TWINGS: Thin Plate Splines Warp-aligned Initialization for Sparse-View Gaussian Splatting

Supplementary Material

A. Time Complexity

All preprocessing timings were measured on a local workstation equipped with an NVIDIA GeForce RTX 3060 GPU and an Intel Core i5-12400 CPU, while 3DGS training was performed on a server equipped with an NVIDIA RTX 6000 Ada GPU and an AMD EPYC 7763 CPU. Table 1 summarizes the runtime of each stage in the preprocessing pipeline. The total preprocessing time is 12.45 s for DTU (3-view), 17.74 s for LLFF (3-view), and 94.50 s for Mip-NeRF360 (12-view). In comparison, our 3DGS training takes 12.75 min on DTU (3-view), 12.77 min on LLFF (3-view), and 14.40 min on Mip-NeRF360 (12-view). The preprocessing time increases roughly linearly with the number of input views, primarily due to multi-view correspondence establishment and TPS deformation. Even with 12 views, preprocessing completes in under 1.6 min, which remains negligible compared to the 3DGS training time. For the challenging 3-view settings, preprocessing accounts for only about 1–2% of the total training time, demonstrating the efficiency and practicality of our pipeline for real-world applications.

Table 1. Runtime of each stage in the preprocessing pipeline across datasets and various view settings.

Stage	**Processing Time (sec)**		
	DTU (3-view)	**LLFF (3-view)**	**Mip-NeRF360 (12-view)**
Multi-view correspondences	4.78	5.96	41.69
Multi-view triangulation	0.97	1.31	5.28
TPS deformation	6.65	10.43	47.36
CBPS	0.05	0.04	0.17
Total	**12.45**	**17.74**	**94.50**

We report the training time of GS-variants with 3-/6-/9-view on the DTU dataset, using the same PCL generated by TWINGS-Init. As reported in Table 2, our method remains highly efficient as the number of views increases, showing a favorable scalability.

Table 2. Training time of GS-variants on the DTU dataset under different training view settings.

Training Time (min)	3-view	6-view	9-view
FSGS	35.48	35.86	32.01
CoR-GS	9.57	10.45	11.35
TWINGS (Ours)	12.74	12.81	13.22

We report the computation time of TWINGS-Init across 3-/6-/9-view on the LLFF and DTU datasets. As illustrated in Fig. 1, the runtime scales approximately linearly with the number of input views, indicating that TWINGS-Init remains computationally efficient as the view count increases.

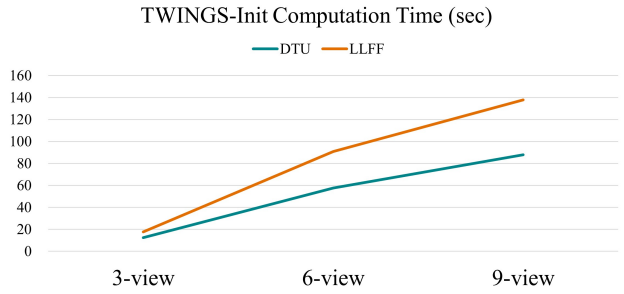

Figure 1. Computation time of TWINGS-Init on the LLFF and DTU datasets. The runtime grows approximately linearly with the number of input views, demonstrating the favorable scalability and efficiency of TWINGS-Init.

B. Comparison with modern dense matchers

Our contribution is to demonstrate how geometrically accurate initialization affects the optimization dynamics of 3DGS under sparse-view supervision. We observe that points triangulated from dense-matcher-derived matches tend to concentrate in foreground regions with high-confidence, while background regions remain sparsely covered. To address this imbalance, we use dense depth via LS or FFD to cover background regions, which results in suboptimal 3DGS performance as shown in Tab. 6 of the manuscript. This observation motivates our TPS with CBPS, which preserves reliable foreground geometry while additionally enabling the sampling of geometrically consistent points in background regions, even when only a sparse number of background matches are available. We find that such balanced coverage of both foreground and background regions is particularly effective in constraining Gaussian optimization in sparse-view scenarios.

Table 3. Comparison with modern dense matchers.

Initial PCL	**DTU (3-view)**		**LLFF (3-view)**		**MipNeRF-360 (12-view)**	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
MASt3R	21.05	0.853	20.26	0.702	19.78	0.573
TWINGS-Init	**21.52**	**0.880**	**21.49**	**0.754**	**20.35**	**0.618**

We compare TWINGS-Init with point clouds triangulated from MASt3R [4] matches on three benchmark

datasets. We evaluate TWINGS with an identical training pipeline, varying only the initial PCL used for Gaussian initialization. As reported in Table 3, using TWINGS-Init consistently yields superior novel view synthesis results compared to modern dense matchers in sparse-view scenarios.

C. Different matching algorithms

We ablate the correspondence module in our pipeline by replacing MASt3R [4] with SIFT [6] as used in COLMAP to assess how sensitive the pipeline is to the quality of feature correspondences. As illustrated in Fig. 2, the SIFT replacement yields sparser and less reliable matches. When a sufficient number of reliable correspondences are established, our method still reconstructs a coarse geometry (CBP). However, when the matches are few or unreliable, the geometry tends to be slightly distorted. In *Flower* scene, the petal shapes appear elongated, and in *Fortress* scene, the frontal region of the fortress is reconstructed with an unnaturally sharp protrusion compared to the ground truth images.

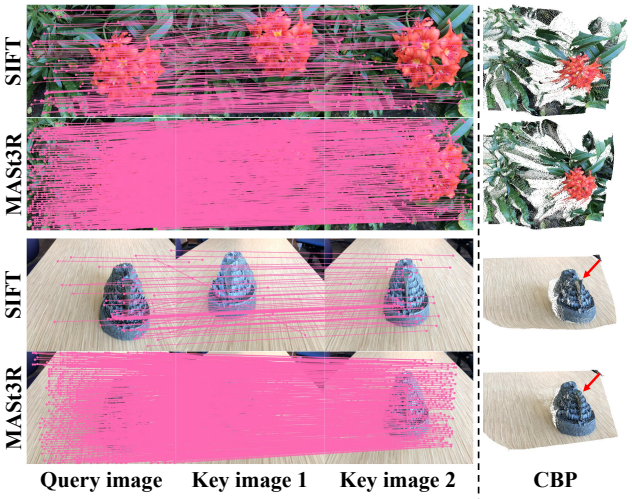

Figure 2. Visualization of CBP with different matching algorithms. The leftmost column shows the query image, the second and third columns show the key images, and the rightmost column shows the CBP of the query view to which TPS deformation is applied, shown from viewpoints that highlight geometric details.

Table 4. Ablation study on the correspondence module on the LLFF dataset (3-view). Improved correspondence quality in our pipeline results in better rendering performance.

Image Matcher	LLFF (3-view)		
	PSNR ↑	SSIM ↑	LPIPS↓
SIFT	21.01	0.732	0.187
MASt3R	**21.49**	**0.754**	**0.167**

In contrast, MASt3R produces denser and more robust correspondences across wide viewpoint changes, thereby enabling TPS model definition to be better established and yielding more accurate 3D geometry. The resulting gains in rendering quality are reported in Table 4.

D. Discussion on the LLFF dataset

To further analyze the diminishing effect of our method on the LLFF dataset with increasing views (Sec. 5.3, "Impact of Sampling Radius"), we introduce a multi-view score inspired by the covisibility map formulation of CoMapGS [3]. Specifically, we compute the proportion of pixels whose covisibility count exceeds two (i.e., observed by at least three views), which measures the extent of reliable multi-view correspondences within the dataset. A higher multi-view score indicates that a greater portion of pixels are consistently observed across multiple views, thereby providing stronger geometric constraints during SfM.

Table 5. Multi-view scores on the benchmark datasets. As the number of views increases, the LLFF dataset exhibits a substantially higher multi-view scores, reflecting its forward-facing camera configuration and high inter-view overlap.

Dataset	Multi-view scores				
	3-view	6-view	9-view	12-view	24-view
LLFF	0.512	**0.828**	**0.906**	-	-
DTU	0.329	0.674	0.784	-	-
MipNeRF-360	-	-	-	0.529	0.765

Unlike DTU and MipNeRF-360 datasets, where camera poses exhibit wider baselines or full 360-degree distributions, the LLFF dataset follows a forward-facing configuration in which cameras are concentrated along a limited arc in front of the scene. This geometric property significantly affects the covisibility pattern, as even modest increases in the number of training views yield a large boost in covisible regions, as adjacent views share a high degree of pixel overlap. Consequently, the multi-view scores of the LLFF dataset grows more rapidly compared to other datasets, as shown in Table 5. When the view count increases in the LLFF dataset, the densified COLMAP point cloud becomes not only denser but also more reliable due to increased multi-view agreement. In this regime, the sampling radius r in CBPS plays a lesser role, as the initial geometry already benefits from robust correspondences across views. In contrast, at very sparse settings (3-view), the lack of overlapping observations produces fewer covisible pixels, and thus a larger sampling radius is required to compensate for missing geometric cues. Our method particularly benefits from such low-covisibility conditions, providing a dense and accurate initialization that greatly stabilizes 3DGS optimization.

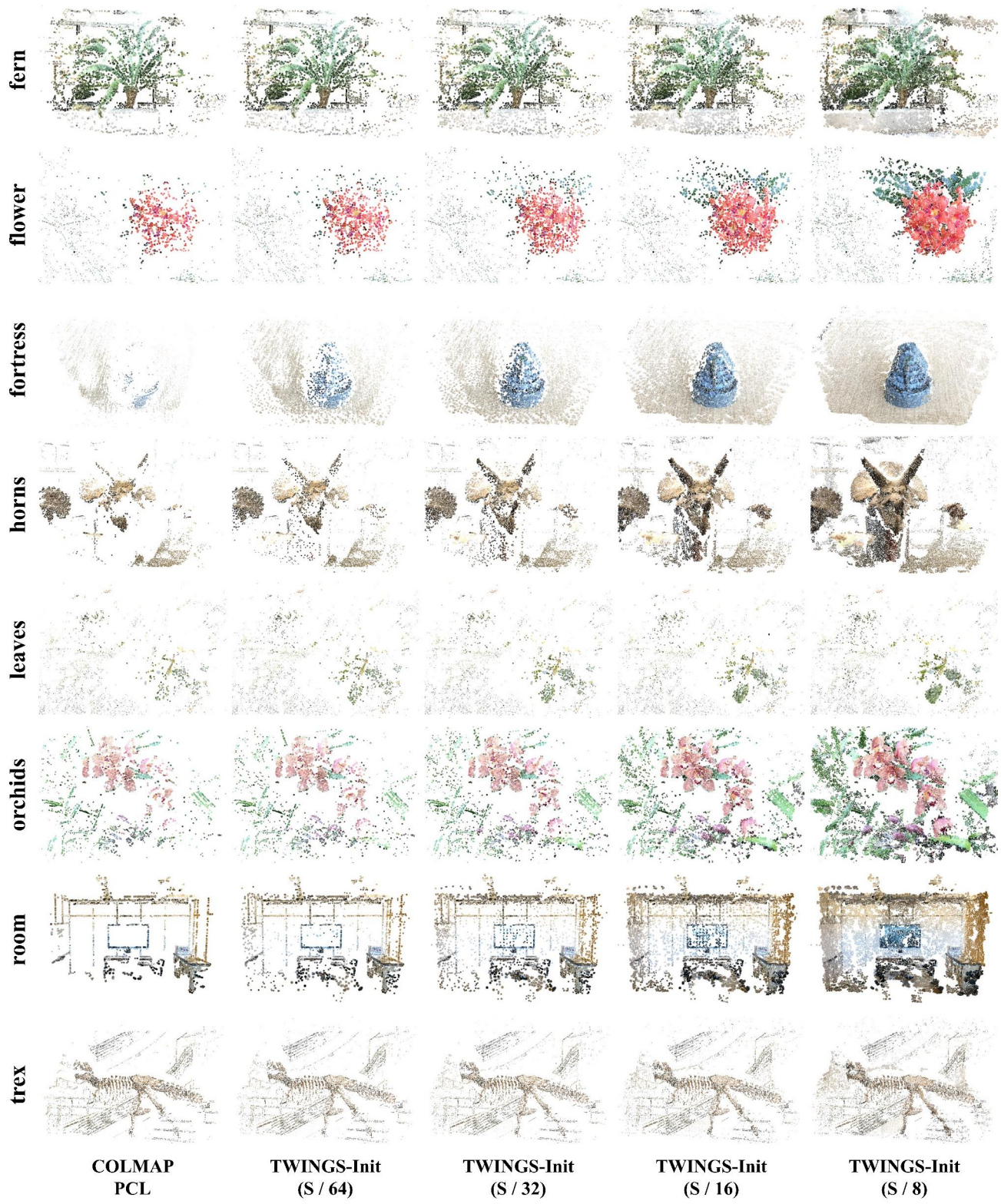

Figure 3. Point cloud comparison with varying CBPS sampling distances on the LLFF dataset. For each scene, given 3 training views, the left-most column shows the COLMAP PCL, while the subsequent columns demonstrate the results from TWINGS-Init using different sampling distances.

E. Visualization of TWINGS-Init

We compare the COLMAP PCL and the point clouds from TWINGS-Init under different CBPS sampling distances. Results on the LLFF dataset with 3 training views are shown in Fig. 3. TWINGS-Init yields richer geometric detail and increased point concentration around scene structures. Reducing the sampling distance further concentrates samples near reliable control points, enhancing local fidelity and suppressing spurious samples.

F. Additional Results

We provide additional rendering results. The examples on the DTU and LLFF datasets with 3 training views and the Mip-NeRF360 dataset with 12 training views are shown in Fig. 6, Fig. 7, and Fig. 8, respectively. We additionally report comparisons against NexusGS [9] and Binocular3DGS [2] using their publicly available code under the same experimental setting. The metrics and rendered results are summarized in Table 6 and Fig. 4, where our method consistently achieves higher PSNR/SSIM and sharper, more faithful reconstructions.

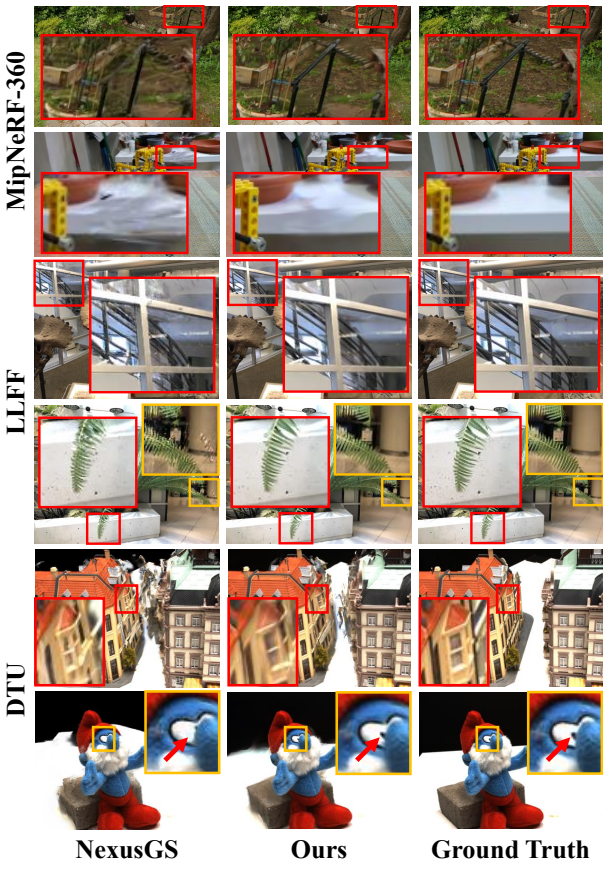

Figure 4. Qualitative comparison on the benchmark datasets. Novel view synthesis results rendered by NexusGS, our approach, and ground truth for comparison.

Table 6. Quantitative comparison of novel view synthesis results on the benchmark datasets.

Method	DTU (3-view)		LLFF (3-view)		MipNeRF-360 (24-view)	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
FSGS	17.24	0.824	20.43	0.682	23.28	0.715
CoR-GS	19.21	0.853	20.45	0.712	23.39	0.727
DropGaussian	18.41	0.849	20.76	0.713	24.13	0.762
NexusGS	20.21	0.869	21.07	0.738	23.86	0.753
Binocular3DGS	20.71	0.862	21.44	0.751	22.26	0.700
TWINGS (Ours)	**21.52**	**0.880**	**21.49**	**0.754**	**24.17**	**0.762**

We additionally compare our method with recent sparse-view initialization methods. We compare our method with Dust-GS [1] and SPARS3R [7] on the same data split. As reported in Table 7, our method consistently achieves higher PSNR/SSIM on the MipNeRF-360 dataset. As illustrated in Fig. 5, SPARS3R exhibits noticeable surface irregularities on planar regions such as walls and floors. We attribute this to residual geometric inconsistencies in its aligned PCL from MASt3R, leading to surface artifacts.

Table 7. Comparisons with dense-initialization methods.

MipNeRF-360 (12-view)	Dust-GS	SPARS3R	TWINGS (Ours)
PSNR↑ / SSIM↑	18.75 / 0.552	19.29 / 0.578	20.35 / 0.618

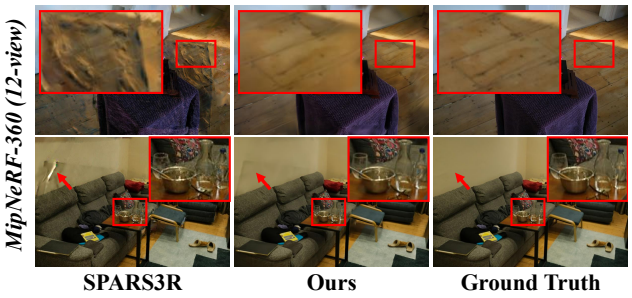

Figure 5. Qualitative comparison with SPARS3R.

G. Implementation Details

Thin Plate Splines. TPS is applied once to each training viewpoint, aligning backprojected points with reconstructed 3D points. This ensures that geometric information from all training viewpoints is incorporated, enhancing scene coverage.

Details of Point Sampling. To enhance the initialization of 3D Gaussian primitives, we combine the sparse 3D reconstruction produced by COLMAP with calibrated back-projected points (CBP). We refine this combined point set by removing points within a margin of 0.05 from the SfM reconstruction using a KD-tree to avoid overlap and applying radius-based clustering with a radius of 0.01 to reduce redundancy. For efficiency we randomly downsample the back-projected points to 30,000, a step that empirically

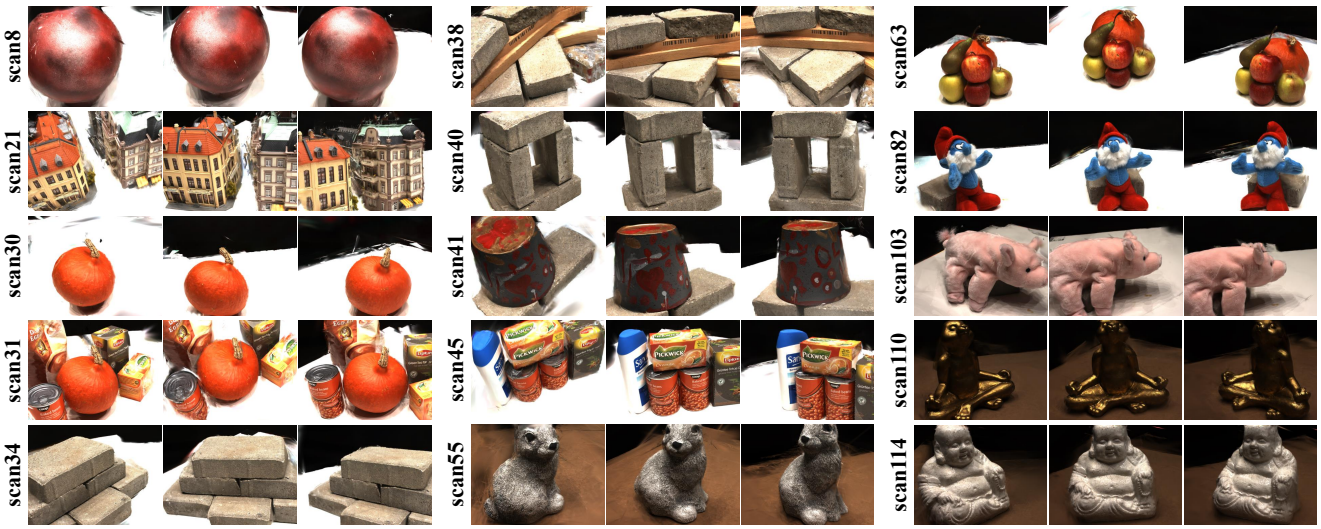

Figure 6. Examples of the rendered novel view results from TWINGS with 3 training views on the DTU dataset.

Figure 7. Examples of the rendered novel view results from TWINGS with 3 training views on the LLFF dataset.

causes no noticeable difference in the final geometric representation while improving computational efficiency.

View Selection. We select key images for each query view using a k-nearest-neighbor search in spherical coordinates. Each camera center in world coordinates is $\mathbf{c}_i = (x_i, y_i, z_i)$ and we define $\alpha_i = \text{atan2}(y_i, x_i)$ for azimuth, $\varepsilon_i = \arcsin(z_i/r_i)$ for elevation, and $r_i = \|\mathbf{c}_i\|_2$ for the distance from the world origin. For a query view i, the distance to every training view j is as follows:

$$d_{ij} = \sqrt{(\alpha_i - \alpha_j)^2 + (\varepsilon_i - \varepsilon_j)^2 + (r_i - r_j)^2}. \quad (1)$$

We compute d_{ij} to all training views and select the k nearest neighbors as key images, with $k = 2$ for the 3-view setting and $k = 4$ for the 6-, 9-, 12-, and 24-view settings, giving a total of three or five images for multi-view triangulation. This distance accounts for both viewing-direction differences and the radial distance of each camera center. Views that share a similar orientation but lie much farther or closer to the world origin than the query view, or that are at a similar radius but face a very different direction, result in larger d_{ij} and are less likely to be selected. The method is particularly effective in sparse-view settings where each

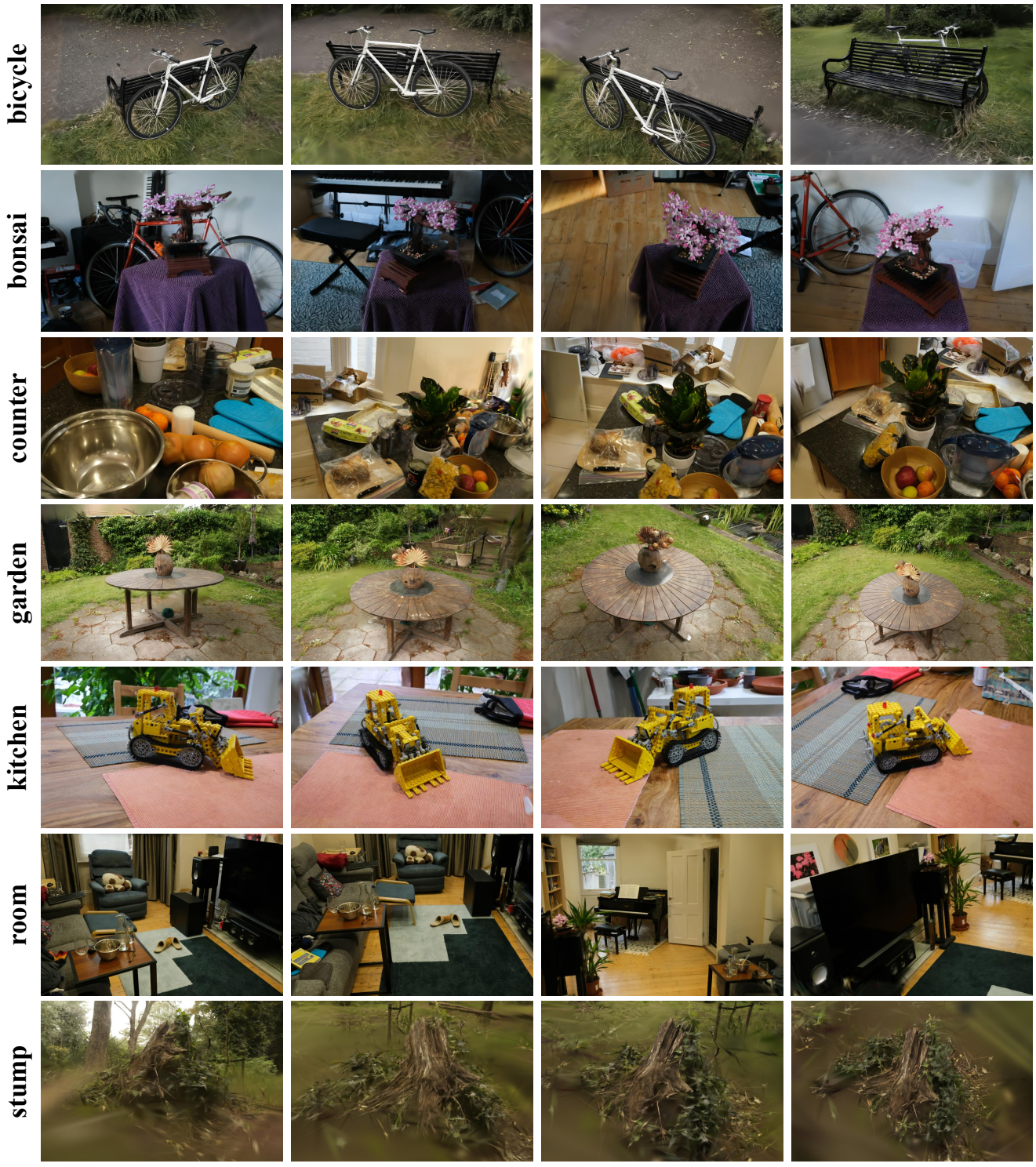

Figure 8. Examples of the rendered novel view results from TWINGS with 12 training views on the Mip-NeRF360 dataset.

selected neighbor must provide strong image overlap and a well-conditioned baseline for reliable multi-view reconstruction.

Impact of View Selection Methods. We perform an ablation on view selection strategies for multi-view triangulation on the Mip-NeRF360 dataset. As shown in Ta-

ble 8, nearest view selection yields consistently better performance than random selection. Nearest neighbors ensure sufficient overlap and moderate parallax, which provide denser and more reliable correspondences for triangulation. In contrast, random selection often introduces wide baselines or large angular differences, leading to fewer valid matches and noisier geometry that degrades subsequent 3DGS optimization. We experimentally find that nearest view selection produces more stable triangulation in sparse-view settings, and adopt this strategy for all experiments in this work.

Table 8. Ablation on view selection for multi-view triangulation on Mip-NeRF360 (12-view). Comparison of novel view synthesis results across different view selection strategies.

View Selection	PSNR↑	SSIM↑	LPIPS↓
Random	20.00	0.605	0.380
Nearest	**20.35**	**0.617**	**0.368**

Camera Poses. Following existing works [5, 8], we assume that all camera poses are known. In practice, for the DTU, LLFF, and Mip-NeRF360 datasets, we utilize the poses provided with the datasets.

Training. We implemented TWINGS using the Pytorch framework. During optimization, we densify the Gaussians every 100 iterations, starting densification after 500 iterations. The total optimization process involves 10,000 steps. Following FSGS, pseudo views are enabled after 2,000 iterations. During training, we initialize the spherical harmonics (SH) degree at 0 and increment it by 1 every 1000 iterations, up to a maximum degree of 3, progressively refining the lighting representation. Simultaneously, opacity values are reset every 3000 iterations to eliminate low-opacity floaters by clamping all opacities to a maximum value of 0.05 using an inverse sigmoid function. The learning rates for position, SH coefficients, opacity, scaling, and rotation are set to 0.00016, 0.0025, 0.05, 0.005, and 0.001, respectively.

References

[1] Shen Chen, Jiale Zhou, and Lei Li. Dense point clouds matter: Dust-gs for scene reconstruction from sparse viewpoints. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 4

[2] Liang Han, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Binocular-guided 3d gaussian splatting with view consistency for sparse view synthesis. *Advances in Neural Information Processing Systems*, 37:68595–68621, 2024. 4

[3] Youngkyoon Jang and Eduardo Pérez-Pellitero. Comapgs: Covisibility map-based gaussian splatting for sparse novel view synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26779–26788, 2025. 2

[4] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 1, 2

[5] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 7

[6] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[7] Yutao Tang, Yuxiang Guo, Deming Li, and Cheng Peng. Spars3r: Semantic prior alignment and regularization for sparse 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26810–26821, 2025. 4

[8] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9065–9076, 2023. 7

[9] Yulong Zheng, Zicheng Jiang, Shengfeng He, Yandu Sun, Junyu Dong, Huaidong Zhang, and Yong Du. Nexusgs: Sparse view synthesis with epipolar depth priors in 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26800–26809, 2025. 4