DroneSplat: 3D Gaussian Splatting for Robust 3D Reconstruction from In-the-Wild Drone Imagery

Supplementary Material

6. Implementation Details

6.1. Datasets

DroneSplat Dataset. DroneSplat Dataset is acquired with a DJI Mavic Pro 3 drone. The drone-captured images have two resolutions: 1920 × 1080 and 3840 × 2160. The dataset contains 24 in-the-wild drone-captured sequences, encompassing both dynamic and static scenes. The dynamic scenes feature a diverse range of moving objects, including cars, trucks, tricycles, pedestrians, strollers, wind-blown flags, etc. Furthermore, dynamic scenes are categorized into three levels based on the number of dynamic objects in the training set: "low dynamic" indicates scenes with only 4-10 dynamic objects, "medium dynamic" includes 10-50 dynamic objects, and "high dynamic" refers to scenes containing more than 50 dynamic objects.

Although previous 3D reconstruction datasets also have drone-captured scenes, such as the *drone* in NeRF On-the-go, their test images still contain dynamic distractors, which can introduce ambiguity in metric evaluation. In contrast, the test images in DroneSplat dataset's dynamic scenes feature only static elements, enabling a more rigorous and precise evaluation of our method and the baselines (Figure 13).

The dataset will be released soon on our project page.

NeRF On-the-go Dataset [29] The NeRF On-the-go dataset comprises 12 casually captured sequences, featuring 10 outdoor scenes and 2 indoor scenes, with occlusion ratios ranging from 5% to over 30%. The images in this dataset are available in two resolutions: most are 4032 × 3024 and a few are 1920 × 1080. To accelerate model training, NeRF On-the-go downsamples these resolutions by a factor of eight and four, respectively, in its experiments.

In the NeRF On-the-go dataset, dynamic distractors are typically limited in number (usually the data collector's companions) but may occupy a significant portion of the image. In contrast, most scenes in the DroneSplat dataset contain numerous dynamic distractors (like cars on the road), but the proportion of dynamic distractors in the image is relatively small. To validate the effectiveness and robustness of our Adaptive Local-Global Masking method on real-world data with diverse characteristics, we perform dynamic distractor elimination on six scenes from this dataset.

6.2. Adaptive Local-Global Masking.

Adaptive Local Masking. The role of Adaptive Local Masking is to identify dynamic distractors in the training image I_t at the current iteration t. Since the residuals across different scenes and iterations can vary significantly, we adaptively adjust the masking threshold based on the current Object-wise Average of Normalized Residuals and statistical approaches.

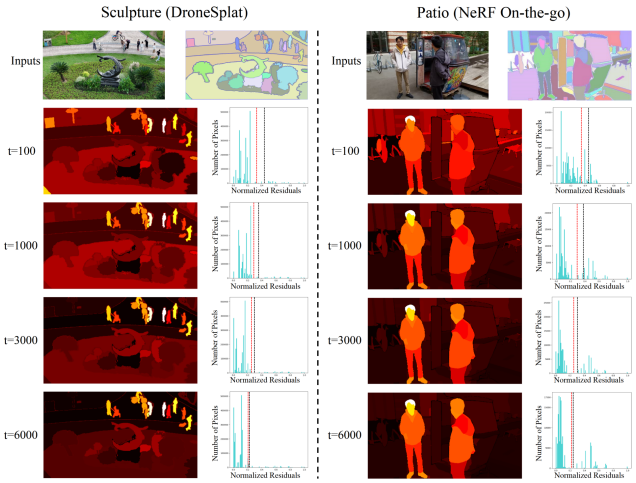


Figure 12. **Adaptive mask thresholding.** The first row of images shows the input training image and its corresponding segmentation result. The subsequent rows illustrate the Object-wise Average of Normalized Residuals and their corresponding histograms at different iterations t. The red dashed line in the histograms represents the sum of the current normalized residual's mathematical expectation and one standard deviation. The black dashed line represents the local masking threshold finally selected.

A key observation is that, for the same image, the residuals of the static scene gradually decrease as t increases while the residuals of dynamic distractors remain almost unchanged. Furthermore, statistical analysis shows that both the mathematical expectation and variance of the Object-wise Average of Normalized Residuals for the image also decrease over time. If dynamic distractors do not dominate the scene, the mathematical expectation of the Object-wise Average of Normalized Residuals declines rapidly as the static scene converges. However, the presence of dynamic distractors causes the variance to decrease at a much slower rate than the expectation.

As shown in Figure 12, whether in the DroneSplat dataset, characterized by numerous small-area dynamic objects, or the NeRF On-the-go dataset, with fewer but larger dynamic objects, nearly all static objects remain within one standard deviation of the expectation, regardless of whether it is the early training stage with higher residuals or the later stage as residuals converge.

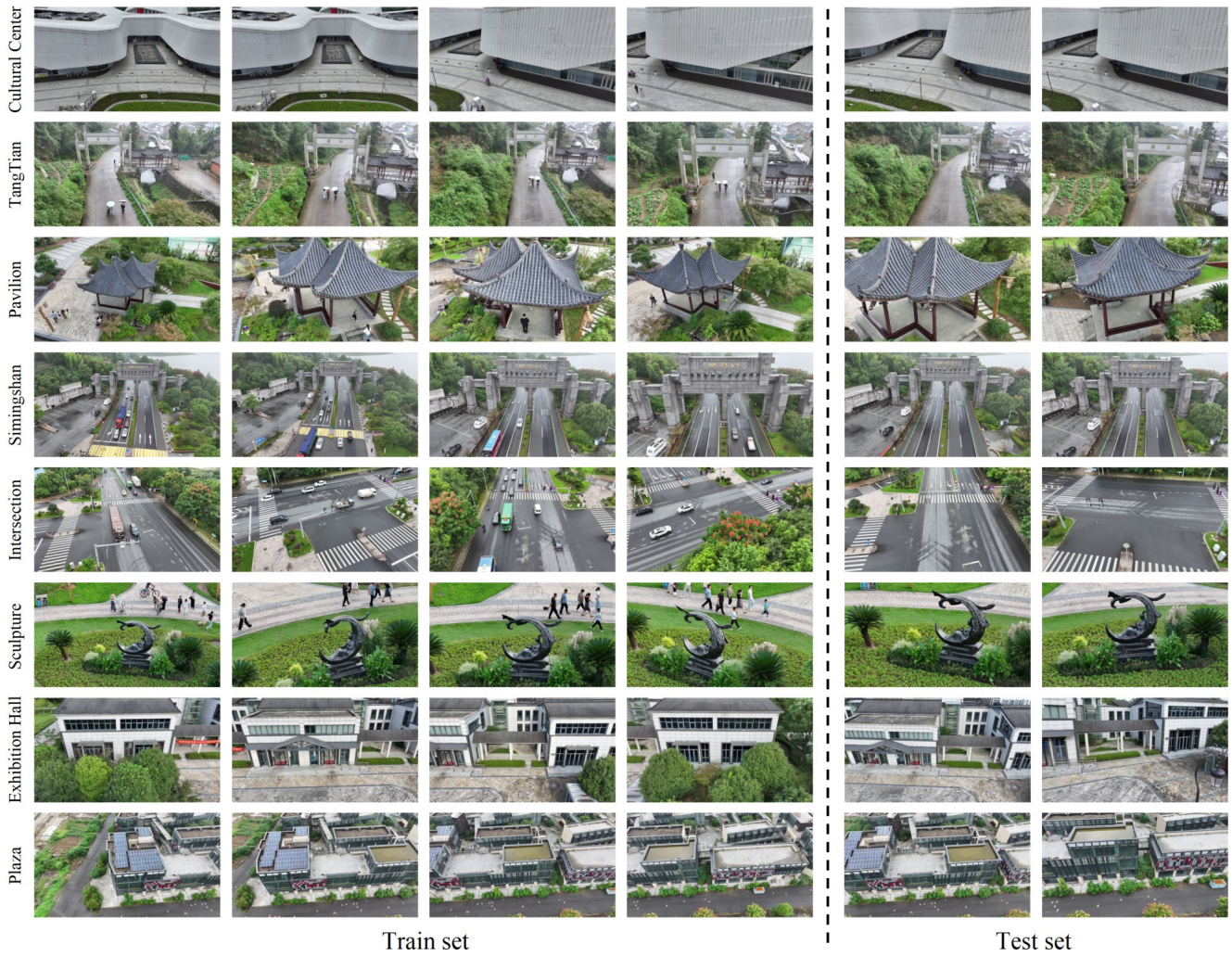


Figure 13. **DroneSplat Dataset.** Our dataset is captured in an uncontrolled wild environment by a drone, distinguishes itself from prior 3D reconstruction datasets by featuring numerous dynamic distractors with small area in each scene.

Complement Global Masking. The role of Complement Global Masking is to identify the corresponding masks in the context of the object with particularly high residual in the current training frame. Specifically, we mark the objects with residuals higher than the threshold $\mathcal{T}^G$ in the Object-wise Average of Normalized Residuals as tracking candidates (there can be multiple candidates in the same iteration). For each candidate, several points are selected as prompts and fed into Segment Anything Model v2 [27] for video segmentation.

As shown in Figure 14, we can obtain the high residual objects (the white car which is highlighted in yellow in the residual image) that needs to be tracked based on the Object-wise Average of Normalized Residuals and the predefined global masking threshold. The blue box on the right represents the tracking results, specifically the mask of the tracked vehicle within the context. By combining these tracking results with the global sets from the previous iteration, we can obtain the updated global sets for the current iteration.

Complement Global Masking is designed to address specific cases that Adaptive Local Masking cannot handle. For example, as discussed in the main paper, when a vehicle stops at a red light at an intersection, Adaptive Local Masking may fail to identify the stationary vehicle as a dynamic distractor in those frames. Unsurprisingly, we observe that lowering the global masking threshold $\mathcal{T}^G$ enables Complement Global Masking to independently and effectively eliminate dynamic distractors. However, due to the significant time cost of video segmentation, relying solely on Complement Global Masking for dynamic object identification would prolong the training process. Therefore, the optimal performance is achieved by combining Adaptive Local Masking with Complement Global Masking.

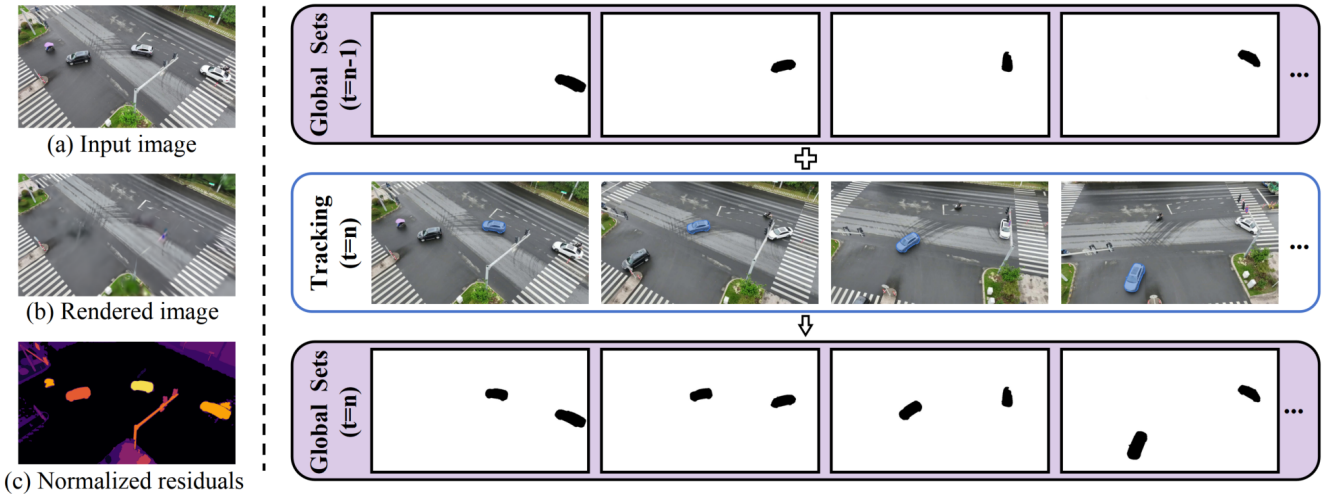


Figure 14. **Complement Global Masking.** At $t = n$, the white vehicle in the center of the image is marked as a tracking candidate due to its residual exceeding the global masking threshold. The tracking results (blue masks) are then incorporated into the global sets from the previous iteration $t = n - 1$ to update the global sets for the current iteration.

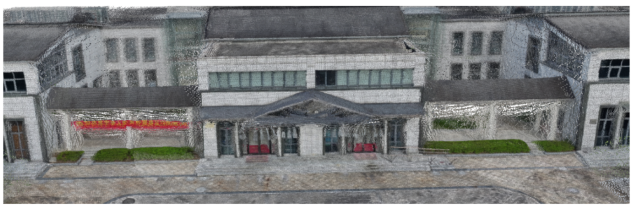
6.3. Voxel-guided Gaussian Splatting.

Multi-view Stereo. DUSt3R [35] is a learning-based framework that takes image pairs as input and outputs corresponding dense point clouds. A post-processing is then used to align the scale across different pairs and obtain a global point cloud. However, as the number of input images increases, the number of image pairs also grows, significantly increasing GPU memory consumption during post-processing. Consequently, the vanilla DUSt3R framework is not well-suited for handling in-the-wild scenes with a large number of images.

To address this issue, we optimize the pipeline by introducing a progressive alignment strategy. Specifically, we divide all images into batches. Suppose there are $b \times N$ images in total, divided into b batches, each containing N images. We first input the N images from the first batch into DUSt3R to generate pair-wise point clouds, followed by post-processing to obtain the aligned point clouds (denoted as P_1) and the corresponding camera parameters C_1. Next, we take the last $N/2$ images from the first batch and the first $N/2$ images from the second batch, inputting them into DUSt3R for pair-wise point cloud prediction, where the poses of the first $N/2$ images are fixed using C_1. Through post-processing, we obtain the aligned point clouds for the second batch (denoted as P_2) and the camera parameters C_2. Note that P_1 and P_2, as well as C_1 and C_2, share $N/2$ overlapping elements, so the duplicated $N/2$ must be removed when merging the outputs of the two batches. Each batch undergoes the aforementioned process, finally resulting in a globally aligned point cloud assembled from multiple batches, along with all corresponding camera parameters. Incorporating the progressive alignment strategy, DUSt3R can handle any number of images by adjusting the number of images per batch.

Geometric-aware Point Sampling. The multi-view stereo method provides rich scene geometry priors, and the number of points in the dense point cloud produced by DUSt3R is still very large even with limited viewpoints. Taking a resolution of 1920 $\times$ 1080 as an example, the images will be automatically resized to 512 $\times$ 288 when input into DUSt3R, and using just six images yields a point cloud containing over 800,000 points. Directly using such a large number of points to initialize Gaussian primitives makes the vanilla optimization strategy ineffective, as each pixel influenced by an excessive number of similar Gaussians, leading to suboptimal reconstruction quality.

Therefore, we propose a geometric-aware point cloud sampling method. The entire scene is divided into smaller voxels, and for each voxel, only a certain number of points are retained for 3DGS initialization, selected based on their geometric features and confidence score. As shown in Figure 16, the sampled point cloud retains the geometric priors, preserving the scene's overall structure. Meanwhile, the number of points is significantly reduced, from over 800,000 to fewer than 100,000. This reduction allows the 3DGS's densification process to fully leverage its remarkable representational capabilities. Notably, the importance of the point sampling method becomes particularly evident when the number of input images increases and overlap regions expand. In such cases, DUSt3R's output may exhibit aliasing, for example, there are many layers of close-fitting ground and walls,, which can severely impact 3DGS's performance. The sampling method effectively mitigates these issues, ensuring high-quality initialization.



(a) Original dense point cloud

(b) Sampled point cloud

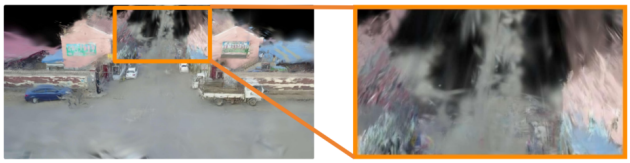
Figure 15. **The effect of point sampling.** Compared to the original point cloud (a), the sampled point cloud (b) also provides sufficient geometric priors using only a fraction of the points.

Voxel-guided Optimization. The role of Voxel-guided Optimization is to overcome the challenge of unconstrained optimization in 3DGS under limited viewpoints. As illustrated in Figure 16, when a drone flies over a street with a fixed posture, the angle between the camera and the ground remains nearly constant (a). Using the original 3DGS optimization results in uncontrolled Gaussian expansion and drifting. Specifically, the unconstrained Gaussian moves toward the camera, which will cause floating Gaussians in the air (b). However, this does not affect optimization for the training views, it highlights a limitation: the Gaussians lack sufficient constraints under limited viewpoints when relying solely on the original optimization.

Our proposed Voxel-guided Optimization addresses the issue of unconstrained optimization by restricting Gaussians within a defined space. Leveraging the rich geometric priors provided by multi-view stereo, we achieve basic overlay of the scene in Gaussian initialization. As mentioned in the main paper, Gaussians that exceed the voxel restricted boundary are identified as unconstrained and have their gradients reduced. Figure 16 (c) demonstrates the effectiveness of our approach. There are no more floating Gaussians in the air, and the surface of the building is reconstructed more accurately. Notably, Voxel-guided Optimization does not interfere with the original optimization process.

Among all the hyperparameters in Voxel-guided Optimization, the most critical is τ, which controls the voxel constraint boundary. A value of τ that is too small prevents Gaussians from adequately fitting the scene, while a value that is too large weakens the voxel's constraint on Gaussian optimization. In practice, we find that setting τ between 3 and 4 strikes an effective balance, enabling accurate Gaussian fitting while maintaining sufficient constraints.

(a) Input images



(b) Vanilla 3DGS optimization

(c) Voxel-guided optimization

Figure 16. **The effect of Voxel-guided Optimization.** Compared to the vanilla optimization, our voxel-based optimization strategy ensure accurate scene geometry. Notably, both (b) and (c) represent the visualization of Gaussians, not the rendering results.

7. Additional Experiments

7.1. Ablation Study

As shown in Table 1, dense point clouds provide a strong geometric prior, accelerating convergence and reducing overall reconstruction time. While initialization of dense points requires more memory, final memory usage and model size remain comparable to sparse point clouds.

As shown in Table 2, among all the designed modules, local masking requires the longest processing time while also providing the most significant performance improvement. Furthermore, although the training time on the 3090 is slightly longer compared to the A100 and 4090, it remains within an acceptable range.

Table 1. **Impact of initial points on memory Usage, training time, model size, and PSNR.** We conduct the experiments on DroneSplat (static) dataset. For each scene, row 1: COLMAP; row 4: DUSt3R; rows 2-3: DUSt3R downsampled points.

Scene	Init Points	Memory Usage	Training Time	Final Size	PSNR↑
Hall	15426	8.17GB	7.72m	667.69MB	15.61
	47489	8.32GB	6.73m	729.29MB	16.48
	237446	8.67GB	6.53m	802.16MB	17.92
	1187233	9.21GB	7.33m	973.62MB	17.63
Plaza	11991	6.74GB	6.67m	481.16MB	16.38
	43885	6.03GB	5.75m	496.41MB	16.74
	219429	6.12GB	5.57m	546.73MB	17.98
	1097146	7.45GB	6.58m	752.92MB	17.73

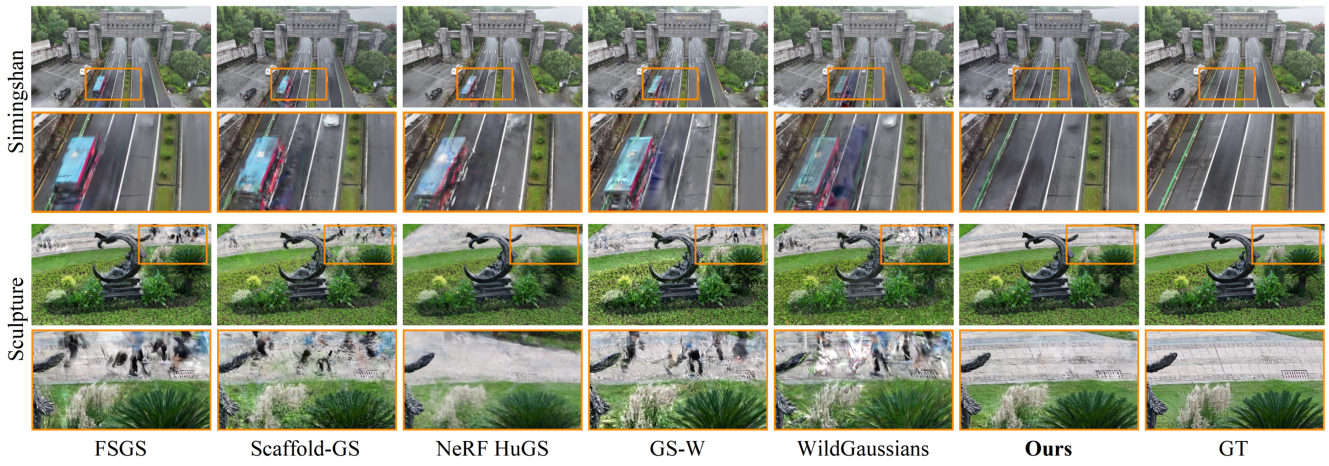


Figure 17. **Qualitative results on DronSplat dataset (dynamic) with limited views.** Each scene contains only six input views

Table 2. **Ablation study of training time for different modules across three GPU platforms.** We conduct the experiments on Simingshan and Sculpture of DroneSplat (dynamic) dataset. For comparison, WildGaussians requires 472 minutes for training, while GS-W takes 138 minutes.

Local Masking	Global Masking	Dense Points	Voxel-guided Optimization	Traing Time A100	4090	3090
✗	✗	✗	✗	7.02m	5.13m	9.63m
✓	✗	✗	✗	15.30m	14.01m	19.87m
✓	✓	✗	✗	18.25m	15.83m	25.93m
✓	✓	✓	✗	16.50m	14.77m	21.70m
✓	✓	✓	✓	16.62m	14.83m	22.18m

7.2. Performance on Two Challenges

In the main paper, we separate the two challenges of *scene dynamics* and *viewpoint sparsity*, and compare them with the most advanced methods in the corresponding fields. Our consideration is that DroneSplat is the only framework capable of tackling both challenges currently. However, to demonstrate the superiority and robustness of our approach in tackling these two challenges simultaneously, we select two representative scenes from the DroneSplat dataset, *Simingshan* and *Sculpture*, to conduct comparative studies on non-static scene reconstruction with limited views. Each scene includes only six input views.

We compare the best-performing baselines from the Distractor Elimination and Limited-View Reconstruction experiments for non-static scene reconstruction with limited views. As shown in Table 3 and Figure 17, our method outperforms the baselines in both scenes. NeRF-HuGS [1], GS-W [45] and WildGaussians [14] fail to effectively eliminate dynamic distractors under sparse-view conditions.

Table 3. **Quantitative results on DronSplat dataset (dynamic) with limited views.** The 1st, 2nd and 3rd best results are highlighted.

Method	Simingshan PSNR↑	SSIM↑	LPIPS↓	Sculpture PSNR↑	SSIM↑	LPIPS↓
FSGS [ECCV'24]	19.97	0.632	0.307	16.55	0.373	0.312
Scaffold-GS [CVPR'24]	19.59	0.637	0.197	16.04	0.321	0.301
NeRF-HuGS [CVPR'24]	20.41	0.656	0.236	16.89	0.249	0.458
GS-W [ECCV'24]	19.21	0.602	0.292	14.84	0.269	0.329
WildGaussians [NIPS'24]	18.91	0.582	0.286	15.58	0.255	0.409
Ours	22.46	0.728	0.156	18.39	0.427	0.253

8. Limitations and Future Work

Our framework leverages effective heuristics and the powerful capabilities of the segmentation model. However, it does have certain limitations in some cases. Firstly, in the video segmentation of Complement Global Masking, we track high-residual objects within their context. In practice, we find that tracking small objects is often unreliable, and tracking errors can sometimes lead to suboptimal results. A potential improvement could involve adding a post-processing to extract features of the tracked targets and filter out results with significant feature discrepancies.

In addition, our approach eliminates the influence of dynamic objects on static scene reconstruction by identifying and masking them. However, if a region is consistently occupied by dynamic distractors, it may lead to underfitting in the reconstruction. To address this, another possible improvement could be the integration of diffusion models to inpaint such regions.

9. Additional Qualitative Results

We additionally show the visualization results of our comparison experiment (Sec 4.2).

9.1. DroneSplat Dataset (dynamic)

As shown in Figure 18, NeRF-based methods, such as RobustNeRF [30] and NeRF On-the-go [29], eliminate dynamic objects in the scene but lack detail, sometimes even missing parts of the scene. A typical example is the *pavilion* scene, where RobustNeRF and NeRF On-the-go successfully remove dynamic pedestrians but fail to reconstruct certain areas accurately, such as the missing the pillars of the pavilion. GS-W [45] and WildGaussians [14] perform poorly in the presence of numerous dynamic distractors, failing to eliminate them effectively. In contrast, our method not only effectively removes dynamic objects but also preserves fine scene details with high fidelity.

9.2. NeRF On-the-go Dataset

Compared to our DroneSplat dataset (dynamic), the NeRF On-the-go dataset [29] features a greater number of viewpoints, with some scenes exhibiting higher levels of occlusion. As shown in the figure 19, NeRF-HuGS [1] struggles to handle highly occluded scenes, resulting in blurring and artifacts. While WildGaussians [14] effectively eliminates dynamic distractors and preserves scene details in most cases, it produces inaccurate reconstructions in areas with sparse viewpoints, such as the roadside in the *fountain* scene. Our method demonstrates superior and robustness performance in handling varying levels of occlusion compared to existing approaches.

9.3. DroneSplat Dataset (static)

To simulate the challenges encountered in practical use, the experimental scenes we select have small overlap between viewpoints. As shown in the figure 20, leveraging rich geometric priors and the Voxel-guided Optimization strategy, our method reconstructs geometrically accurate scenes even under limited views constraint.

9.4. UrbanScene3D Dataset (static)

The results and conclusions are similar to those on the DroneSplat dataset (static).

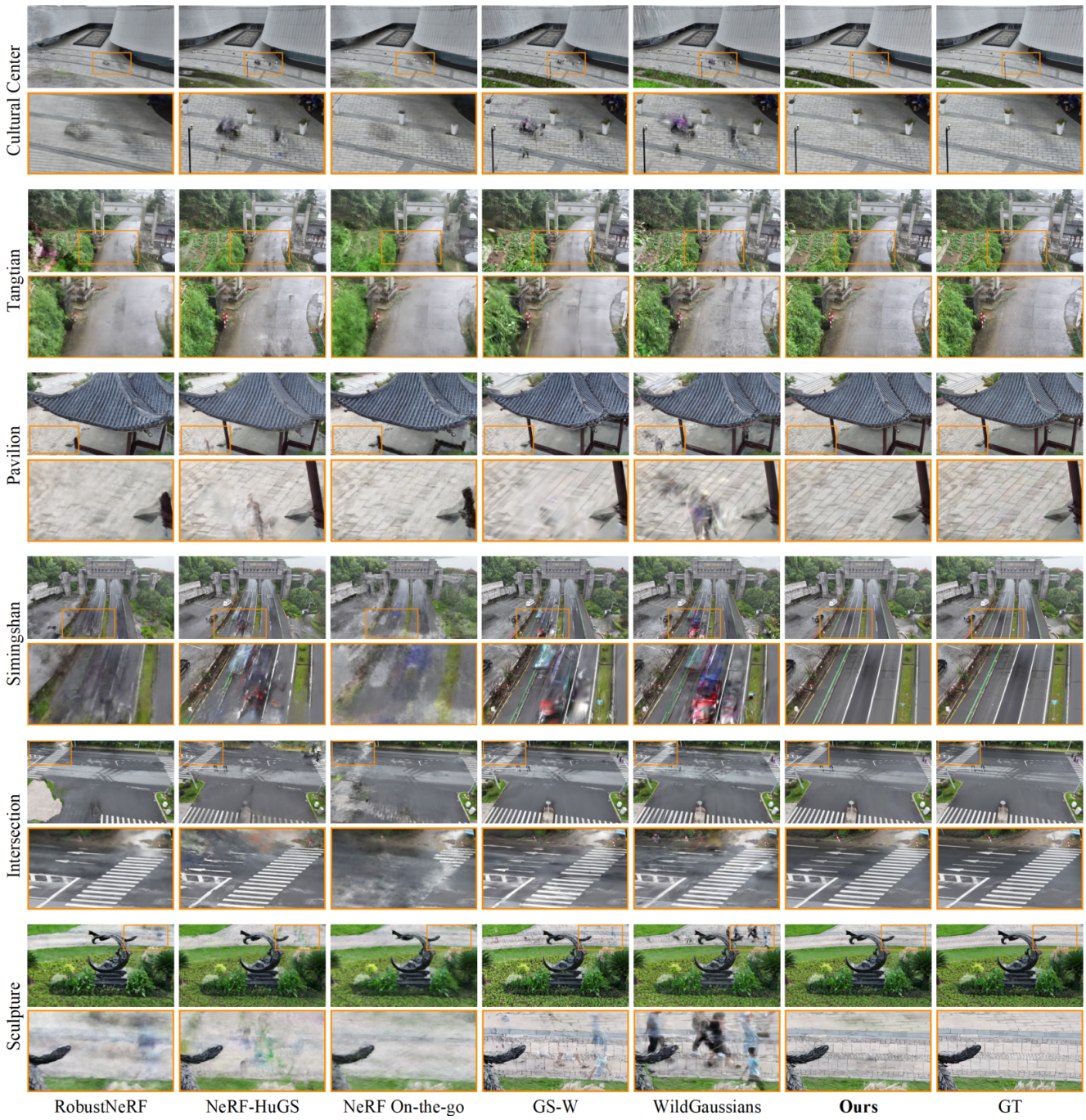


Figure 18. **Qualitative results of distractor elimination on DronSplat dataset (dynamic).**

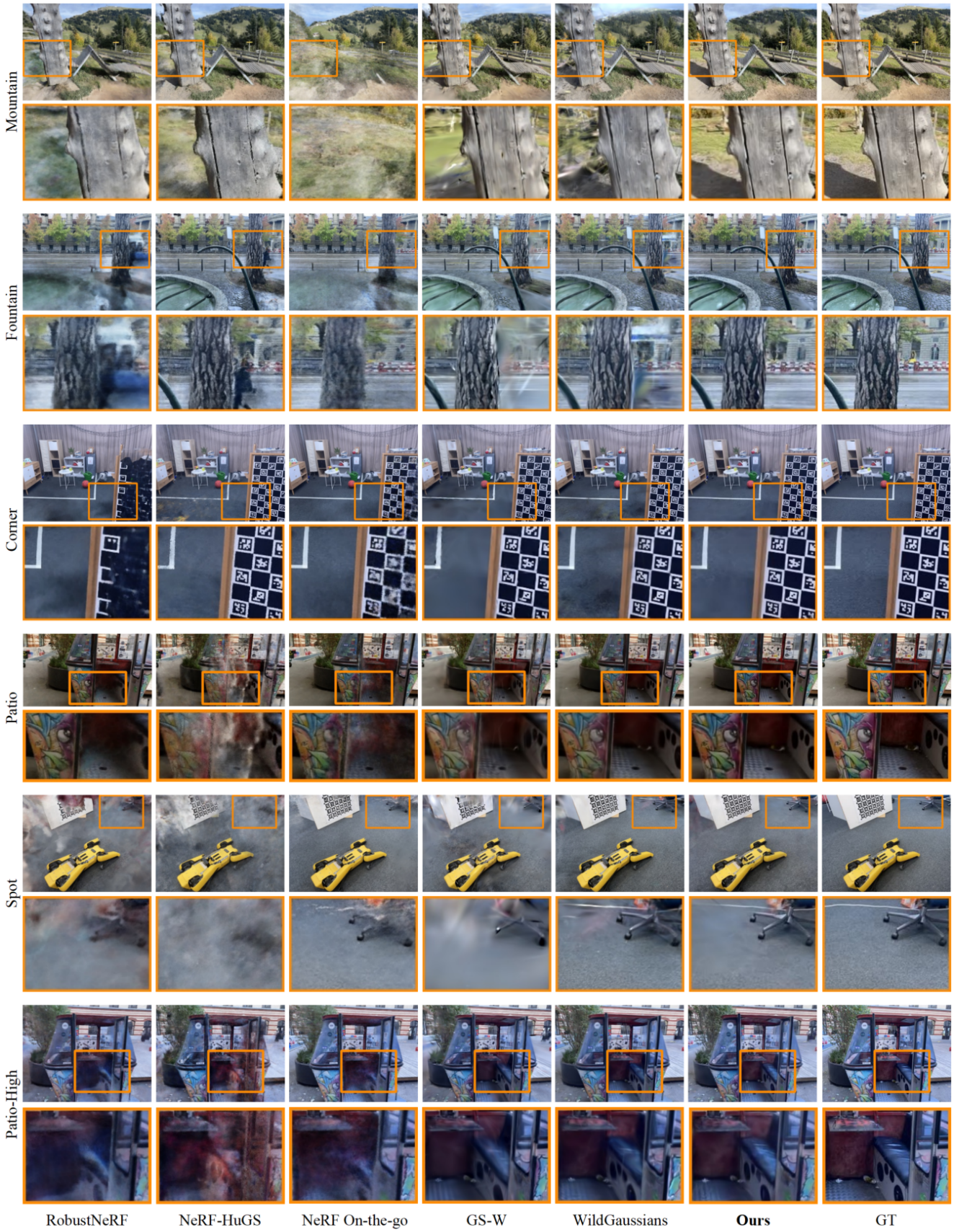


Figure 19. **Qualitative results of distractor elimination on NeRF On-the-go dataset.**

Figure 20. **Qualitative results of limited-view reconstruction on DronSplat dataset (static) and UrbanScene3D dataset.**