
Supplementary: Geometrically Consistent Generalizable Splatting

Anonymous Author(s)

Affiliation

Address

email

1 A Supplemental Material

2 A.1 Architecture

3 Our approach adapts the N-view pose-free generalizable splatting architecture proposed by [16]
4 and consists of three main modules: a transformer-based image encoder, two cross-attention-based
5 asymmetric multi-view feature aggregators and a set of dense prediction transformer decoders to
6 predict parameters of per-pixel Gaussian splats. Specifically, the image encoder transforms RGB
7 image patches into a sequence of tokens and are augmented with the intrinsic parameters of the camera.
8 These pre-image token sequences are processed by a ViT encoder independently. The resulting per-
9 image features are subsequently fused via two sets of feature aggregators, each comprising of
10 multi-head cross-attention layers, effectively aggregating information across multiple views. These
11 transformer-based networks provide features for the prediction of Gaussian splats for the reference
12 (first) and subsequent images, respectively. Parameters of the Gaussians are estimated by two separate
13 DPT prediction heads: the first head directly infers Gaussian centers for input image pixels, the
14 second head predicts the rest of the Gaussian parameters and combines fine image-level features
15 for accurate appearance estimation. Following [14], we use two asymmetric feature aggregators to
16 provide the Gaussian centers for the pixels of each frame to be aligned with the first image of the
17 sequence. As all the recovered Gaussians are aligned, the union of them can be directly rendered
18 to an arbitrary viewpoint without requiring any warping. Notably, both image tokenizer and feature
19 aggregator employ Vision Transformer (ViT) backbones exclusively, without specialized architectures
20 such as epipolar attention or multi-view cost-volumes. This geometry-free architecture does not
21 require relative camera poses during training and inference and has been shown to perform on par or
22 better than the generalizable splatting methods deploying these specialized architectures [16].

23 A.2 Implementation Details

24 All of our generalizable Gaussian splatting models are trained using the setup described in NoPoS-
25 plat [16]. Training is performed on a cluster of 24 NVIDIA A100 (40 GB) GPUs with a batch size of
26 6 per-GPU (144 total), while all evaluations are performed on a single NVIDIA A6000 GPU. Models
27 are trained on the RE10K [17] training split for 18751 iterations using two 256×256 input images
28 and rendering three virtual views per sample to minimize the view-synthesis loss. Regularization
29 weights are set to $\lambda_o = 0.05$ and $\lambda_a = 0.1$ for both the full model and all applicable ablations. We
30 employ a base learning rate of 2×10^{-4} , while layers of the DUST3R backbone are updated with a
31 reduced rate of 2×10^{-5} . We also note that, like NoPoSplat, DUST3R-based splatting extensions
32 are unable to be trained from scratch on RE10K without weight distillation; instead, we initialize
33 them with MAST3R-pretrained weights, and allow competing baselines to adopt similar supervised
34 backbone initialization, where available.

35 To evaluate zero-shot generalization in pose estimation, geometry reconstruction, and novel-view
36 synthesis, we test all models and baselines on the ACID [11] split from [16] and the ScanNet-V1

Table 1: **Comparison of our retrained NoPoSplat against the public checkpoint.** (a) Pose evaluation (with photometric optimization) on RE10K [17], ScanNet-V1 [5] and ACID [11]. (b) Depth estimation for novel views (with pose refinement) on ScanNet-V1. (c) Novel-view synthesis on RE10K.

(a) Pose evaluation									
Method	RE10K			ScanNet-V1			ACID		
	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑
NoPoSplat*	0.672	0.792	0.869	0.111	0.254	0.465	0.454	0.591	0.709
NoPoSplat	0.672	0.791	0.868	0.109	0.256	0.463	0.456	0.593	0.705

(b) Depth estimation			
Method	Rendered Depth (Novel Views)		
	Abs Rel↓	$\delta_1 < 1.10$ ↑	$\delta_1 < 1.25$ ↑
NoPoSplat*	0.127	0.564	0.859
NoPoSplat	0.126	0.567	0.861

(c) Novel-view synthesis												
Method	Small			Medium			Large			Average		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NoPoSplat*	21.086	0.721	0.237	23.134	0.776	0.185	25.086	0.818	0.141	23.189	0.775	0.185
NoPoSplat	21.097	0.723	0.237	23.191	0.779	0.187	25.107	0.817	0.144	23.244	0.778	0.187

test set [5], which comprises 2000 indoor RGB-D image pairs. To select novel views in ScanNet-V1 [5], we uniformly sample four intermediate viewpoints (if applicable) along the camera trajectory between each pair of source views used for pose evaluation, resulting in 1592 samples (out of 2000). These novel-views are employed for both novel-view depth evaluation and novel-view synthesis experiments.

We retrained NoPoSplat [16] under our training setup and hyperparameters for a fair comparison. Minor deviations from its published results are detailed in Table 1. The source code and pretrained models will be released to ensure reproducibility.

A.3 Rendered Normal Consistency in 2DGS

For every 3D Gaussian defined in the 3D space corresponding to a pixel j in image t , $\mathcal{G}_t^j = (\mu_t^j, \alpha_t^j, \Sigma_t^j, c_t^j)$, 2DGS [9] first projects the center and the covariance to the image plane of the novel view. Let $\mathbf{P}_f = \mathbf{K}_f[\mathbf{R}_f | \mathbf{T}_f] \in \mathbb{R}^{3 \times 4}$ be the novel-view’s projection matrix. The homogeneous image of the mean is $\bar{\mu} = \mathbf{P}[\mu | 1]^\top$, $\mu' = (\bar{\mu}_x / \bar{\mu}_z, \bar{\mu}_y / \bar{\mu}_z)^\top$. Denoting by $\mathbf{J} = \partial(\mu' \bar{\mu}_z) / \partial \mu$ the Jacobian of the local affine approximation of the perspective map, the (unnormalized) screen-space covariance is obtained by $\Sigma' = \mathbf{J} \mathbf{P} \Sigma \mathbf{P}^\top \mathbf{J}^\top$, and $\Sigma'_{uv} = (\Sigma')_{1:2, 1:2}$ after which only its upper-left 2×2 block Σ'_{uv} is kept. The resulting projected 2D Gaussian $\mathcal{G}'(u, v) = \exp[-\frac{1}{2}((u, v)^\top - \mu')^\top \Sigma'_{uv}^{-1}((u, v)^\top - \mu')]$ is splatted with per-pixel front-to-back α -blending

$$\hat{\mathbf{I}}_f(u, v) = \sum_{k=1}^K c_k \alpha_k \mathcal{G}'_k(u, v) \prod_{j < k} (1 - \alpha_j \mathcal{G}'_j(u, v)), \quad (1)$$

where the 3D Gaussians are pre-sorted per tile by depth and rasterized with the tile-based GPU pipeline introduced in [9]. Equation (1) is used to render the novel-view RGB image via α -blending; the same blending operation is also applied to render depth and surface normals for that view.

2DGS [8] introduces a *rendered normal–depth consistency* loss that enforces agreement between rendered surface normals and corresponding rendered depth maps. Let $D_r(\mathbf{x})$ be the depth rendered with the same weights as color, where $\mathbf{x} = (u, v)^\top$, and let $\mathbf{N}_r(\mathbf{x}) = \sum_k \omega_k(\mathbf{x}) \mathbf{n}_k$ be the correspondingly blended normal, with $\omega_k = \alpha_k \mathcal{G}'_k \prod_{j < k} (1 - \alpha_j \mathcal{G}'_j)$. A surface normal can also be estimated from the rendered depth map ($D_r(\mathbf{x})$) through finite differences. The loss adopted from [8]

Table 2: **Ablation of our Gaussian orientation losses.** Comparison between models trained with \mathcal{L}_{align} and \mathcal{L}_{RNC} versus our full model trained with \mathcal{L}_{align} and \mathcal{L}_{orient} . (a) Pose evaluation (with photometric optimization) on RE10K [17], ScanNet-V1 [5] and ACID [11]. (b) Depth estimation for novel views (with pose refinement) on ScanNet-V1. (c) Novel-view synthesis on RE10K.

(a) Pose evaluation

Method	RE10K			ScanNet-V1			ACID		
	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20° ↑
Ours (2DGS+Align+RNC)	0.681	0.799	0.870	0.137	0.313	0.521	0.476	0.609	0.720
Ours (2DGS+Align+Orient)	0.689	0.804	0.876	0.156	0.334	0.539	0.488	0.619	0.726

(b) Depth evaluation

Method	Rendered Depth (Novel Views)		
	Abs Rel↓	$\delta_1 < 1.10 \uparrow$	$\delta_1 < 1.25 \uparrow$
Ours (2DGS+Align+RNC)	0.105	0.706	0.900
Ours (2DGS+Align+Orient)	0.100	0.707	0.904

(c) Novel-view synthesis

Method	Small			Medium			Large			Average		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Ours (2DGS+Align+RNC)	21.340	0.734	0.236	23.451	0.787	0.185	25.522	0.829	0.141	23.506	0.788	0.185
Ours (2DGS+Align+Orient)	21.349	0.739	0.234	23.463	0.789	0.184	25.628	0.832	0.141	23.564	0.789	0.184

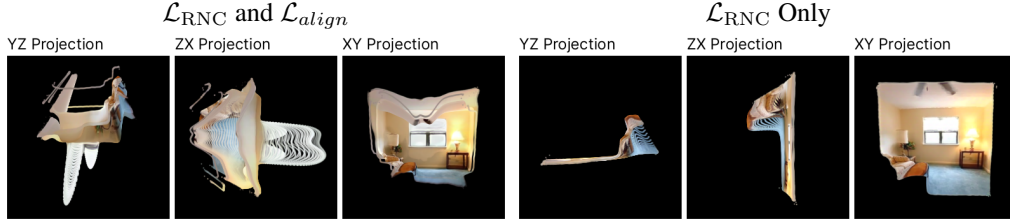


Figure 1: **Performance drops drastically when the proposed orientation loss in the main paper is replaced with a commonly used consistency loss between rendered depth and normal per-view.** The reconstructed 3D scene (trained on RE10K [17]) is projected onto three axis-aligned planes, comparing results with and without the alignment loss in place when the orientations are learned by enforcing consistency between rendered depth and normal [8]. Using the consistency between depth and normal without \mathcal{L}_{align} provides degenerate reconstructions. With the \mathcal{L}_{align} network learns meaningful structure, but remains inferior to the reported results in the main manuscript using \mathcal{L}_{orient} loss.

62 penalizes the angular mismatch between rendered normals and those derived from the depth map.

$$\mathcal{L}_{RNC} = \frac{1}{|\Omega|} \sum_{\mathbf{x} \in \Omega} \omega(\mathbf{x}) (1 - \langle \mathbf{N}_r(\mathbf{x}), \widehat{\mathbf{N}}(\mathbf{x}) \rangle), \quad (2)$$

63 with $\omega(\mathbf{x}) = \sum_k \omega_k(\mathbf{x})$.

64 \mathcal{L}_{RNC} acts *after* rasterization and therefore enforces coherence between the *rendered* depth and
65 normal, while our own \mathcal{L}_{orient} operates directly on the Gaussian parameters, avoiding any dependence
66 on the rasterizer. This consistency term enforces each 2D Gaussian splat contributing to a pixel’s
67 rendering to be perpendicular to that pixel’s rendered normal. While this prior proved effective
68 in [8] and its few-shot variants, naively applying it in a generalizable splatting network causes
69 the optimization of Gaussian means and orientations to stall in a near-planar local minimum (see
70 Figure 1). Detaching the rendered depth from the computation graph resolves this by treating depth
71 rendering as pseudo-label generation for Gaussian orientations, yielding sensible Gaussian normals
72 and centers. It should be noted that successful training of the Gaussians orientation with \mathcal{L}_{RNC} still
73 requires the alignment loss \mathcal{L}_{align} . Table 2 presents an ablation comparing models trained with
74 \mathcal{L}_{align} plus \mathcal{L}_{RNC} against our full model using both \mathcal{L}_{align} and \mathcal{L}_{orient} .

Table 3: **Single-view depth estimation results on NYUD-v2 [12]**. The scene is reconstructed from a single image which is copied to be used as input of our two frame splat estimators. We report “best depth estimate” for every splat based approach which could be rendered or intermediate depth. The best overall results are highlighted in **bold**. We Compare results with a few state of art self-supervised single view depth estimation approaches alongside the splatting based baselines.

Training scheme	Method	Best <i>Source-View</i> Depths		
		Abs Rel↓	$\delta_1 < 1.10 \uparrow$	$\delta_1 < 1.25 \uparrow$
<i>Two-view Supervised</i>	DUST3R [14]	0.065	-	0.941
<i>Single-view Self-Supervised</i>	MonoDepth V2 [7]	0.162	-	0.745
	SC-SfM-Learners [1]	0.138	-	0.796
	SC-DepthV3 [13]	0.123	-	0.848
<i>Two-view Self-Supervised Pose Req.</i>	pixelSplat [2]	0.746	0.138	0.314
	MVSplat [3]	0.130	0.534	0.823
	DepthSplat [15]	0.143	0.525	0.802
<i>Two-view Self-Supervised Pose Free</i>	NoPoSplat [16]	0.114	0.575	0.855
	Ours (2DGS+Align)	0.116	0.581	0.849
	Ours (2DGS+Orient)	0.109	0.593	0.867

75 A.4 Additional Experiments

76 Source View Depth from Single Image.

77 Using Gaussian splats as a scene representation enables both novel-view synthesis and holistic 3D
 78 reconstruction. While most generalizable splatting methods focus on image interpolation between
 79 input views, our work emphasizes geometric consistency and full 3D reconstruction, with interpolation
 80 capabilities. We also demonstrate that our Gaussian splat based approach outperforms all other
 81 baselines in source-view depth estimation (Table 3 in the main paper). However, we also analyze
 82 whether switching from depth-map or point-cloud representations to Gaussian splats offers advantages
 83 in source-view depth prediction.

84 As our method is self-supervised, the fairest comparison is against other self-supervised approaches
 85 that use two uncalibrated views and are trained on RE10k [17]. However, given the scarcity of two-
 86 view self-supervised structure estimators, we instead compare on the well-studied task of single-view
 87 depth estimation, for which a range of strong baselines exists. We adopt the DUST3R [14] single-view
 88 depth evaluation protocol by duplicating each input into both views. As in Table 3 (main paper), for
 89 each splatting-based baseline we report its best depth estimate—either from rendered Gaussians or
 90 directly from pixel-aligned 3D Gaussian means, in Table 3.

91 In addition to splatting-based baselines, we include three state-of-the-art self-supervised single-
 92 view depth estimators: MonoDepth-v2 [7], SC-SfM-Learners [1], and SC-Depth-v3 [13]. Despite
 93 this, some self-supervised, generalizable splatting methods achieve depth-prediction performance
 94 comparable to direct single-view networks such as [7], [1], and [13]. pixelSplat [2], which employs
 95 an epipolar transformer to encode multi-view geometry, fails to produce meaningful depths at
 96 zero baseline. MVSplat [3], with its cost-volume-based multi-view aggregation, outperforms self-
 97 supervised single-view methods such as MonoDepth v2 [7] and SC-SfM-Learners [1]. NoPoSplat [16]
 98 remains the strongest prior splatting-based approach. In this zero-baseline setup, the alignment loss
 99 proposed in the paper did not improve depth-prediction performance, nor did source image synthesize
 100 based pose optimization. Our 2D surfel-like Gaussian splats trained with the orientation loss achieve
 101 a substantial improvement over the NoPoSplat [16] baseline and outperform all in-domain self-
 102 supervised methods on depth estimation.

103 **Novel View Synthesis Evaluation.** While novel-view synthesis is not the primary focus of this work,
 104 we evaluate both in-domain and zero-shot out-of-domain performance against relevant baselines
 105 (Table 4 and Table 5). For pose-free methods, novel views are rendered directly from all reconstructed

Table 4: **Novel-view synthesis performance on RealEstate10K [17].** We compare pose-required and pose-free methods against our proposed variants. The best pose-free results (without target-pose optimization) are in **bold**, and the top pose-required method is underlined.

		Small			Medium			Large			Average		
Method		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Pose-Required	pixelNeRF	18.417	0.601	0.526	19.930	0.632	0.480	20.869	0.639	0.458	19.824	0.626	0.485
	AttnRend	19.151	0.663	0.368	22.532	0.763	0.269	25.897	0.845	0.186	22.664	0.762	0.269
	pixelSplat	20.263	0.717	0.266	23.711	0.809	0.181	27.151	0.879	0.122	23.848	0.806	0.185
	MVSplat	<u>20.353</u>	<u>0.724</u>	<u>0.250</u>	<u>23.778</u>	<u>0.812</u>	<u>0.173</u>	<u>27.408</u>	<u>0.884</u>	<u>0.116</u>	<u>23.977</u>	<u>0.811</u>	<u>0.176</u>
Pose-Free	Splatt3R	14.352	0.475	0.472	15.529	0.502	0.425	15.817	0.483	0.421	15.318	0.490	0.436
	CoPoNeRF	17.393	0.585	0.462	18.813	0.616	0.392	20.464	0.652	0.318	18.938	0.619	0.388
	NoPoSplat	21.097	0.723	0.237	23.191	0.779	0.187	25.107	0.817	0.144	23.244	0.778	0.187
	Ours (2DGS)	21.051	0.725	0.245	23.338	0.784	0.187	25.541	0.830	0.140	23.407	0.782	0.188
	Ours (2DGS+Align)	21.341	0.736	0.236	23.453	0.788	0.185	25.529	0.829	0.141	23.528	0.787	0.185
	Ours (2DGS+Orient)	21.275	0.733	0.236	23.198	0.780	0.187	25.205	0.823	0.144	23.300	0.781	0.187
	Ours (2DGS+Align+Orient)	21.349	0.739	0.234	23.463	0.789	0.184	25.628	0.832	0.141	23.564	0.789	0.184

Table 5: **Zero-shot out-of-distribution novel-view synthesis performance on ACID [11] and ScanNet-V1 [5], compared against state-of-the-art methods.** All models are trained exclusively on the RE10k dataset. The overall best results are highlighted in **bold**. Note that these results are obtained without optimizing camera poses for target views.

		ACID			ScanNet-V1		
Method		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
<i>Pose-Free</i>	NoPoSplat	23.379	0.683	0.238	21.068	0.646	0.268
	Ours (2DGS)	23.914	0.704	0.233	21.167	0.648	0.267
	Ours (2DGS+Align)	23.745	0.697	0.235	20.919	0.644	0.271
	Ours (2DGS+Orient)	23.848	0.702	0.235	21.178	0.648	0.271
	Ours (2DGS+Align+Orient)	23.950	0.707	0.233	21.163	0.648	0.268

Gaussian splats at a fixed target pose relative to the first input image. Methods that predict Gaussian means as depth maps require an additional warping step using the ground-truth relative pose to align Gaussians to the first-view frame. Although some works (e.g., [?]) report novel-view results for DUS3R [14] and MAST3R [10], these models are not designed for view synthesis, so we omit them from our depth and image-synthesis comparisons. Note that we do not optimize camera poses for target-view image synthesis. Although common in some NeRF and 3DGS methods, such pose refinement can obscure geometric inconsistencies and relies on “peeking” at ground truth during synthesis [4, 6], which we explicitly avoid.

Our warping-free approach outperforms prior art in novel-view synthesis. We observe improvements over NoPoSplat [16] in novel-view synthesis when training with our proposed losses. On ScanNet-V1 [5] (out-of-domain), improvements are more modest than on RE10K [17] (in-domain), a trend also evident in the qualitative comparisons (Figure 5 and Figure 6); for example, observe the window rendering in the first row of Figure 5. The greatest improvement is observed on the out-of-domain ACID [11].

Additional geometry evaluation. In addition to the results in the main paper, we provide further qualitative comparisons of mesh reconstructions from two input views on the ScanNet-V1 [5] dataset. We compare pose-required baselines MVSplat [3] and DepthSplat [15], the pose-free NoPoSplat [16], and our method. Meshes are reconstructed by fusing *virtual* (novel-view) rendered depth maps via TSDF. For each method, we also visualize Gaussian-splat orientations (surface normals) for the first input view alongside the rendered depth for a novel view, and its ground-truth depth. Our approach consistently produces geometrically coherent meshes, whereas the other methods’ inconsistent novel-view depths lead to deformed planar regions (rows one and three) and large holes (row two), losing fine scene details.

We additionally visualize novel-view rendered depth maps from our method on RE10K [17] (Fig. 3) and on ScanNet [5] (Fig. 4).

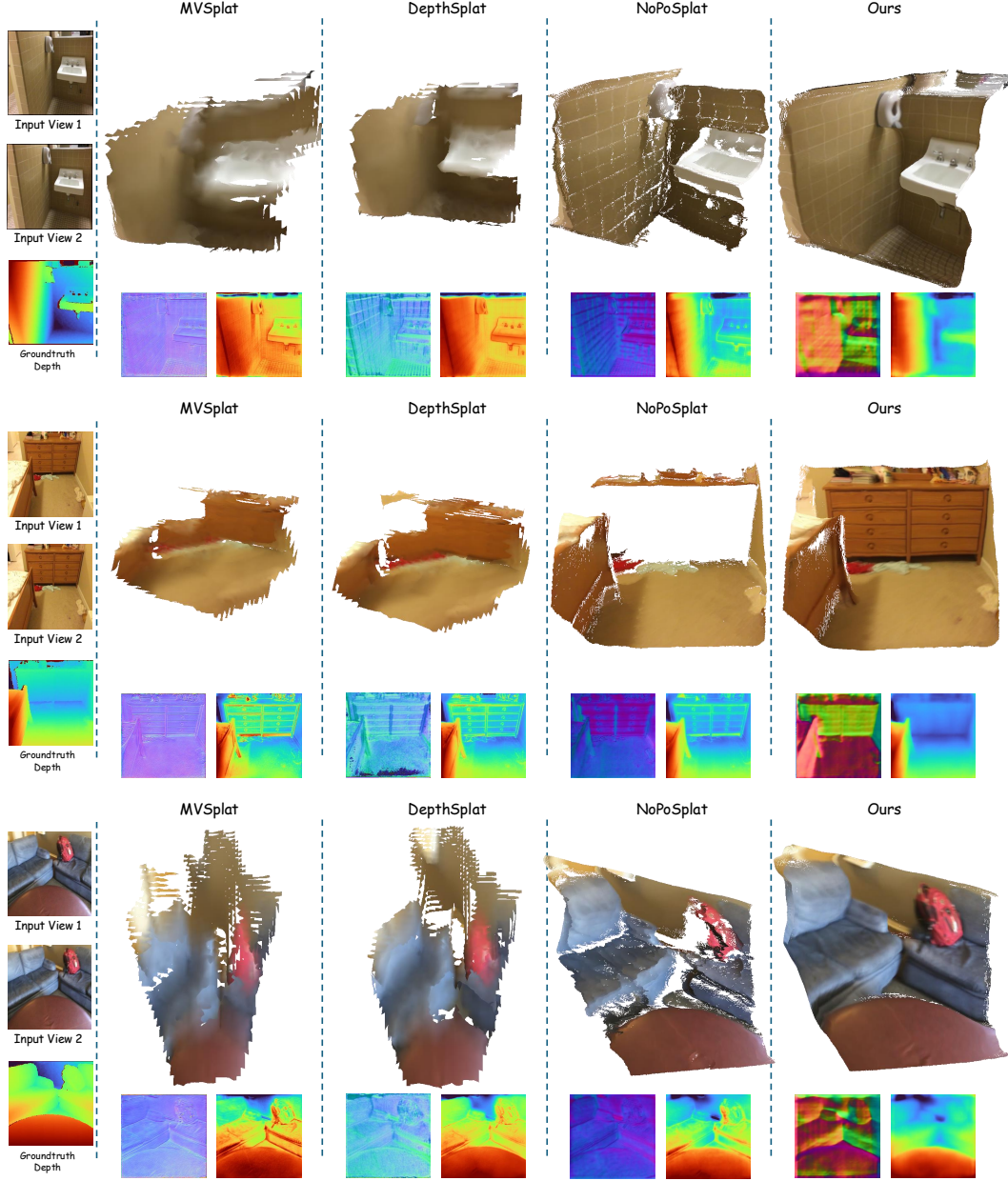
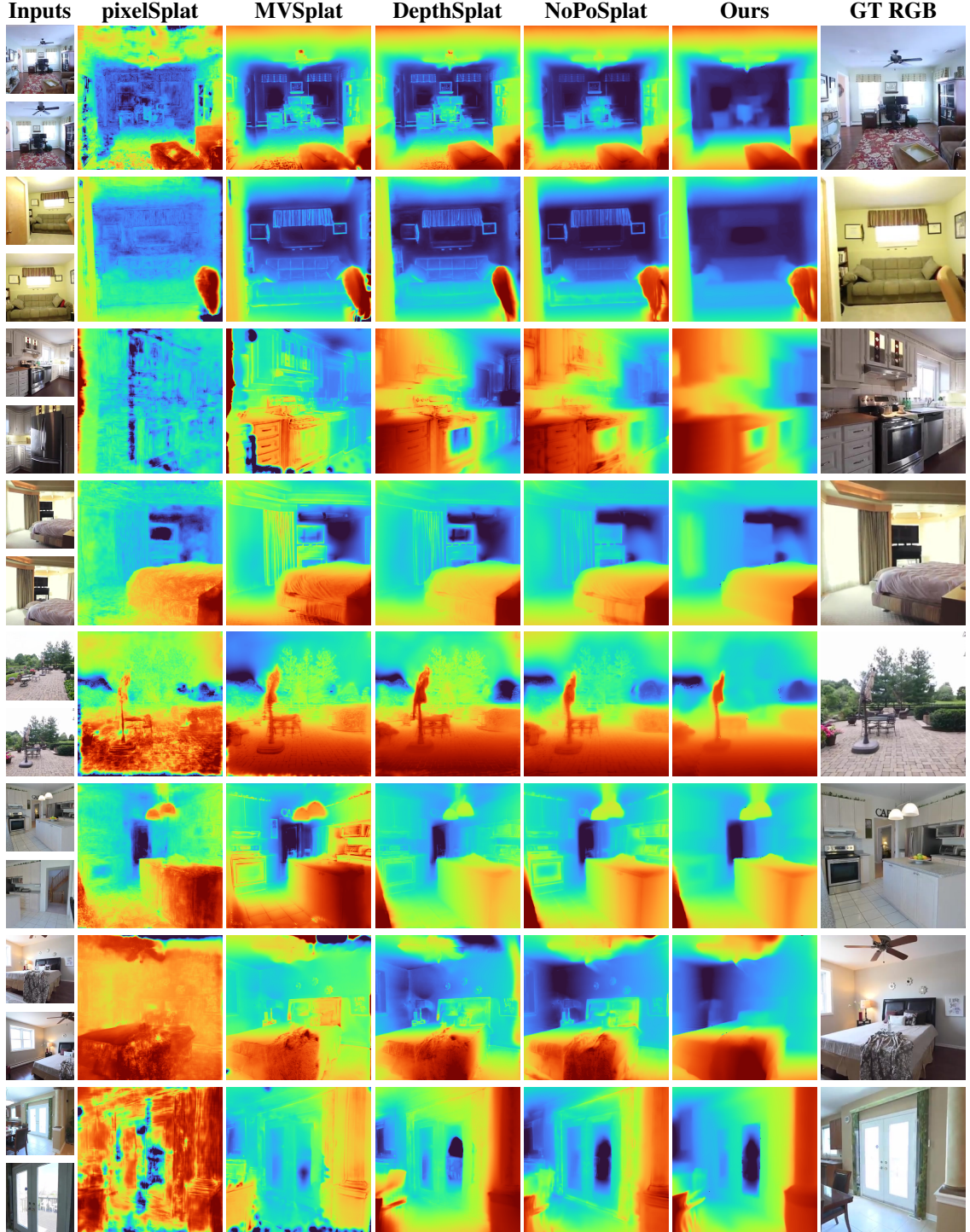


Figure 2: More qualitative comparison of mesh on ScanNet-V1 [5].

References

- [1] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Tat-Jun Chin, Chunhua Shen, and Ian Reid. Auto-rectify network for unsupervised indoor depth estimation. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):9802–9813, 2021. 4
- [2] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 4
- [3] 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. In *ECCV*, 2024. 4, 5
- [4] Shin-Fang Chng, Ravi Garg, Hemanth Saratchandran, and Simon Lucey. Invertible neural warp for nerf. In *ECCV*, 2024. 5

- 142 [5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner.
143 Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. [2](#), [3](#), [5](#), [6](#), [9](#), [11](#)
- 144 [6] Ravi Garg, Shin-Fang Chng, and Simon Lucey. Direct alignment for robust nerf learning. In *ACCV*, 2024.
145 [5](#)
- 146 [7] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised
147 monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*,
148 pages 3828–3838, 2019. [4](#)
- 149 [8] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for
150 geometrically accurate radiance fields. In *ACM SIGGRAPH 2024*, 2024. [2](#), [3](#)
- 151 [9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for
152 real-time radiance field rendering. *ACM TOG*, 2023. [2](#)
- 153 [10] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv*
154 *preprint arXiv:2406.09756*, 2024. [5](#)
- 155 [11] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa.
156 Infinite nature: Perpetual view generation of natural scenes from a single image. In *ICCV*, 2021. [1](#), [2](#), [3](#), [5](#)
- 157 [12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support
158 inference from rgb-d images. In *Computer Vision—ECCV 2012: 12th European Conference on Computer*
159 *Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. [4](#)
- 160 [13] Libo Sun, Jia-Wang Bian, Huangying Zhan, Wei Yin, Ian Reid, and Chunhua Shen. Sc-depthv3: Robust
161 self-supervised monocular depth estimation for dynamic scenes. *IEEE transactions on pattern analysis*
162 *and machine intelligence*, 46(1):497–508, 2023. [4](#)
- 163 [14] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric
164 3d vision made easy. In *CVPR*, 2024. [1](#), [4](#), [5](#)
- 165 [15] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc
166 Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *CVPR*, 2025. [4](#), [5](#)
- 167 [16] Botao Ye, Sifei Liu, Haofei Xu, Li Xueting, Marc Pollefeys, Ming-Hsuan Yang, and Peng Songyou. No
168 pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *ICLR*, 2025. [1](#),
169 [2](#), [4](#), [5](#)
- 170 [17] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification:
171 learning view synthesis using multiplane images. *ACM TOG*, 2018. [1](#), [2](#), [3](#), [4](#), [5](#), [8](#), [10](#)



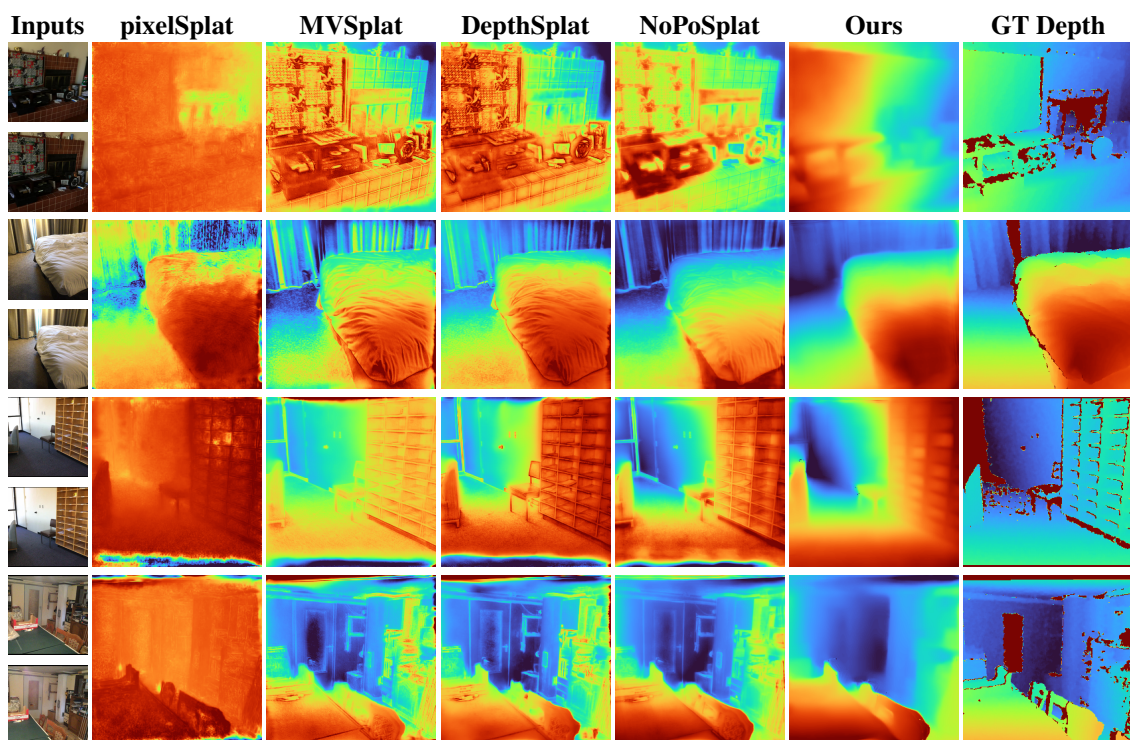


Figure 4: More qualitative comparison of novel-view rendered depth on ScanNet-V1 [5].

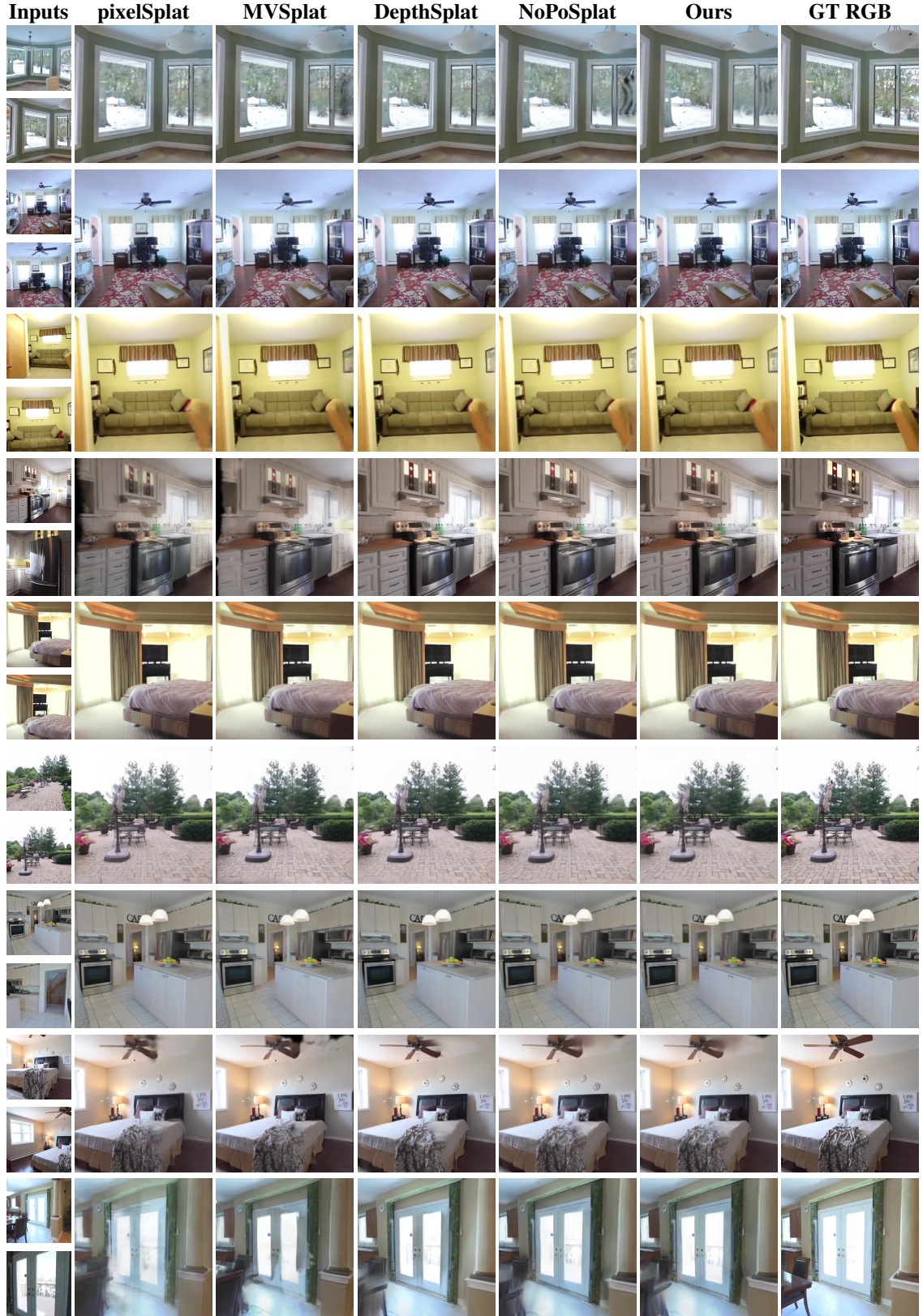


Figure 5: Qualitative comparison of rendered novel-view RGB on RE10k [17].

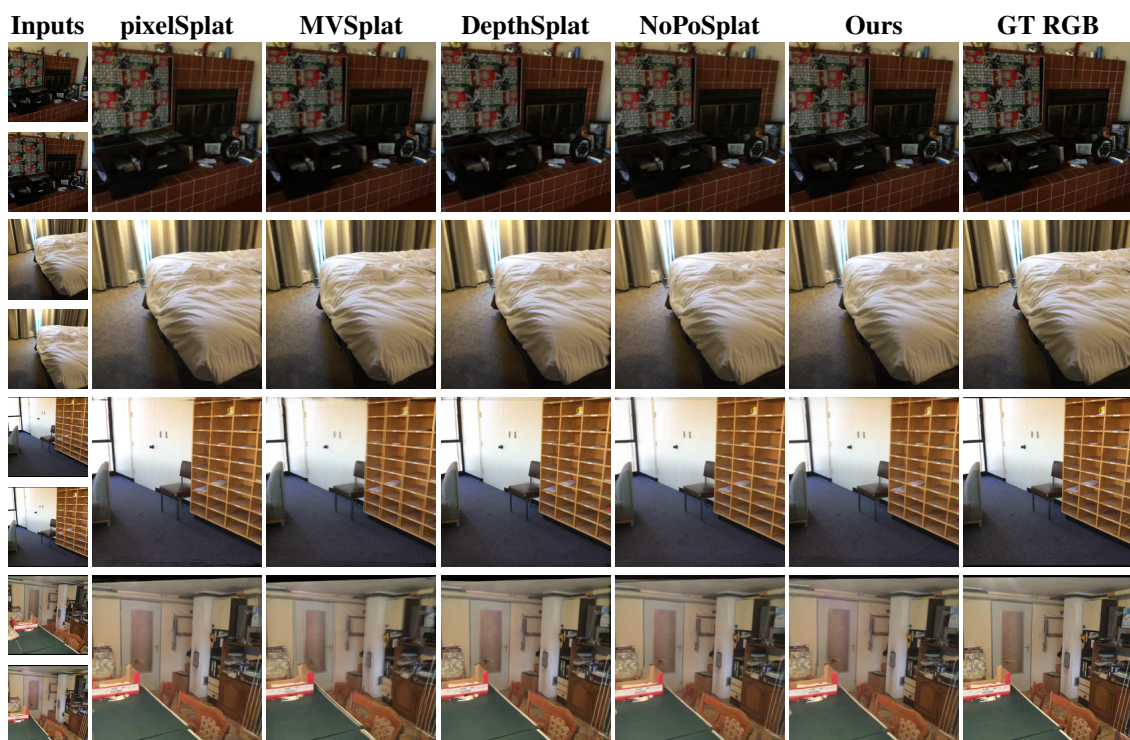


Figure 6: Qualitative comparison of rendered novel-view RGB on ScanNet-V1 [5].