RobustSplat: Decoupling Densification and Dynamics for Transient-Free 3DGS

Supplementary Material

1. Discussions

Sparse Gaussian Initialization and Gaussian Densification The optimization of 3D Gaussian Splatting (3DGS) relies on an initial set of points obtained via Structure-from-Motion (SfM). Since SfM reconstructs sparse point clouds based on multi-view consistency, transient objects that remain stationary in multiple captured images before moving can introduce noisy points into the reconstruction. As a result, 3DGS may initially fit these transient regions, even before Gaussian densification takes place.

As illustrated in Fig. S1, in the *Patio* scene from the NeRF On-the-go dataset, moving subjects remained stationary for a period, leading to COLMAP reconstructing noisy points corresponding to these transient objects. As a result, 3DGS initially fits to these transient regions. However, with longer optimization, our transient mask estimation progressively removes these artifacts. This observation highlights that by applying a transient mask to filter dynamic regions, our method effectively mitigates the impact of noisy initialization, leading to improved reconstruction quality.

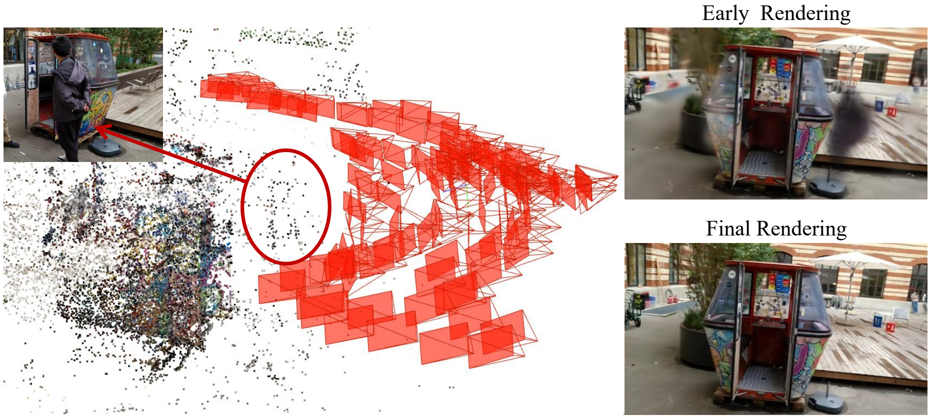

Figure S1. Gaussians initialization with inaccurate COLMAP SfM point clouds may affect the early optimization stage.

Illumination Variations In real-world environments, besides transient disturbances, illumination changes can introduce multi-view inconsistencies, leading to floating artifacts. Our method mainly addresses transient object interference. However, when abrupt illumination changes occur in a scene, our approach fails to correctly model the actual lighting variations due to the absence of an explicit illumination model Fig. S2. A promising direction for future work is to incorporate illumination modeling into our method, enabling the handling of more complex outdoor datasets.

Feature Extraction for Mask Estimation In the main text, we discuss the impact of different feature types on mask learning. DINOv2 performs well due to its efficiency and the reliable consistency of features within similar object categories. However, its patch-based nature introduces inconsistencies at the edges when extended to high-resolution settings, limiting the effectiveness of our mask predictor. In this work, we slightly expand the mask by applying dilation with a kernel size of 7. In the future, we will explore integrating more expressive and efficient feature extractors for mask learning.

2. More Details for the Method

Training Details The original 3DGS [1] resets the opacity starting from the 3000 iterations while maintaining an interval of 3000 iterations. This operation aims to eliminate the accumulation of low-opacity Gaussian primitives in regions close to the camera, which can interfere with gradient backpropagation and manifest as artifacts. However, the opacity reset is no longer suitable for our method due to the delayed Gaussian growth. Therefore, we delay the opacity reset to start from the 15000 iterations while maintaining the same interval of 3000 iterations. Meanwhile, the start of pruning is also delayed to 10000 iterations to align with delayed Gaussian growth.

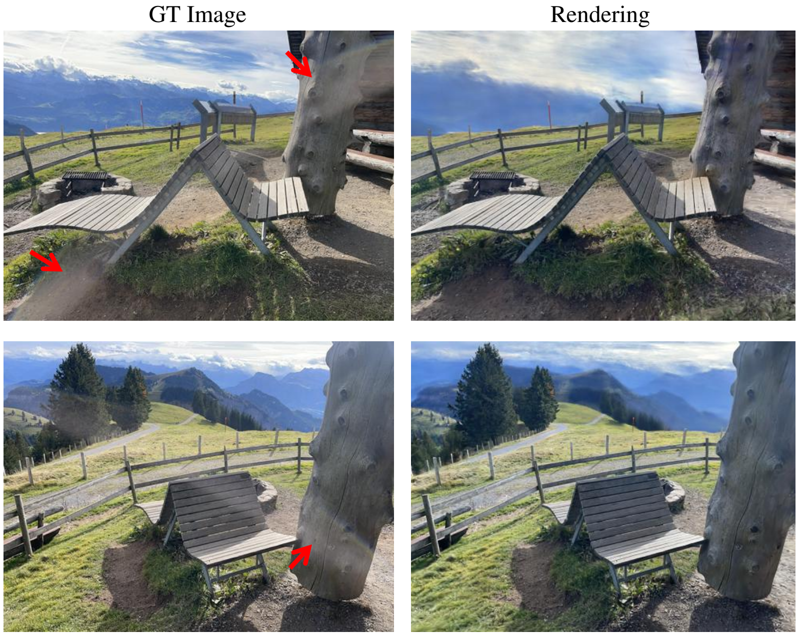

Figure S2. Illumination changes in real-world scenes.

Robust Loss based on Image Residuals The image robust loss used in our mask predictor follows [5]:

$$\mathcal{L}_{\text{residual}} = max\left(\left(\mathrm{U}-\mathrm{M}\right),0\right) + max\left(\left(\mathrm{M}-\mathrm{L}\right),0\right), \tag{1}$$

where M is the mask we predicted, U and L are upper and lower bound of the dynamic residual mask, respectively, which determined by different values of the parameter τ. In our method, the parameters are set to $\tau_u = 0.6$ and $\tau_l = 0.8$ for all experiments.

3. Runtime Evaluation

Our method adopts the lightweight DINOv2 model *ViT-S/14-distilled*, with a feature dimensionality of 384, for feature extraction. As shown in Table S1, our method runs slightly slower than 3DGS but remains faster than other methods. SpotLessSplats achieves similar optimization time without iterative feature extraction, but its SD features, with a dimensionality of 1280, require a long processing time before training.

Table S1. Runtime evaluation on an NVIDIA RTX 3090 (unit: minutes). The runtime of SpotLessSplats is divided into two parts: training and SD feature extraction.

Method	Mountain #Img 120	Fountain #Img 169	Corner #Img 101	Patio #Img 99	Spot #Img 169	Patio-High #Img 222
3DGS	12.21	14.37	9.986	7.707	11.68	12.82
SpotLessSplats	13.48+6.9	16.07+9.8	14.15+6.4	13.82+6.4	13.03+9.5	14.07+13.7
WildGaussians	32.63	52.90	33.58	29.93	27.32	33.86
Ours	15.43	17.33	13.32	12.82	12.95	14.35

4. More Ablation Study

Effects of Mask Regularization. Initial mask estimation yields suboptimal results in most scenes due to unconverged reconstruction at early training stages. To address this challenge, we introduce a mask regularization for stabilizing early-stage mask training. Table S2 shows that removing the proposed mask regularization leads to a decrease in overall performance.

Effects of Delayed Gaussian Growth. We discussed the effectiveness of Delayed Gaussian Growth in Section 4.3 of the main paper. To further validate its effects, we extend the Delayed Gaussian Growth to 3DGS in this supplementary material. Table S3 shows that integrating the delayed gaussian growth into 3DGS leads to improve results, but its performance is limited by the lack of predicting the transient masks.

Table S2. Effects of Mask Regularization. We denote *Mask Regularization* as “MR”.

Method	Mountain		Fountain		Corner		Patio		Spot		Patio-high	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ours w/o MR	21.09	0.728	20.87	**0.701**	26.18	0.889	21.61	0.826	25.63	**0.907**	22.68	**0.837**
Ours	**21.15**	**0.737**	**21.01**	**0.701**	**26.42**	**0.897**	**21.63**	**0.827**	**26.21**	**0.907**	**22.87**	**0.837**

Table S3. Effects of Extending Delayed Gaussian Growth to 3DGS. We denote *Delayed Gaussian Growth* as “DG”.

Method	Mountain		Fountain		Corner		Patio		Spot		Patio-high	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
3DGS	19.21	0.691	20.08	0.686	22.65	0.835	17.04	0.713	18.54	0.717	17.04	0.657
3DGS+DG	20.14	0.693	20.35	0.683	23.54	0.864	17.46	0.728	23.42	0.854	18.87	0.728
Ours	**21.15**	**0.737**	**21.01**	**0.701**	**26.42**	**0.897**	**21.63**	**0.827**	**26.21**	**0.907**	**22.87**	**0.837**

5. Evaluation on On-the-go II Dataset

The NeRF On-the-go II dataset [4] is more challenging compared to the other scenes of NeRF On-the-go, as it consists of outdoor scenes that include not only dynamic objects but also motion blur and varying lighting conditions. Since the testing images in the On-the-go II dataset contain moving objects, we manually segment and exclude these objects when computing the metrics to ensure a fair evaluation.

We can see from Table S4 that our method achieves nearly the best results across all six scenes, except for the second-best performance in the PSNR metric on *Statue*. Moreover, our method outperforms existing methods and achieves state-of-the-art regarding average metrics. Figure S3, Figure S4, and Figure S5 present qualitative comparisons with existing methods on the NeRF On-the-go II dataset. Our method successfully eliminates artifacts (*e.g.*, vehicles in the *Drone*) and recovers finer details (*e.g.*, thin cables in the *Train-station*), further demonstrating its effectiveness in handling complex scenarios.

Table S4. Quantitative comparison on NeRF On-the-go II Dataset. The best results are highlighted in **bold**, and the second in underline.

Method	Arcdetriomphe		Drone		Statue		Train		Train-station		Tree		Mean	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
3DGS [1]	25.57	0.926	21.37	0.830	15.95	0.751	22.49	0.847	21.43	**0.871**	22.44	0.846	21.54	0.845
SpotLessSplats [5]	28.70	0.940	20.87	0.800	16.01	0.737	23.28	0.841	21.37	0.815	23.00	0.834	22.21	0.828
WildGaussians [2]	24.25	0.898	21.31	0.815	**17.32**	0.795	23.81	0.852	22.50	0.846	22.77	0.832	21.99	0.840
Robust3DGS [6]	26.36	0.933	18.69	0.785	14.66	0.724	23.79	0.860	20.67	0.833	22.73	0.868	21.15	0.834
T-3DGS [3]	28.86	0.943	21.08	0.820	16.57	0.756	**24.34**	0.870	21.87	0.851	23.14	**0.870**	22.63	0.852
Ours	**29.43**	**0.949**	**21.62**	**0.844**	16.65	**0.802**	24.07	**0.871**	**22.78**	**0.871**	**23.57**	0.868	**23.02**	**0.868**

6. Comparison of Mask Estimation

Figure S6 compares the transient mask estimation results of our method with existing methods. Our method can better filter the transient objects while keeping the static regions, leading to less artifacts and sharp details in the rendering images.

References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 2023. 1, 3

[2] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *arXiv preprint arXiv:2407.08447*, 2024. 3

[3] Vadim Pryadilshchikov, Alexander Markin, Artem Komarichev, Ruslan Rakhimov, Peter Wonka, and Evgeny Burnaev. T-3dgs: Removing transient objects for 3d scene reconstruction. *arXiv preprint arXiv:2412.00155*, 2024. 3

[4] Weining Ren, Zihan Zhu, Boyang Sun, Jiaqi Chen, Marc Pollefeys, and Songyou Peng. Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In *CVPR*, 2024. 3

[5] Sara Sabour, Lily Goli, George Kopanas, Mark Matthews, Dmitry Lagun, Leonidas Guibas, Alec Jacobson, David J Fleet, and Andrea Tagliasacchi. Spotlesssplats: Ignoring distractors in 3d gaussian splatting. *arXiv preprint arXiv:2406.20055*, 2024. 2, 3

[6] Paul Ungermann, Armin Ettenhofer, Matthias Nießner, and Barbara Roessle. Robust 3d gaussian splatting for novel view synthesis in presence of distractors. *arXiv preprint arXiv:2408.11697*, 2024. 3

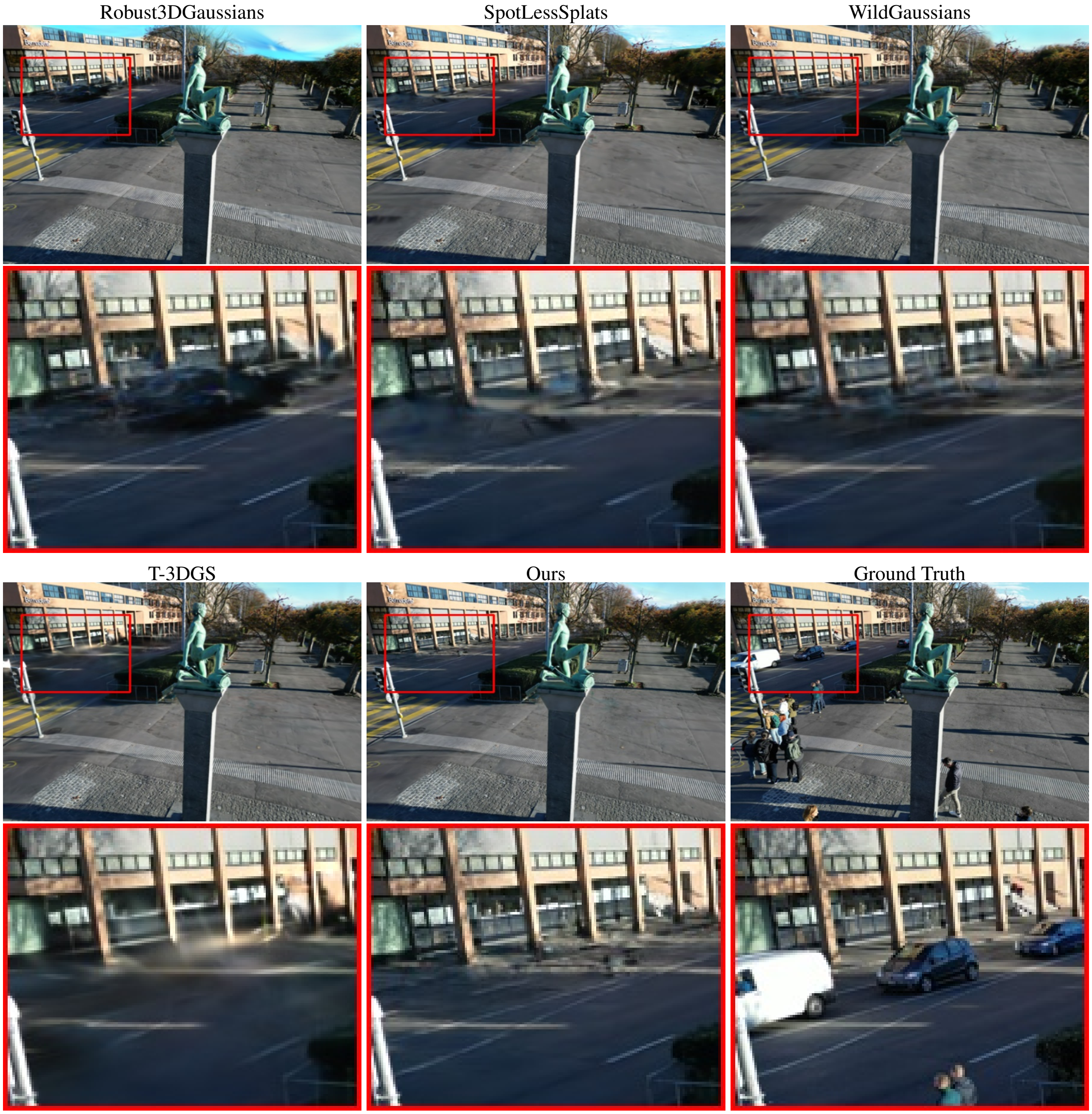

Figure S3. Qualitative results on *Drone* in NeRF On-the-go II dataset.

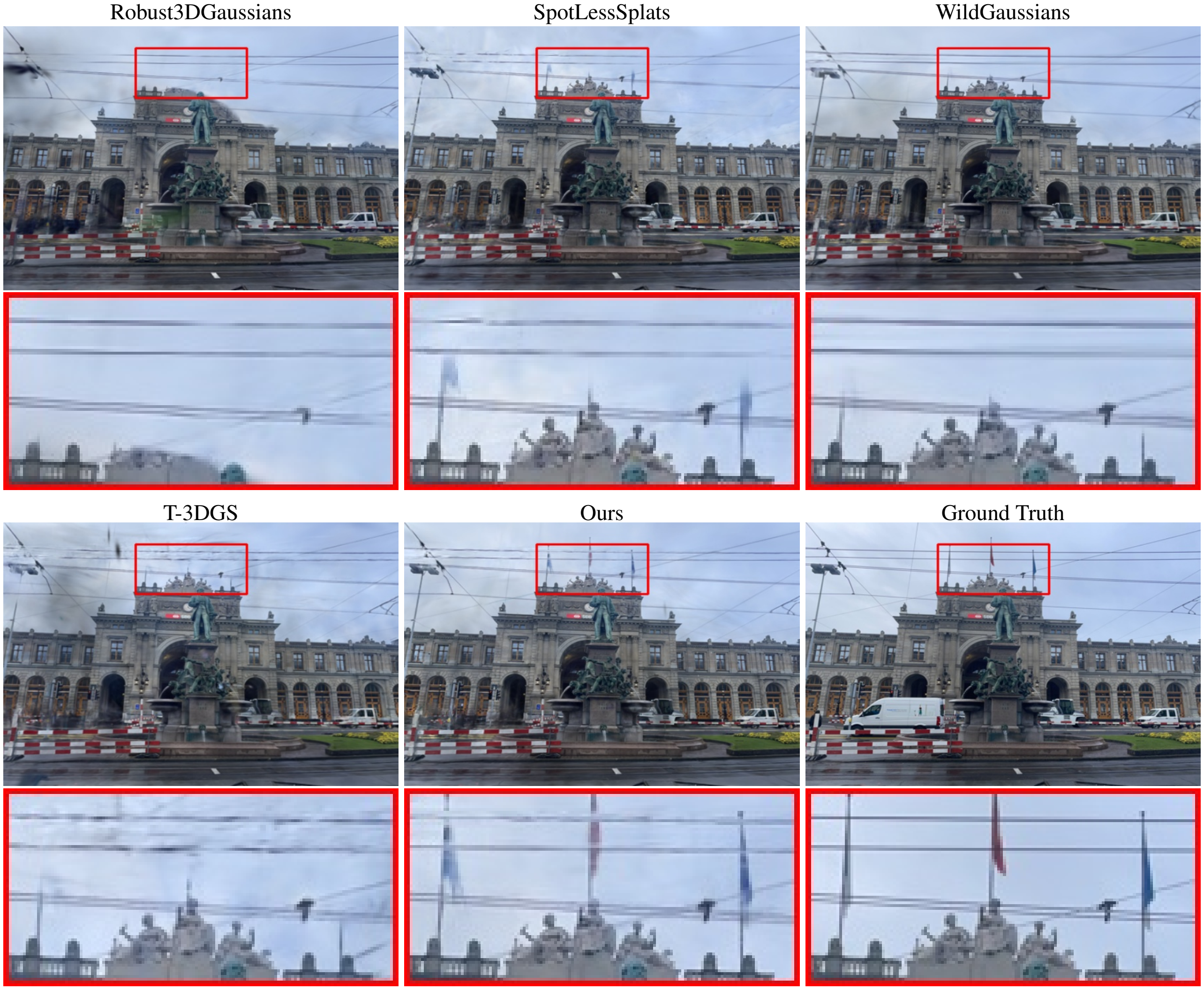

Figure S4. Qualitative results on *Train-station* in NeRF On-the-go II dataset.

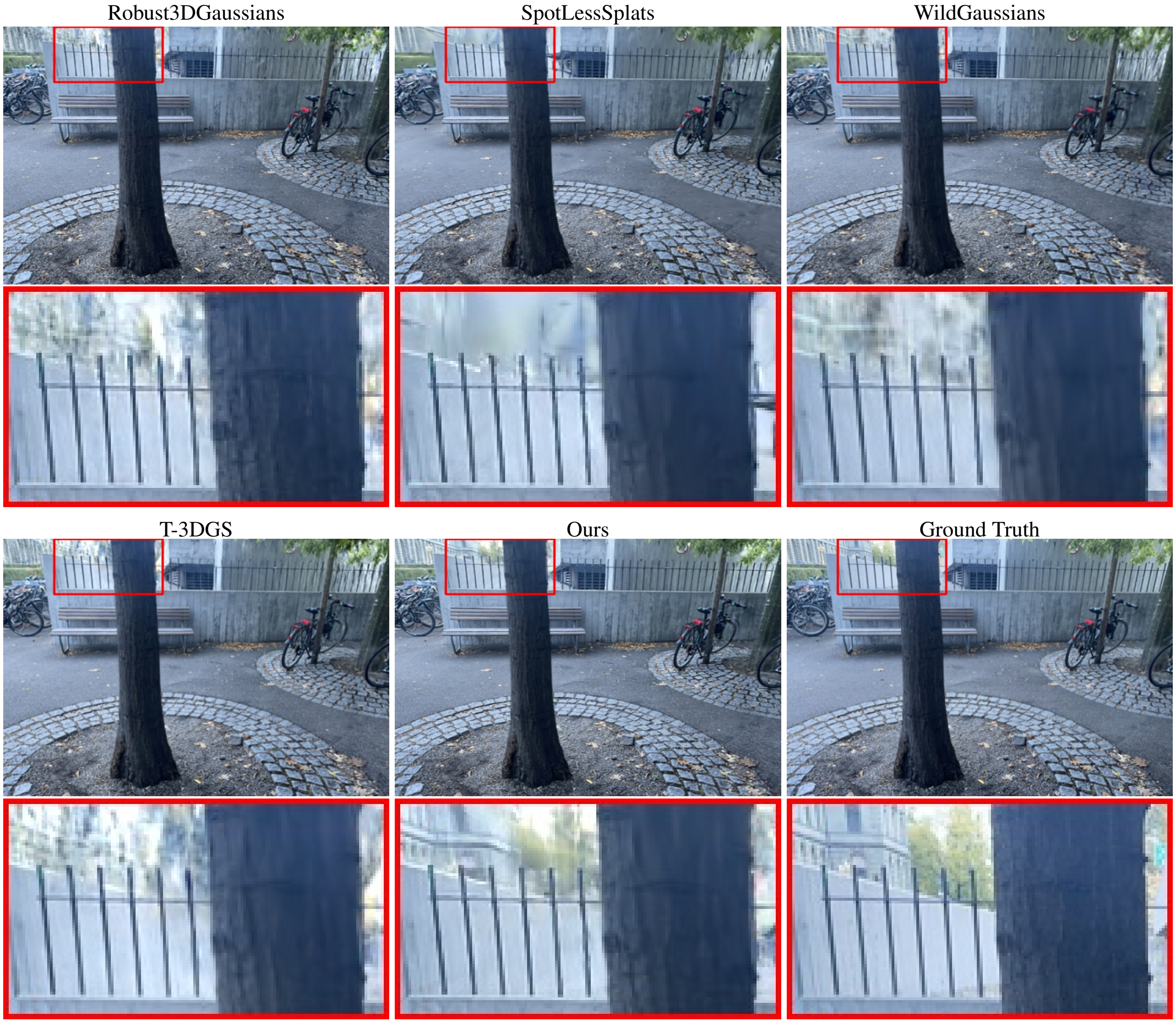

Figure S5. Qualitative results on *Tree* in NeRF On-the-go II dataset.

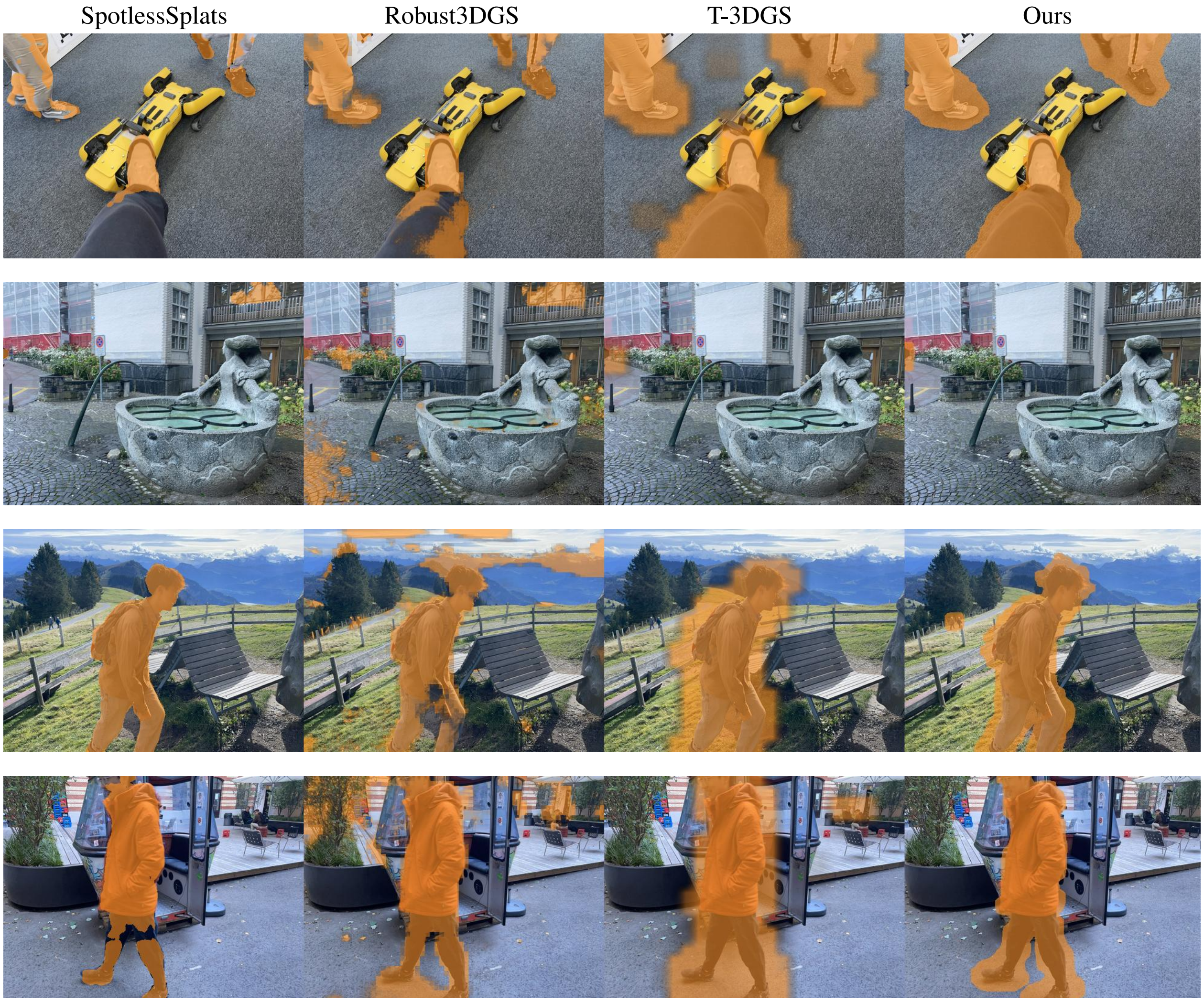

Figure S6. Comparison of transient mask in NeRF On-the-go dataset.