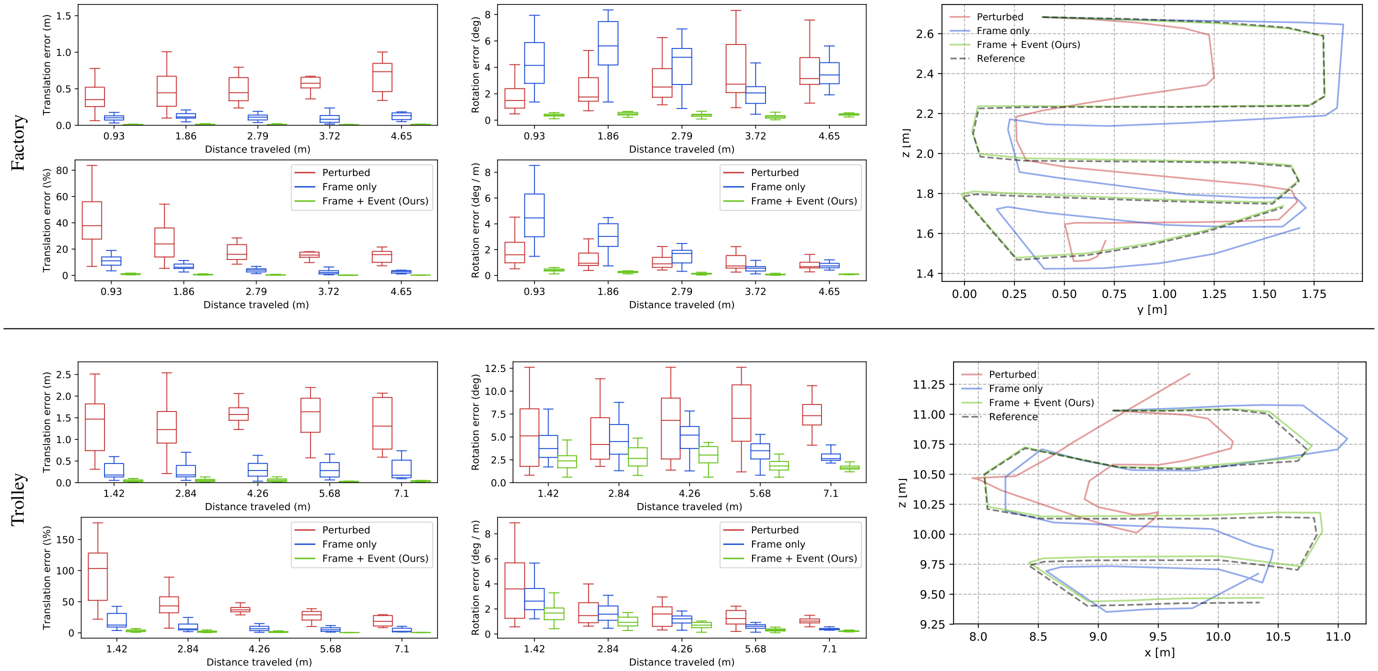


Fig. 4: Factory (top) and Trolley (bottom) trajectory analysis on NoisyPose-EvDeblurBlender at noise level 4. We present the change of trajectory errors with traveled distance (left), for both translation (meters and percentage) and rotation (degrees and degrees per meter), along with a visual comparison of the trajectories (right).

commonly observed in visual-inertial odometry systems during prolonged motion. Results are reported in Table III and illustrated qualitatively in Figure 6.

For this study, we compare against Ev-DeblurNeRF [12], the best-performing baseline, which also shares the most architectural similarities with our method and thus serves as the most meaningful baseline to isolate the effect of trajectory refinement. Our pose refinement strategy is fully learned and unconstrained by fixed coordinate frames. As such, it can result in globally shifted or rotated trajectories that are photometrically valid but misaligned with the ground-truth frame. To account for this, we first align ground-truth poses to the learned trajectory via Procrustes alignment on the training views. We then further register the test poses to the learned NeRF by minimizing a photometric loss between the reference and rendered views, following BAD-NeRF protocol [28].

Ev-DeblurNeRF [12] struggles to recover meaningful geometry, with PSNR dropping to 14.82 dB on average in the mildest noise case (see Table III), compared to the 30.42 dB achieved when using ground-truth poses (see Table I). In contrast, our method consistently compensates for the drift, achieving an average of 29.88 dB PSNR across scenes. As the noise level increases, performance degrades but remains robust even in the most challenging setting, yielding an average of 26.49 dB despite substantial initial trajectory drift. Notably, in the longest sequences, *trolley* and *tanabata*, where drift affects the perturbed trajectory the most (108.12 ATE and 99.50 ATE, respectively; see Table IV), our method still successfully recovers geometry, achieving 25.93 dB and 23.97 dB PSNR.

The impact of our joint pose refiner module is further

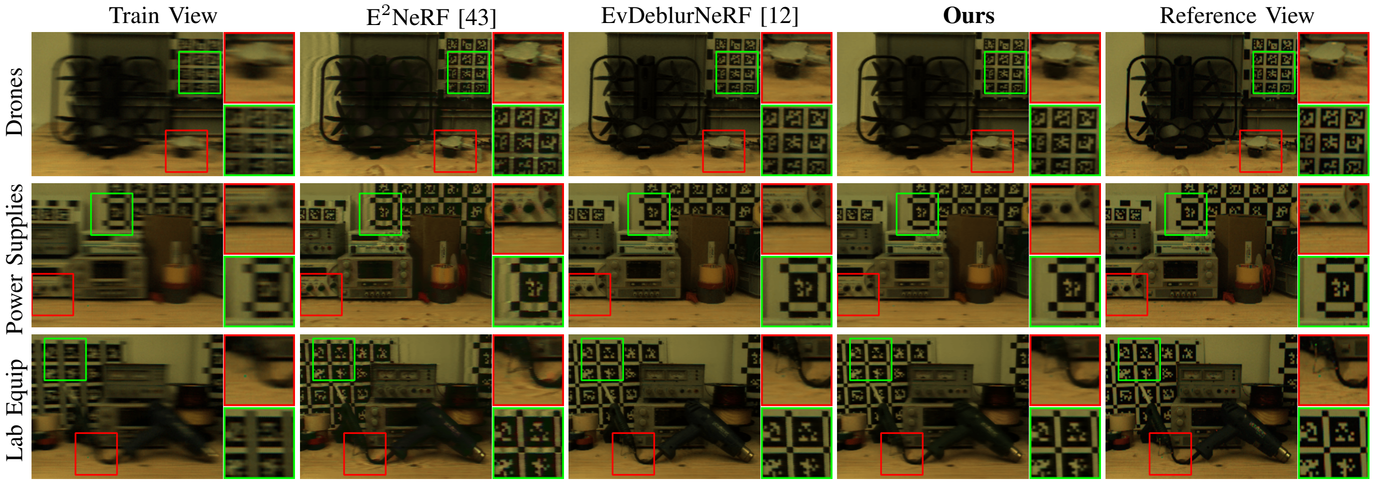


Fig. 5: Novel view synthesis comparison on the Ev-DeblurCDAVIS dataset.

TABLE V: Quantitative comparison on the new Gen3-HandHeld dataset, and ablation of speed vs. exposure time. USLAM stands for UltimateSLAM [33].

| Exposure | Poses | Method | Max Speed 0.8∼1 m/s | | | Max Speed 1.2∼1.4 m/s | | | Max Speed 1.4∼1.8 m/s | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ |
| 10ms | USLAM | Ev-DeblurNeRF [12] | 16.92 | 0.57 | 0.45 | 16.30 | 0.59 | 0.42 | 15.99 | 0.64 | 0.42 | 16.40 | 0.60 | 0.43 |
| | USLAM | Ours | **28.14** | **0.21** | **0.77** | **27.34** | **0.23** | **0.76** | **26.69** | **0.27** | **0.74** | **27.39** | **0.24** | **0.76** |
| | our refined | Ev-DeblurNeRF [12] | 24.03 | 0.24 | 0.72 | 24.28 | 0.26 | 0.72 | 23.17 | 0.32 | 0.69 | 23.83 | 0.27 | 0.71 |
| | our refined | E$^2$NeRF [43] | 23.77 | 0.24 | 0.71 | 23.47 | 0.30 | 0.68 | 21.46 | 0.37 | 0.62 | 22.90 | 0.30 | 0.67 |
| 30ms | USLAM | Ev-DeblurNeRF [12] | 16.49 | 0.46 | 0.43 | 16.33 | 0.51 | 0.45 | 15.69 | 0.55 | 0.43 | 16.17 | 0.51 | 0.44 |
| | USLAM | Ours | **24.57** | **0.14** | **0.77** | **24.73** | **0.18** | **0.77** | **23.27** | **0.21** | **0.73** | **24.19** | **0.18** | **0.76** |
| | our refined | Ev-DeblurNeRF [12] | 24.04 | 0.15 | 0.77 | 23.79 | 0.19 | 0.76 | 22.57 | 0.25 | 0.71 | 23.47 | 0.20 | 0.74 |
| | our refined | E$^2$NeRF [43] | 22.99 | 0.19 | 0.72 | 23.09 | 0.33 | 0.61 | 21.22 | 0.35 | 0.63 | 22.43 | 0.29 | 0.65 |
| 50ms | USLAM | Ev-DeblurNeRF [12] | 14.34 | 0.53 | 0.36 | 14.00 | 0.60 | 0.37 | 13.16 | 0.63 | 0.33 | 13.84 | 0.59 | 0.35 |
| | USLAM | Ours | **23.80** | **0.13** | **0.78** | **22.52** | **0.18** | **0.74** | **21.31** | **0.26** | **0.68** | **22.54** | **0.19** | **0.73** |
| | our refined | Ev-DeblurNeRF [12] | 22.96 | 0.14 | 0.76 | 21.50 | 0.21 | 0.71 | 18.28 | 0.35 | 0.58 | 20.91 | 0.23 | 0.68 |
| | our refined | E$^2$NeRF [43] | 21.95 | 0.22 | 0.69 | 19.12 | 0.32 | 0.60 | 20.26 | 0.40 | 0.62 | 20.44 | 0.31 | 0.64 |

demonstrated in Table IV, where we analyze the trajectory error of the poses recovered by our method after the full training procedure has converged. We denote the noisy input trajectory as *perturbed*, and report performance for *frame only* and *frame + events* configurations, where the event-based branch of our network is disabled or enabled, respectively. We measure Absolute Trajectory Error RMSE (ATE RMSE) in centimeters, Relative Pose Error in translation ($\text{RPE}_{\text{trans}}$) as a percentage of the traveled distance, and Relative Pose Error in rotation ($\text{RPE}_{\text{rot}}$) in degrees per meter, using the toolbox proposed in [48]. For RPE, we divide the ground-truth trajectory into six equal-length segments, use the five segment midpoints as reference poses, compute both translational and rotational errors over the segment length at each reference pose, and average the five resulting errors.

The results in Table IV show that combining both frames and events leads to significantly improved performance, as the pose refiner benefits from supervision not only from the frames but also from the high-temporal-resolution information provided by events. On average, using frames alone improves the initial perturbed trajectory from 67.69 cm to 18.22 cm ATE in the most challenging setting, while incorporating events yields a further improvement, lowering ATE to only 7.47 cm. When the initial poses are within 30 cm (noise levels 1 and 2), our full method with events achieves sub-centimeter ATE, highlighting the contribution of the event-based supervision.

While precise trajectory recovery is not the primary objective of our method, these results demonstrate its ability to achieve accurate poses by exploiting dense 3D reconstruction as a way to jointly refine poses during training. Additional results are provided in Figure 4, where we qualitatively visualize the refined trajectories and report the changes in trajectory error with the distance traveled.

**Gen3-HandHeld Results.** We continue our evaluation on real-world sequences captured via a beamsplitter setup. Compared to the CDAVIS dataset, this setup presents greater challenges, as it consists of two independent sensors with distinct responses to light. This necessitates modeling these differences explicitly, which we achieve through the learnable camera response function (CRF). We first focus on the Gen3-HandHeld dataset, which comprises handheld scenes recorded with varying camera motion speeds and RGB exposure times. To ensure comparable brightness across settings, we manually adjust the lens aperture when increasing

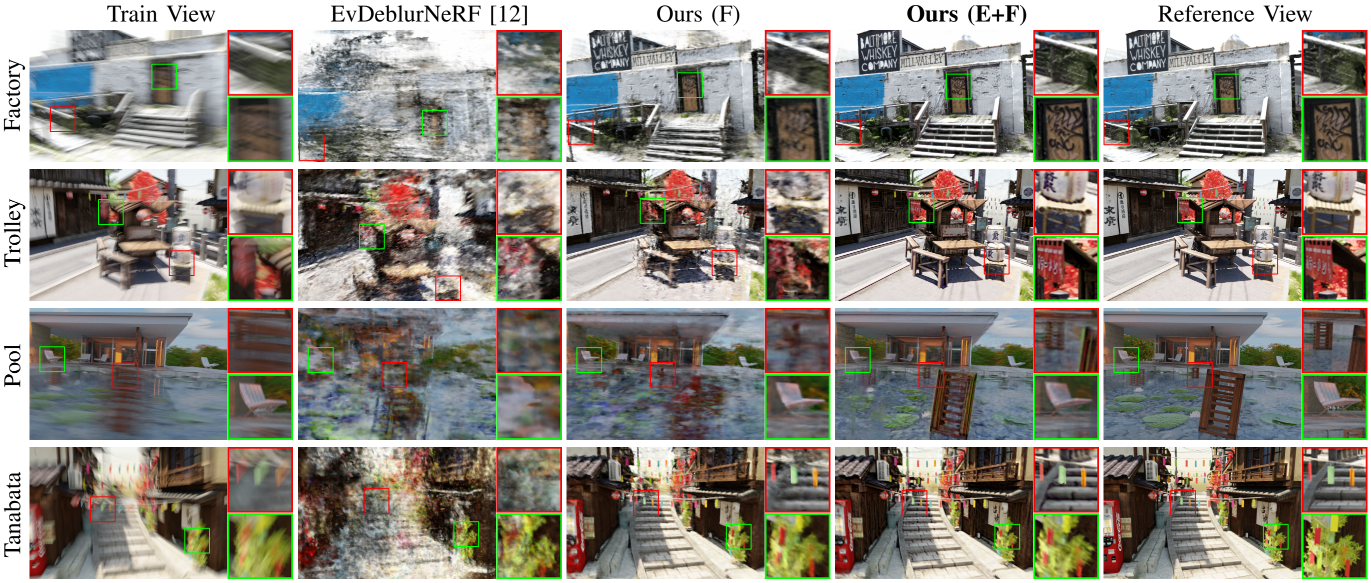


Fig. 6: Novel view synthesis comparison on the NoisyPose-EvDeblurBlender dataset under noise level four. We indicate *frame-only* methods with (F) and *frames+events* with (F+E), while *Train view* refers to the closest blurry image in the training set.

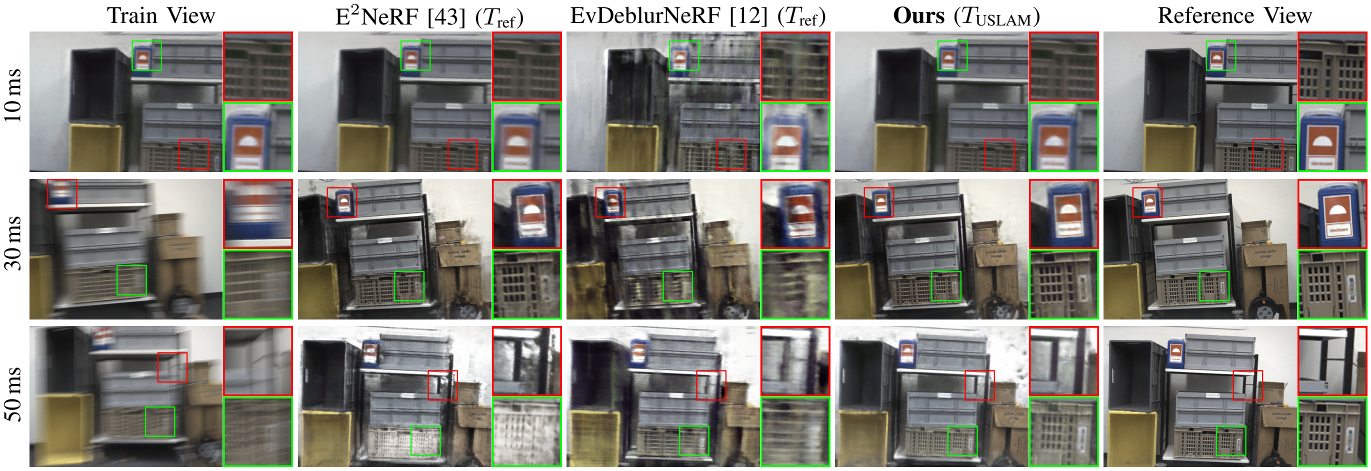

Train View
E$^2$NeRF [43] ($T_{\text{ref}}$)
EvDeblurNeRF [12] ($T_{\text{ref}}$)
**Ours** ($T_{\text{USLAM}}$)
Reference View
10 ms
30 ms
50 ms

Fig. 7: Novel view synthesis comparison on the Gen3-HandHeld dataset under the fastest speed profile. We indicate in parentheses the poses each method uses as initialization, $T_{\text{ref}}$ for our refined poses, $T_{\text{USLAM}}$ for UltimateSLAM [33] poses.

exposure durations. Camera poses are initially estimated using UltimateSLAM [33] and subsequently refined during training. For comparison, we include the best-performing event-based baselines, namely Ev-DeblurNeRF [12] and E$^2$NeRF [43]. To ensure a fair comparison, we provide these baselines with the refined poses obtained from our method at convergence, representing a best-case scenario where they benefit from more accurate trajectories, although such supervision would not be available in practice. To further highlight the contribution of our joint pose refinement, we also report results for Ev-DeblurNeRF [12] trained directly on the original UltimateSLAM [33] poses.

The results are shown in Table V. Despite using noisy UltimateSLAM poses [33] as initialization, our method maintains high performance across different exposure times and speed configurations, achieving 21.31 dB PSNR in the most challenging setting. In contrast, Ev-DeblurNeRF [12] using the same poses fails to produce competitive results in all settings due to the lack of a pose refinement stage. The performance gap between Ev-DeblurNeRF [12] and our method is consistently greater than 7 dB in PSNR. Crucially, even when provided with our refined poses, both Ev-DeblurNeRF [12] and E$^2$NeRF [43] baselines consistently underperform relative to our approach. The first is limited to linearly interpolating poses at event timestamps, while the second only considers events during image exposures, thus underutilizing high-frequency supervision. Qualitative results illustrating these findings are in Figure 7.

**Gen3-DroneFlight Results.** We present the results for drone flight sequences in Table VI and Figure 8. These results conclude our analysis by demonstrating the successful deployment of our algorithm on data collected onboard a real aerial robotic platform. The Gen3-DroneFlight dataset presents the

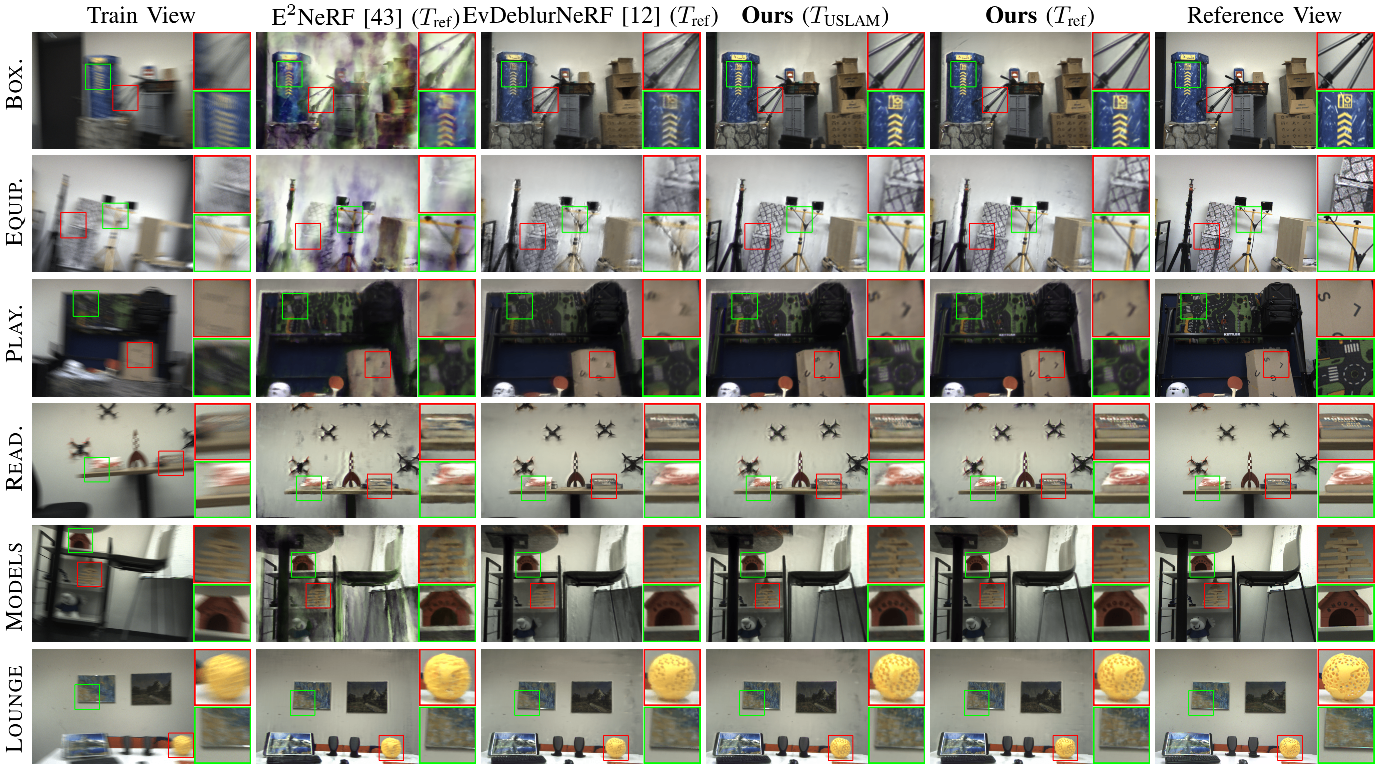

Train View
E$^2$NeRF [43] ($T_{\text{ref}}$)
EvDeblurNeRF [12] ($T_{\text{ref}}$)
**Ours** ($T_{\text{USLAM}}$)
**Ours** ($T_{\text{ref}}$)
Reference View
Box.
Equip.
Play.
Read.
Models
Lounge

Fig. 8: Novel view synthesis comparison on the Gen3-DroneFlight dataset under the fastest speed profile. We indicate in parentheses the poses each method uses as initialization, $T_{\text{ref}}$ for our refined poses, and $T_{\text{USLAM}}$ for UltimateSLAM [33] poses.

most challenging scenarios in our evaluation: aerial sequences recorded at 30 ms exposure, where the drone follows a zig-zag trajectory in front of three different scenes (a visual representation is provided in Figure 1). This motion induces substantial lateral blur across the entire image, unlike circular trajectories where regions near the rotation center remain relatively sharp. The challenge is further amplified by high-frequency vibrations from the drone's motors, which introduce additional disturbances to both images and events.

From Table VI we can see that our method maintains the best performance across varying scenes and speed profiles. We continue to outperform the Ev-DeblurNeRF [12] and E$^2$NeRF [43] baselines, even when they are provided with our refined poses as input. While in this case, the baselines benefit from starting with accurate trajectories, our method begins from noisy, inaccurate poses and must learn to refine them during training, thus experiencing not only motion blur but also visual distortions caused by inaccurate camera trajectories. Despite this additional challenge, our method achieves average improvements of 4.3% in PSNR, 7.1% in LPIPS, and 6% in SSIM, highlighting the strength of our joint trajectory and radiance field optimization. When provided with our refined poses as initialization, our method achieves even better reconstruction quality. Qualitative results are provided in Figure 8, where our method clearly recovers fine textures and structural details that are lost in the baseline reconstructions.

**Robustness to Trajectory Length.** We evaluate the robustness of our approach to varying trajectory lengths using the Gen3-DroneFlight dataset. The trajectories in scenes BoxStack, Equipment, and PlayCorner measure approximately 7.5–8 meters, while those in scenes Models, ReadingCorner, and Lounge are roughly twice as long, at 15–16 meters. The results in Table VI indicate that our method reliably outperforms all baselines across trajectories of both lengths.

Results from different scenes are not directly comparable due to scene-specific factors such as different textures and object layout. To mitigate these scene-dependent variations and enable a more reliable scalability evaluation, we captured a larger-scale scene covering the combined area of the Models and ReadingCorner scenes. The drone was flown along a 30-meter trajectory at the fastest speed profile (maximum speed of 2 m/s). The results are reported in Table VII. As shown, our method maintains stable reconstruction quality as the trajectory length increases, demonstrating robustness to larger-scale trajectories.

**Component Validation and Ablation Study.** We validate the technical components of our method on the playCorner scene from the Gen3-DroneFlight dataset. Specifically, we ablate the loss terms in Eq. 6 (blur loss $\mathcal{L}_{\text{blur}}$), Eq. 7 (prior loss $\mathcal{L}_{\text{prior}}$) and Eq. 9 (event loss $\mathcal{L}_{\text{event}}$), and we experimentally evaluate the eCRF module of Eq. 8. Note that we do not include an event-only variant, since events lack absolute intensity and color information and a radiance field trained solely with event supervision is not comparable to RGB-supervised variants. Therefore, we demonstrate the contribution of event supervision by adding or removing the event loss to models trained with at least one RGB loss. Results are presented in

TABLE VI: Quantitative comparison on the new Gen3-DroneFlight dataset. USLAM stands for UltimateSLAM [33].

| Sequence | Poses | Method | Max Speed 1 m/s | | | Max Speed 1.5 m/s | | | Max Speed 2 m/s | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ |
| BoxStack | USLAM | Ev-DeblurNeRF [12] | 13.34 | 0.58 | 0.30 | 12.08 | 0.64 | 0.25 | 13.58 | 0.66 | 0.30 | 13.00 | 0.63 | 0.28 |
| | USLAM | Ours | **23.89** | **0.21** | **0.68** | **23.49** | **0.26** | **0.67** | **23.05** | **0.33** | **0.66** | **23.48** | **0.26** | **0.67** |
| | our refined | Ev-DeblurNeRF [12] | 21.83 | 0.23 | 0.61 | 22.37 | 0.27 | 0.63 | 21.12 | 0.36 | 0.59 | 21.78 | 0.29 | 0.61 |
| | our refined | E$^2$NeRF [43] | 22.23 | 0.28 | 0.61 | 18.41 | 0.45 | 0.45 | 14.38 | 0.62 | 0.33 | 18.34 | 0.45 | 0.46 |
| | our refined | Ours | **24.40** | **0.21** | **0.71** | **23.58** | **0.25** | **0.67** | **23.31** | **0.30** | **0.67** | **23.76** | **0.26** | **0.68** |
| Equipment | USLAM | Ev-DeblurNeRF [12] | 13.86 | 0.53 | 0.30 | 13.49 | 0.63 | 0.27 | 13.08 | 0.73 | 0.24 | 13.48 | 0.63 | 0.27 |
| | USLAM | Ours | **23.78** | **0.23** | **0.72** | **22.90** | **0.26** | **0.69** | **22.07** | **0.36** | **0.65** | **22.91** | **0.28** | **0.69** |
| | our refined | Ev-DeblurNeRF [12] | 21.54 | 0.28 | 0.66 | 21.02 | 0.31 | 0.62 | 19.21 | 0.44 | 0.53 | 20.59 | 0.34 | 0.60 |
| | our refined | E$^2$NeRF [43] | 21.09 | 0.32 | 0.62 | 17.14 | 0.46 | 0.43 | 12.70 | 0.68 | 0.28 | 16.97 | 0.49 | 0.44 |
| | our refined | Ours | **23.77** | **0.23** | **0.73** | **23.01** | **0.28** | **0.69** | **22.07** | **0.35** | **0.65** | **22.95** | **0.29** | **0.69** |
| PlayCorner | USLAM | Ev-DeblurNeRF [12] | 13.30 | 0.70 | 0.31 | 11.66 | 0.78 | 0.20 | 11.92 | 0.76 | 0.26 | 12.29 | 0.75 | 0.26 |
| | USLAM | Ours | **25.47** | **0.34** | **0.75** | **23.91** | **0.39** | **0.69** | **23.40** | **0.41** | **0.66** | **24.26** | **0.38** | **0.70** |
| | our refined | Ev-DeblurNeRF [12] | 23.68 | 0.36 | 0.66 | 21.16 | 0.44 | 0.60 | 22.51 | 0.43 | 0.63 | 22.45 | 0.41 | 0.63 |
| | our refined | E$^2$NeRF [43] | 22.63 | 0.38 | 0.63 | 21.35 | 0.47 | 0.58 | 22.41 | 0.51 | 0.61 | 22.13 | 0.45 | 0.60 |
| | our refined | Ours | **25.54** | **0.33** | **0.74** | **24.20** | **0.36** | **0.69** | **23.55** | **0.40** | **0.67** | **24.43** | **0.37** | **0.70** |
| Models | USLAM | Ev-DeblurNeRF [12] | 10.50 | 0.74 | 0.19 | 10.71 | 0.77 | 0.16 | 11.48 | 0.78 | 0.18 | 10.90 | 0.76 | 0.18 |
| | USLAM | Ours | **26.60** | **0.16** | **0.79** | **25.04** | **0.20** | **0.74** | **24.81** | **0.24** | **0.70** | **25.48** | **0.20** | **0.74** |
| | our refined | Ev-DeblurNeRF [12] | 26.43 | 0.17 | 0.79 | 24.95 | 0.21 | 0.75 | 24.09 | 0.28 | 0.68 | 25.16 | 0.22 | 0.74 |
| | our refined | E$^2$NeRF [43] | 24.02 | 0.18 | 0.73 | 22.67 | 0.25 | 0.68 | 21.81 | 0.32 | 0.62 | 22.83 | 0.25 | 0.68 |
| | our refined | Ours | **26.90** | **0.15** | **0.80** | **25.76** | **0.18** | **0.76** | **25.11** | **0.22** | **0.72** | **25.93** | **0.18** | **0.76** |
| ReadingCorner | USLAM | Ev-DeblurNeRF [12] | 13.23 | 0.69 | 0.36 | 13.76 | 0.67 | 0.39 | 11.77 | 0.70 | 0.13 | 12.92 | 0.69 | 0.29 |
| | USLAM | Ours | **27.40** | **0.20** | **0.76** | **27.22** | **0.22** | **0.75** | **25.70** | **0.29** | **0.72** | **26.77** | **0.24** | **0.74** |
| | our refined | Ev-DeblurNeRF [12] | 27.60 | 0.19 | 0.76 | 26.81 | 0.22 | 0.74 | 26.46 | 0.25 | **0.76** | 26.96 | 0.22 | 0.75 |
| | our refined | E$^2$NeRF [43] | 25.66 | 0.20 | 0.72 | 26.20 | 0.23 | 0.72 | 24.40 | 0.31 | 0.68 | 25.42 | 0.25 | 0.71 |
| | our refined | Ours | **27.63** | **0.19** | **0.76** | **27.42** | **0.21** | **0.76** | **26.62** | **0.24** | 0.75 | **27.22** | **0.21** | **0.76** |
| Lounge | USLAM | Ev-DeblurNeRF [12] | 12.26 | 0.68 | 0.17 | 12.65 | 0.69 | 0.18 | 11.74 | 0.69 | 0.13 | 12.22 | 0.69 | 0.16 |
| | USLAM | Ours | **27.88** | **0.23** | **0.75** | **27.33** | **0.18** | **0.72** | **25.61** | **0.22** | **0.67** | **26.94** | **0.21** | **0.71** |
| | our refined | Ev-DeblurNeRF [12] | 27.65 | 0.23 | 0.74 | 27.32 | 0.18 | 0.72 | 25.28 | 0.22 | 0.66 | 26.75 | 0.21 | 0.71 |
| | our refined | E$^2$NeRF [43] | 27.13 | **0.19** | 0.74 | 26.60 | 0.17 | 0.71 | 24.84 | 0.21 | 0.64 | 26.19 | 0.19 | 0.70 |
| | our refined | Ours | **28.76** | 0.20 | **0.77** | **27.57** | **0.17** | **0.73** | **25.68** | **0.20** | **0.67** | **27.34** | **0.19** | **0.72** |
| Average | USLAM | Ev-DeblurNeRF [12] | 12.75 | 0.65 | 0.27 | 12.39 | 0.70 | 0.24 | 12.26 | 0.72 | 0.21 | 12.47 | 0.69 | 0.24 |
| | USLAM | Ours | **25.84** | **0.23** | **0.74** | **24.98** | **0.25** | **0.71** | **24.11** | **0.31** | **0.68** | **24.98** | **0.26** | **0.71** |
| | our refined | Ev-DeblurNeRF [12] | 24.79 | 0.24 | 0.70 | 23.94 | 0.27 | 0.68 | 23.11 | 0.33 | 0.64 | 23.95 | 0.28 | 0.67 |
| | our refined | E$^2$NeRF [43] | 23.79 | 0.26 | 0.68 | 22.06 | 0.34 | 0.60 | 20.09 | 0.44 | 0.53 | 21.98 | 0.35 | 0.60 |
| | our refined | Ours | **26.17** | **0.22** | **0.75** | **25.26** | **0.24** | **0.72** | **24.39** | **0.29** | **0.69** | **25.27** | **0.25** | **0.72** |

Table VIII.

Adding the event loss on top of the blur loss yields large improvements (+20% PSNR, +21% LPIPS, +31% SSIM on average), confirming the strong benefit of event supervision. When using only the prior loss, compared to using only the blur loss, the performance gap increases significantly from +0.21dB at 1 m/s to +5.51dB in the 2 m/s case, with an average gap of +2.53dB PSNR, demonstrating the usefulness of the pseudo ground-truth target obtained from the model-based deblurring process, particularly in high-speed scenarios. Similarly, after adding the event loss on top of the prior loss, the average performance increases considerably (+7% PSNR, +21% LPIPS and +17% SSIM), which again demonstrates the significance of event supervision. Combining the blur loss and the prior loss yields better results than using either loss alone. Nonetheless, configurations that include the event loss (e.g., blur+event or prior+event) consistently outperform the blur+prior combination. The highest performance is achieved when all three loss terms are used jointly. Finally, removing the eCRF module leads to noticeable performance degradation (-2% PSNR, -11% LPIPS, -3% SSIM), demonstrating the effectiveness of modeling the event camera response function.

## V. Discussion and Limitations

**Extension to Gaussian Splatting.** Our focus in this work is accurate and stable reconstruction under the fast motion of aerial drones, where robustness to sparse input frames and generalization to unseen viewpoints are critical due to high motion speed and limited capture frequency. Recent analyses [49], [50] indicate that NeRFs are more stable to train and provide stronger geometric consistency and better generalization than 3D Gaussian Splatting with limited training views. Our choice of a NeRF backbone therefore prioritizes reconstruction fidelity and robustness under sparse training views. However, our framework is agnostic to the underlying representation of the scene. In scenarios where rendering speed is more important, our method can be naturally extended to 3DGS. Compared to NeRF, which represents the scene implicitly through a neural network, 3DGS models it explicitly as a set of Gaussian splats parameterized by position, covariance, and appearance. By rendering these splats differentiably with

TABLE VII: Quantitative comparison on trajectories of different lengths. All trajectories were flown by a real drone under the fastest speed profile. Scene 1: MODELS, Scene 2: READINGCORNER.

| Poses | Method | Scene 1 (15 m Traj.) | | | Scene 2 (15 m Traj.) | | | Scene 1+2 (30 m Traj.) | | | AVERAGE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ |
| USLAM | Ev-DeblurNeRF [12] | 11.48 | 0.78 | 0.18 | 11.77 | 0.70 | 0.13 | 10.22 | 0.77 | 0.15 | 11.16 | 0.75 | 0.15 |
| USLAM | Ours | **24.81** | **0.24** | **0.70** | **25.70** | **0.29** | **0.72** | **25.59** | **0.25** | **0.74** | **25.37** | **0.26** | **0.72** |
| our refined | Ev-DeblurNeRF [12] | 24.09 | 0.28 | 0.68 | 26.46 | 0.25 | **0.76** | 25.93 | 0.24 | 0.75 | 25.49 | 0.26 | 0.73 |
| our refined | $E^2$NeRF [43] | 21.81 | 0.32 | 0.62 | 24.40 | 0.31 | 0.68 | 24.17 | 0.27 | 0.70 | 23.46 | 0.30 | 0.67 |
| our refined | Ours | **25.11** | **0.22** | **0.72** | **26.62** | **0.24** | 0.75 | **26.13** | **0.23** | **0.75** | **25.95** | **0.23** | **0.74** |

TABLE VIII: Ablation study on the technical components of our method.

| $\mathcal{L}_{\text{blur}}$ | $\mathcal{L}_{\text{prior}}$ | $\mathcal{L}_{\text{event}}$ | Max Speed 1 m/s | | | Max Speed 1.5 m/s | | | Max Speed 2 m/s | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ | PSNR↑ | LPIPS↓ | SSIM↑ |
| ✓ | | | 23.60 | 0.44 | 0.66 | 18.65 | 0.60 | 0.46 | 15.74 | 0.68 | 0.36 | 19.33 | 0.57 | 0.49 |
| ✓ | | ✓ | 25.10 | 0.37 | 0.71 | 21.75 | 0.52 | 0.59 | 23.01 | 0.46 | 0.63 | 23.29 | 0.45 | 0.64 |
| | ✓ | | 23.81 | 0.43 | 0.67 | 20.53 | 0.57 | 0.55 | 21.25 | 0.55 | 0.56 | 21.86 | 0.52 | 0.59 |
| | ✓ | ✓ | 24.73 | 0.35 | 0.74 | 23.53 | 0.41 | 0.71 | 21.75 | 0.47 | 0.62 | 23.34 | 0.41 | 0.69 |
| ✓ | ✓ | | 24.11 | 0.42 | 0.67 | 21.48 | 0.52 | 0.57 | 21.86 | 0.51 | 0.58 | 22.49 | 0.48 | 0.61 |
| ✓ | ✓ | ✓ | **25.47** | **0.34** | **0.75** | **23.91** | **0.39** | **0.69** | **23.40** | **0.41** | **0.66** | **24.26** | **0.38** | **0.70** |
| w/o eCRF | | | 24.96 | 0.36 | 0.72 | 23.37 | 0.46 | 0.66 | 23.24 | 0.45 | 0.65 | 23.85 | 0.42 | 0.68 |

respect to the continuous-time camera trajectory analogous to our current formulation, the same image- and event-based supervision can be used to jointly optimize both the splat parameters and the camera poses.

**Alternative Hardware Settings.** In this work, we use a beamsplitter-based sensor rig to obtain synchronized RGB images and events. This hardware setting can be replaced in several ways in real-world deployment. A promising hardware alternative is the emerging field of hybrid vision sensors (HVS) that provide co-registered frame and event outputs on the same chip [51]–[53], eliminating synchronization and co-location issues as the image and event outputs share the same pixel array. When such hybrid sensors are unavailable, non-synchronized cameras (independently mounted RGB and event cameras) can still be used, provided that accurate timestamps and extrinsic calibrations are available. In this case, unsynchronized captures could be handled by introducing time offset parameters between sensors and optimizing them jointly with the continuous-time trajectory and scene parameters, ensuring temporal alignment during training. Similarly, small extrinsic errors can also be refined during optimization.

**Motion Capture Poses for Evaluation.** Our current quantitative evaluation uses motion capture poses for rendering to compute accurate metrics and ensure fair comparison across methods. However, such infrastructure is often unavailable in practical deployment. We propose several alternatives that do not rely on motion capture for evaluating reconstruction quality: (i) adopting no-reference image quality metrics to assess the visual fidelity of reconstructed views [54]–[56]; (ii) measuring multi-view geometric consistency within the reconstructed scene to infer potential inconsistency; and (iii) evaluating reconstruction quality through downstream tasks, where task performance serves as an indirect indicator of reconstruction accuracy. These motion-capture-free evaluation strategies could be used for assessing the quality of our reconstructed radiance field in real-world scenarios.

**Presence of Dynamic Objects.** Another limitation of the current work is the assumption that the observed scene is static, with motion blur arising solely from the rapid movement of the camera. This assumption is consistent with concurrent works [5], [40], [43] and with traditional Structure-from-Motion pipelines, where scene dynamics are not explicitly modeled. A promising direction for future work would be to relax this assumption by incorporating ideas from recent dynamic NeRF and dynamic 3DGS methods [21], [57], [58], enabling the system to distinguish between static and dynamic regions of the scene. Leveraging event-based motion cues together with on-board IMU measurements could further help disentangle object motion from camera ego-motion, improving reconstruction fidelity in dynamic environments.

## VI. CONCLUSION

We introduced a unified framework for radiance field reconstruction under fast motion, specifically tailored to the challenges of aerial robotics. By leveraging the complementary sensing of motion-blurred RGB images and asynchronous event data, our method enables accurate scene reconstruction without requiring ground-truth poses from e.g., a motion capture system, which is often unavailable in practice. Central to our approach is a continuous-time pose refinement module that improves initial VIO estimates, allowing both events and frames to jointly supervise the radiance field and the underlying motion.

We validated our framework on two newly collected real-world datasets, including onboard drone flights at high speed. Our results on synthetic and real-world datasets demonstrate that the proposed system can recover sharp radiance fields even under high-dynamic motion, where RGB frames are heavily degraded by motion blur and pose priors are unreliable.

Our work represents a significant step forward toward reliable, high-fidelity scene reconstruction for aerial robots operating under agile motion, opening the door to practical deployment of NeRF-like representations for robotic tasks such as infrastructure inspection, terrain exploration, and search-and-rescue, where rapid coverage and accurate spatial modeling are essential for operational success.

## VII. Acknowledgements

We sincerely thank Leonard Bauersfeld for his invaluable support throughout this research.

## References

[1] L. Bauersfeld and D. Scaramuzza, "Range, endurance, and optimal speed estimates for multicopters," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2953–2960, 2022.

[2] 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," in *ECCV*, 2020.

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

[4] Y. Qi, J. Li, Y. Zhao, Y. Zhang, and L. Zhu, "E $^{3}$ nerf: Efficient event-enhanced neural radiance fields from blurry images," *arXiv preprint arXiv:2408.01840*, 2024.

[5] 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.

[6] Y. Weng, Z. Shen, R. Chen, Q. Wang, and J. Wang, "Eadeblur-gs: Event assisted 3d deblur reconstruction with gaussian splatting," *arXiv preprint arXiv:2407.13520*, 2024.

[7] Z. Zhang, K. Chen, and L. Wang, "Elite-evgs: Learning event-based 3d gaussian splatting by distilling event-to-video priors," *arXiv preprint arXiv:2409.13392*, 2024.

[8] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019, pp. 6820–6829.

[9] L. Sun, C. Sakaridis, J. Liang, Q. Jiang, K. Yang, P. Sun, Y. Ye, K. Wang, and L. V. Gool, "Event-based fusion for motion deblurring with cross-modal attention," in *Lecture Notes in Computer Science*, Springer. Springer Nature Switzerland, 2022, pp. 412–428.

[10] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[11] Q. Ma, D. P. Paudel, A. Chhatkuli, and L. Van Gool, "Continuous pose for monocular cameras in neural implicit representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5291–5301.

[12] M. Cannici and D. 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*, 2024, pp. 9286–9296.

[13] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv preprint arXiv:2210.13641*, 2022.

[14] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu *et al.*, "Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping," *IEEE Robotics and Automation Letters*, 2024.

[15] T. Chen, O. Shorinwa, J. Bruno, A. Swann, J. Yu, W. Zeng, K. Nagami, P. Dames, and M. Schwager, "Splat-nav: Safe real-time robot navigation in gaussian splatting maps," *IEEE Transactions on Robotics*, 2025.

[16] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–15, Jul. 2022.

[17] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "TensoRF: Tensorial radiance fields," in *Lecture Notes in Computer Science*. Springer Nature Switzerland, 2022, pp. 333–350.

[18] A. Hanson, A. Tu, G. Lin, V. Singla, M. Zwicker, and T. Goldstein, "Speedy-splat: Fast 3d gaussian splatting with sparse pixels and sparse primitives," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 537–21 546.

[19] G. Fang and B. Wang, "Mini-splatting: Representing scenes with a constrained number of gaussians," in *European Conference on Computer Vision*. Springer, 2024, pp. 165–181.

[20] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2021, pp. 5865–5874.

[21] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe, and Z. Lv, "Neural 3d video synthesis from multi-view video," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2022, pp. 5521–5531.

[22] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "BARF: Bundle-adjusting neural radiance fields," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2021, pp. 5741–5751.

[23] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "Nope-nerf: Optimising neural radiance field with no pose prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4160–4169.

[24] Y. Fu, S. Liu, A. Kulkarni, J. Kautz, A. A. Efros, and X. Wang, "Colmap-free 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 796–20 805.

[25] 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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2022, pp. 12 861–12 870.

[26] C. Peng and R. Chellappa, "Pdrf: Progressively deblurring radiance field for fast and robust scene reconstruction from blurry images," *The 37th AAAI Conference on Artificial Intelligence*, 2023.

[27] D. Lee, M. Lee, C. Shin, and S. Lee, "DP-NeRF: Deblurred neural radiance field with physical scene priors," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2023, pp. 12 386–12 396.

[28] P. Wang, L. Zhao, R. Ma, and P. Liu, "Bad-nerf: Bundle adjusted deblur neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4170–4179.

[29] W. Chen and L. Liu, "Deblur-gs: 3d gaussian splatting from camera motion blurred images," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 7, no. 1, pp. 1–15, 2024.

[30] L. Zhao, P. Wang, and P. Liu, "Bad-gaussians: Bundle adjusted deblur gaussian splatting," in *European Conference on Computer Vision*. Springer, 2024, pp. 233–250.

[31] C. Peng, Y. Tang, Y. Zhou, N. Wang, X. Liu, D. Li, and R. Chellappa, "Bags: Blur agnostic gaussian splatting through multi-scale kernel modeling," in *European Conference on Computer Vision*. Springer, 2024, pp. 293–310.

[32] O. Seiskari, J. Ylilammi, V. Kaatrasalo, P. Rantalankila, M. Turkulainen, J. Kannala, E. Rahtu, and A. Solin, "Gaussian splatting on the move: Blur and rolling shutter compensation for natural camera motion," in *European Conference on Computer Vision*. Springer, 2024, pp. 160–177.

[33] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.

[34] S. Tulyakov, A. Bochicchio, D. Gehrig, S. Georgoulis, Y. Li, and D. Scaramuzza, "Time lens++: Event-based frame interpolation with parametric nonlinear flow and multi-scale fusion," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2022, pp. 17 755–17 764.

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

[36] V. Rudnev, M. Elgharib, C. Theobalt, and V. Golyanik, "Eventnerf: Neural radiance fields from a single colour event camera," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[37] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[38] 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.

[39] J. Wu, S. Zhu, C. Wang, and E. Y. Lam, "Ev-gs: Event-based gaussian splatting for efficient and accurate radiance field rendering," in *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2024, pp. 1–6.

[40] 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.

[41] J. Wang, J. He, Z. Zhang, M. Sun, J. Sun, and R. Xu, "Evggs: A collaborative learning framework for event-based generalizable gaussian splatting," *arXiv preprint arXiv:2405.14959*, 2024.

[42] S. Klenk, L. Koestler, D. Scaramuzza, and D. Cremers, "E-NeRF: Neural radiance fields from a moving event camera," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1587–1594, Mar. 2023.

[43] Y. Qi, L. Zhu, Y. Zhang, and J. Li, "E2nerf: 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.

[44] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2021, pp. 14 821–14 831.

[45] H. Son, J. Lee, J. Lee, S. Cho, and S. Lee, "Recurrent video deblurring with blur-invariant motion estimation and pixel volumes," *ACM Transactions on Graphics*, vol. 40, no. 5, pp. 1–18, Aug. 2021.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] H. Rebecq, D. Gehrig, and D. Scaramuzza, "Esim: an open event camera simulator," in *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, ser. Proceedings of Machine Learning Research, vol. 87. PMLR, 2018, pp. 969–982.

[48] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[49] S. He, Z. Osman, and P. Chaudhari, "From nerfs to gaussian splats, and back," *arXiv preprint arXiv:2405.09717*, 2024.

[50] S. Fang, I. Shen, T. Igarashi, Y. Wang, Z. Wang, Y. Yang, W. Ding, S. Zhou *et al.*, "Nerf is a valuable assistant for 3d gaussian splatting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 26 230–26 240.

[51] AlpsenTek. (2023) Alpsentek. hybrid vision® technology. Accessed: 2025-10-31. [Online]. Available: https://alpsentek.com

[52] M. Guo, S. Chen, Z. Gao, W. Yang, P. Bartkovjak, Q. Qin, X. Hu, D. Zhou, Q. Huang, M. Uchiyama *et al.*, "A three-wafer-stacked hybrid 15-mpixel cis+ 1-mpixel evs with 4.6-gevent/s readout, in-pixel tdc, and on-chip isp and esp function," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 11, pp. 2955–2964, 2023.

[53] K. Kodama, Y. Sato, Y. Yorikado, R. Berner, K. Mizoguchi, T. Miyazaki, M. Tsukamoto, Y. Matoba, H. Shinozaki, A. Niwa *et al.*, "1.22 $\mu$m 35.6 mpixel rgb hybrid event-based vision sensor with 4.88 $\mu$m-pitch event pixels and up to 10k event frame rate by adaptive control on event sparsity," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2023, pp. 92–94.

[54] Y. Liu, J. Wang, S. Cho, A. Finkelstein, and S. Rusinkiewicz, "A no-reference metric for evaluating the quality of motion deblurring." *ACM Trans. Graph.*, vol. 32, no. 6, pp. 175–1, 2013.

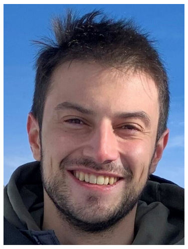
[55] Z. Wang and A. C. Bovik, "Reduced-and no-reference image quality assessment," *IEEE Signal Processing Magazine*, vol. 28, no. 6, pp. 29–40, 2011.

[56] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

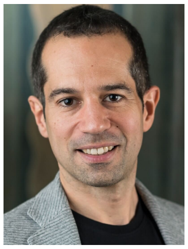
[57] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 20 310–20 320.

[58] J. Lee, C. Won, H. Jung, I. Bae, and H.-G. Jeon, "Fully explicit dynamic gaussian splatting," in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024, pp. 5384–5409.

**Rong Zou** received the M.Sc. degree in robotics, systems and control from ETH Zurich, Zurich, Switzerland, in 2024. He is currently pursuing the Ph.D. degree in robotic perception with the Robotics and Perception Group led by Prof. Davide Scaramuzza. He is also an Associated Doctoral Researcher with the ETH AI Center, Zurich. His research interests lie at the intersection of vision, learning and robotics, with a focus on event-based visual sensing and transferable representation learning for various robotic tasks such as motion estimation, scene understanding, and robust perception in challenging conditions.



**Marco Cannici** received the M.Sc. degree in 2018 and the Ph.D. degree in 2022 from Politecnico di Milano, Italy. From 2022 to 2025, he was a post-doctoral researcher with the Robotics and Perception Group at the University of Zurich, under the supervision of Prof. Davide Scaramuzza. His research focuses on event-based vision and perception for autonomous systems, spanning from visual odometry and monocular obstacle avoidance to high-speed and low-latency perception and 3D reconstruction in challenging conditions. He is currently a researcher at the Smart Eyewear Lab, a joint platform initiative between EssilorLuxottica and the Politecnico di Milano, where he develops event-based perception and sensing systems for smart glasses.



**Davide Scaramuzza** is a Professor of Robotics and Perception at the University of Zurich. He did his Ph.D. at ETH Zurich, a postdoc at the University of Pennsylvania, and was a visiting professor at Stanford University and NASA Jet Propulsion Laboratory. His research focuses on autonomous, agile navigation of mobile robots using standard and event-based cameras. He made fundamental contributions to visual-inertial state estimation, autonomous vision-based agile navigation of micro flying robots, and low-latency perception with event cameras, which were transferred to many products, from drones to automobiles, cameras, AR/VR headsets, and mobile devices. He pioneered autonomous, vision-based navigation of drones, which inspired the algorithm of the NASA Mars helicopter. In 2022, his team demonstrated that an AI-powered drone could outperform the world champions of drone racing. He received several awards, including an IEEE Technical Field Award, the IEEE Fellowship, the IEEE Robotics and Automation Society Early Career Award, a European Research Council Consolidator Grant, a Google Research Award, and many paper awards. In 2015, he co-founded Zurich-Eye, today Meta Zurich, which developed the head-tracking software of the Meta Quest. In 2020, he co-founded SUIND, which builds autonomous drones for precision agriculture. Many aspects of his research have been featured in the media, such as The New York Times, The Guardian, The Economist, and Forbes. He co-authored the book "Introduction to Autonomous Mobile Robots," published by MIT Press, which has sold over 10 thousand copies worldwide and is among the most used textbooks for teaching mobile robotics. He has been consulting the United Nations on disaster response, the Fukushima Action Plan, disarmament, and AI for good.