Supplementary Materials

This document provides supplementary materials for the paper "POSE-FREE 3D GAUSSIAN SPLATTING VIA SHAPE-RAY ESTIMATION." In this document, we present the followings:

0.1. Detailed Discussion on Ray Guidance (Pose Embedding)

Most conventional Multi-view Stereo (MVS) methods rely on camera poses to establish geometric relationships across input views. However, when pose estimates are noisy, these relationships become unreliable, leading to errors in 3D reconstruction. We argue that embedding camera pose awareness into image features is important for mitigating the impact of such noise and maintaining geometric consistency.

To achieve this, we combine predicted Plücker rays with image features to construct the cost volume, leveraging the advantages of using a generic camera representation. This design choice aims to explicitly encode multi-view geometric relationships within image features, ensuring that pose information is directly integrated into the reconstruction process.

Specifically, features from different viewpoints are projected and correlated by converting rays into camera poses and performing homography warping. While this process introduces some pose-induced misalignment in the feature space, we address these issues through a pose-aware cost aggregation process, as detailed in the main paper. As shown in Figure 1, removing pose embedding results in significant discrepancies in geometry estimation, producing blurry reconstructions and visible artifacts. These results emphasize the importance of pose embedding in preserving geometric consistency and improving reconstruction quality during feature fusion.

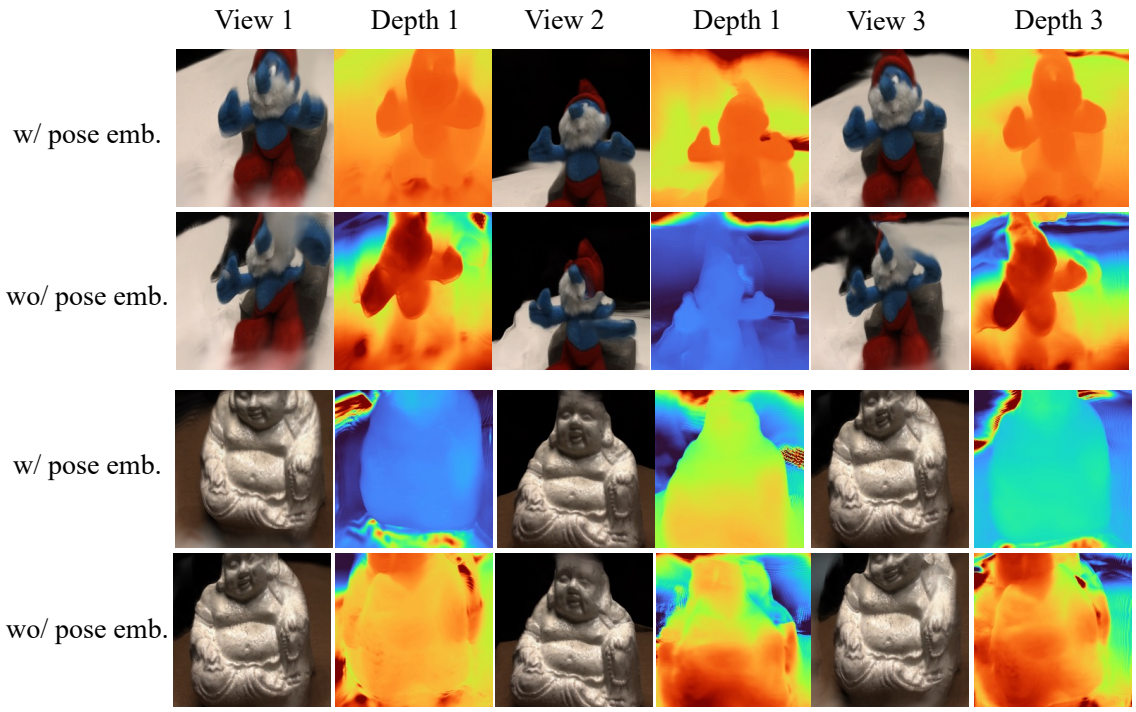

Fig. 1: **Qualitative Ablation Results on Pose Embedding.** Omitting pose embedding leads to blurry artifacts and misaligned structures in 3D reconstructions, while incorporating pose embeddings significantly improves accuracy and sharpness. This demonstrates the critical role of pose-aware fusion in mitigating pose errors and ensuring geometric consistency in multi-view reconstruction.

Table 1: **Comparison on baselines with different pose prediction methods on DTU dataset.**

Method	Pose	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓
PixelSplat	GT	–	–	20.96	0.65	0.31
	COLMAP	7.10	31.62	13.49	0.34	0.66
	MASt3R	2.40	**3.52**	15.69	0.40	0.50
	DUSt3R	**1.77**	13.66	15.98	0.42	0.47
	Ours	2.74	6.28	13.29	0.31	0.66
MVSplat	GT	–	–	21.00	0.69	0.24
	COLMAP	7.10	31.62	14.69	0.44	0.46
	MASt3R	2.40	**3.52**	13.31	0.31	0.58
	DUSt3R	**1.77**	13.66	13.22	0.32	0.58
	Ours	2.74	6.28	14.08	0.33	0.51
Ours	–	2.74	6.28	**19.94**	**0.63**	**0.28**

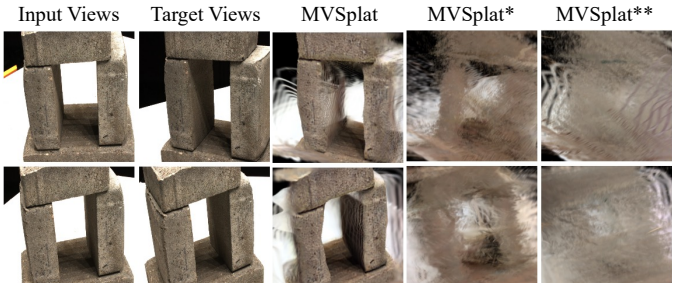

Fig. 2: **Rendering of MVSplat Trained with Predicted and Noisy Poses.** The row labeled MVSplat shows the results of training with ground-truth poses, while MVSplat* and MVSplat** refer to the MVSplat model trained with predicted poses from DUSt3R and noisy poses with minor errors, respectively.

0.2. Details of baseline implementation

For the small-scale DTU dataset [1], we compared and validated our method against the pose-free baseline LEAP [2]. The LEAP model was trained on the DTU 3-view dataset for 140K iterations. Since our evaluation on DTU uses three input views, we also trained pose-dependent state-of-the-art generalizable 3D reconstruction methods, including PixelSplat [3] and MVSplat [4], with a batch size of 1 for 140K iterations.

For the large-scale RealEstate10K dataset [5], we compared our method against pose-free baselines CoPoNeRF [6] and FlowCam [7]. Since CoPoNeRF and FlowCam use the same train-test split as our method, we directly compared our results with the reported values. Additionally, PixelSplat and MVSplat were evaluated using their pretrained checkpoints on the same 2-view train-test split settings.

We evaluated pose-dependent baselines under two conditions: using predicted poses and poses perturbed by random noise. For predicted poses, we used one of the state-of-the-art pose estimator, DUSt3R [8], to estimate poses from the input images. To ensure fair comparisons, we also evaluated the baselines with various pose estimators, including COLMAP, DUSt3R [8], MASt3R [9] and SHARE. For DUSt3R and MASt3R, we utilized pre-trained model weights provided in their official GitHub repositories. As shown in Table 1, our method consistently outperformed these combinations. Furthermore, the results for noisy poses shown in main paper (Table1 and Table2 of main paper) highlight that even minor errors—currently unavoidable by state-of-the-art pose estimators—can introduce significant instability in reconstruction quality.

We trained the baseline models using ground-truth (GT) poses, as training with noisy poses lacking specific noise patterns often resulted in instability, divergence, or failure to converge. Figure 2 illustrates a comparison of MVSplat models trained on DTU with GT poses versus those trained with predicted poses from DUSt3R [8] and slightly perturbed poses ($\sigma = 0.01$, rotation error 0.95°, translation error 1.05°). These findings demonstrate that even small amounts of noise during training can destabilize models by introducing subtle misalignments between views, leading to a decline in reconstruction quality.

0.3. Details of model implementation

In this section, we'll discuss our framework in more detail. Given sparse-view unposed images, our goal is to build comprehensive Gaussians in a canonical space. The output of the multi-view feature extractor is $V \times C \times H \times W$, where we set C as 128 in all experiments. Given these features, we estimate the relative Plücker rays $V \times 6 \times H \times W$ with two additional transformer blocks following the U-Net structure of [10]. Then, we embed ray with a lightweight MLP to latent space and modulate multi-view features using AdaLN [11], following LaRa [12]. In the ray-guided multi-view fusion process, we first build the cost volumes from all input views, where the depth candidates D are all set to 128. We warp all the features to the reference views with the estimated pose (converted from Plücker rays). Then, we build the geometry volume V_g. The geometry volume is used to estimate the anchor points $3 \times \frac{H}{4} \times \frac{W}{4}$. Simultaneously, we build the feature volume V_f in a similar manner, but with the upscaled multi-view features, to estimate the offset vectors and Gaussian parameters necessary for finer detail reconstruction.

We divide channels of $\mathbf{V}_f$ for displacement prediction of anchor points (32), and the remaining channels (96) encode texture-related Gaussian parameters. The geometry channels of $\mathbf{V}_f$ are passed through the offset prediction MLP head f_o, which predicts the offset vectors $\Delta\mathbf{p}_k = f_o(\mathbf{V}_f)$, for the Gaussian positions. We set $K = 3$ for all experiments. These offset vectors are then concatenated with the remaining channels of $\mathbf{V}_f$ Another MLP head, f_p, processes the concatenated features to estimate the remaining Gaussian parameters.

0.4. Additional Results

Cross-dataset generalization Figure 3 present the quantitative results of cross-dataset generalization, comparing our proposed method, SHARE, with baseline approaches. Models trained on the RealEstate10K [5] dataset were evaluated on the ACID [13] dataset, while those trained on the DTU [1] dataset were tested on BlendedMVS [14]. The ACID dataset comprises natural large-scaled scenes captured using aerial drones, divided into 11,075 scenes for training and 1,972 scenes for testing, with accompanying camera extrinsic and intrinsic parameters. The BlendedMVS dataset consists of 3D models of diverse scenes, including outdoor and indoor environments. In our experiments, we utilize a subset of BlendedMVS as a cross-dataset evaluation benchmark to assess the generalization ability of our method.

Table 2: **Quantitative Comparison with Concurrent Work.** We compare our method with Splatt3R on DTU and RealEstate10K using two input views. Splatt3R results are from pretrained weights on ScanNet++, while ours are trained per dataset. Best results are in bold. The best results are highlighted in bold.

	DTU (2-views)			RealEstate10K		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Splatt3R	11.78	0.28	0.57	15.80	0.53	0.30
Ours	**17.50**	**0.34**	**0.48**	**21.23**	**0.71**	**0.26**

Table 3: **Quantitative Comparison with Concurrent Work: Cross-Dataset Generalization.** We evaluate and compare the cross-dataset generalization performance of our method and Splatt3R. The best results are highlighted in bold.

Method	Training data	ACID		
		PSNR ↑	SSIM ↑	LPIPS ↓
Splatt3R	ScanNet++	17.49	0.63	**0.26**
Ours	RealEstate10K	**23.47**	**0.69**	**0.26**

Table 4: **Quantitative results of pose estimation performance.** We evaluate pose estimation on DTU with small baselines using three input views. Bold indicates the lowest error.

Method	Rot. ↓	Trans. ↓
DUSt3R	1.77	13.66
MASt3R	2.40	3.52
COLMAP	7.10	31.62
Relpose++	19.56	44.18
RayRegression	3.10	6.57
Ours	**2.74**	**6.28**

Table 5: **Model Efficiency Measurements.** Each metric is evaluated across models using the same dataset configuration and averaged for consistency. Bold indicates the lowest inference time and memory consumption.

Method	Inference time (s)	GPU Memory (MB)
CoPoNeRF	3.37	9587.22
MVSplat + MASt3R	0.22	**4376.94**
Splatt3R	0.26	6198.00
Ours	**0.17**	5887.18

Notably, under the challenging conditions of pose error $\sigma = 0.01$, which remains difficult even for state-of-the-art pose estimators, SHARE consistently outperforms all baseline methods across all metrics. These findings underscore the robustness of SHARE, particularly in realistic scenarios where pose estimation inaccuracies are inevitable.

Comparision with the Concurrent Work. We compare SHARE with our concurrent work, Splatt3R [15] which utilizes pretrained MASt3R [9] weights for geometry estimation. Since Splatt3R requires ground-truth dense depth maps for training, it is not directly applicable to our datasets. RealEstate10K [5] lacks ground-truth depths, and DTU [1] provides masked depths, which we found to be incompatible with Splatt3R's pixel-aligned dense prediction without modifications. Instead, we directly compare with the pretrained Splatt3R model trained on ScanNet++ [16]. We note that Splatt3R employs a "masking loss" (refer to Section 3.4 in their paper) to render only valid pixels for the target view based on input images. Measuring metrics across all regions would significantly lower PSNR and misrepresent its performance. To ensure fairness, we evaluate PSNR and other metrics only on valid pixels produced by Splatt3R (pixels with > 0 values).

In Table 2 and Figure 4, we present comparisons both on the DTU and RealEstate10K datasets, where SHARE outperforms Splatt3R. To ensure fairness, as comparing Splatt3R trained on ScanNet++ with SHARE trained on each dataset may introduce biases, we conducted additional evaluations in a cross-dataset setting. Specifically, we compared Splatt3R trained on ScanNet++ and SHARE trained on RealEstate10K in the ACID [13] dataset. As illustrated in Table 3 and Figure 5, SHARE demonstrates superior rendering quality compared to Splatt3R. We measure metrics only for the valid pixels produced by Splatt3R (pixels with > 0 values). Including entire regions would lead to significant drops in PSNR and thus would not reflect the method's intended performance. Splatt3R exhibits scale ambiguity in its predicted scenes, which can lead to a substantial drop in performance when applied to datasets with unseen scale distributions.

Discussion on large baseline inputs We visualized large-baseline camera scenarios (Figure 6). We compare our method with PixelSplat [3] and MVSplat [4] using both our predicted poses and perturbed poses with Gaussian noise, which exhibit similar or lower pose errors compared to predicted poses.

Results of pose estimation We evaluated our pose estimation performance in terms of rotation error (degrees) and translation error (degrees), as detailed in the main paper. Comparisons were made against state-of-the-art pose estimators, including DUSt3R [8], MASt3R [9], and RayRegression from Cameras-as-Rays [10]. Additionally, we compared our method with COLMAP [17] for primitive pose estimation and RelPose++ [18] as a direct 6D pose estimator. The evaluation used three small-baseline views from the DTU [1] dataset as input images.

While our primary objective is high-quality novel view synthesis rather than pose estimation, our method achieves pose estimation performance comparable to state-of-the-art methods, further demonstrating its robustness and versatility.

Discussion on Efficiency. We evaluated inference time (seconds) and GPU memory usage (MB) of our method against baselines on the RealEstate10K dataset (Table 5). Inference time measures the end-to-end duration for novel view synthesis using two unposed input images, while GPU memory usage accounts for both static and dynamic allocations during inference. SHARE achieves superior efficiency in both inference time and GPU memory usage compared to the pose-free, generalizable NVS baseline CoPoNeRF [6] and the concurrent method Splatt3R. Furthermore, our approach delivers the highest rendering quality among the compared methods, underscoring its effectiveness. All experiments were conducted on an NVIDIA RTX 4080 GPU.

Qualitative results of novel view synthesis We present our additional qualitative results on the DTU [1] dataset (Figure 7) and RealEstate10K [5] dataset (Figure 8).

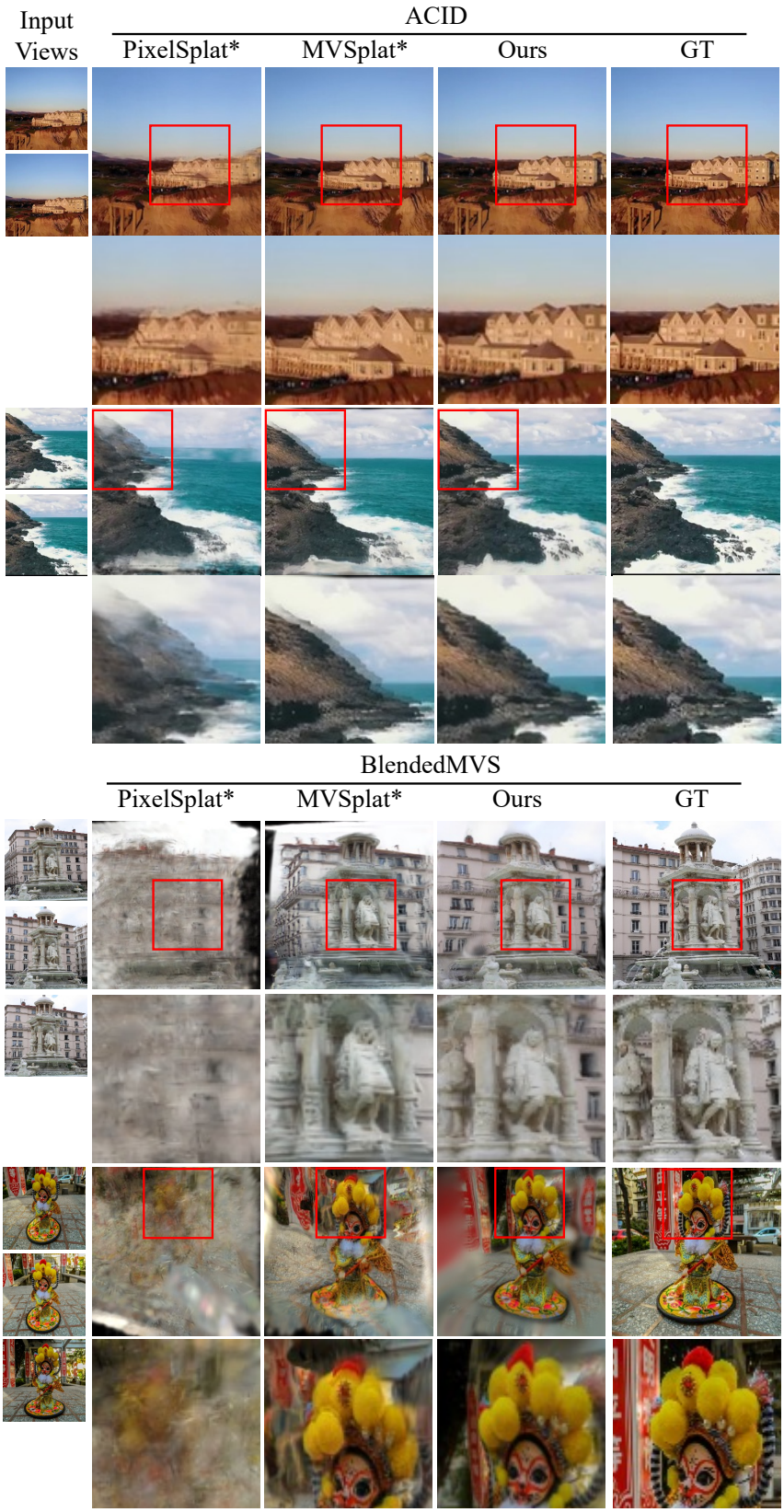

Fig. 3: **Qualitative Results for Novel View Synthesis in Cross-Dataset Generalization.** PixelSplat* and MVSplat* denote methods combined with noisy camera settings ($\sigma = 0.01$). To aid visibility, we highlight the regions of interest with red boxes and provide close-up visualizations of these areas for detailed comparison.

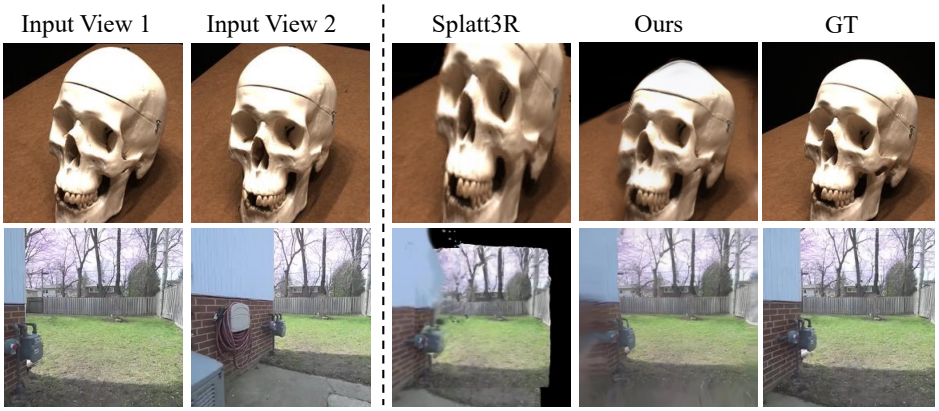

Fig. 4: **Qualitative Comparision with the Concurrent Work.**

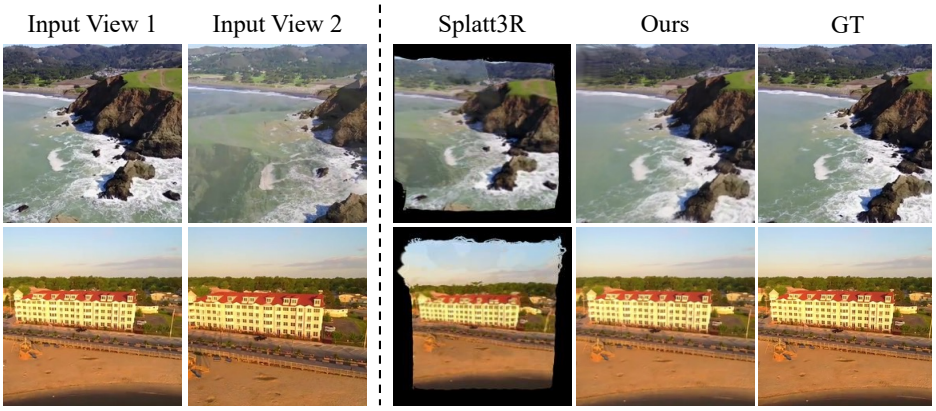

Fig. 5: **Qualitative Comparision with the Concurrent Work: Cross-dataset Generalization.**

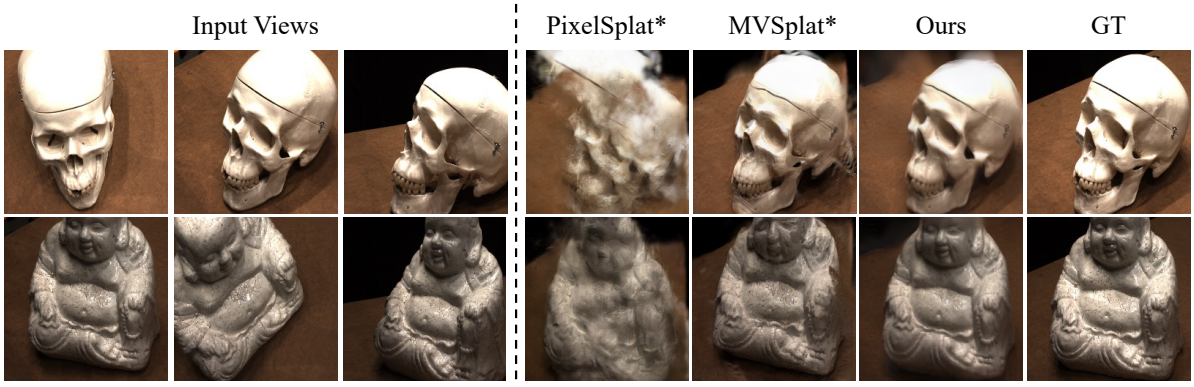

Fig. 6: **Qualitative Results of Novel View Synthesis with Large-Baseline View Sets.** PixelSplat and MVSplat denote methods combined with a noisy camera setup, incorporating Gaussian noise with a standard deviation of 0.01.

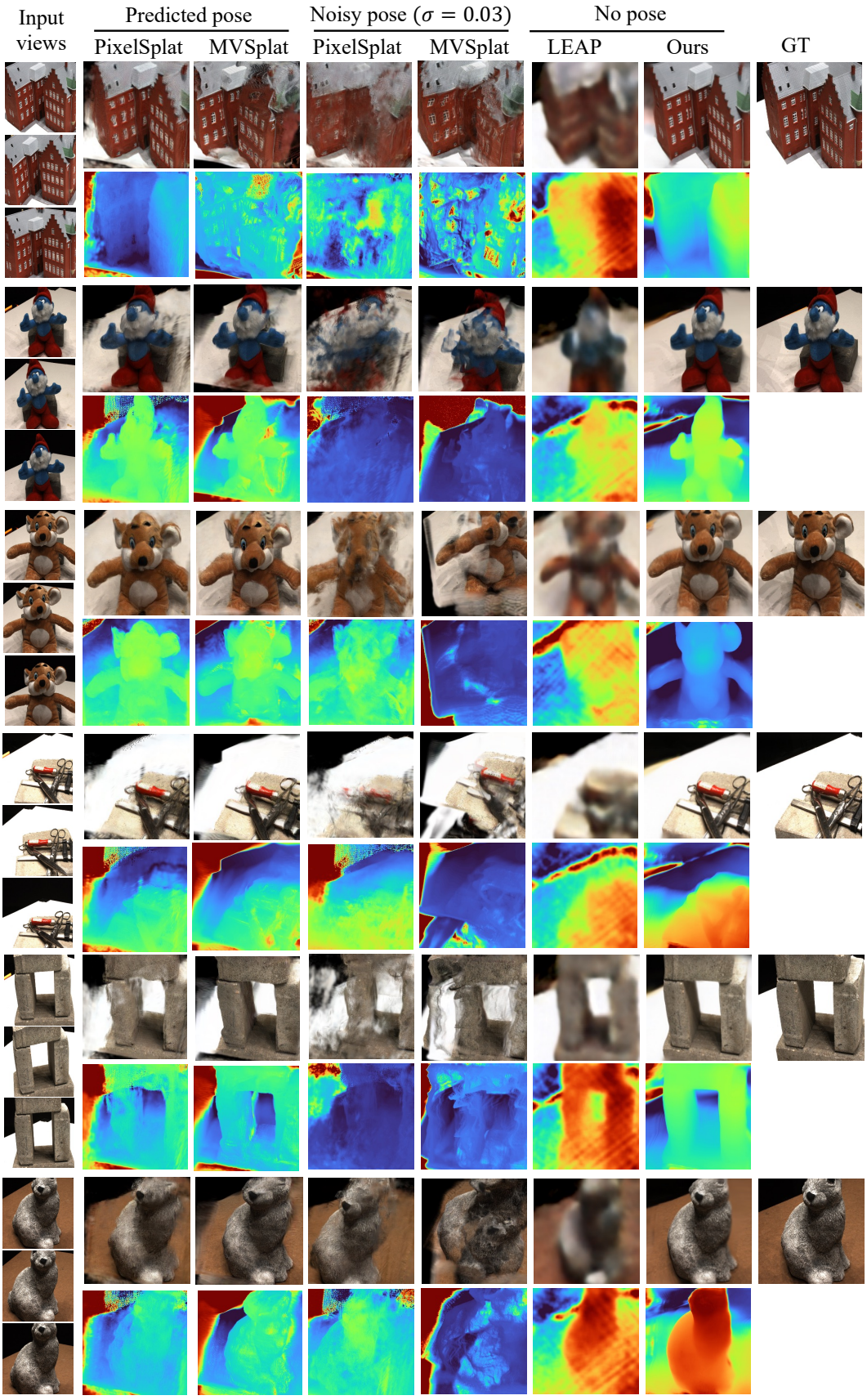

Fig. 7: **Additional Qualitative Results on the DTU Dataset.** Rendered target images are shown based on three input views. The predicted pose indicates poses predicted using DUSt3R [8].

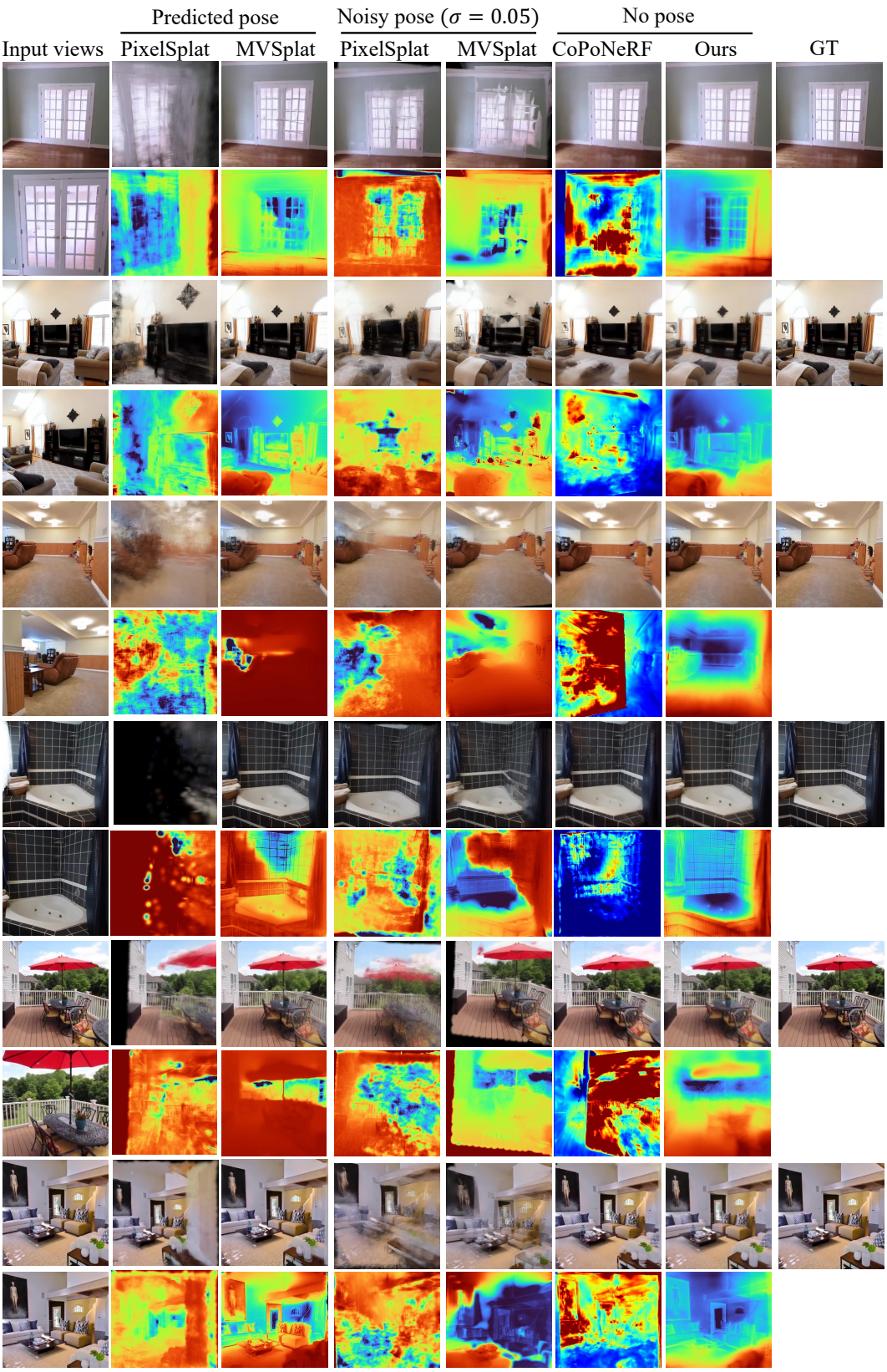

Fig. 8: **Rendering and Depth comparison on RealEstate10K** The visualized images are rendered target images given 2 input views. The predicted pose indicates poses predicted using DUSt3R [8].

1. REFERENCES

[1] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs, "Large scale multi-view stereopsis evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 406–413.

[2] Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang, "Leap: Liberate sparse-view 3d modeling from camera poses," *arXiv preprint arXiv:2310.01410*, 2023.

[3] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann, "pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19457–19467.

[4] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai, "Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images," *arXiv preprint arXiv:2403.14627*, 2024.

[5] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *ACM Trans. Graph*, vol. 37, 2018.

[6] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jiaolong Yang, Seungryong Kim, and Chong Luo, "Unifying correspondence pose and nerf for generalized pose-free novel view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20196–20206.

[7] Cameron Smith, Yilun Du, Ayush Tewari, and Vincent Sitzmann, "Flowcam: training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow," *arXiv preprint arXiv:2306.00180*, 2023.

[8] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20697–20709.

[9] Vincent Leroy, Yohann Cabon, and Jerome Revaud, "Grounding image matching in 3d with mast3r," 2024.

[10] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani, "Cameras as rays: Pose estimation via ray diffusion," in *International Conference on Learning Representations (ICLR)*, 2024.

[11] William Peebles and Saining Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

[12] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger, "Lara: Efficient large-baseline radiance fields," in *European Conference on Computer Vision (ECCV)*, 2024.

[13] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa, "Infinite nature: Perpetual view generation of natural scenes from a single image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14458–14467.

[14] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan, "Blendedmvs: A large-scale dataset for generalized multi-view stereo networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1790–1799.

[15] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu, "Splatt3r: Zero-shot gaussian splatting from uncalibarated image pairs," *arXiv preprint arXiv:2408.13912*, 2024.

[16] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai, "Scannet++: A high-fidelity dataset of 3d indoor scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12–22.

[17] Johannes L Schonberger and Jan-Michael Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[18] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani, "Relpose++: Recovering 6d poses from sparse-view observations," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 106–115.