AnchorSplat: Feed-Forward 3D Gaussian Splatting With 3D Geometric Priors

Supplementary Material

A. Appendix

This supplementary material provides the following additional information: (B) additional ablation experiments and results and (C) additional visualizations.

B. Ablation Study

B.1. Ablation Study on Input Information

We perform an ablation study on the ScanNet++ V2 dataset to investigate how different input modalities influence the performance of the GS decoder. The following configurations are examined: (i) RGB-only, (ii) RGB with camera ray embeddings, (iii) RGB with depth, and (iv) RGB supplemented with both camera information and depth. To isolate the effect of input modalities, this study evaluates the Gaussian decoder alone without any refinement, and thus only **Ours (w/o Refiner)** is reported. As summarized in Table. 1, incorporating additional geometric and camera-related cues leads to progressively better reconstruction quality, demonstrating the importance of informed inputs for effective Gaussian decoding.

Table 1. Ablation study on input information for the Gaussian decoder. This experiment presents the evaluation results on the novel 4 views using ScanNet++ V2 with 32 input views. In this table, RGB denotes using only RGB images as input; Depth indicates the use of depth maps; CamRay represents Plücker ray embeddings derived from camera intrinsics and extrinsics.

Setting	RGB			Depth	
	PSNR↑	SSIM↑	LPIPS↓	δ_1 ↑	AbsRel↓
RGB	20.53	0.78	0.46	0.92	0.076
RGB+Depth	20.67	0.78	0.48	0.94	0.057
RGB+CamRay	20.34	0.78	0.48	0.92	0.080
RGB+CamRay+Depth	20.96	0.78	0.47	0.94	0.068

B.2. Ablation Study on number of Gaussians

We further conduct an ablation study to analyze the effect of the number of Gaussians predicted per anchor in the Gaussian decoder on the ScanNet++ V2 dataset. Specifically, we vary the Gaussian count per anchor across 1, 2, 4, 8, 16 and evaluate how this design choice influences reconstruction fidelity and geometric accuracy, as shown in Table. 2. To isolate the capacity of the decoder itself, this experiment evaluates only **Ours (w/o Refiner)**, without applying the Gaussian refiner.

The results show that the number of Gaussians per anchor plays a critical role in balancing expressiveness and stability. Very small configurations lack sufficient representational capacity, leading to incomplete or under-detailed geometry. In contrast, overly large configurations yield only marginal accuracy gains while incurring significantly higher computational cost. In practice, we therefore adopt 4 Gaussians per anchor as the default configuration, as it achieves the best overall performance while keeping computational cost low. This ablation study is conducted using a smaller batch size (bs = 32) while keeping the same number of training iterations (2.5k) as in the main experiments. As a result, the absolute performance values may differ from those in the main paper, but the overall trend regarding Gaussian multiplicity remains consistent. Moreover, when incorporating our Gaussian Refiner, the reconstruction quality surpasses even the decoder-only configuration with 16 Gaussians per anchor, further demonstrating the effectiveness of the refinement module.

Table 2. Ablation on the number of Gaussians per anchor. We evaluate the effect of varying the Gaussian multiplicity in the Gaussian decoder on the ScanNet++ V2 dataset. All results are obtained using the decoder only (w/o Refiner). Using too few Gaussians restricts representational capacity, while very large counts yield diminishing returns with substantially higher computational cost. A moderate setting of 4 Gaussians per anchor achieves the best overall balance between accuracy and efficiency.

Setting	RGB			Depth		numGS
	PSNR↑	SSIM↑	LPIPS↓	δ_1 ↑	AbsRel↓	
numGS=1	20.14	0.78	0.50	0.93	0.077	61,788
numGS=2	20.61	0.79	0.48	0.93	0.075	123,577
numGS=4	20.66	0.79	0.48	0.93	0.076	247153
numGS=8	20.46	0.78	0.49	0.93	0.073	494,307
numGS=16	20.79	0.79	0.49	0.94	0.072	988,613

B.3. Ablation Study on Variant Datasets

We also conducted ablation experiments on additional datasets, including indoor datasets (Replica) and the outdoor dataset Tanks and Temples (T&T), as shown in Table 3. Here, AnchorSplat* denotes AnchorSplat without the Refiner. The results demonstrate that our method achieves superior RGB and depth rendering quality, as well as higher reconstruction efficiency, compared to AnySplat. These results further validate the generalization ability and robustness of our approach across diverse scenes.

Table 3. Comparison of AnchorSplat variants on different datasets (32 sampled views / 4 novel views). Experiments are conducted on indoor datasets (ARkitScenes and Replica) and an outdoor dataset (T&T), using the same number of training steps, learning rate, and backbone settings. AnchorSplat* denotes AnchorSplat without the Refiner. Performance is evaluated in terms of RGB and depth rendering quality as well as reconstruction efficiency.

Dataset	Method	RGB			Depth		NumGS	ReconTime(s)
		PSNR↑	SSIM↑	LPIPS↓	δ_1 ↑	AbsRel↓		
ARKitScenes	AnySplat	21.35	0.75	0.30	0.88	0.12	3,237,113	6.63
	AnchorSplat*	21.00	0.77	0.41	0.96	0.06	400,000	2.23
Replica	AnySplat	21.19	0.70	0.32	0.86	0.11	5,740,107	4.72
	AnchorSplat*	23.48	0.78	0.31	0.95	0.066	800,000	2.40
T&T	AnySplat	15.30	0.45	0.46	0.44	0.39	3,952,092	6.74
	AnchorSplat*	16.53	0.57	0.59	0.75	0.23	800,000	1.98

B.4. Ablation Study on Aggregation Method

For each anchor, multiple projected image features may be obtained from different views that map to the same anchor. We aggregate these features into a single C-dimensional anchor feature by applying a pooling operation over all valid projections that pass the visibility and depth-consistency checks. We conducted an early-stage ablation study to compare different aggregation strategies, including average pooling, max pooling, and a FIFO (first in fist out) selection baseline. We first visualized the aggregated features using PCA, as shown in Fig. 1, and then evaluated the three pooling methods quantitatively. The results indicate PSNR values of 20.96, 20.81, and 20.28 for average, max, and FIFO pooling, respectively. Based on these results, we adopt average pooling as the default aggregation method.

For each anchor, we may obtain multiple projected image features (from different views mapping to the same anchor). We aggregate these features into a single C-dimensional anchor feature by applying a pooling operation over all valid projected features associated with that anchor, and we only pool over projections that pass the visibility/depth-consistency checks. We conducted an early-stage ablation comparing different aggregation strategies, including average pooling, max pooling, and a FIFO/first-in selection baseline. We initially performe PCA visualization shown as Fig. 1 and experiments on features aggregated using three pooling methods (average, max, FIFO). The results showed PSNR values of 20.96, 20.81, and 20.28 for average, max, and FIFO pooling, respectively. Therefore, we utilize the average pooling as default method.

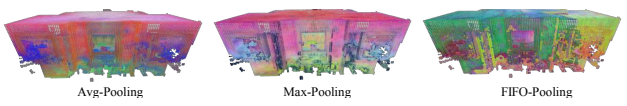

Figure 1. PCA visualization of three feature aggregations.

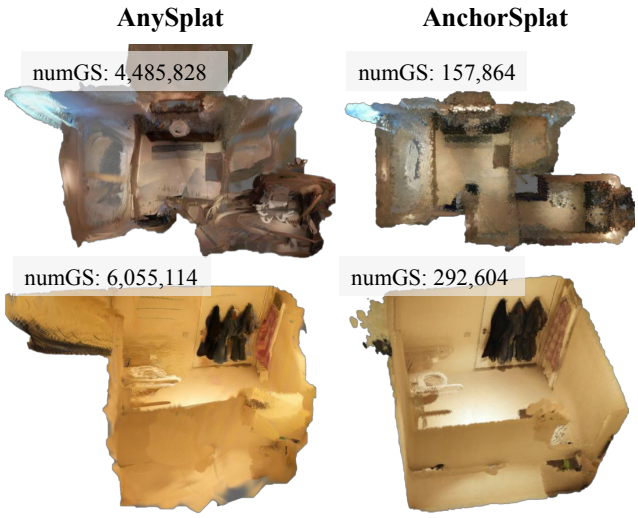

Figure 2. Comparison of reconstructed Gaussians between AnySplat and AnchorSplat

C. Ablation Study on Backbones

In the AnchorPredictor module, we use MapAnything as the default backbone for depth prediction. This backbone estimates the depth at each anchor location, providing a strong geometric prior for the subsequent AnchorSplat reconstruction. To evaluate the impact of the backbone choice, we compare MapAnything with DA3 as the AnchorPredictor backbone on the Replica dataset. As shown in Table 4, both backbones achieve comparable performance given the same number of training steps, demonstrating that MapAnything serves as an effective and reliable default for depth prediction.

Table 4. Ablation study of AnchorPredictor backbones on the Replica dataset (32 sampled views / 32 novel views). We compare MapAnything and DA3 as backbones under the same training configurations, including identical number of training steps, learning rate, and network settings. Performance is measured in terms of RGB and depth rendering quality as well as reconstruction efficiency.

Backbone	RGB			Depth		NumGS	ReconTime(s)
	PSNR↑	SSIM↑	LPIPS↓	δ_1 ↑	AbsRel↓		
MapAnything	22.41	0.79	0.33	0.92	0.084	800,000	1.89
DA3	22.04	0.75	0.35	0.91	0.085	800,000	1.72

D. Additional Visualizations

When converting the depth maps produced by the MVS estimator into 3D points via back-projection, we observe that some predicted depths are unreliable, resulting in outlier 3D points such as flying points, points lying far behind actual

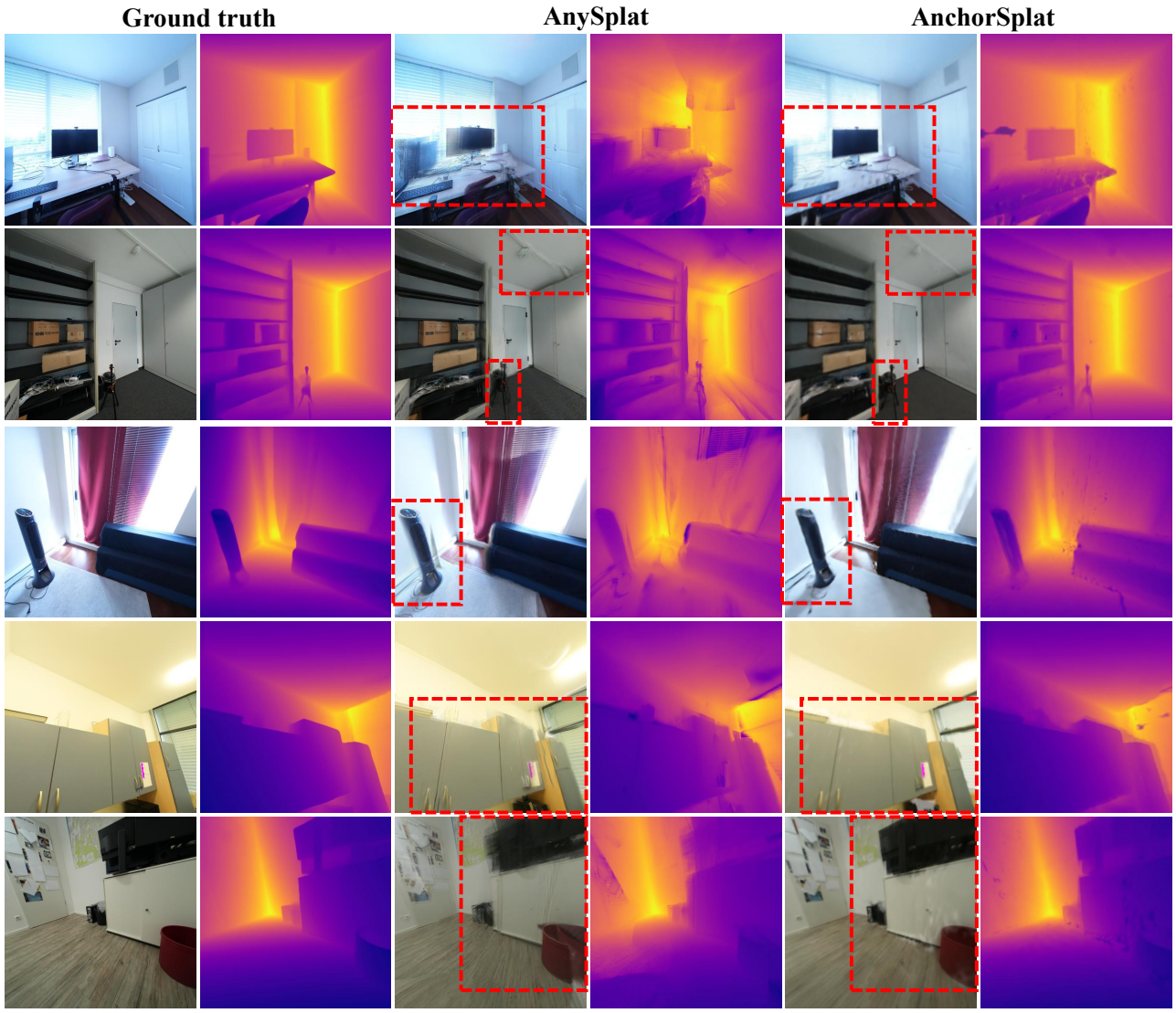

Figure 3. Comparison of rendered RGB images and depth images between AnySplat and AnchorSplat

surfaces, or points drifting outside the valid scene region. To address these artifacts, we apply a 3D clipping operation that restricts all back-projected points to a predefined spatial boundary. As illustrated in Fig. 2, this clipping step effectively removes extreme outliers, stabilizes the initial geometry, and ensures that the Gaussian initialization remains structurally valid without being affected by large-magnitude depth errors.

In addition, we provide extended visual comparisons against AnySplat to more clearly demonstrate the advantages of our anchor-aligned representation. As shown in Fig. 3, both the RGB and depth rendering results reveal that our method consistently produces sharper appearance details and significantly cleaner geometry, free from the floaters, ghosting artifacts, and structural distortions commonly observed in voxel-aligned approaches. The depth visualizations are particularly indicative of this improvement: our predictions preserve crisp geometric boundaries and exhibit strong view-consistency, suggesting that the underlying 3D structure is substantially more accurate and stable.

These observations are further corroborated by the reconstructed Gaussian visualizations in Fig. 2. Our method generates compact, well-structured, and spatially coherent Gaussian distributions, whereas AnySplat tends to produce fragmented, noisy, and geometrically inconsistent Gaussians. Together, these comparisons highlight that AnchorSplat not only improves rendering quality but also delivers a more faithful and robust 3D representation.