Learning Scene Coordinate Reconstruction from Unposed Images via Pose Graph Optimization – Supplementary Material

Tze Ho Elden Tse[1,2], Jizong Peng[1], Angela Yao[2]
[1]dConstruct Robotics [2]National University of Singapore

In this supplemental document, we provide:

- extended implementation details and discussion (Sec A);
- additional analysis (Sec B);
- additional qualitative examples (Sec C).

A. Extended implementation details and discussion

Overall algorithm. We present the complete pseudocode algorithm in Algorithm 1. Our key contribution is to suppress incorrectly refined camera poses, enable a more accurate and globally consistent set of poses. This improved alignment allows the scene coordinate regression network to be trained on a more reliable pose set in subsequent iterations. Consequently, the overall reconstruction process converges faster by reaching accurate camera poses earlier, supported by pose graph optimization.

Impact of graph sparsity or connectivity on optimization quality. Pose graph connectivity plays a critical role in the success of global optimization. A sparse graph, where only a few reliable edges exist, limits the propagation of global constraints and can lead to fragmented or weakly consistent pose estimates. Conversely, an overly dense graph may include noisy or incorrect edges, which can degrade optimization quality by introducing conflicting constraints. We observed that:

- **Sparse Graphs:** When the geometric consistency threshold τ is set too low, many valid correspondences are rejected, resulting in fewer edges. This reduces the optimizer's ability to enforce global consistency, especially in large-scale scenes with repetitive structures.
- **Dense Graphs:** A high τ admits more matches, increasing connectivity but also introducing outliers. These noisy edges can bias the optimization, even when robust loss functions are applied.
- **Balanced Connectivity:** Our experiments (see Fig. 5(b) in the main paper) show that $\tau = 0.1$ m achieves the best trade-off between accuracy and connectivity. This setting provides sufficient edges for global consistency while minimizing the inclusion of unreliable matches.

Can the uncertainty-aware metrics provide supervision signal to model training? Our current framework uses uncertainty-aware metrics exclusively for weighting constraints during pose graph optimization. These metrics—epipolar error and optical flow consistency—quantify geometric reliability between image pairs without requiring ground truth poses. This raises an interesting question: can these metrics also serve as supervision signals for training the ACE model? We summarize the potential benefits:

- **Self-Supervised Learning:** Epipolar and flow-based consistency could act as additional loss terms, encouraging the ACE model to predict scene coordinates that satisfy multi-view geometric constraints.
- **Improved Robustness:** Incorporating these metrics during training may reduce reliance on pseudo-labels and improve performance in ambiguous or textureless regions.
- **Dynamic Weighting:** Confidence scores could guide curriculum learning, where reliable correspondences are emphasized early in training.

One major challenge is non-differentiability: epipolar distance and optical flow errors involve discrete matching and RANSAC steps, making direct backpropagation difficult. Additionally, these metrics depend on estimated poses, which may be inaccurate during early iterations, potentially introducing bias into the training process. Finally, computing optical flow and epipolar constraints for all image pairs during training would significantly increase computational overhead, making the approach less practical for large-scale datasets.

Future work could explore differentiable approximations of epipolar and flow consistency, enabling their integration into end-to-end training. Another promising direction is to incorporate confidence-based weighting into ACE's reprojection loss, allowing uncertainty-aware metrics to influence optimization directly. Finally, curriculum strategies could be investigated, where uncertainty-aware metrics gradually gain importance as pose estimates become more reliable, balancing computational cost and training stability.

Algorithm 1 Overall algorithm

Input: Unordered image set $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$

Output: Refined camera poses $\{T_i\}$ and trained scene coordinate regression network Φ_θ

Initialization:

Select seed images and predict depth maps using ZoeDepth.

Generate initial pseudo-labels for scene coordinates.

Iterative Reconstruction Loop:

for *each iteration* t **do**

(1) Model training

Train scene coordinate regression network Φ_θ on current set of images with pseudo-poses.

(2) Pose estimation with previously trained model

Use trained scene coordinate regression network Φ_θ to estimate poses for all images.

Update pseudo-poses for unregistered images.

(3) Scene coordinate prediction

Predict dense scene coordinates $X_i = \Phi_\theta(I_i)$ for each image.

(4) Pose Graph Construction

for *each image pair* (I_i, I_j) **do**

Find geometrically consistent matches: project X_i into I_j, compute $e_{\text{match}} = \|X_j - X_i\|$, accept if $e_{\text{match}} < \tau$.

Estimate relative pose Z_{ij} using PnP + RANSAC.

Add edge (i, j) with Z_{ij} to pose graph.

end

(5) Uncertainty estimation

Compute epipolar error e_{epi} and optical flow error e_{flow}.

Confidence score $\sigma = \alpha_1 e_{\text{epi}} + \alpha_2 e_{\text{flow}}$.

Aggregate σ to edge-level uncertainty σ_{ij} and compute $\Omega_{ij} = \text{diag}(1/(\sigma_{ij}^2 + \epsilon))$.

(6) Pose Graph Optimization

Solve $\min_{\{T_i\}} \sum_{(i,j)\in E} \|\text{Log}(Z_{ij}^{-1} T_i^{-1} T_j)\|^2_{\Omega_{ij}}$ using LM optimizer with Huber loss.

(7) Check stopping criteria

Stop if 99% images registered or $< 1\%$ new.

end

B. Additional analysis

Details of the teaser example. For the teaser figure, we use the the example video *office loop* from [3] which consists of 473 frames capturing movement around an office environment. We provide the full qualitative comparison in Fig. 6 with sample image frames in Fig. 7. As illustrated in Fig. 7, ACE-Zero exhibits suboptimal performance, achieving PSNR values of 19.1 and 15.6 with and without COLMAP initialization, respectively. This highlights that the vanilla ACE-Zero configuration can incorrectly refine camera poses and fails to recover, even with additional iterations and accurate initialization from COLMAP. Furthermore, VGGT-SLAM, despite correctly identifying loop closures, does not recover sufficiently accurate camera poses, yielding a PSNR of 17.1. In contrast, our method achieves a PSNR of 25.6, which is comparable to COLMAP's PSNR of 28.1. In addition, we provide qualitative comparisons in Fig. 8.

	Frames	Pseudo Ground Truth		All Frames			
		COLMAP SSIM (↑)	COLMAP LPIPS (↓)	ACE0 SSIM (↑)	ACE0 LPIPS (↓)	Ours SSIM (↑)	Ours LPIPS (↓)
Chess	6k	0.86	0.17	0.84	0.19	0.85	0.18
Fire	4k	0.73	0.27	0.68	0.30	0.73	0.26
Heads	2k	0.82	0.26	0.80	0.28	0.84	0.25
Office	10k	0.85	0.25	0.82	0.27	0.86	0.24
Pumpkin	6k	0.84	0.23	0.83	0.24	0.87	0.22
RedKitchen	12k	0.80	0.25	0.77	0.29	0.78	0.27
Stairs	3k	0.59	0.50	0.62	0.45	0.76	0.29
Average		0.78	0.28	0.76	0.29	0.81	0.24

Table 4. Quantitative comparisons on 7-Scenes dataset. We report the pose accuracy via view synthesis as **SSIM** (↑) and **LPIPS** (↓). Dark green indicate the best results. We report results for COLMAP (default).

	SSIM (↑)				LPIPS (↓)			
	COLMAP (default)	VGGT-SLAM [3]	ACE0 [2]	Ours (full)	COLMAP (default)	VGGT-SLAM [3]	ACE0 [2]	Ours (full)
Bicycle	0.68	0.15	0.46	0.70	0.20	0.71	0.36	0.18
Bonsai	0.89	0.37	0.84	0.85	0.06	0.59	0.10	0.08
Counter	0.82	0.22	0.77	0.80	0.10	0.70	0.13	0.12
Garden	0.86	0.27	0.77	0.87	0.08	0.51	0.12	0.07
Kitchen	0.87	0.29	0.80	0.84	0.06	0.77	0.10	0.08
Room	0.91	0.30	0.58	0.68	0.05	0.73	0.43	0.22
Stump	0.40	0.15	0.20	0.65	0.45	0.61	0.54	0.29
Average	0.78	0.25	0.63	0.77	0.14	0.66	0.25	0.15

Table 5. Quantitative comparisons on Mip-NeRF 360 dataset. We report the pose accuracy via view synthesis as **SSIM** (↑) and **LPIPS** (↓). Dark green indicate the best results. We report results for COLMAP (default).

Full experimental results. We present both SSIM and LPIPS scores for 7-Scenes and Mip-NeRF 360 datasets in Table 4 and 5, respectively. We also present SSIM and LPIPS scores for Tanks and Temples dataset in Table 6 and 7, respectively. We observe that both SSIM and LPIPS scores behave similarly to the PSNR results presented in the main script.

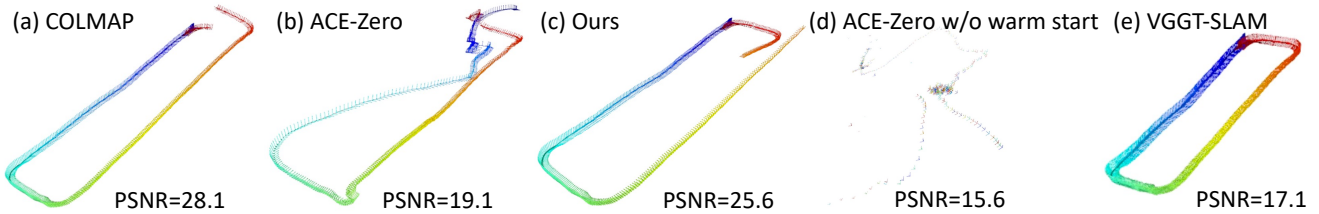

Figure 6. Despite being initialized from (a) COLMAP generated camera poses, local refinements in (b) often introduce global inconsistencies. In contrast, (c) our hybrid-framework, which integrates pose graph optimization into ACE-Zero, enforces multi-view consistency and suppress incorrect refinements. Furthermore, we demonstrate that (d) ACE-Zero fails without COLMAP initialization, and (e) the low PSNR indicates that VGGT-SLAM lacks accuracy despite its ability to perform loop closure.

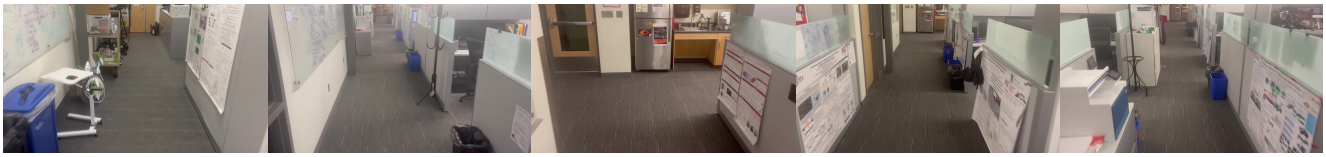

Figure 7. We provide several sample image frames from the *office loop* video that were used for the teaser example.

		Frames	COLMAP (default)	VGGT-SLAM [3]	ACE0 [2]	Ours (full)	Frames	COLMAP (fast)	DROID-SLAM [4]	ACE0 [2]	Ours (full)
Training	Barn	410	0.77	0.50	0.54	0.68	12.2k	0.77	0.49	0.60	0.68
	Caterpillar	383	0.63	0.34	0.56	0.62	11.4k	0.72	0.62	0.67	0.67
	Church	507	0.64	0.36	0.57	0.57	19.3k	0.29	0.34	0.56	0.63
	Ignatius	264	0.69	0.32	0.65	0.65	7.8k	0.77	0.61	0.74	0.62
	Meetingroom	371	0.67	0.41	0.55	0.59	11.1k	0.67	0.55	0.62	0.67
	Truck	251	0.76	0.44	0.71	0.73	7.5k	0.83	0.69	0.79	0.80
	Average	364	0.69	0.40	0.60	0.64	14.6k	0.68	0.55	0.66	0.68
	Avg Time		1h	5min	1.1h	30min		74h	18min	2.2h	1h
Intermediate	Family	152	0.77	0.53	0.69	0.72	4.4k	0.83	0.67	0.69	0.71
	Francis	302	0.84	0.62	0.75	0.78	7.8k	0.77	0.80	0.75	0.81
	Horse	151	0.77	0.51	0.73	0.71	6.0k	0.79	0.69	0.73	0.78
	Lighthouse	309	0.65	0.47	0.62	0.69	8.3k	0.70	0.68	0.62	0.67
	Playground	307	0.54	0.33	0.48	0.52	7.7k	0.53	0.32	0.48	0.63
	Train	301	0.62	0.30	0.52	0.59	12.6k	0.73	0.51	0.52	0.62
	Average	254	0.70	0.46	0.63	0.66	7.8k	0.73	0.61	0.63	0.70
	Avg Time		32min	5min	1.3h	30min		48h	14min	2.2h	1.3h
Advanced	Auditorium	302	0.74	0.54	0.63	0.69	13.6k	0.50	0.58	0.63	0.66
	Ballroom	324	0.46	0.24	0.49	0.65	10.8k	0.48	0.25	0.49	0.57
	Courtroom	301	0.61	0.36	0.51	0.57	12.6k	0.43	0.32	0.51	0.59
	Palace	509	0.54	0.45	0.24	0.49	21.9k	0.50	0.38	0.24	0.50
	Temple	302	0.68	0.45	0.25	0.53	17.5k	0.44	0.45	0.25	0.63
	Average	348	0.61	0.41	0.42	0.59	15.6k	0.47	0.39	0.42	0.59
	Avg Time		1h	5min	1h	30min		71h	27min	2.8h	1.5h

Table 6. Quantitative comparisons on Tanks and Temples dataset. We report the pose accuracy via view synthesis as **SSIM** (↑). Dark green indicate the best results, **excluding** COLMAP.

		Frames	COLMAP (default)	VGGT-SLAM [3]	ACE0 [2]	Ours (full)	Frames	COLMAP (fast)	DROID-SLAM [4]	ACE0 [2]	Ours (full)
Training	Barn	410	0.21	0.57	0.51	0.34	12.2k	0.19	0.63	0.39	0.30
	Caterpillar	383	0.29	0.61	0.34	0.32	11.4k	0.18	0.26	0.23	0.23
	Church	507	0.31	0.73	0.39	0.40	19.3k	0.79	0.94	0.38	0.36
	Ignatius	264	0.21	0.54	0.25	0.25	7.8k	0.15	0.25	0.17	0.20
	Meetingroom	371	0.33	0.71	0.48	0.42	11.1k	0.31	0.50	0.35	0.31
	Truck	251	0.18	0.52	0.22	0.20	7.5k	0.11	0.24	0.14	0.15
	Average	364	0.26	0.61	0.37	0.32	14.6k	0.29	0.47	0.28	0.26
	Avg Time		1h	5min	1.1h	30min		74h	18min	2.2h	1h
Intermediate	Family	152	0.17	0.36	0.24	0.22	4.4k	0.12	0.23	0.24	0.22
	Francis	302	0.15	0.43	0.25	0.20	7.8k	0.24	0.17	0.25	0.23
	Horse	151	0.21	0.49	0.24	0.25	6.0k	0.17	0.27	0.24	0.24
	Lighthouse	309	0.36	0.67	0.38	0.32	8.3k	0.29	0.32	0.38	0.33
	Playground	307	0.45	0.68	0.53	0.46	7.7k	0.45	0.82	0.53	0.33
	Train	301	0.29	0.70	0.39	0.32	12.6k	0.20	0.45	0.39	0.29
	Average	254	0.27	0.55	0.34	0.30	7.8k	0.24	0.38	0.34	0.27
	Avg Time		32min	5min	1.3h	30min		48h	14min	2.2h	1.3h
Advanced	Auditorium	302	0.31	0.69	0.50	0.39	13.6k	0.75	0.64	0.50	0.34
	Ballroom	324	0.43	0.68	0.40	0.37	10.8k	0.44	0.92	0.40	0.33
	Courtroom	301	0.34	0.67	0.46	0.41	12.6k	0.61	0.79	0.46	0.34
	Palace	509	0.56	0.67	0.83	0.67	21.9k	0.60	0.84	0.83	0.62
	Temple	302	0.30	0.62	0.86	0.51	17.5k	0.65	0.70	0.86	0.44
	Average	348	0.39	0.56	0.51	0.47	15.6k	0.61	0.78	0.51	0.41
	Avg Time		1h	5min	1h	30min		71h	27min	2.8h	1.5h

Table 7. Quantitative comparisons on Tanks and Temples dataset. We report the pose accuracy via view synthesis as **LPIPS** (↓). Dark green indicate the best results, **excluding** COLMAP.

Reolcalization performance on 7-Scenes (Table 8). We compare with the supervised relocalizer ACE [1] by training with both KinectFusion and COLMAP mapping poses. As pointed out in [2], ACE achieves almost perfect relocalization rate under the 5cm, 5° error threshold as the evaluation is performed against COLMAP query poses. We show that our approach can further improve the relocalization performance by providing a more globally aligned poses.

Uncertainty metrics weight sensitivity (Table 9). We evaluate the impact of varying the weighting parameters α_1 and α_2 in Eq. (9), which combine epipolar and optical flow errors into the confidence score. Our best-performing configuration is $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$, achieving **21.7 dB** PSNR on the 7-Scenes dataset and **24.3 dB** PSNR on the Mip-NeRF 360 dataset. Table 9 reports performance for different weight ratios. We observe that moderate variations in α (e.g., 0.3:0.7 or 0.5:0.5) have minimal impact on performance, indicating robustness to weight changes. However, extreme imbalance significantly degrades accuracy. When epipolar weighting dominates (e.g., $\alpha_1 \geq 0.7$), performance drops due to sensitivity to pose initialization

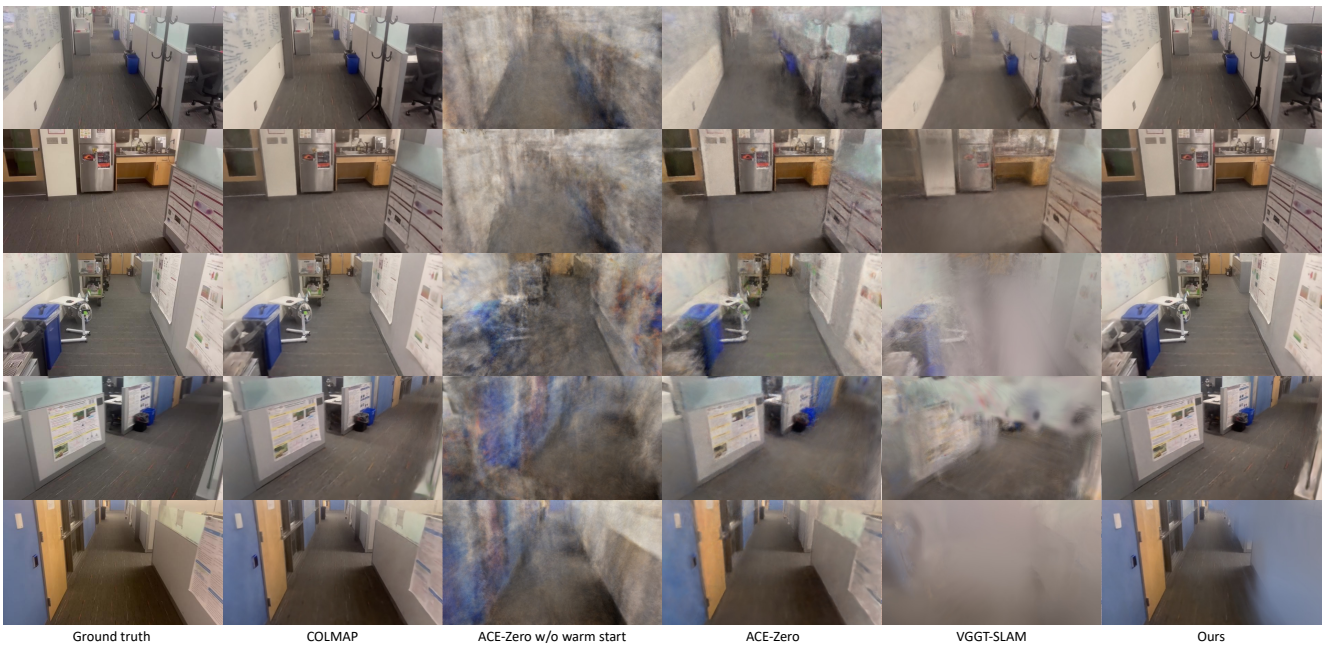

Figure 8. Qualitative comparison from the *office loop* video that were used for the teaser example. Although ACE-Zero, even when initialized with COLMAP-generated camera poses, tends to exhibit artifacts and blurriness in synthesized novel views, our proposed method consistently delivers high-fidelity renderings. This improvement reflects a more globally consistent and structurally coherent scene reconstruction.

Supervision	ACE [1] KinextFusion	ACE [1] COLMAP	ACE0[2] self-supervised	Ours self-supervised
Chess	96.0%	100.0%	100.0%	100.0%
Fire	98.4%	99.5%	98.8%	99.0%
Heads	100.0%	100.0%	100.0%	100.0%
Office	36.9%	100.0%	99.1%	99.1%
Pumpkin	47.3%	100.0%	99.9%	100.0%
Redkitchen	47.8%	98.9%	98.1%	98.5%
Stairs	74.1%	85.0%	61.0%	75.1%
Average	71.5%	97.6%	93.8%	96.0%

Table 8. Relocalization performance comparisons on 7-Scenes dataset. We follow [2] and report percentage below 5cm, 5° error which are computed with respect to COLMAP pseudo ground truth.

errors and lack of local consistency. Conversely, optical flow only configurations ($\alpha_2 = 1.0$) improve robustness in indoor scenes but fail to enforce global consistency, leading to drift in large-scale or mixed environments. The best trade-off is achieved with $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$, which balances global and local geometric cues effectively across both datasets. This suggests that optical flow provides strong local evidence for texture-rich indoor scenes, while epipolar constraints remain essential for maintaining global structure in complex environments.

$\alpha_1 : \alpha_2$	7-Scenes	Mip-NeRF 360
0.4 : 0.6 (Best)	**21.7**	**24.3**
0.5 : 0.5	21.5	24.1
0.3 : 0.7	21.6	24.2
0.6 : 0.4	21.3	23.8
0.7 : 0.3	21.0	23.5
1.0 : 0.0	20.5	22.8
0.0 : 1.0	21.2	23.6

Table 9. Ablations on different α ratios on 7-Scenes and Mip-NeRF 360 datasets. We report the pose accuracy via view synthesis as PSNR in dB.

Full runtime breakdown. For a 200 images frames dataset, our additional pose graph optimization step requires approximately 30 seconds per reconstruction iteration, compared to 150 seconds for vanilla ACE-Zero. Of this 30-second overhead, the actual optimization accounts for only 0.016 seconds, while the remaining time is spent on graph construction and uncertainty metric computation.

The scalability of this step depends primarily on graph construction rather than optimization. The optimization itself is highly efficient, as it operates on a sparse graph and optimizes only camera poses, resulting in near-linear complexity with respect to the number of nodes and edges. In contrast, graph construction involves pairwise consistency

checks and uncertainty metric computation, which can scale quadratically with the number of images if all pairs are considered. This explains why graph construction dominates the runtime: it requires dense correspondence verification and optical flow estimation across multiple image pairs, whereas the optimization step leverages sparse factor graphs and converges rapidly. For larger datasets, pruning strategies or retrieval-based filtering will be essential to maintain scalability.

Effect of removing optical flow or epipolar metric individually (Table 10). We analyze the effect of incorporating uncertainty-aware metrics into pose graph optimization. Starting from vanilla PGO with graph construction, we progressively add epipolar and optical flow metrics. The full configuration uses both metrics. Table 10 summarizes the results for the 7-Scenes and Mip-NeRF 360 datasets. We observe that adding uncertainty-aware metrics significantly improves pose graph optimization over vanilla PGO. Epipolar constraints provide global geometric consistency, yielding a 1.0 dB improvement on both datasets. Optical flow offers dense local correspondences, which are particularly beneficial in texture-rich indoor scenes, leading to 1.6 dB improvement over vanilla PGO. Combining both metrics achieves the highest gains (2.3 dB and 2.2 dB): epipolar constraints enforce global structure, while optical flow enhances local robustness. This complementary effect highlights the importance of integrating multiple geometric priors for reliable pose refinement.

Configuration	7-Scenes	Mip 360
Vanilla PGO (with construction)	18.5	21.0
+ Epipolar metric	19.5	22.0
+ Optical flow metric	20.1	22.3
+ Both metrics	**20.8**	**23.2**

Table 10. Ablations on different uncertainty-aware configurations. We report the pose accuracy via view synthesis as PSNR in dB.

Full impact of spare data (Fig. 9). After including the results from the Tanks and Temples dataset, we identified an error in the previously generated figure. The figure has now been corrected, and the version in the main text should be disregarded. The updated figure presented here demonstrates that our method consistently exhibits superior robustness under limited training data conditions.

C. Additional qualitative examples

Epipolar-based uncertainty metric (Fig. 10). To further analyze the behavior of the epipolar-based uncertainty metric, we apply synthetic perturbations to estimated camera poses. Specifically, we introduced rotational perturbations

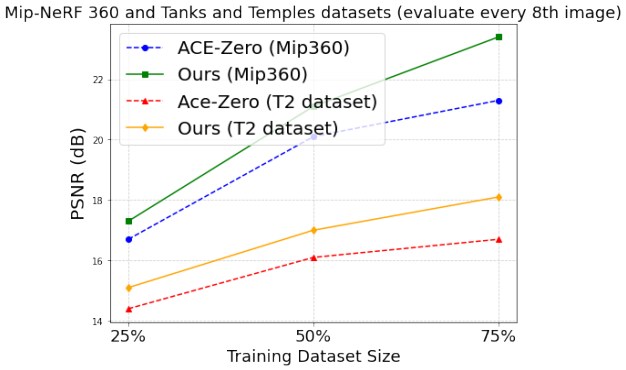

Figure 9. Ablations on sparse training data for both Mip-NeRF 360 and Tanks and Temples (T2) datasets. We show that our method consistently demonstrates better robustness to limited training data.

around the z-axis and translational perturbations along the z-axis to simulate varying degrees of pose deviation. For each pose, we randomly sampled 100 predicted 3D scene coordinates and reprojected them into the paired image. We then drew the corresponding epipolar lines and discarded any second-view projections that fell outside the image boundaries to ensure valid comparisons. Fig. 10 illustrates qualitative results on two datasets (7-Scenes and Tanks and Temples), where reprojected points and their epipolar lines are color-coded by error magnitude (green = low, red = high). Interestingly, in some cases, the highest epipolar error occurs when no perturbation is applied, suggesting that the original pose estimate is inaccurate. Introducing small rotational perturbations can reduce this error, while translation perturbations occasionally lead to similar effects. These observations indicate that epipolar error is sensitive to interactions between pose components and does not consistently correlate with absolute pose accuracy. Consequently, we interpret this metric as a heuristic confidence indicator rather than a strict geometric validator.

Optical flow-based uncertainty metric. We provide additional qualitative examples in Fig. 11.

Additional qualitative examples. We provide additional qualitative examples in Fig. 12.

References

[1] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, 2023. 3, 4

[2] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Pos-

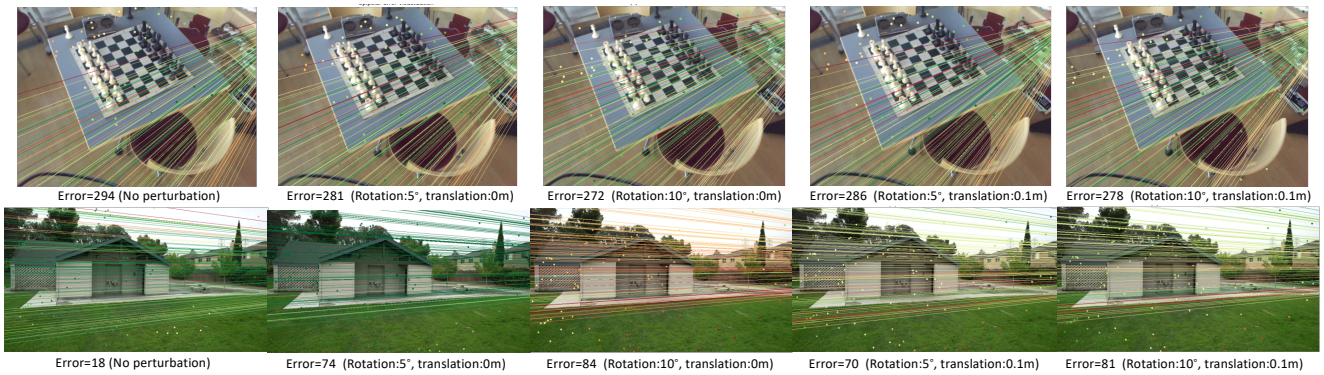

Figure 10. Qualitative example of the epipolar-based uncertainty metric (top row: 7-Scenes dataset; bottom row: Tanks and Temples dataset). We overlay reprojected points with their corresponding epipolar lines, color-coded by error (green = low, red = high), to indicate geometric alignment quality under perturbed poses. In the top row, the highest error occurs when no perturbation is applied, suggesting the original camera pose is inaccurate; introducing rotation perturbations reduces the error. In the bottom row, from left to right, we observe a similar trend as in Fig. 2 of the main text: errors remain low without perturbation, increase with pose perturbations, and occasionally decrease when translation perturbations are added. This behavior highlights that epipolar error is sensitive to interactions between pose components and does not always correlate directly with pose accuracy. Consequently, we treat it as a heuristic confidence indicator rather than a strict geometric validator.

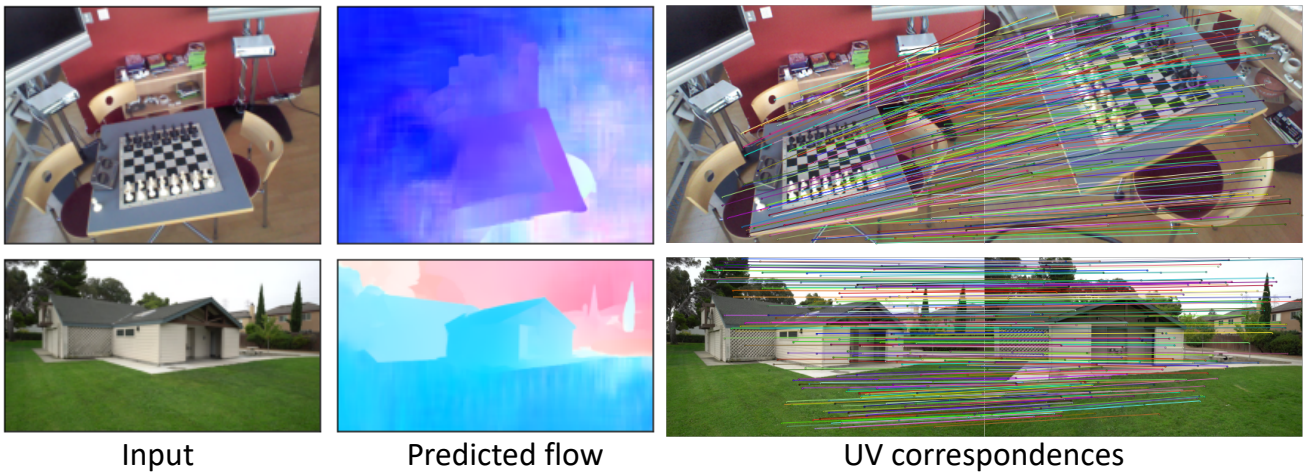

Figure 11. Qualitative examples of optical flow predictions are shown (top row: 7-Scenes dataset; bottom row: Tanks and Temples dataset). From left to right: (i) an image from the input pair, (ii) predicted flow visualization where similar colors indicate similar motion directions, and (iii) UV correspondences overlaid on the image pair, illustrating that optical flow provides robust pixel-level matches across views.

ing of image collections via incremental learning of a relocalizer. *ECCV*, 2024. 2, 3, 4, 7

[3] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025. 2, 3, 7

[4] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *NeurIPS*, 2021. 3

Figure 12. Qualitative comparison on Mip-NeRF 360 (top two rows) and Tanks and Temples datasets (bottom two rows) with VGGT-SLAM [3] and ACE-Zero [2]. While ACE-Zero performs well on small-scale scenes such as Mip-NeRF 360, it struggles with large, complex environments like Tanks and Temples, where our method maintains high rendering quality across diverse scenes.