

Figure 6: Randomness of artifacts across training runs. Row 1 shows the target view (with the presence of distractors); Rows 2 and 3 present results from two independent runs of Mip-Splatting; Row 4 shows the result of our method with the mutual consistency regularization.

A Random nature of artifacts

As shown in Figure 1 and 6, different runs of 3DGS on the same scene (with only the view order randomized) result in different artifacts, particularly in uncertain regions. The mutual consistency loss helps suppress these artifacts in both models. On one hand, the shared static regions remain consistent and act as a strong regularizer. On the other hand, the differing artifacts in uncertain areas provide complementary supervision signals, allowing regions affected by artifacts in one model to be recovered by the other.

For the effect of distinct masking strategies, Table 4 in the main paper presents a quantitative comparison. The performance degrades when both models use the same mask, for both GS-GS and EMA-GS settings. This supports our claim that using separate masks helps prevent convergence to the same erroneous reconstruction patterns.

B Comparison with HybridGS

Our Asymmetric Dual 3DGS framework differs fundamentally from HybridGS [13] in both design and training. HybridGS separates static and dynamic content using two models (3DGS for static, 2DGS for dynamic) and requires a staged training process with a learnable blending mask. In contrast, our method uses a dual 3DGS setup with mutual supervision to improve robustness against dynamic noise, all within the standard 3DGS training pipeline. While both methods use masking, HybridGS blends outputs based on 2DGS-derived uncertainty, whereas we apply two distinct masking strategies to reduce confirmation bias from a single, potentially inaccurate mask.

C Multi-cue adaptive mask

Some prior works [19, 18] use residuals between ground-truth and rendered images to detect distractors, assuming static regions are learned first. However, this can misclassify object boundaries and miss distractors resembling the background. Others [14, 17] use pretrained semantic segmentation to mask known distractors, such as people or sky, but these methods rely on task-specific priors and lack generality across diverse scenes. We propose Multi-Cue Adaptive Masking to combine the strengths

of residual-based and segmentation-based methods, while also providing a complementary hard mask that captures distinct error patterns compared to the self-supervised soft mask.

Algorithm 1 Multi-Cue Adaptive Masking

Require: Rendered image $\tilde{\mathbf{I}}$, ground-truth image \mathbf{I} , semantic masks $\{\mathbf{M}_k\}$ from SAM, stereo correspondence map $\mathbf{S}_{\geq 3}$ from COLMAP

- 1: $\mathbf{E}_{\text{pix}} = \|\tilde{\mathbf{I}} - \mathbf{I}\|_1$ ▷ Pixel-level residual
- 2: $\mathbf{F} = \text{DINOv2}(\mathbf{I}); \tilde{\mathbf{F}} = \text{DINOv2}(\tilde{\mathbf{I}})$ ▷ DINOv2 features
- 3: $\mathbf{E}_{\text{feat}} = 1 - \text{CosineSimilarity}(\tilde{\mathbf{F}}, \mathbf{F})$ ▷ Feature-level residual
- 4: $\bar{e}_{\text{pix}} = \sum \mathbf{E}_{\text{pix}} / \text{Area}(\mathbf{I})$ ▷ Average residuals over \mathbf{I}
- 5: $\bar{e}_{\text{feat}} = \sum \mathbf{E}_{\text{feat}} / \text{Area}(\mathbf{I})$
- 6: $\bar{s} = \sum \mathbf{S}_{\geq 3} / \text{Area}(\mathbf{I})$ ▷ Stereo correspondence density over \mathbf{I}
- 7: **for** each mask \mathbf{M}_k **do**
- 8: $e_{\text{pix},k} = \sum \mathbf{M}_k \odot \mathbf{E}_{\text{pix}} / \sum \mathbf{M}_k$ ▷ Average residuals over \mathbf{M}_k
- 9: $e_{\text{feat},k} = \sum \mathbf{M}_k \odot \mathbf{E}_{\text{feat}} / \sum \mathbf{M}_k$
- 10: $s = \sum \mathbf{M}_k \odot \mathbf{S}_{\geq 3} / \sum \mathbf{M}_k$ ▷ Stereo correspondence density over \mathbf{M}_k
- 11: **if** $e_{\text{pix},k} > \bar{e}_{\text{pix}}$ **and** $e_{\text{feat},k} > \bar{e}_{\text{feat}}$ **and** $s < 0.1 \cdot \bar{s}$ **then**
- 12: Mark \mathbf{M}_k as a distractor mask
- 13: **end if**
- 14: **end for**
- 15: **return** $\mathbf{M}_h = 1 - \bigcup \{\mathbf{M}_k\}_{\text{selected}}$ ▷ 0 for distractor

Here, the stereo-based correspondence records the number of matches each pixel in the given image has, based on SIFT feature correspondences proposed in COLMAP [21]. A pixel is considered a valid correspondence (with the stereo correspondence map value set to true at the pixel location) if its match count exceeds a threshold, indicating it likely belongs to a static region. In contrast, distractors typically yield fewer matches due to their limited presence across images. In Algorithm 1, $\mathbf{S}_{\geq 3}$ denotes the stereo correspondence map, where a pixel is considered a valid correspondence if it has more than three matches.

D Datasets and metrics

We evaluate our method on three in-the-wild datasets with varying challenges, as shown in Table 5. NeRF On-the-go dataset [18] features indoor and outdoor sequences with consistent appearance but varying distractor ratios (5%–30%). RobustNeRF dataset [19] provides indoor scenes with static geometry and controlled distractor placement (from single-type to 150 varied distractors), where training is done on cluttered views and testing on clean, unseen ones. We use the undistorted versions of these datasets, following the protocols of WildGaussian [11] and HybridGS [13]. The PhotoTourism dataset [6] includes landmark scenes (Brandenburg Gate, Sacre Coeur, Trevi Fountain) captured under diverse lighting, weather, and viewpoints, with both significant appearance variation and real-world distractors. We report PSNR, SSIM, and LPIPS [27] to assess reconstruction accuracy and perceptual quality.

E Implementation details

Our base model is built on Mip-Splatting [25]. Following its default settings, we recompute the sampling rate of each Gaussian every 100 iterations, with a 2D Mip filter variance of 0.1 and a 3D smoothing filter variance of 0.2. We train for 30,000 iterations on NeRF On-the-go and RobustNeRF, with densification and pruning every 1,000 steps until iteration 15,000; and for 100,000 iterations on PhotoTourism, with densification and pruning every 1,000 steps until iteration 50,000. We omit the opacity reset and apply a 1,000-step warm-up before the mutual consistency regularization begins. The consistency regularization weight is set to 0.1. The learnable mask is optimized by a loss weighted $\lambda_{\text{mask}} = 1.0$ with a learning rate of 0.1. For EMA, we use a smoothing factor of $\beta = 0.8$. Semantic regions for the multi-cue adaptive mask are generated using Semantic SAM [12] to create instance-level segmentations and apply Algorithm 1 to select distractor regions as masks.

Table 5: In-the-wild 3D reconstruction datasets.

Dataset	Scene	# Train	# Test	Distractor	Appear. change
NeRF On-the-go [18]	Patio-high	222	45	~30%	No
	Spot	168	10	~30%	No
	Patio	98	26	15%~20%	No
	Corner	101	20	15%~20%	No
	Fountain	168	17	5%~10%	No
	Mountain	119	12	5%~10%	No
RobustNeRF [19]	Statue	255	19	1 type	No
	Android	122	19	1 type	No
	Yoda	109	202	100 types	No
	Crab	109	194	150 types	No
PhotoTourism [6]	Brandenburg Gate	763	10	~3.5%	Yes
	Sacre Coeur	830	21	~3.5%	Yes
	Trevi Fountain	1689	19	~3.5%	Yes

Additionally, we use a 32-dimensional per-view appearance embedding and a 24-dimensional per-Gaussian embedding. Color transformation is performed using a three-layer MLP with hidden size 128, outputting a scale and bias for each RGB channel. The learning rates are set to 0.001 for the per-view embedding, 0.005 for the per-Gaussian embedding, and 0.0005 for the MLP. The other 3DGS-related hyperparameters follow the setup from origin work shown in Table 6.

Table 6: The other 3DGS-related hyperparameters.

Parameter	Value
position_lr_init	0.00016
position_lr_final	0.0000016
position_lr_delay_mult	0.01
feature_lr	0.0025
opacity_lr	0.1
scaling_lr	0.005
rotation_lr	0.001
percent_dense	0.01
lambda_dssim	0.2
densification_interval	1000
opacity_reset_interval	No opacity reset
densify_from_iter	500
densify_grad_threshold	0.0002

Table 7: The code repo and licenses.

Method	Link	License
3DGS [8]	https://github.com/graphdeco-inria/gaussian-splatting	Custom
Mip-Splatting [25]	https://github.com/autonomousvision/mip-splatting	Custom
WildGaussians [11]	https://github.com/jkulhanek/wild-gaussians/	MIT License
NerfBaselines [10]	https://github.com/nerfbaselines/nerfbaselines	MIT License
COLMAP [21]	https://github.com/colmap/colmap	BSD License
Semantic-SAM [12]	https://github.com/UX-Decoder/Semantic-SAM	Apache 2.0 License
NeRF On-the-go dataset [18]	https://github.com/cvg/nerf-on-the-go	Apache 2.0 License
RobustNeRF dataset [19]	https://robustnerf.github.io/	Custom
PhotoTourism dataset [6]	https://github.com/ubc-vision/image-matching-benchmark	Apache 2.0 License

Table 8: Quantitative results on the NeRF On-the-go dataset [18]. The best and second-best results are highlighted in **bold** and underline, respectively.

Scene	Mountain			Fountain			Corner			Patio			Spot			Patio-High		
Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
RobustNeRF [19]	17.54	0.496	0.383	15.65	0.318	0.576	23.04	0.764	0.244	20.39	0.718	0.251	20.65	0.625	0.391	20.54	0.578	0.366
NeRF On-the-go [18]	20.15	0.644	0.259	20.11	0.609	0.314	24.22	0.806	0.190	20.78	0.754	0.219	23.33	0.787	0.189	21.41	0.718	0.235
3DGS [8]	19.40	0.638	0.213	19.96	0.659	0.185	20.90	0.713	0.241	17.48	0.704	0.199	20.77	0.693	0.316	17.29	0.604	0.363
Mip-Splatting [25]	19.86	0.649	0.200	20.19	0.672	0.189	21.15	0.728	0.230	18.31	0.639	0.328	20.18	0.689	0.338	18.31	0.639	0.328
WildGaussian [11]	20.43	0.653	0.255	20.81	0.662	0.215	24.16	0.822	0.045	21.44	0.800	0.138	23.82	0.816	<u>0.138</u>	22.23	0.725	0.206
SLS-nlp [20]	19.84	0.580	0.294	20.19	0.612	0.258	24.03	0.795	0.258	21.55	0.838	0.065	23.52	0.756	0.185	20.31	0.664	0.259
HybridGS [13]	21.73	0.693	0.284	21.11	0.674	0.252	25.03	0.847	0.151	21.98	0.812	0.169	24.33	0.794	0.196	21.77	0.741	0.211
Ours (GS-GS)	22.00	0.740	0.199	21.83	0.717	<u>0.180</u>	26.15	0.885	0.085	22.97	0.860	0.096	25.52	0.854	0.135	23.17	0.796	<u>0.164</u>
Ours (EMA-GS)	<u>21.93</u>	<u>0.735</u>	0.162	<u>21.61</u>	<u>0.709</u>	0.162	<u>25.77</u>	<u>0.876</u>	0.089	<u>22.87</u>	<u>0.853</u>	<u>0.091</u>	<u>25.09</u>	<u>0.839</u>	0.152	<u>23.14</u>	0.797	0.156

F More results

F.1 NeRF On-the-go and RobustNeRF

In Table 1 and Table 8, our method (GS-GS) outperforms all baseline methods by more than 1 dB in scenes with medium to high occlusion ratios. The margin is smaller in low-occlusion scenes, where 3DGS-based methods already perform well due to strong geometric priors from the initial point cloud. A similar trend is observed in Table 2: while the proposed method surpasses the SOTA by approximately 0.4 dB in simpler scenes containing a single distractor type (e.g., Statue and Android), it outperforms others by more than 1 dB in complex scenes with a large number of diverse distractors (e.g., Yoda and Crab). The rendering results in Figure 7 and 8 further demonstrate the superiority of our method, as competing approaches exhibit distractor remains and missing details.

F.2 PhotoTourism

The Asymmetric Dual 3DGS achieves an average improvement of 0.8 dB on the PhotoTourism dataset (Table 3), demonstrating its effectiveness under challenging appearance variations. Furthermore, proper appearance modeling is essential for handling in-the-wild data with diverse visual conditions. This is supported by a significant performance gap of more than 4 dB between methods with and without appearance modeling, as shown in Table 3, and further illustrated by the visual differences in Figure 9. Therefore, we apply appearance modeling for the PhotoTourism dataset by default. As the importance of appearance modeling is addressed here, we omit further discussion in the following ablation section and apply appearance modeling by default for the PhotoTourism dataset.

F.3 Statistical significance of the main result

Table 9: Quantitative results on the NeRF On-the-go dataset. Each experiment is repeated five times, and we report the mean and standard deviation.

Setting	GS-GS			EMA-GS		
Scene	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
High Occlusion	24.36 \pm 0.02	0.823 \pm 0.001	0.151 \pm 0.001	24.11 \pm 0.05	0.819 \pm 0.002	0.152 \pm 0.004
Medium Occlusion	24.52 \pm 0.06	0.871 \pm 0.001	0.090 \pm 0.001	24.26 \pm 0.08	0.864 \pm 0.001	0.092 \pm 0.002
Low Occlusion	21.99 \pm 0.04	0.730 \pm 0.001	0.184 \pm 0.004	21.81 \pm 0.09	0.723 \pm 0.002	0.166 \pm 0.007

Table 10: Quantitative results on the RobustNeRF dataset. Each experiment is repeated five times, and we report the mean and standard deviation.

Setting	GS-GS			EMA-GS		
Scene	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Statue	23.44 \pm 0.05	0.893 \pm 0.001	0.098 \pm 0.001	23.46 \pm 0.06	0.890 \pm 0.001	0.097 \pm 0.001
Android	25.58 \pm 0.05	0.856 \pm 0.001	0.070 \pm 0.003	25.47 \pm 0.06	0.849 \pm 0.002	0.070 \pm 0.002
Yoda	37.12 \pm 0.09	0.969 \pm 0.001	0.074 \pm 0.001	36.46 \pm 0.06	0.967 \pm 0.001	0.078 \pm 0.001
Crab	36.11 \pm 0.07	0.963 \pm 0.001	0.079 \pm 0.001	35.52 \pm 0.07	0.961 \pm 0.001	0.080 \pm 0.001

We repeated the experiment five times. Based on the results in Table 9 and 10, our method shows statistically significant improvements.

F.4 Hyperparameters

We perform hyperparameter tuning on the NeRF On-the-go dataset [18] to optimize the performance of our method (GS-GS and EMA-GS). As shown in Table 11, we tune the EMA smoothing factor β and find that $\beta = 0.8$ yields the highest PSNR and SSIM with the lowest LPIPS. In Table 12, we evaluate different densification intervals and observe that an interval of 1000 offers the best overall performance. Similarly, Table 13 presents the results of tuning the warm-up interval, where 1000 again emerges as the optimal choice, outperforming both shorter and longer intervals. Lastly, Table 14 shows that removing opacity reset improves reconstruction quality, suggesting that preserving learned opacity leads to more stable and effective training.

Table 11: Tuning the EMA smoothing factor according to the average performance on the NeRF On-the-go dataset [18].

β	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
0.5	22.80	0.797	0.136
0.6	22.93	0.797	0.137
0.7	23.12	0.799	0.136
0.8	23.40	0.801	0.135
0.9	23.05	0.798	0.136

Table 12: Tuning the densification interval according to the average performance on the NeRF On-the-go dataset [18].

Setting	GS-GS			EMA-GS		
Densification Interval	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
500	23.60	0.810	0.129	23.00	0.797	0.134
1000	23.61	0.810	0.135	23.40	0.801	0.135
1500	23.58	0.807	0.146	23.15	0.796	0.143
2000	23.56	0.806	0.152	22.96	0.797	0.145

Table 13: Tuning the warm-up interval according to the average performance on the NeRF On-the-go dataset [18].

Setting	GS-GS			EMA-GS		
Warm-up Interval	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
0	23.55	0.808	0.137	22.96	0.798	0.135
500	23.55	0.809	0.137	23.08	0.799	0.134
1000	23.61	0.810	0.135	23.40	0.801	0.135
1500	23.58	0.809	0.137	23.10	0.799	0.135
2000	23.60	0.810	0.135	22.88	0.798	0.136

In Table 15 and 16, although the best performance is generally achieved at our default setting ($\lambda_m = 1.0$ and $\lambda_{mask} = 1.0$ for GS-GS; $\lambda_m = 0.1$ and $\lambda_{mask} = 1.0$ for EMA-GS), the differences across settings are minimal (less than 0.1 dB). This indicates that the performance is not highly sensitive to the values of λ_m and λ_{mask} .

G Limitations

We adopt the appearance modeling approach from WildGaussian [11], using a per-view appearance embedding to control global appearance and a per-Gaussian embedding to model the appearance of individual Gaussian primitives. However, this model struggles to capture fine-grained effects such as object highlights. A likely reason is the limited diversity in training data. To address this, we plan to introduce data augmentation with randomized illumination variations.

Table 14: Impact of opacity reset on reconstruction quality, evaluated on the NeRF On-the-go dataset [18].

Setting	GS-GS			EMA-GS		
Opacity Reset	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
w/o	23.61	0.810	0.135	23.40	0.801	0.135
w/	22.87	0.790	0.176	22.43	0.786	0.158

Table 15: Performance under varying weights of the mutual consistency loss, evaluated on the NeRF On-the-go dataset [18].

GS-GS				EMA-GS			
λ_m	PSNR	SSIM	LPIPS	λ_m	PSNR	SSIM	LPIPS
0.0	23.13	0.808	0.135	0.0	23.10	0.801	0.132
0.5	23.66	0.810	0.130	0.05	23.39	0.803	0.136
1.0	23.61	0.810	0.135	0.1	23.40	0.801	0.135
1.5	23.54	0.807	0.142	0.2	23.43	0.805	0.133
2.0	23.44	0.803	0.149	0.3	23.47	0.805	0.134

Table 16: Performance under varying weights of the learnable mask loss, evaluated on the NeRF On-the-go dataset [18].

Setting	GS-GS			EMA-GS		
λ_{mask}	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
0.5	23.62	0.809	0.136	23.33	0.802	0.134
1.0	23.61	0.810	0.135	23.40	0.801	0.135
1.5	23.59	0.809	0.137	23.41	0.804	0.135
2.0	23.63	0.811	0.135	23.32	0.802	0.134

H Social impact

Notre-Dame de Paris suffered a devastating fire in 2019. Although the building was severely damaged, restoration was aided by a 3D model originally created for a video game, highlighting the importance of preserving 3D models of cultural landmarks. However, such sites are often crowded with people, and photos taken at different times may exhibit varying lighting conditions. This highlights the broader societal benefit of accessible and robust 3D scene reconstruction technologies. Our method contributes positively by enabling the creation of high-quality 3D models from in-the-wild images, which are often affected by distractors and lighting variations. By making it feasible to reconstruct cultural landmarks from everyday photos, our approach supports digital preservation, education, and historical restoration efforts.

There are potential negative impacts, such as misuse in surveillance or privacy-invading applications. In particular, in-the-wild image collections often contain individuals who are unintentionally captured. To mitigate this risk, we recommend removing or anonymizing identifiable information, such as faces or bodies, from the reconstructed scenes. This can be achieved through automated segmentation or masking techniques applied before or during training.



Figure 7: Qualitative results on the NeRF On-the-go dataset [18]. The scenes shown are, from top to bottom: Patio-high (high occlusion), Spot (high occlusion), Patio (medium occlusion), Corner (medium occlusion), Mountain (low occlusion), and Fountain (low occlusion).

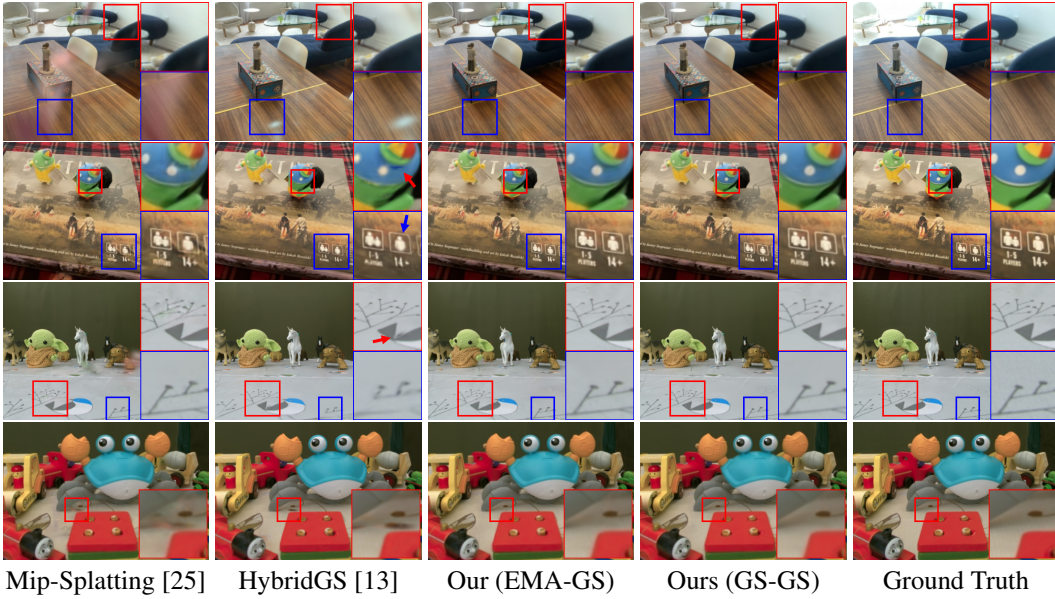


Figure 8: Qualitative results on the RobustNeRF dataset [19]. The scenes shown are, from top to bottom: Statue, Android, Yoda, and Crab.



Figure 9: Qualitative results on the PhotoTourism dataset [6]. The scenes shown are, from top to bottom: Sacre Coeur, Brandenburg Gate, and Trevi Fountain.