

Appendix A Real-world Quadrotor Experiment

To validate the effectiveness of Event3DGS in real-world robotic applications, we incorporate it into a custom-designed quadrotor platform. As illustrated in Fig 7(B), we employ an iPhone 13 Pro Max as the data collection device. The drone captures video at 240 FPS with a resolution of 1920×1080 , which is subsequently converted into an event stream via v2e[61]. We utilize COLMAP[33] to estimate the corresponding camera matrices. Our experimental setting is challenging and aggressive, involving extreme maneuvering conditions: the drone reaches a maximum horizontal acceleration of over 6 m/s^2 , a maximum roll angular velocity of 87 deg/s , and a maximum pitch angular velocity of 48 deg/s . Details of these maneuvers are available in our supplementary video.

Experimental results demonstrate that Event3DGS significantly improves both the qualitative and quantitative aspects of event-based 3D dense reconstruction. In Tab. 4, Event3DGS clearly surpasses the baseline across all evaluation metrics. In Fig. 7, Event3DGS accurately reconstructs the sharp geometric structure of the table and trees, whereas EventNeRF[9] cannot preserve those details.

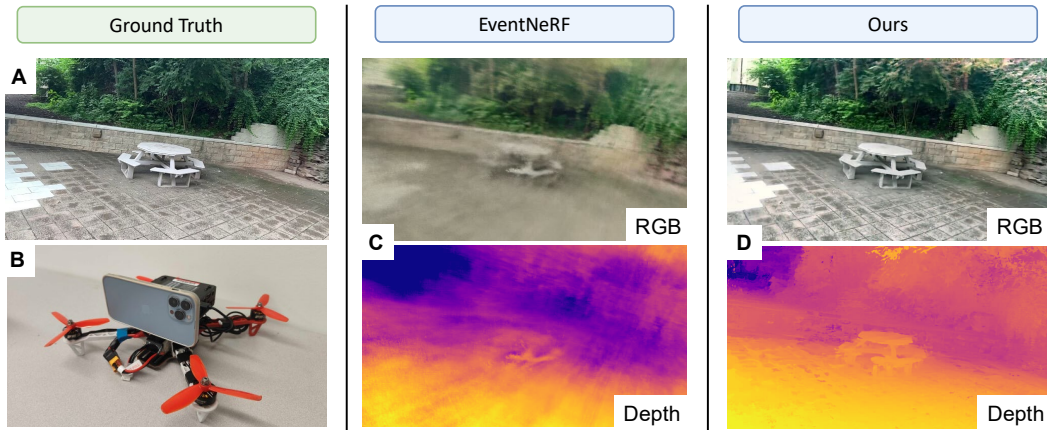


Figure 7: **A:** Ground truth RGB image. **B:** Demonstration of the custom-designed quadrotor. **C:** Rendered RGB and depth of Event3DGS. **D:** Rendered RGB and depth of EventNeRF[9].

Scene	EventNeRF			Event3DGS		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Quadrotor Flight	16.72	0.26	0.77	19.66	0.61	0.31

Table 4: Quantitative comparison on real-world quadrotor experiment. Due to more complex geometrical structures and larger scale, PSNR of the reported scenes is lower than PSNR of other real-world scenes. However, our method still outperforms EventNeRF[9] by a clear margin.

Appendix B Comparison with Deblurring Baselines

In this section, we compare Event3DGS with blur-aware 3DGS baselines: 1) 3DGS + Blur, i.e. vanilla 3D Gaussian Splatting[2] trained with motion-blurred RGB images; 2) DeblurGS[4], a novel method that reconstructs sharp 3D scenes from blurry images via estimating camera motions. We combine the consecutive frames within an event window of length 40 to be a blurry image, and generate 100 blurry training views for each scene. For fair comparison, we set all the hyper-parameters as default for baseline methods.

As DeblurGS[4] fails to reconstruct the 3D structure of synthetic scenes, we only report the visualization results in Fig. 8. Under high-speed rotations, 3DGS[2] is unable to accurately capture sharp details, and DeblurGS fails to estimate camera motions under severe motion blurs. In contrast, Event3DGS leverages high temporal resolution event data to accurately reconstruct the structure and appearance of the target scene. For real-world sequences, we report the numerical and visualization

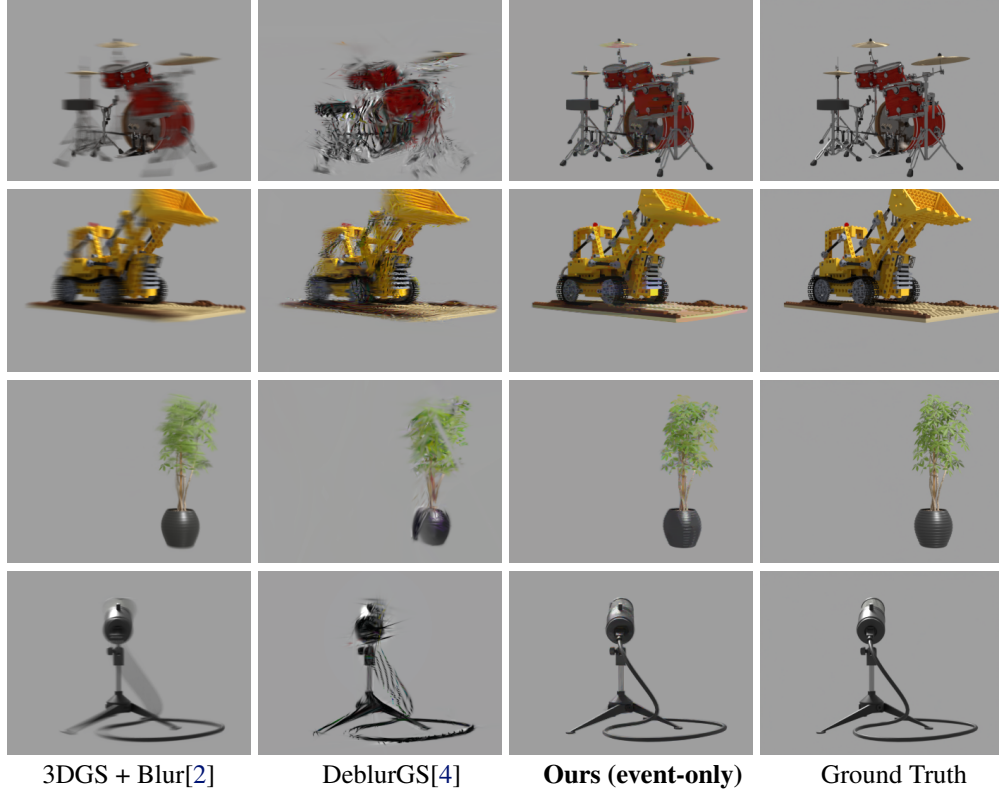


Figure 8: Qualitative comparison with deblurring baselines on synthetic dataset. We only report the scenes where rendering of DeblurGS[4] can align with the test views. Event3DGS demonstrates more accurate structural details and better multi-view consistency than baseline methods.

458 results in Tab. 5 and Fig. 9 respectively. Although DeblurGS roughly deblurs the input images and
 459 achieves higher reconstruction quality than the vanilla 3DGS, it fails to preserve multi-view con-
 460 sistency due to the existence of motion blur, causing under-representation in structural details (e.g.
 461 bicycle spokes, keyboard, edges of leaves, shoelaces in Fig. 9). As shown in Tab. 5, Event3DGS
 462 clearly outperforms baseline methods by an average of $+0.44dB$ higher PSNR, 19% higher SSIM
 463 and 33% lower LPIPS.

Scene	3DGS[9] + Blur			DeblurGS[4]			Event3DGS (event-only)		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Bike	21.0	0.42	0.62	23.90	0.54	0.42	23.06	0.71	0.26
Computer	20.75	0.64	0.42	24.58	0.80	0.13	24.11	0.87	0.08
Drum	23.79	0.68	0.41	25.48	0.76	0.18	24.8	0.83	0.15
Plant	17.05	0.34	0.57	19.28	0.52	0.28	22.53	0.8	0.13
Shoes	24.49	0.78	0.43	27.15	0.83	0.21	28.08	0.89	0.16
Average	21.42	0.57	0.49	24.08	0.69	0.24	24.52	0.82	0.16

Table 5: Quantitative comparison with deblurring baselines on real-world dataset. Due to the inherent radiance scale ambiguity of event data and the absence of direct color-wise supervision, Event3DGS does not achieve superior PSNR across all scenes. However, it demonstrates the highest structural and perceptual accuracy.

464 Notably, DeblurGS[8] requires an average of 3.5 hours for training on a synthetic scene due to the
 465 high computational cost of motion-blur formation and long training rounds. Event3DGS converges
 466 in just 18 minutes with the same hardware (a single NVIDIA RTX 6000Ada GPU), demonstrating
 467 significantly higher efficiency.

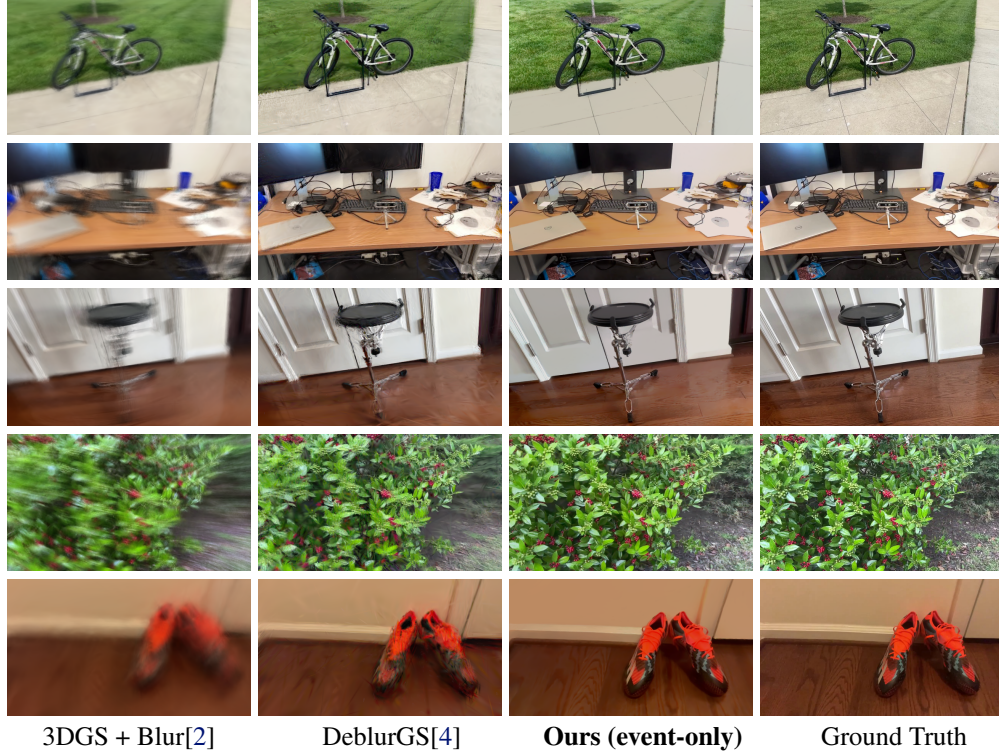


Figure 9: Qualitative comparison with deblurring baselines on real-world dataset. Event3DGS reconstructs sharpest details with least motion-blur effects across all scenes.

Appendix C Additional Implementation Details

Real-world Data Capture For each real-world scene, we first capture a video from a fast-moving RGB camera, then extract frames and use COLMAP[33] to estimate the corresponding camera extrinsics and intrinsics. We utilize v2e[61] with bayes filter [9] to simulate the colorful event stream.

Point-cloud Initialization Following [2], we start training from 100K uniformly random Gaussians inside a volumetric cube that bounds the scene. For synthetic and low-light sequences proposed in EventNeRF[9], we initialize the scale of points as $l = 0.2$; for our real-world sequences, we set $l = 10$ and move the points to the positive half-axis of z .

Appendix D Additional Low-light Visualization

For the low-light scenes proposed in [9], objects are placed on a spinning table rotating at a consistent speed of 45 RPM, then event sequences are captured with a DAVIS-346C color event camera under the illumination from a 5W light source. As ground-truth images are not provided in this dataset, we report additional visualization results in Fig. 10. With low-light real sequences, Event3DGS exhibits superior performance in accurately reconstructing sharp geometric details (e.g. edges of the objects) and removing noises on non-event background pixels.

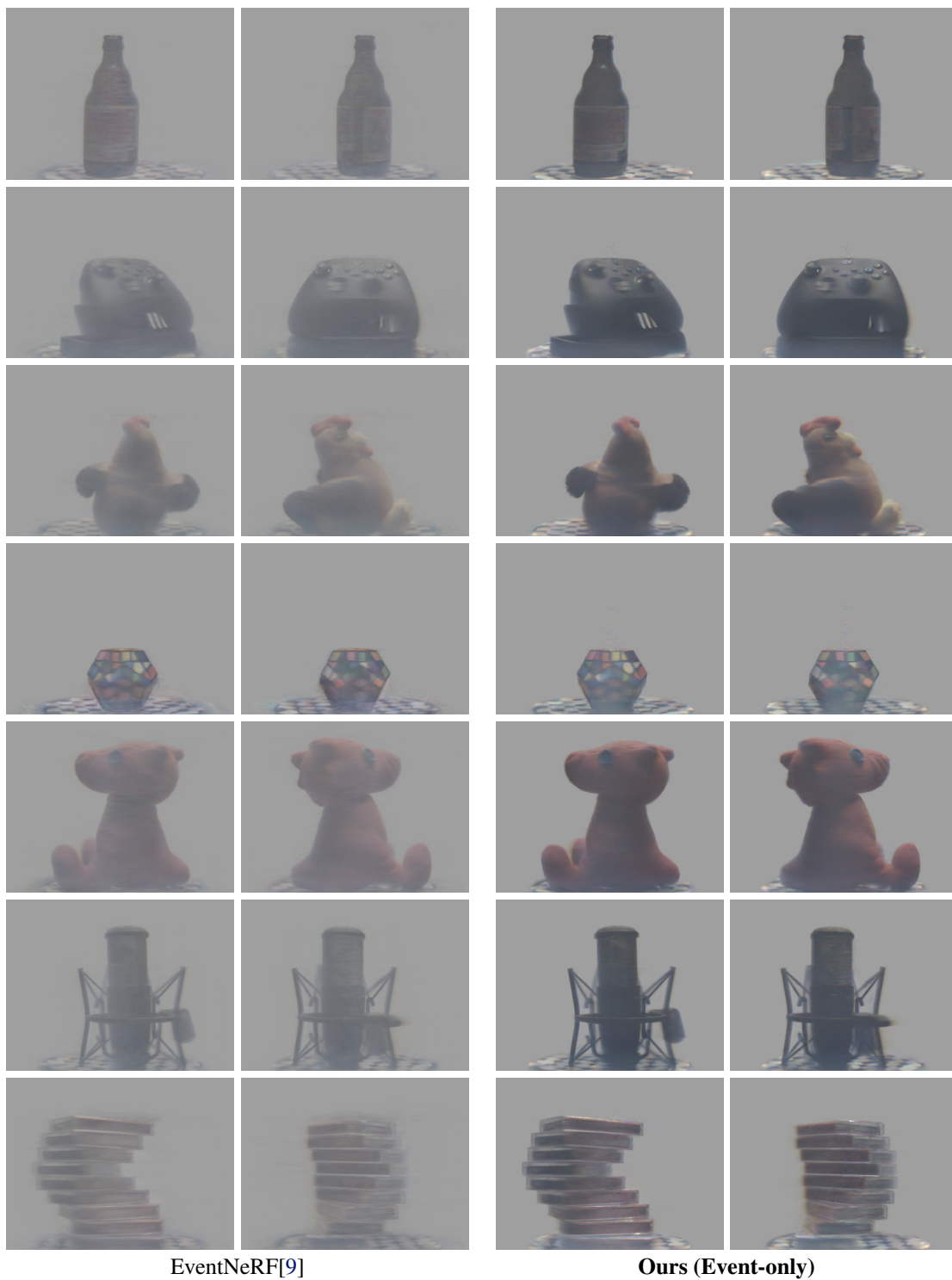


Figure 10: Visualization results on low-light scenes. We randomly select two rendered views for each scene. For EventNeRF[9], we directly render images from their official checkpoints.