DualSplat: Robust 3D Gaussian Splatting via Pseudo-Mask Bootstrapping from Reconstruction Failures

Supplementary Material

6. More Details

6.1. Data Preprocessing and First-Pass Setup

For the stage one reconstruction, we down-sample all training frames by $4\times$. The only exception is *Patio*, for which we use a factor $2\times$. The same resizing is consistently applied to ground-truth images, first-pass renderings, feature maps, and masks to avoid scale mismatch. After computing the cosine-similarity map from FiT3D features at their native stride, we *bilinearly resample* the similarity map back to the preprocessing resolution used by our pipeline.

6.2. SAM2 Instance Proposals

Rather than using SAM2 with default settings, we bias it toward more masks so that small or partially occluded distractions are not missed. This yields more instance candidates, while the actual transient decision is deferred to our feature and residual filter. In all experiments, instance masks are exported as integer-labeled arrays for efficient per-instance statistics. In all experiments reported in the main paper, we ran SAM2 in automatic mask generation mode. We set SAM2 with

$$
\begin{aligned}
&points\_per\_side = 32\\
&points\_per\_batch = 128\\
&pred\_iou\_thresh = 0.8\\
&stability\_score\_thresh = 0.9\\
&box\_nms\_thresh = 0.8\\
&crop\_n\_layers = 2\\
&crop\_nms\_thresh = 0.8\\
&crop\_overlap\_ratio = 0.6\\
&crop\_n\_points\_downscale\_factor = 1\\
&min\_mask\_region\_area = 0\\
&use\_m2m = False\\
&multimask\_output = True\\
&mask\_threshold = 0.0\\
&output\_mode = binary\_mask
\end{aligned}
$$

Although we used this setting in the paper, further experiments show that simply setting $crop\_n\_layers = 1$ already covers most cases and is faster.

6.3. Mask Filter: Implementation Specifics

Our filter operates per instance and fuses feature consistency and photometric residuals, constrained by instance boundaries and geometry sanity checks. Below are the inputs to the mask filter:

- A pixel-wise L_1 map between the first-pass rendering and its ground truth, computed at the same spatial resolution.
- Dense features for both GT and rendering, followed by a pixelwise cosine-similarity map $S \in [-1, 1]^{H\times W}$. We apply min-max normalization to $\hat{S} \in [0, 1]^{H\times W}$ before aggregation.
- SAM2 instance masks.

Geometry sanity checks. Before any photometric and feature reasoning, we discard spurious proposals using four methods:

- **Minimum area:** remove masks with < 32 pixels.
- **Minimum short side:** remove masks whose short side of the bbox < 5 px.
- **Minimum fill ratio** (mask area / bbox area): remove if < 0.05.
- **Minimum global fill ratio** (mask area / image area): remove if $< 10^{-4}$.

Only proposals that pass all four checks are carried forward. For each remaining instance m with the set of pixels Ω_m, we accumulate:

$$
\mu_m = \frac{1}{|\Omega_m|}\sum_{p\in\Omega_m}\hat{S}(p), \qquad \bar{\ell}_m = \frac{1}{|\Omega_m|}\sum_{p\in\Omega_m} L_1(p). \tag{17}
$$

An instance is accepted as *transient* if $\mu_m \le \tau_{\text{sim}}$ and $\bar{\ell}_m \ge \tau_{L1}$. The final pseudo-masks M_{pseudo} are the union of the retained instances. We also sweep $\tau_{\text{sim}} \in \{0.65, 0.70, 0.75, 0.80, 0.85\}, \tau_{L1} \in \{0, 0.05, 0.10\}$. And we report mask-level Accuracy, Precision, Recall, and IoU (all in %) in Tab. 7. From the ablation, increasing τ_{sim} makes the similarity gate looser, so more instances are flagged as transient. This typically raises the recall and overall detection capacity, but also introduces more false positives, lowering the precision. In contrast, increasing τ_{L1} makes the residual requirement stricter, which filters out hard-to-render yet static regions; this improves precision and IoU at the cost of reduced recall. Across the sweep over τ_{sim} and τ_{L1}, we observe stable trends and no sensitivity spikes, indicating that the filter is reasonably robust to threshold choices. We adopt $(\tau_{\text{sim}}, \tau_{L1}) = (0.75, 0.05)$ as a balanced operating point that trades off missed detections and false alarms.

Table 7. Parameter sensitivity of τ_{sim} and τ_{L1}

Accuracy			
	0	0.05	0.1
0.65	0.981	0.984	0.988
0.7	0.974	0.981	0.983
0.75	0.965	0.976	0.981
0.8	0.950	0.967	0.979
0.85	0.924	0.957	0.977
Precision			
	0	0.05	0.1
0.65	0.856	0.868	0.873
0.7	0.830	0.854	0.866
0.75	0.802	0.841	0.861
0.8	0.767	0.817	0.848
0.85	0.710	0.784	0.834
Recall			
	0	0.05	0.1
0.65	0.946	0.945	0.928
0.7	0.949	0.948	0.931
0.75	0.952	0.951	0.934
0.8	0.955	0.953	0.935
0.85	0.957	0.955	0.938
IoU			
	0	0.05	0.1
0.65	0.846	0.858	0.848
0.7	0.822	0.846	0.844
0.75	0.798	0.835	0.841
0.8	0.763	0.813	0.830
0.85	0.708	0.781	0.816

6.4. Lightweight Mask Estimator

Inputs and Notation. We use the DINOv2 [17] model to extract image features and resize them to the training view resolution with bilinear sampling. Let the per-pixel feature map be $F \in \mathbb{R}^{C\times H\times W}$ with C=384, and the per-pixel depth residual map be $D \in \mathbb{R}^{H\times W}$. Denote the pixel set by $\mathcal{P}$ with $|\mathcal{P}|$=HW. For any pixel $p \in \mathcal{P}$, let $f_p \in \mathbb{R}^{384}$ be the feature vector and $d_p \in \mathbb{R}$ the depth residual. F is reshaped to $(|\mathcal{P}|, C)$ to apply the network pixelwise; D is reshaped to $(|\mathcal{P}|, 1)$ and concatenated with an intermediate representation.

Network Architecture. The model consists of two fully-connected (per-pixel) submodules in series, with ReLU/Sigmoid activations:

$$
\begin{aligned}
h_p &= \mathrm{ReLU}(W_1 f_p + b_1)\,, && W_1 \in \mathbb{R}^{16\times 384},\\
z_p &= \sigma(W_2 h_p + b_2)\,, && W_2 \in \mathbb{R}^{8\times 16},\\
u_p &= [\,z_p;\; d_p\,] \in \mathbb{R}^9, &&\\
\hat{m}_p &= \sigma\big(w_3^\top u_p + b_3\big) > 0.2, && w_3 \in \mathbb{R}^9.
\end{aligned}
$$

Here $\sigma(\cdot)$ denotes the Sigmoid. After reshaping $\{\hat{m}_p\}_{p\in\mathcal{P}}$ back to the image resolution and dilating, it yields the final mask $\hat{M} \in \{0,1\}^{1\times H\times W}$ which is applied to reconstruction.

7. Additional Experiments

7.1. More scenes of NeRF On-the-go dataset

Beyond the main benchmarks, we further evaluate on an extended subset of the NeRF On-the-go dataset that contains in-the-wild handheld captures with natural distractions. We select five scenes: drone, train station, train, statue, and tree. We select these scenes because they do not require additional processing on the test set. Unless otherwise stated, we follow the same training schedule and hyperparameters as in the main paper. We report per-scene PSNR, SSIM and LPIPS in Tab. 8 to quantify reconstruction quality.

8. Runtime

We also present the runtime efficiency of DualSplat. All methods are executed on a single RTX 3090 GPU and follow the identical optimization schedule. To isolate our algorithmic overhead, the reported times exclude SAM2 preprocessing, as the mask generation time can vary substantially with its hyperparameters. Tab. 9 summarizes the runtime comparison.

SAM2 Preprocessing Sensitivity. Since SAM2 preprocessing constitutes the main external cost, we report the SAM2 preprocessing time under different hyperparameter settings. Tab. 10 reports the average processing time per image under varying configurations together with the corresponding mean reconstruction metrics on the NeRF On-the-go dataset. The additional SAM2 preprocessing cost can be flexibly adjusted according to available computational resources without materially affecting the reconstruction quality.

9. Mask Visualization

Fig. 7 compares our transient-mask estimates with existing methods. Our approach more effectively filters transient objects while preserving static regions, thereby reducing artifacts in the rendered images and improving detail sharpness.

Table 8. **Comparison on additional scenes of the NeRF On-the-go Dataset.** For better visualization, the 1st, 2nd and 3rd best results are highlighted.

Method	drone			statue			train			train_station			tree			Mean		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
3DGS [8]	20.05	0.793	0.150	15.10	0.616	0.285	21.40	0.766	0.126	20.56	0.814	0.119	19.62	0.634	0.146	19.35	0.725	0.165
WildGaussians [10]	19.84	0.768	0.182	17.04	0.795	0.213	23.20	0.806	0.138	21.12	0.814	0.162	22.66	0.829	0.101	20.77	0.802	0.159
DeSplat [31]	19.18	0.768	0.190	15.98	0.799	0.258	22.13	0.784	0.168	19.18	0.790	0.177	22.55	0.845	0.095	19.80	0.797	0.178
SpotLessSplats [24]	19.43	0.751	0.188	15.32	0.749	0.235	22.52	0.791	0.133	20.54	0.783	0.166	22.40	0.817	0.096	20.04	0.778	0.164
RobustSplat [6]	19.77	0.793	0.159	16.43	0.812	0.157	22.86	0.820	0.110	21.62	0.841	0.114	23.06	0.862	0.070	20.75	0.826	0.122
3DGS with pseudo-masks	19.74	0.787	0.153	15.10	0.726	0.247	22.52	0.806	0.105	20.75	0.823	0.121	22.48	0.850	0.075	20.12	0.798	0.140
Ours	19.95	0.794	0.157	16.70	0.808	0.159	23.07	0.821	0.110	21.68	0.841	0.114	22.99	0.862	0.071	20.88	0.825	0.122

Table 9. **Time consumption of different stages.** We present the consumption of 3DGS as a baseline.

Method	Corner	Fountain	Mountain	Patio-High	Spot	Patio	Mean
3DGS	803.489s	1042.167s	881.698s	994.384s	862.851s	731.072s	885.943s
Stage 1	512.684s	727.496s	552.997s	1066.476s	657.683s	441.534s	659.812s
Stage 2	1114.166s	1276.067s	1325.501s	1346.641s	1141.181s	1332.23s	1255.964s

Table 10. Comparison of different settings.

Method	Low Occlusion						Medium Occlusion						High Occlusion						Mean		
	Mountain			Fountain			Corner			Patio			Spot			Patio-High					
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Ours with default SAM2 setting	21.85	0.747	0.127	**21.37**	**0.706**	0.131	26.29	0.891	0.056	21.53	0.828	0.081	25.69	0.907	0.046	23.25	0.836	**0.094**	23.33	0.819	0.089
Ours	**21.86**	**0.747**	**0.127**	21.28	0.706	**0.130**	**26.53**	**0.897**	**0.053**	**21.64**	**0.828**	**0.080**	**25.96**	**0.908**	**0.046**	**23.26**	**0.836**	0.095	**23.42**	**0.820**	**0.088**

Settings	Time consumption per image
SAM2 with default setting	1.6155s
SAM2 with ours setting	26.4248s

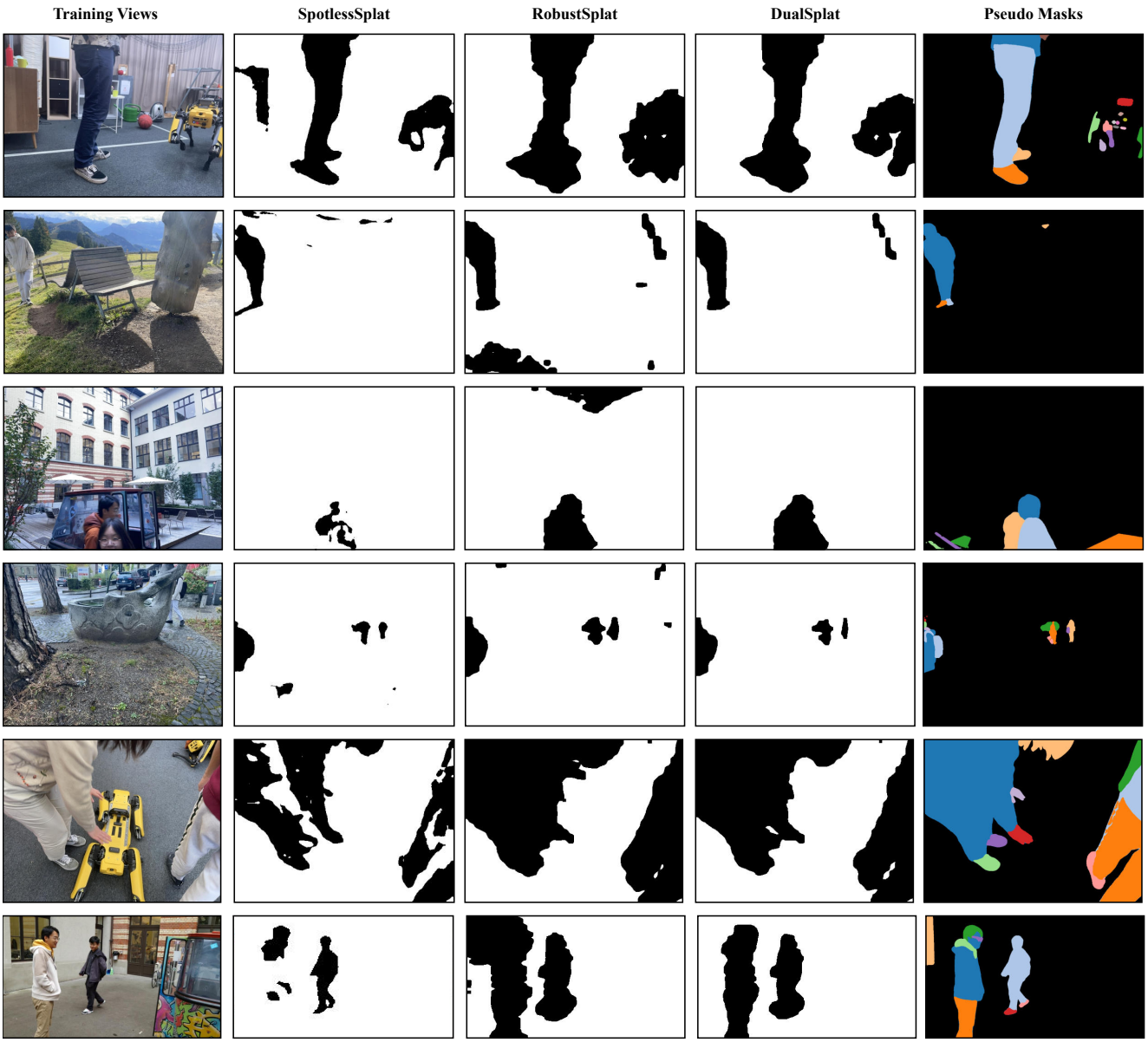

Figure 7. Mask comparison.