Continuous Pose for Monocular Cameras in Neural Implicit Representation

Supplementary Material

In this document, we provide additional details of our method and implementations. We further provide qualitative examples corresponding to the main paper results. Please also refer to the supplementary video for additional qualitative visualizations. We have also attached an example code in a separate file.

6. PoseNet Coordinate Frames

In this section, we report the details of the PoseNet outputs concerning the coordinate frame under different applications. For the first two applications involving inaccurate pose and asynchronous events, we follow the work [29] and output refined pose from the noisy pose with respect to the ith camera frame. $T_{c'_i,w} = T_{init_i} \circ T_{refine_i}$, $T_{init_i} = T_{c_i,w} \circ T_{noise_i}$, while $T_{a,b}$ represents the rigid-body transformation matrix that transforms homogeneous points defined in frame b to the equivalent points in frame a. Note $c_i i$ refers to pose estimation of camera i and c_i denotes the ground truth. The target of PoseNet is to learn the cancellation of noise perturbations, essentially to serve as the inverse of T_{noise_i}. In the real experiment, we assume unknown initial pose so $T_{c'_i,w} = I \circ T_{refine_i}$ making the objective of PoseNet to directly estimate $T_{c_i,w}$.

In the RGB-D SLAM application in Table 5, we analyze the impact of varying reference coordinates on tracking. We denote the PoseNet output with respect to frames x. So the estimation of $i$the camera pose: $T_{w,c_i} = T_{w,c_{i-1}} \circ T_{c_{i-1},x} \circ P(f(\theta_p, t_i))$. The output with respect to different reference frames is shown in Table 9. Note we get the random frame by perturbing the pose of c_{i-1}.

For the experiments of IMU, PoseNet outputs the pose of the agent which is fixed as the IMU sensor. Then we transform the pose to camera frame with $T_{w,c_i} = T_{w,b_i} \circ T_{b_i,c_i}$ while $_{b_i,c_i}$ is constant and read from the sensor extrinsic.

7. NeRF from Inaccurate Pose

Implementation details. Compared to [29] we make the following modifications and extensions: (1) BARF perturbs the ground truth pose in synthetic datasets by independently sampling 6 dimensions Gaussian noise in $SE(3)$. We introduce time-dependent noise which is closer to the real-world scenario, for monocular cameras, where the error of pose estimation increases with time due to drift and error accumulation. Furthermore, we also separate the rotation and translation perturbation and instead of sampling noise across all frames we only sample a subset of frames and interpolate the poses for the rest. By doing so we can explicitly set the maximal deviation on translation or rotation. (2) Unlike BARF when we optimize one camera pose it also affects the surrounding poses, therefore a larger batch size is important for stable training. We use 4096 random rays for each iteration to optimize camera poses collectively.

For joint training with the radiance field, we use the Adam [23] optimizer for both translation and rotation networks with different learning rates. We use a smaller learning rate for rotation since quaternion rotation expression is highly nonlinear and difficult to train compared to translation [28]. We use 1e-3 for TransNet and 2e-4 for RotsNet and exponentially decay the schedule to 1e-5 and 1e-6 respectively for stable training.

Reference Frame	Transformation
Default	T_{c_{i-1},c_i}
World	T_{w,c_i}
Random	T_{r,c_i}
Intrinsic	T_{I,c_i}
IMU	T_{b_{i-1},b_i}

Table 9. **PoseNet on different reference coordinates.**

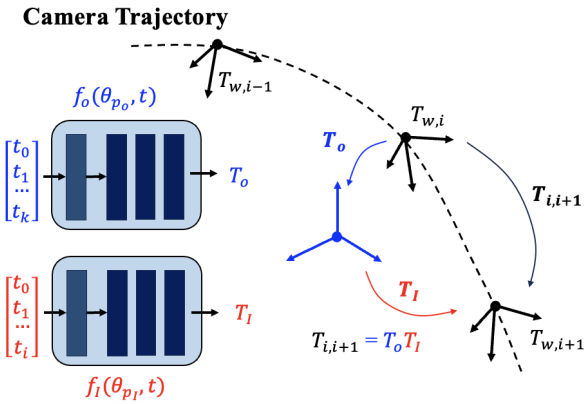


Figure 7. **Intrinsic Motion Frame.** We decompose the relative motion $T_{i,i+1}$ with a slowly changing rigid transform T_0 and a low dimensional frame-wise motion T_I using two separate PoseNets.

More results on the synthetic dataset. From Table 11 we can see our method is robust to large translation noise of up to 40% of the whole scene and is also robust to large rotation deviations of up to 90 degrees. BARF fails to register the camera frame under 20% translation and 60-degree rotation perturbation and although the 3D object is correctly reconstructed with largely correct poses, certain novel view synthesis yields bad PSNR as the object deviates from the image centre. This can be clearly seen in qualitative results

comparison in Figure 8.

More results of real dataset. More results on other real scenes as well as qualitative results can be found in Table 16 and Figure 9. Benefiting from neighboring temporal information our proposed pose representation performs consistently well on different speeds of camera motion. Similar to the above experiments we can find the novel view deviates from the image center in Fort/2(19) results. Furthermore, our method is robust to high-speed scenarios with slight artifacts while BARF diverges and provides very inaccurate results. Note that in the reported results we disable the test-time photometric optimization for better comparison of camera pose registration performance.

B-spline baseline experiments on 3D. Similar to 2D planar experiments we report also the results using classical continuous B-spline to enforce continuity between neighbouring poses. We experimented with various parameter configurations to illustrate the challenge of tuning classical methods in the context of neural radiance fields.

Ablation on network size. We report the performance evaluations with different network sizes. The reduction in network size affects camera localization performance. We use the 8-layer and 256 width model for other applications.

8. Continuous Pose for Asynchronous Events

Implementation details. [47] shows EventNeRF reconstruction quality cannot handle inaccurate camera poses over 1 °. The real sequences angle offset reported by EventNeRF however can reach up to 2.85 °. Following its noise perturbation method, we introduce different magnitudes of pose inaccuracies in the real datasets. Furthermore, we also consider the pose inaccuracies due to unknown asynchronous event poses. [47] provides in total 1000 ground truth poses from Blender which describes a circumferential movement. We uniformly sample different numbers of poses to linearly interpolate the whole circular path position and keep the orientation unchanged. Similarly to above, we use the Adam optimizer with an exponential learning rate schedule which decays from 2e-4 to 2e-6 for TransNet and 5e-5 to 5e-7 for RotsNet.

Parameter	Rotation error ↓	Translation error ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Regularize = 1e-3	26.815	14.5	8.87	0.62	0.60
Regularize = 1e-2	74.586	350.301	9.37	0.71	0.55
Regularize = 1e-1	115..65	581.81	4.51	0.39	0.73
Regularize = 1	94.81	284.61	9.85	0.70	0.56
knots = 75	50.779	199.2	8.46	0.61	0.60
knots = 50	3.009	9.523	14.46	0.69	0.21
knots = 25	3.01	9.53	14.46	0.69	0.21

Table 10. **Quantitative results of BARF with B-Spline.** We use scipy B-spline interpolation implementation splrep. On top part we use knots = 25 and for bottom part we use s = 1e-3.

TM	80 (10%)				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	0.06	0.254	27.72	0.92	0.04
Ours	0.03	0.196	27.91	0.92	0.04
TM	160 (20%)				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	24.76	57.342	9.79	0.61	0.52
Ours	0.05	0.292	26.74	0.91	0.06
TM	240 (30%)				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	19.77	95.631	6.97	0.50	0.73
Ours	0.03	0.178	28.44	0.93	0.04
TM	320 (40%)				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	18.66	127.3	7.39	0.53	0.71
Ours	0.03	0.200	28.25	0.93	0.04

(a) **Interpolated translational noise experiments.**

RM	30 °				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	0.067	0.265	27.75	0.92	0.05
Ours	0.049	0.105	28.22	0.93	0.04
RM	60 °				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	0.101	0.378	26.82	0.91	0.06
Ours	0.050	0.141	28.13	0.93	0.04
RM	90 °				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	12.103	37.380	10.40	0.61	0.42
Ours	0.061	0.181	28.03	0.93	0.04
RM	120 °				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	40.526	122.454	6.62	0.54	0.66
Ours	19.279	66.572	8.79	0.56	0.52

(b) **Interpolated rotational noise experiments.**

TM+RM $R\|t$	30 ° + 80(10%)				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	0.062	0.306	27.78	0.92	0.04
Ours	0.064	0.266	28.97	0.93	0.04
TM+RM $R\|t$	60 ° + 160(20%)				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	5.835	29.560	11.63	0.63	0.35
Ours	0.077	0.293	26.64	0.91	0.06
TM+RM $R\|t$	90 ° + 240(30%)				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	46.352	160.639	8.17	0.63	0.60
Ours	0.378	2.813	22.10	0.83	0.09
TM+RM $R\|t$t	120 ° + 320(40%)				
Method	RE	TE	PSNR	SSIM	LPIPS
BaRF[29]	55.640	195.134	7.7	0.63	0.63
Ours	16.122	52.527	9.16	0.56	0.51

(c) **Interpolated translational and rotation noise experiments.**

Table 11. **Interpolated pose noise experiments.** TM refers to Translational maximal deviation and RM refers to Rotational maximal deviation. The diameter of the circular trajectory is 800, the maximal deviation of the translation perturbation is set to be 10%, 20%, 30%, and 40%.

Method	Rotation error ↓	Translation error ↓	PSNR ↑	SSIM ↑	LPIPS ↓
8-layer, width 256	**0.07**	**0.28**	**27.30**	**0.92**	**0.06**
8-layer, width 128	0.09	0.31	27.33	0.90	0.09
4-layer, width 256	0.10	0.32	27.13	0.90	0.10
4-layer, width 128	0.11	0.33	27.15	0.91	0.11

Table 12. **Ablation study on network sizes**. The performance of camera localization drops only slightly with decreased network size. The experiments is conducted using our synthetic dataset, consistent with Table 3.

Qualitative results in interpolation error experiments. In Figure 10 we report the qualitative results of novel view synthesis on synthetic sequences of chair and hotdog which correspond to Table 4 of the main text. We can see EventNeRF suffers from strong fuzzy artifacts and the depth seems to dilate around the object while our method correctly learns the depth and reconstructs clearer 3D objects.

Qualitative results on the synthetic datasets in angle offset calibration experiments. In Figure 11 we report 3 more real dataset experiments on sequences of multimeter and plant. Similar to Figure 5 in the main text, EventNeRF suffers from trailing artifacts and at large angle offsets it nearly reconstructs 2 separate objects around the image center. In contrast, our method learns the offset angle and repositions the object back to the image center.

9. Visual SLAM with Depth and IMUs

Full IMU fusion. In the main text, we elaborate on harnessing gyroscope readings through both loose and tight coupling methods. However, direct utilization of accelerometer readings poses challenges as it provides acceleration instead of velocity in the body frame, resulting in a significant error when integrating with an unknown initial speed. Additionally, effective processing of acceleration data necessitates critical steps such as gravity removal and denoising [5, 11, 35, 42]. Therefore we first show the experiment with simulated IMU on the modified ScanNet dataset with simulated IMU as shown in Table 13. Given accelerator reading on time t, $\hat{\alpha}_t = (\hat{\alpha}_x, \hat{\alpha}_y, \hat{\alpha}_z)$. We first transform the reading to the last body frame with captured image, $\hat{\alpha}_{t_{i-1},t} = R_{t_{i-1},t} \circ \hat{\alpha}_t$. We calculate $R_{t_i,t}$ from loosely coupled method mentioned above. We then use auto-differentiation to calculate the second derivative of TransNet with respect to input time and supervise it with $\mathcal{L}1$ loss:

$$\mathcal{L}_{acc} = |\ddot{f}(\theta_p, t) - \hat{\alpha}_{t_{i-1},t}|; \quad (8)$$

Implementation details. In the NICE-SLAM experiments, we follow the original work [64] and the bundle adjustment is disabled. Learning rate for TransNet is set to 1e-3 and for RotsNet 2e-4. For IMU experiment we use

	With IMU				
	scan/059	scan/106	scan/181	scan/207	Average
Nice-SLAM [64]	37.28	174.27	71.94	80.00	89.75
Ours(Gyro)	14.51	12.78	43.98	18.23	22.36
Ours(Acceleration)	14.98	11.49	44.13	19.38	22.49
Ours(Combined)	**13.80**	**10.68**	**38.20**	**14.80**	**19.37**

Table 13. **Tracking performance on challenging Scannet [9].** Our PoseNet improves the tracking performance of NICE-SLAM significantly by fusing the IMU tightly. Using full IMU reading yields the best results over all experiment sequences.

Method	v101	v102	v103	v201	v202	v203	Avg
No IMU	2.17	N/A	5.82	7.76	5.04	N/A	N/A
Gyro	**1.98**	6.09	5.55	**4.99**	**3.03**	15.34	6.16
Accelerator	2.16	4.76	5.10	6.72	4.14	15.10	6.33
Combined	2.40	**5.33**	**3.63**	5.84	3.46	**13.63**	**5.71**

Table 14. **Tracking performance on EUROC [4].** Note that here we report only PoseNet based results. Utilizing both gyroscope and accelerometer data proves beneficial, particularly in challenging scenes, as compared to not using IMU.

Method	GFLOPs	Params[10^3]	Time-cost[s/it]
BARF	65.60	514	0.133
Ours	65.62	791	0.138

Method	Tracking time-Cost[ms/iter]	Convergence rate[iter]
NICE-SLAM	27.1	11.96
Ours	31.5	13.21

Table 15. **Left–Computation & Runtime.** Computation of a 1024 batch ray using RTX 3090, with the negligible inclusion of extra computation and time expense. **Right–Runtime & Convergence rate.** We follow the default setting of Replica.yaml. We assume convergence when the tracking loss remains unchanged.

$\lambda_{gyro} = 1$ and $\lambda_{acc} = 1$. For IMU simulation we interpolate the ground truth from 20 Hz to 200 Hz and calculate the numerical derivatives. We use the cubic interpolator for translation and $SLERP$ for rotation. We downsample the dataset from 20Hz to 5 Hz to highlight the importance of using IMU which is 100 Hz.

More experiments on RGB-D SLAM with IMU. As Table 13 shows, by fusing the acceleration and angular velocity we improve NICE-SLAM significantly and can maintain tracking to the end on challenging ScanNet. Taking advantage of both temporal information yields the best tracking performance. Qualitative results can be seen in Figure 12. We then use our method on EUROC [4]. We first use EKF-SLAM to denoise accelerator readings with sensor-fusion from gyroscope and Vicon Pose. As Table 14 demonstrates, fusing accelerator is beneficial especially under challenging scenes such as v103 and v203, and combining both sensor data yields the best results on average.

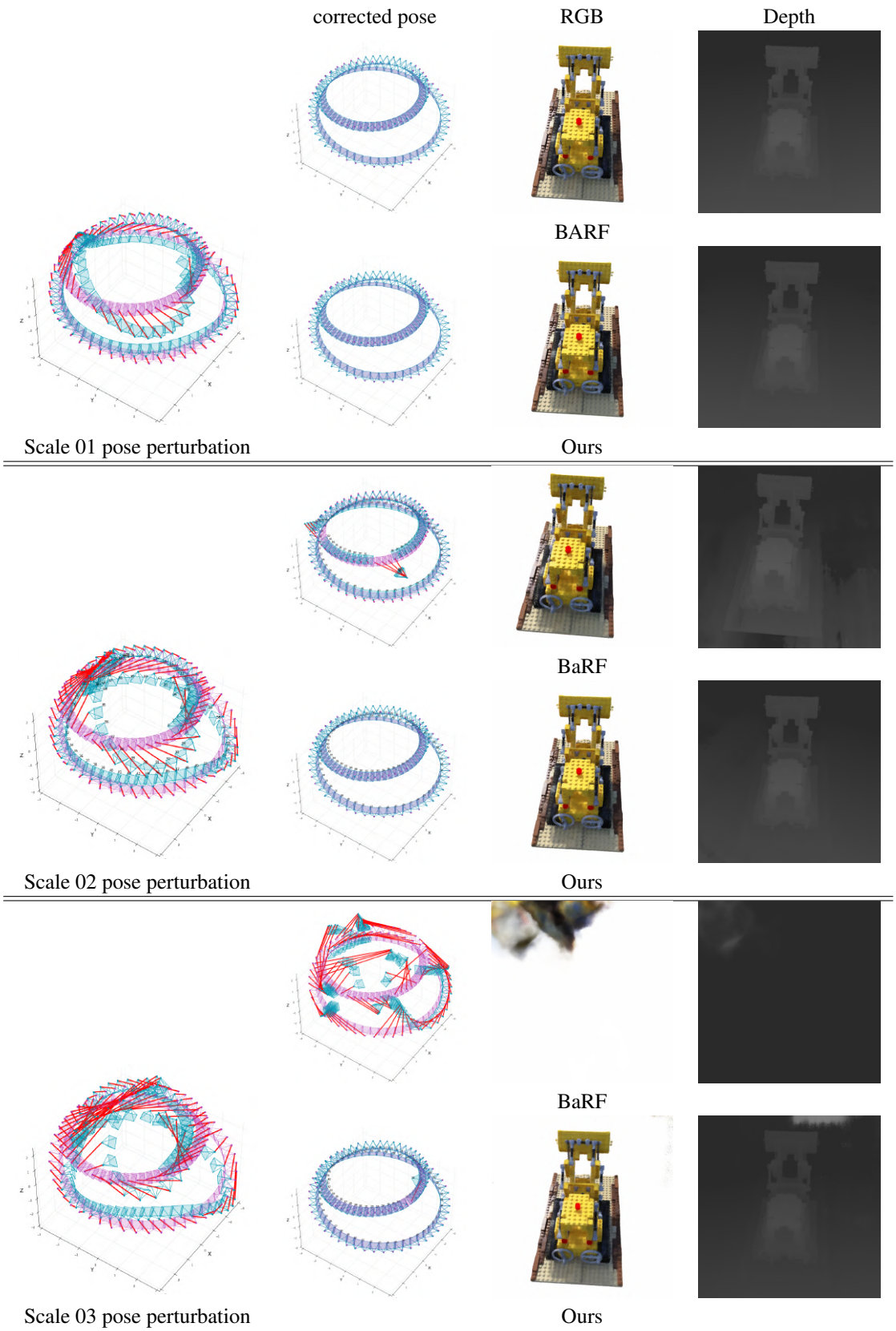
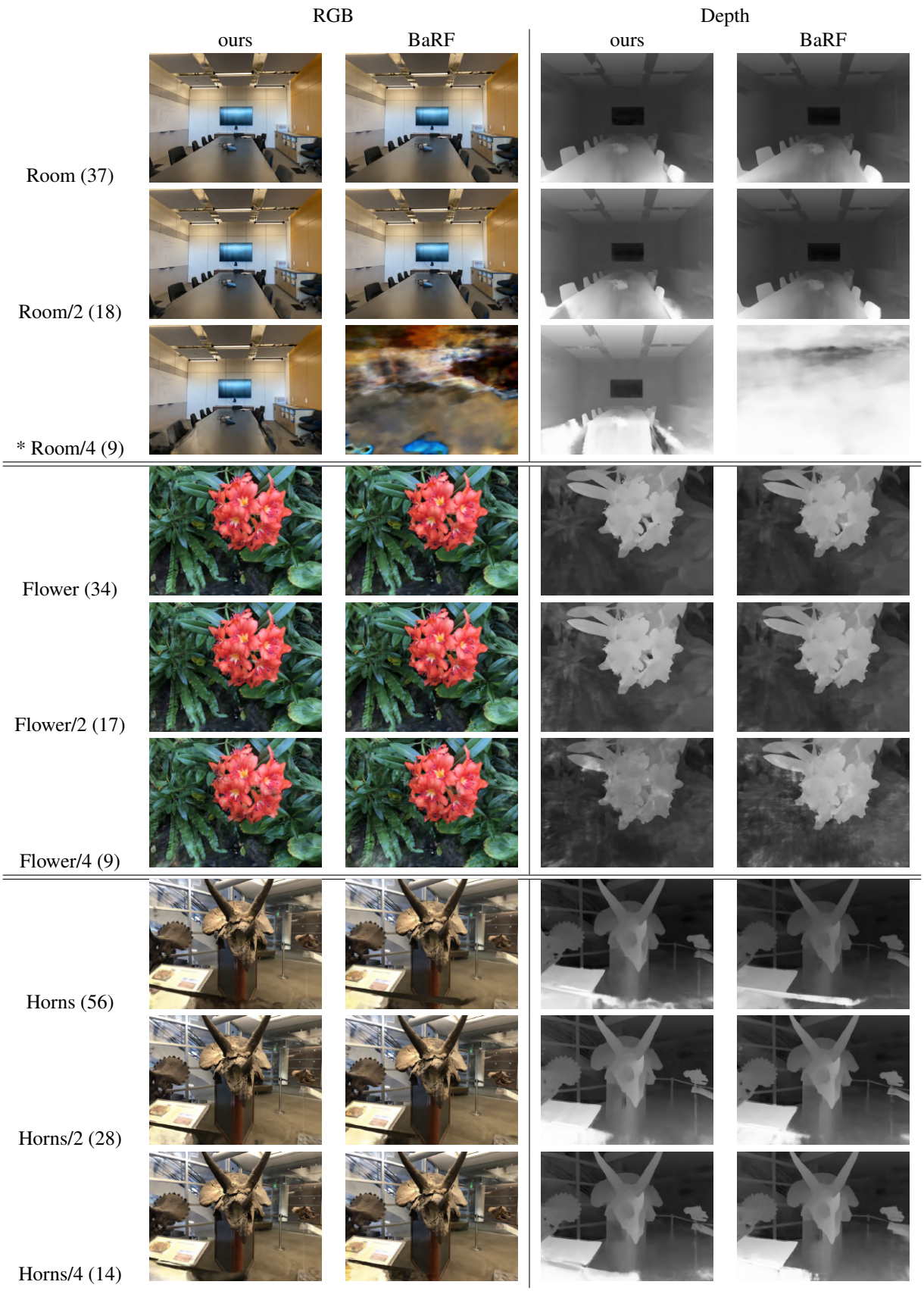


Figure 8. **Qualitative results of interpolated pose noise.** Our method can handle large pose noise and render images in the centre with the correct detphs.


RGB
Depth
ours
BaRF
ours
BaRF
Fern (18)
Fern/2 (9)
Fern/4 (5)
Fort (38)
* Fort/2 (19)
Fort/4 (10)
Orchids (23)
Orchids/2 (12)
* Orchids/4 (6)


RGB
Depth
ours
BaRF
ours
BaRF
Room (37)
Room/2 (18)
* Room/4 (9)
Flower (34)
Flower/2 (17)
Flower/4 (9)
Horns (56)
Horns/2 (28)
Horns/4 (14)

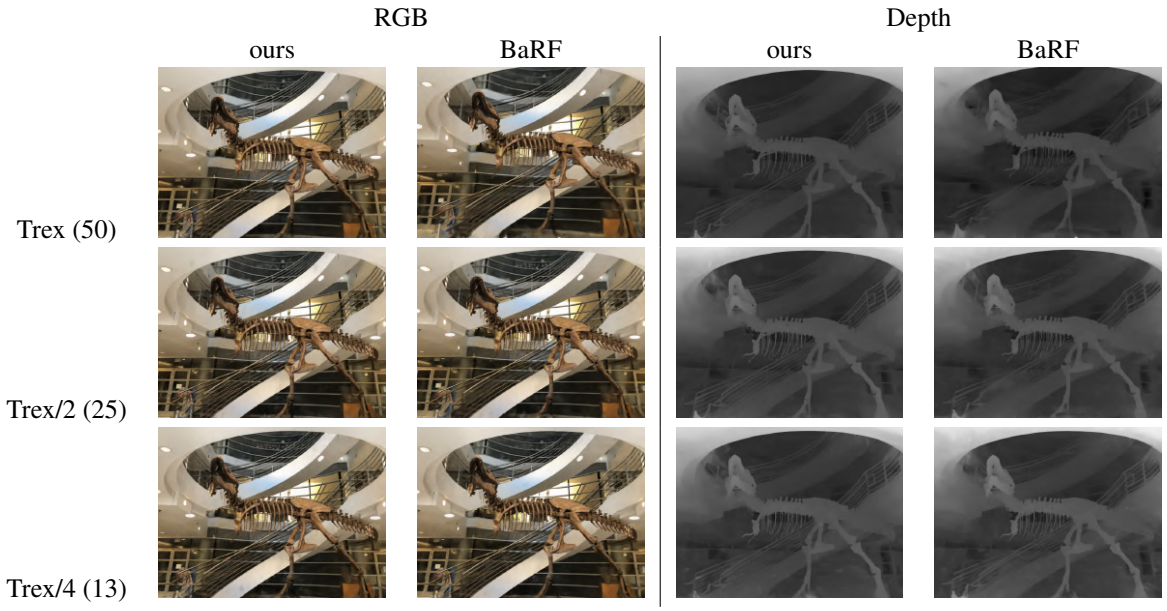
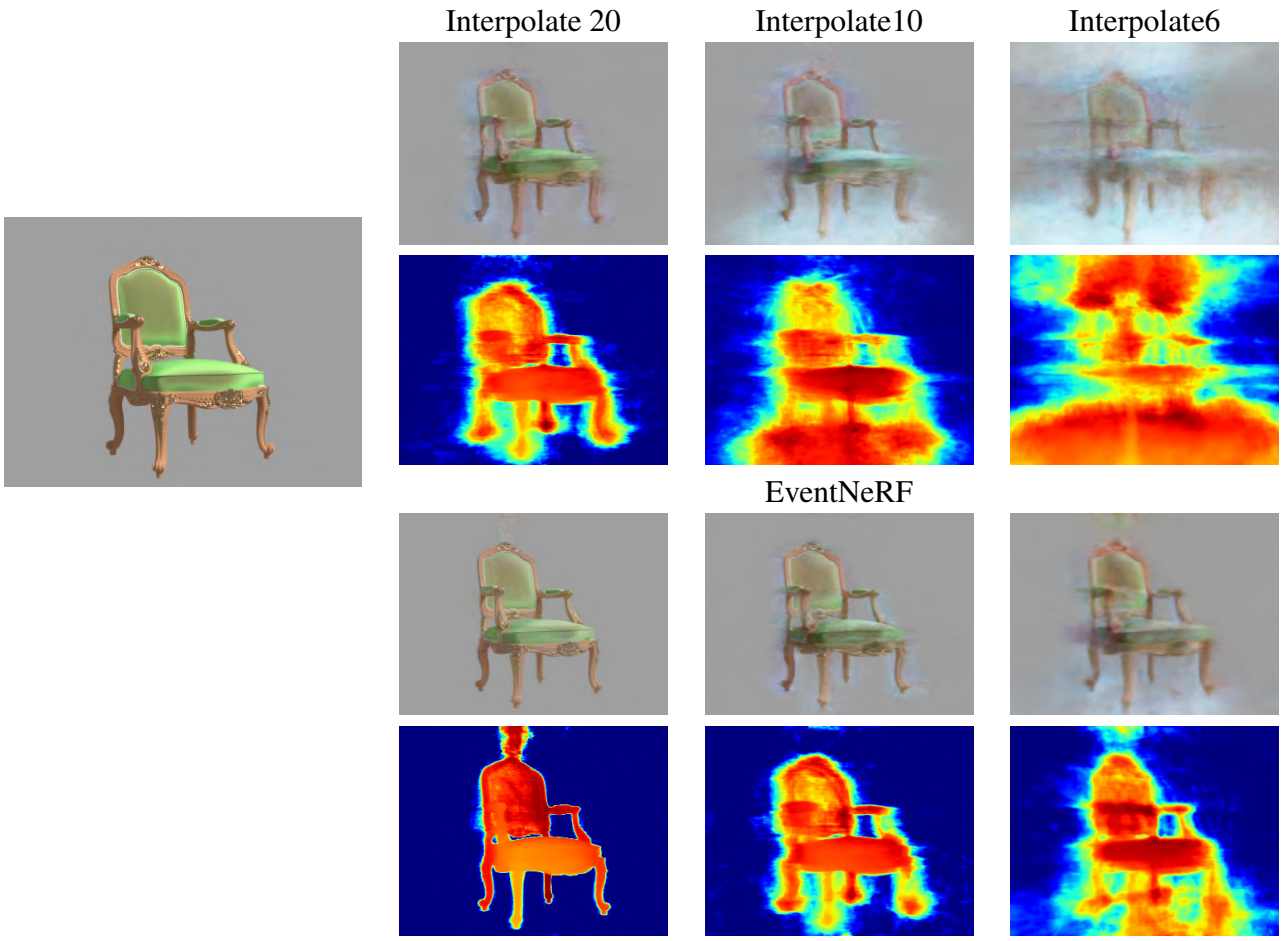
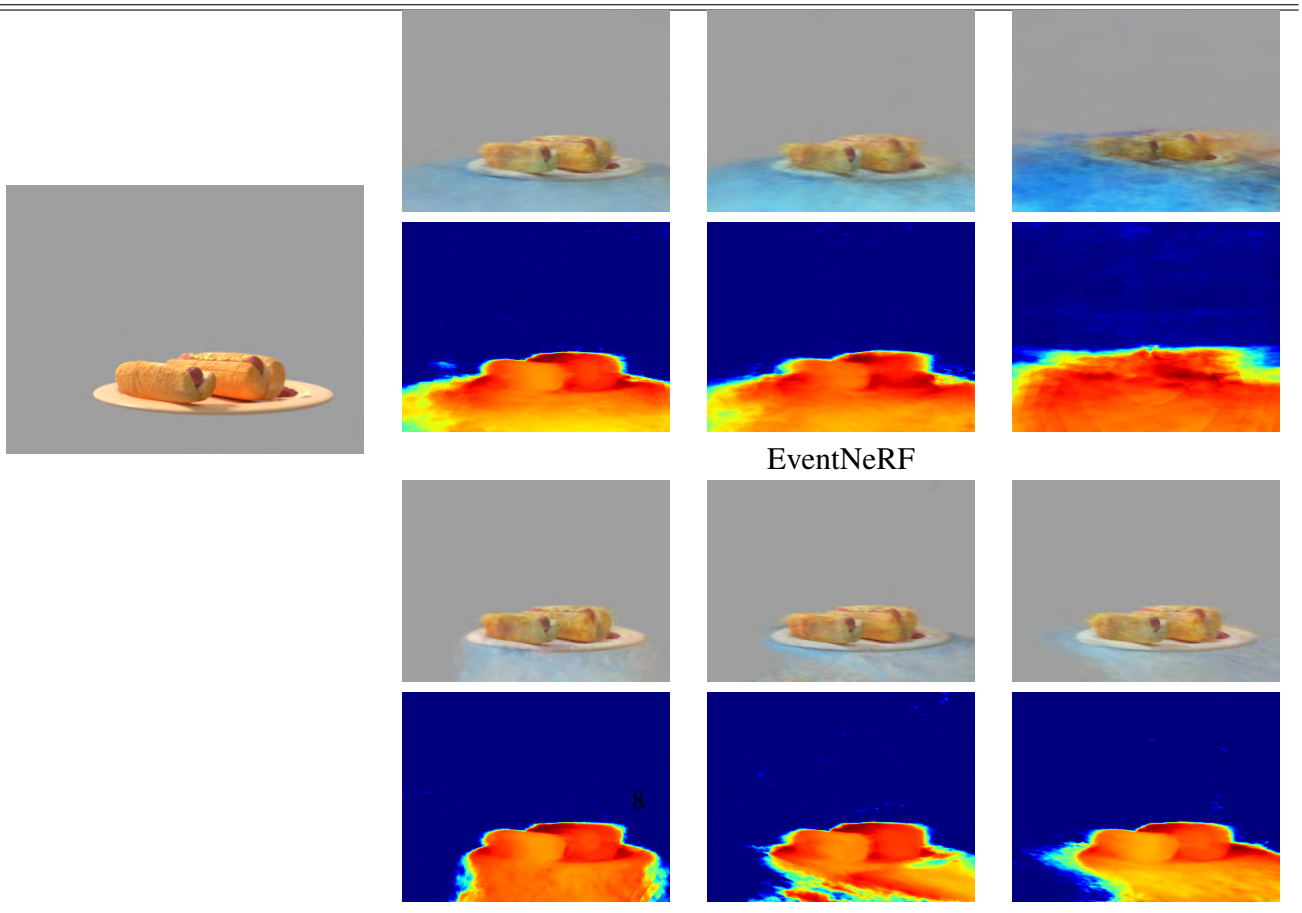


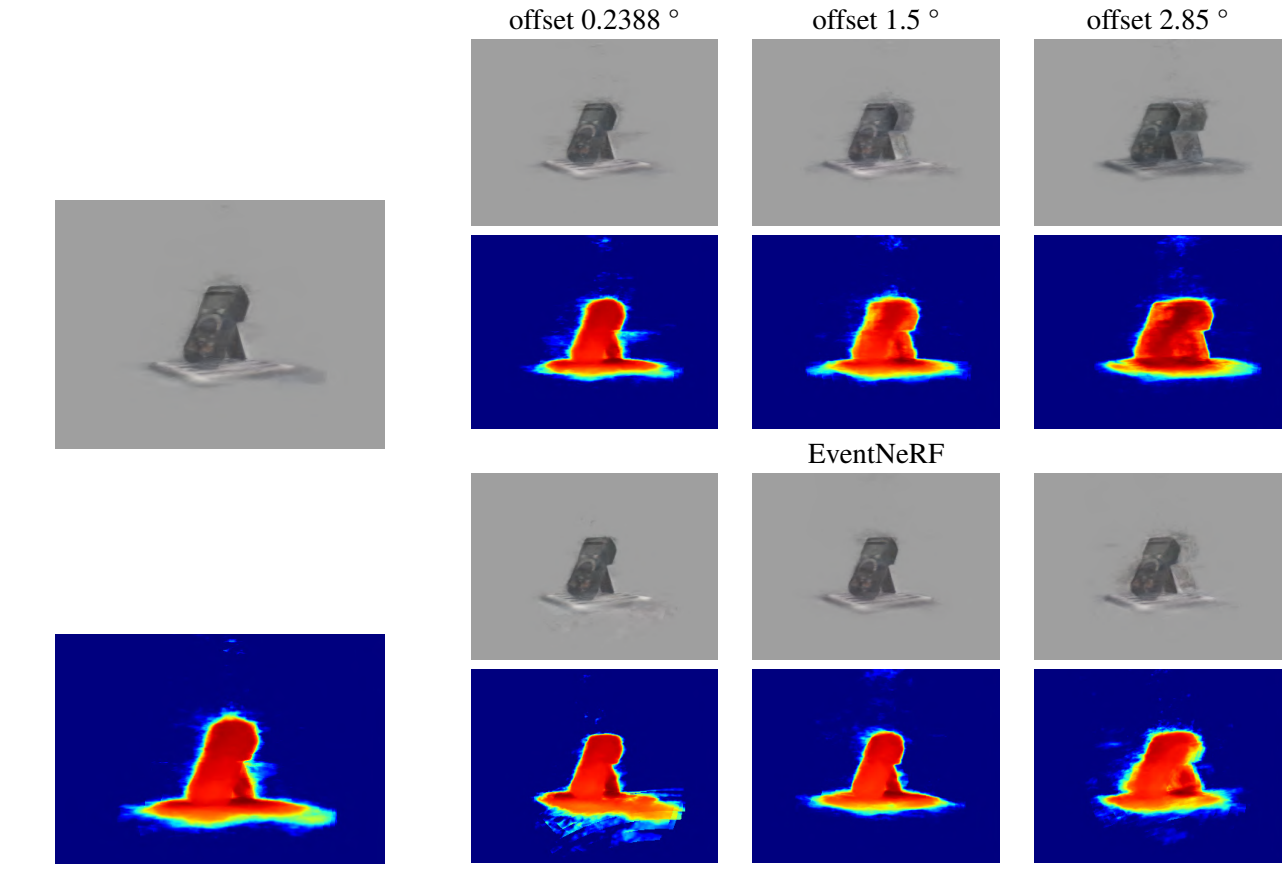
Figure 9. **Qualitative results of novel view synthesis real datasets [32] with unknown pose.** Corresponding to Table 2 in the main text we report the real dataset results in which the camera moves in a smooth trajectory. We denote BaRF failure cases with * and the number in the parentheses is the number of frames used in training.

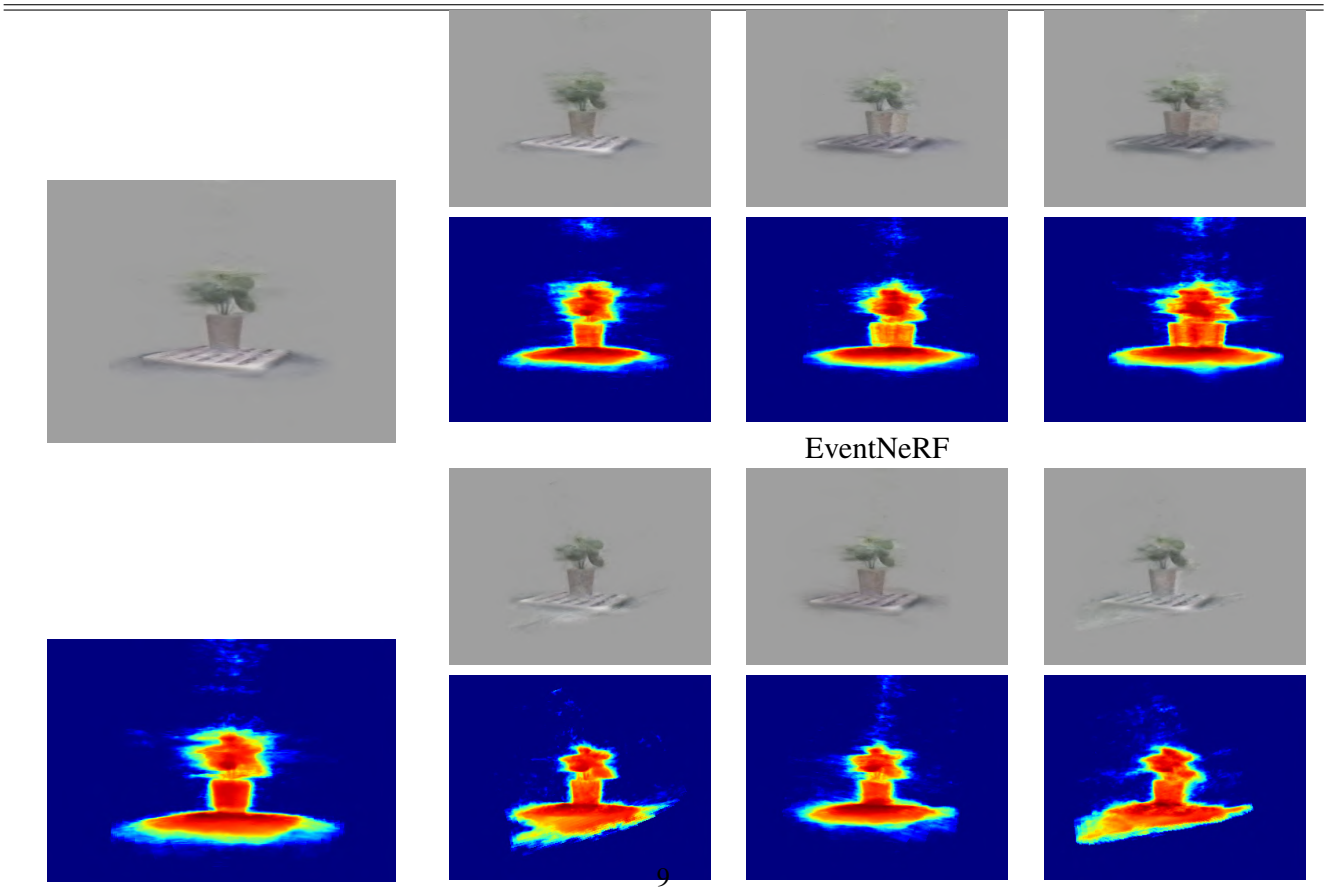
Scene	Rotation ↓		Translation ↓		PSNR ↑		SSIM ↑		LPIPS ↓	
	BARF	ours	BARF	ours	BARF	ours	BARF	ours	BARF	ours
Flower	0.64	0.49	0.27	0.25	17.18	17.93	0.34	0.41	0.27	0.21
Flower/2	0.62	0.46	0.28	0.25	17.18	17.94	0.34	0.36	0.27	0.23
Flower/4	0.59	0.54	0.31	0.29	17.07	17.38	0.33	0.38	0.29	0.27
Horns	0.18	0.19	0.18	0.18	19.58	18.89	0.59	0.55	0.32	0.27
Horns/2	0.27	0.33	0.20	0.17	16.24	16.09	0.49	0.45	0.31	0.28
Horns/4	0.21	0.24	0.16	0.17	16.91	16.85	0.54	0.53	0.32	0.32
Trex	0.49	0.41	0.38	0.35	16.53	17.04	0.42	0.45	0.21	0.19
Trex/2	0.56	0.26	0.43	0.29	16.37	18.96	0.40	0.61	0.23	0.16
Trex/4	0.19	0.20	0.24	0.26	21.62	20.74	0.73	0.70	0.17	0.15
Average	0.42	**0.34**	0.27	**0.24**	17.63	**17.95**	0.46	**0.49**	0.27	**0.23**

Table 16. **More quantitative results on real datasets**. In addition to Table 2 we report more results on LLFF [32] dataset. Note that in this dataset images are captured in a top-down, left-right manner rather than following a continuous trajectory. Consequently, our method may not be fully leveraged. Nevertheless, when considering average values, our approach outperforms the baseline.


Interpolate 20
Interpolate10
Interpolate6
EventNeRF
Chair Dataset ground truth
Ours


EventNeRF
Hotdog Dataset ground truth
Ours


offset 0.2388 °
offset 1.5 °
offset 2.85 °
EventNeRF
Multimeter Dataset (with calibration)
Ours


EventNeRF
Plant Dataset (with calibration)
Ours

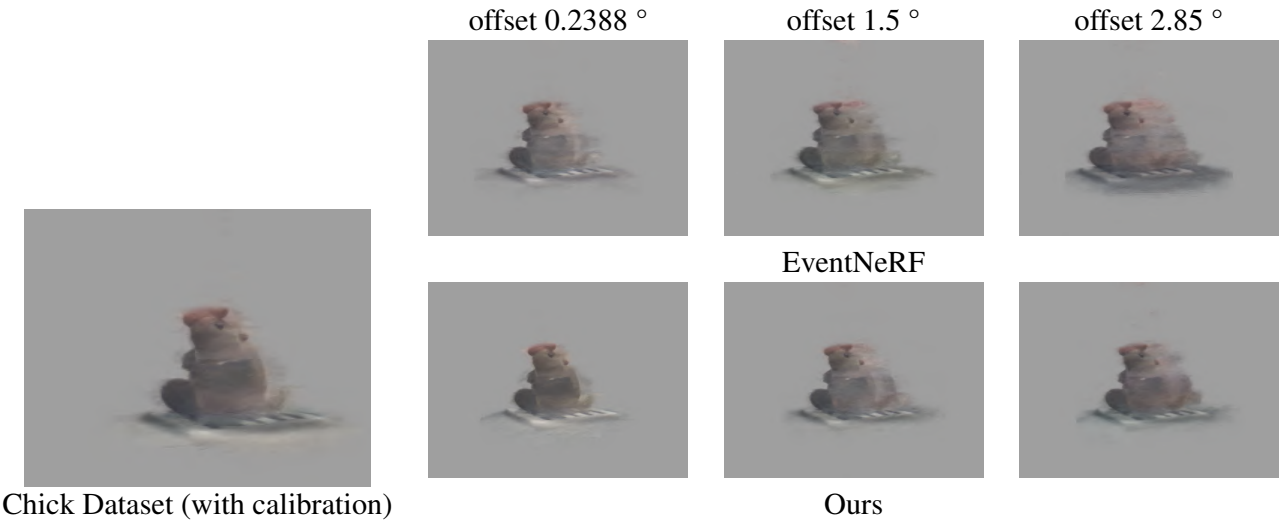
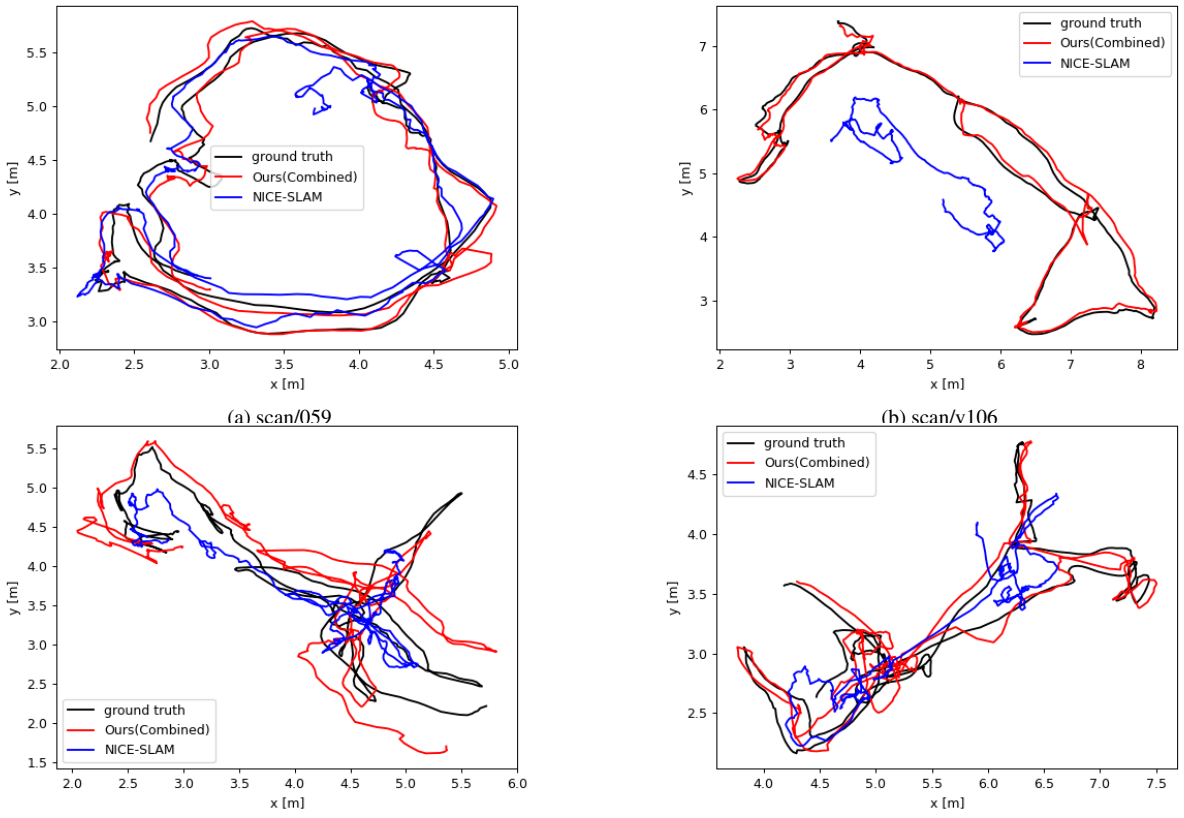


Figure 11. **Qualitative results of novel view depth and rgb synthesis in angle offset calibration experiments.** Our method improves EventNeRF significantly in all six experimental setups.

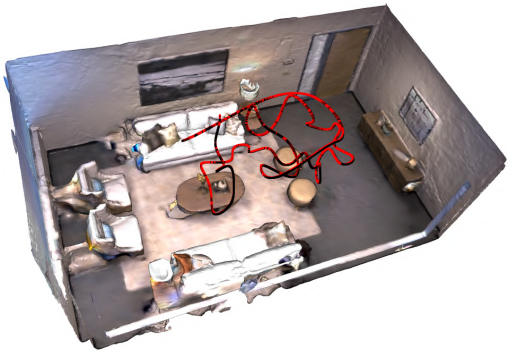


(a) scan/059 (b) scan/v106

(c) scan/181 (d) scan/207

Figure 12. **Qualitative results of tracking on challenging ScanNet** With the assistance of simulated IMU information, our method maintains robust tracking and preserves scale accuracy.

(a) NICE-SLAM

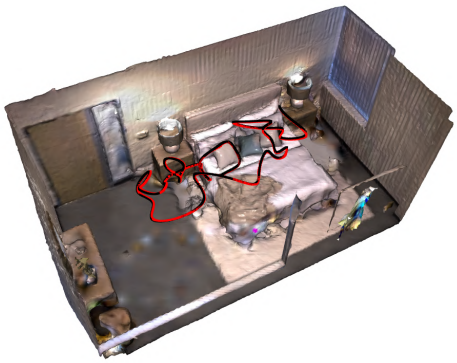


(b) Ours

(c) NICE-SLAM PSNR: 33.9

(d) Ours PSNR: 36.9

Figure 13. **Reconstruction and Rendering of Replica Room0.** Thanks to the improvement of tracking performance, our method is able to substantially increase the fidelity of the renderings. This is also supported by the quantitative results PSNR. We reconstruct clean details compared to NICE-SLAM.



(a) NICE-SLAM

(b) Ours

(c) NICE-SLAM PSNR: 32.7

(d) Ours PSNR: 33.3

Figure 14. **Reconstruction and Rendering of Replica Room1.** In this relatively easier scene, we perform slightly better than NICE-SLAM in rendering and reconstruction with less artifacts.

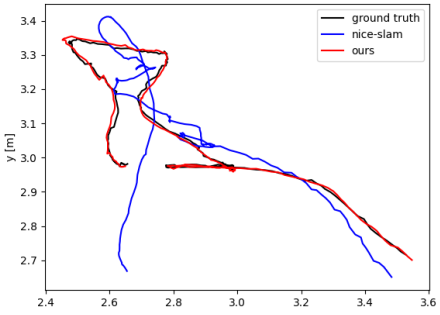
(a) NICE-SLAM

(b) Ours

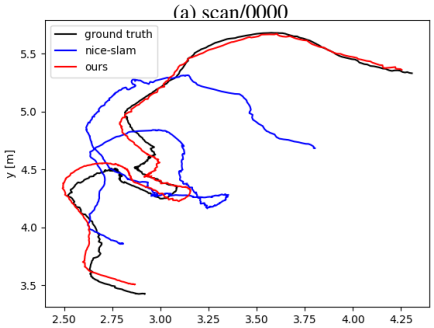
(c) NICE-SLAM PSNR: 33.3

(d) Ours PSNR: 36.8

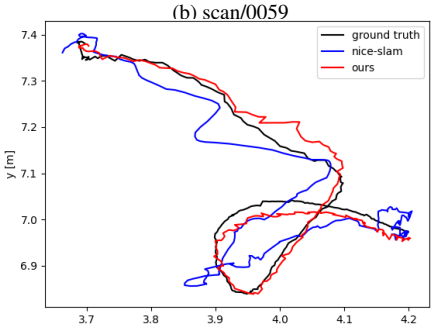
Figure 15. **Reconstruction and Rendering of Replica Room2.** While the reconstruction demonstrates that the NICE-SLAM trajectory is highly aligned with the ground truth, it adversely affects rendering performance, resulting in lower fidelity. In contrast, our method maintains high-fidelity rendering.



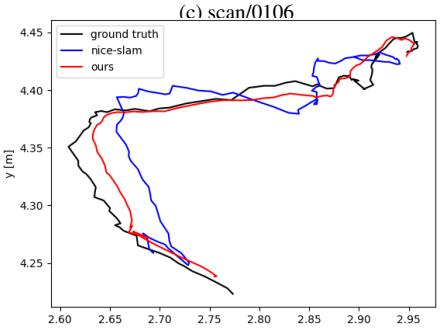
(a) scan/0000



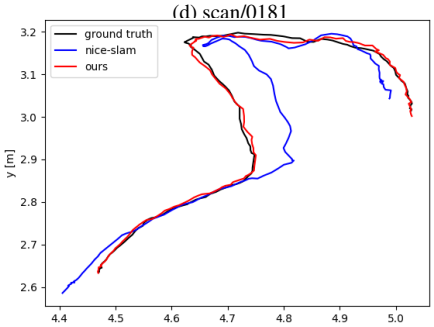
(b) scan/0059



(c) scan/0106

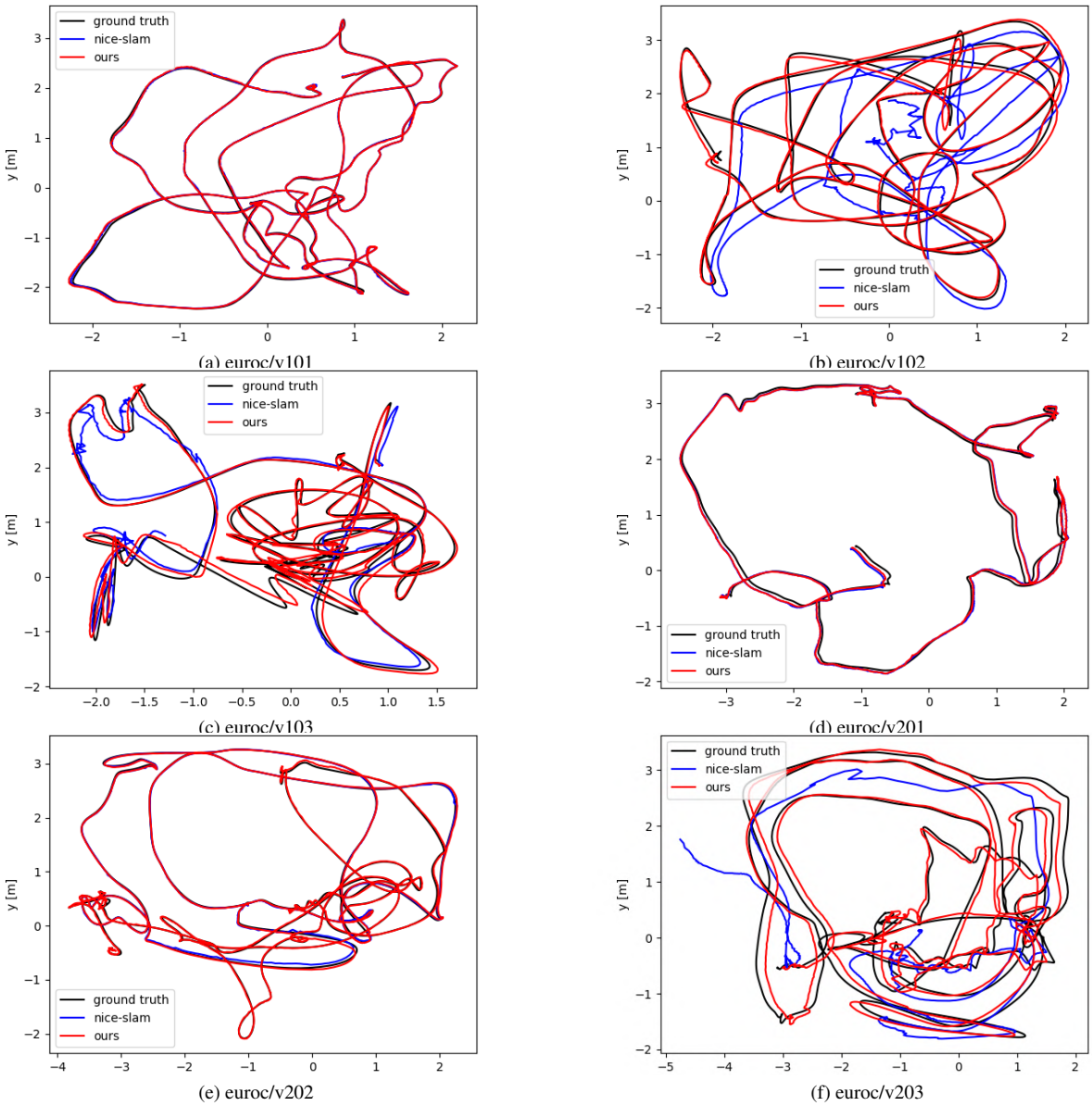


(d) scan/0181



(e) scan/0207

Figure 16. **Qualitative results of tracking on ScanNet[9].**The initial trajectories diverge in the NICE-SLAM trajectory from the ground truth, while ours align with it.



(a) euroc/v101 (b) euroc/v102

(c) euroc/v103 (d) euroc/v201

(e) euroc/v202 (f) euroc/v203

Figure 17. **Qualitative results of tracking on EUROC[4].** We compare the trajectories of our method to NICE-SLAM. Notably, NICE-SLAM encounters failures at v102, v202, and v203, so only part of trajectories are displayed. The results indicate that our method significantly aligns with the ground truth trajectory.