Distractor-Aware COLMAP-Free 3D Gaussian Splatting

Namtae Kim Haebeom Jung Jaesik Park
Seoul National University, Republic of Korea
{knt0613, haebeom.jung, jaesik.park}@snu.ac.kr

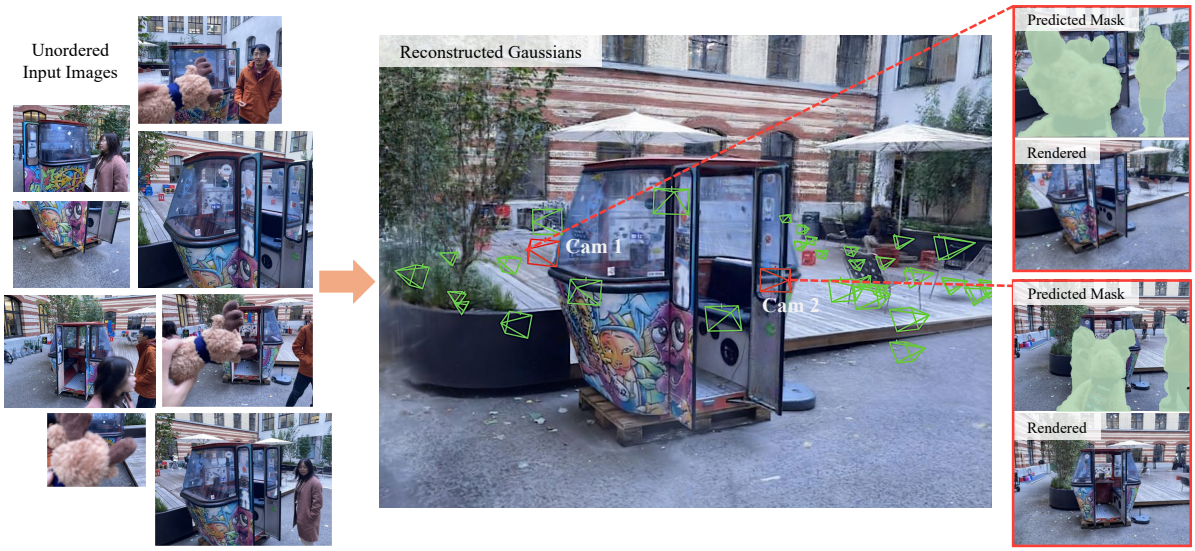


Figure 1. From an in-the-wild image collection that is unordered, unposed, and contaminated by dynamic distractors, our method effectively estimates both a static 3D Gaussian map and camera poses. The illustration shows the reconstructed Gaussians, rendered images, and predicted distractor masks.

Abstract

Pose-free 3D Gaussian Splatting (3DGS) has demonstrated remarkable potential in reconstructing high-fidelity 3D scenes from unposed images. However, current frameworks degrade severely in real-world environments with dynamic distractors, as these transient elements violate the strict photometric consistency required for joint pose-scene optimization and often trap the model in suboptimal local minima. We propose a novel framework that mitigates this ambiguity using a dual coarse-to-fine curriculum. First, we introduce a coarse-to-fine anchor-growing strategy, starting optimization with a sparse, low-frequency anchor grid to stabilize initial pose alignment and geometry. Second, we present a DINOv2-guided uncertainty-aware masking module to dynamically isolate transient objects. The masking module is controlled by a coarse-to-fine uncertainty regularization schedule that initially enforces static-scene assumptions to stabilize poses before progressively allowing dynamic masking. Experiments show that our method reduces pose error among the compared methods in cluttered, unordered captures, while maintaining competitive Novel View Synthesis quality.

1. Introduction

3D Gaussian Splatting (3DGS) [17] has rapidly emerged as a dominant paradigm for high-quality 3D reconstruction, offering an unprecedented combination of photorealistic rendering and real-time performance. However, its success critically depends on accurate camera poses, which are typically obtained through external Structure-from-Motion (SfM) pipelines such as COLMAP [44]. This preprocessing introduces substantial computational overhead and can fail in low-texture regions or repetitive structures, making it a major bottleneck for scalable real-world deployment.

To remove this dependency, recent works have explored *pose-free* 3DGS. Methods such as CF-3DGS [11] jointly optimize scene representations and camera poses without relying on external SfM. However, these approaches [11, 15, 26, 54] typically assume temporally ordered inputs, where strong inter-frame continuity stabilizes optimization. Another line of work leverages learning-based coarse pose initialization [22], followed by joint refinement from unordered image sets. While these approaches [9, 12] relax the need for sequential images, they still implicitly assume static scenes and do not explicitly address dynamic distractors. Meanwhile, distractor-aware 3DGS methods [29, 42, 50] improve robustness by modeling transient objects. However, they rely on accurate camera poses estimated by external SfM pipelines. These limitations leave unordered, pose-free, and distractor-rich image collections largely unresolved.

This gap becomes critical in realistic captures, where images are unordered and scenes are rarely static. Without temporal continuity or external pose supervision, dynamic distractors introduce a structural ambiguity: photometric residuals from pose misalignment become indistinguishable from those caused by transient objects. Even slight pose perturbations can distort photometric losses, making loss-driven dynamic masking unreliable. To address this challenge, we propose **D**istractor-Aware **C**OLMAP-**F**ree **3D** **G**aussian **S**platting (**DCF-3DGS**), a framework that mitigates the ambiguity between pose-induced and distractor-induced residuals in a coarse-initialized, order-agnostic setting. As previewed in Fig. 1, our method reconstructs a static Gaussian scene, camera poses, and distractor masks directly from unordered cluttered images.

To this end, we propose a unified framework based on a dual coarse-to-fine optimization strategy. First, we introduce *coarse-to-fine anchor growing*, which stabilizes geometry and pose alignment by beginning with a sparse, low-frequency anchor representation and progressively increasing model capacity. This prevents premature overfitting to high-frequency artifacts caused by inaccurate poses. Second, we present uncertainty-aware masking with a *coarse-to-fine uncertainty regularization* schedule. A DINOv2 [35]-guided MLP predicts per-pixel observation uncertainty to suppress unreliable photometric supervision, while the regularization ensures that pose stabilization precedes dynamic masking, effectively breaking the chicken-and-egg dependency between pose errors and distractor-induced residuals. We further use a *transient-aware matching prior* and *outlier relocalization* to prevent corrupted initial poses from destabilizing joint optimization.

Our contributions are summarized as follows:

- We present a distractor-aware COLMAP-free 3D Gaussian Splatting framework for coarsely initialized, order-agnostic image reconstruction.
- We mitigate ambiguity between pose-induced and distractor-induced photometric residuals via uncertainty-aware masking and coarse-to-fine scheduling.
- We introduce transient-aware matching and outlier relocalization to improve joint pose-scene convergence under dynamic corruption and imperfect initialization.

2. Related Work

2.1. Novel View Synthesis

Neural Radiance Fields (NeRF) [33] initiated modern Novel View Synthesis (NVS) by implicitly representing scenes with MLPs. This spurred research into improving NeRF's quality [1], speed [10, 34], and data efficiency [13, 59]. More recently, 3D Gaussian Splatting (3DGS) [17] achieved real-time rendering and faster training using an explicit Gaussian representation, which has also seen subsequent quality enhancements [18, 28, 60]. However, a key limitation shared by both NeRF and 3DGS is their strong dependency on accurate camera poses, typically pre-estimated by Structure-from-Motion (SfM) [36, 44].

2.2. Distractor-Aware Novel View Synthesis

Data collected '*in-the-wild*' often contains transient objects (e.g., people, vehicles) that violate static-scene assumptions and degrade reconstruction. NeRF-based methods address this issue through transient modeling, robust losses, uncertainty, or segmentation [4, 30, 40, 41]. This problem has also been studied in 3DGS, where methods handle visual inconsistencies with appearance embeddings [8, 21, 23, 56, 61] or explicitly model transient objects through scene decomposition, diffusion priors, or segmentation [27, 29, 42, 48, 50]. However, most of these methods assume accurate camera poses and treat distractor handling independently from pose estimation, making them brittle when pose errors and transient residuals are entangled.

2.3. Joint Pose and Scene Optimization

To reduce reliance on traditional SfM pipelines [44], research has explored joint optimization of camera poses and scene representations. In NeRF, BARF [25] introduced coarse-to-fine pose optimization, followed by pose-free, registration-based, and camera-optimization methods [2, 5, 7, 14, 19, 37, 49, 53, 55]. In 3DGS, pose-free methods progressively estimate poses from video or ordered image streams [11, 15, 26, 54], while related sparse-view, feed-forward, large-scale, LiDAR-assisted, and robust optimization methods address complementary settings [6, 9, 12, 16, 32, 38, 45]. SLAM-style 3DGS methods, including dynamic-scene variants [24, 31, 39, 43, 57, 58, 63], still assume ordered monocular or RGB-D streams, unlike our unordered setting.

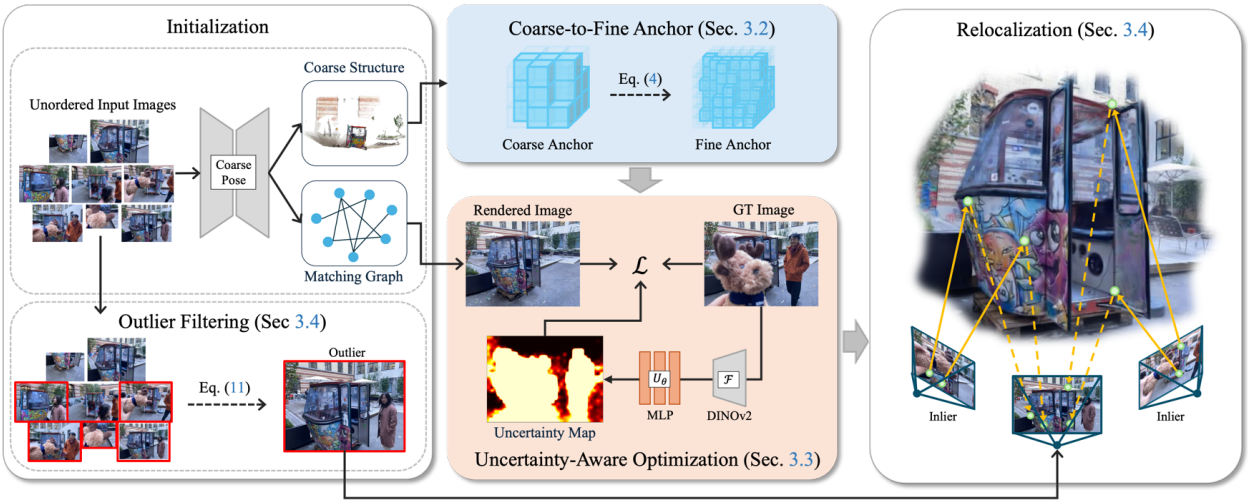


Figure 2. **Overall Pipeline.** From unordered images, we extract a coarse structure and filter unreliable camera poses. The resulting geometry initializes anchor Gaussians that progressively increase in resolution. A DINO-guided MLP masks dynamic distractors under a coarse-to-fine regularization schedule, enabling robust joint pose-scene optimization. Finally, outlier cameras are relocalized using the optimized scene.

3. Distractor-Aware Pose-Scene Optimization

3.1. Preliminaries

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) represents a 3D scene with anisotropic Gaussians parameterized by a mean $\mu \in \mathbb{R}^3$, opacity, color, and covariance Σ decomposed into rotation and scaling components. Given a camera pose $\mathcal{P}$, the projected Gaussians are composited with volumetric alpha blending to produce color $\hat{C}$ and depth $\hat{D}$:

$$\hat{C} = \sum_{k=1}^{K} c_k \, \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j), \quad \hat{D} = \sum_{k=1}^{K} d_k \, \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j), \tag{1}$$

where c_k, α_k, and d_k are the color, opacity, and center depth of the k-th Gaussian, respectively.

Anchor-based Gaussian Splatting. To improve memory efficiency and spatial regularity in large-scale scenes, Scaffold-GS [28] introduces an anchor-based representation, discretizing the scene into sparse voxel anchors. Let x_v denote an anchor center with learnable feature f_v that generates k Gaussians with centers:

$$\{\mu_i\}_{i=0}^{k-1} = x_v + \{O_i\}_{i=0}^{k-1} \cdot l_v, \tag{2}$$

where $\{O_i\}$ are learnable offsets and l_v denotes the voxel scale. Gaussian attributes are decoded from f_v via lightweight MLPs:

$$\{\alpha_i\}_{i=0}^{k-1} = \Phi_\alpha(f_v, \Delta v_c, d_v), \tag{3}$$

where Δv_c is the relative camera-anchor distance and d_v is the viewing direction.

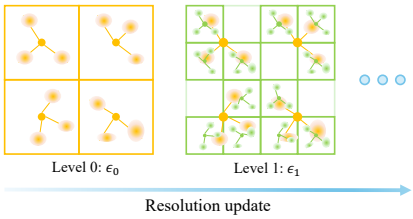


Figure 3. **Coarse-to-fine anchor growing.** We progressively refine the anchor resolution ($\epsilon_{l+1} = \frac{1}{2}\epsilon_l$) to stabilize pose convergence and enable coarse-to-fine learning.

3.2. Coarse-to-Fine Anchor Formulation

While anchor-based Gaussian representations introduce spatial structure, they do not prevent instability during joint pose-geometry optimization. When poses are inaccurate, large rendering residuals may arise even with correct geometry, encouraging unnecessary fine-scale Gaussian growth and leading to distorted structures. To address this issue, we propose a coarse-to-fine anchor formulation that regulates geometric capacity during training. Inspired by the pose convergence principle of BARF [25], we progressively increase spatial resolution, allowing geometry to become more expressive only as pose estimates stabilize. As illustrated in Fig. 3, this design can be interpreted as an implicit pose-aware capacity scheduling mechanism that prevents high-frequency geometric adaptation before pose stabilization.

Coarse-to-fine anchor growing. We adopt a scheduled growing strategy over a multi-resolution voxel hierarchy. Initially, anchors are instantiated only at locations supported by the MASt3R [22] point cloud. Given 3D points P, we voxelize them using a coarse voxel size ϵ_0 and define the

initial anchor set:

$$V = \{v \mid v = \lfloor p/\epsilon_0 \rfloor \cdot \epsilon_0, \forall p \in P\}, \tag{4}$$

with duplicates removed.

Let N be the total training iterations and L the number of resolution levels. We increase resolution every N/L iterations by halving the voxel size, $\epsilon_{l+1} = \frac{1}{2}\epsilon_l$. Anchors from all levels are retained and jointly optimized, so coarse anchors preserve global structure while finer anchors progressively model high-frequency details.

Optimization perspective. The coarse representation restricts the solution space to low-frequency geometry, which is less sensitive to pose perturbations and thus stabilizes early pose convergence. As poses improve, finer anchors gradually expand representational capacity, enabling detailed reconstruction without compensating for pose misalignment through spurious geometry. This delayed capacity growth stabilizes joint pose-scene optimization and limits excessive growth of Gaussians.

3.3. Uncertainty-Aware Joint Optimization

Our goal is to jointly optimize 3D Gaussian Splatting (3DGS) and camera poses from unordered in-the-wild image collections. Dynamic distractors and coarse initial poses coexist, creating a structural ambiguity: photometric residuals caused by pose misalignment are indistinguishable from those introduced by transient objects. This ambiguity destabilizes joint optimization and often leads to pose drift or corrupted geometry. To address this issue, we propose an *uncertainty-aware joint optimization* framework. Our method learns per-pixel observation uncertainty to identify unreliable regions and selectively suppress their influence during optimization. The predicted uncertainty is integrated with a masking strategy tailored to preserve stable 3DGS density control and is further regulated through a coarse-to-fine schedule that prevents early degeneracy. In addition, we incorporate a transient-aware geometric matching prior that provides explicit multi-view constraints for camera pose refinement. Figure 2 summarizes this overall optimization pipeline.

Uncertainty MLP training. We train a lightweight MLP U_θ alongside the 3D Gaussian scene representation $\mathcal{G}$ and camera poses $\mathcal{P}$, while using separate update targets for uncertainty prediction and scene optimization. The MLP U_θ maps DINOv2 [35] features $\mathcal{F}$ to a per-pixel log-uncertainty value, and we set $\beta = \exp(U_\theta(\mathcal{F}))$ to ensure $\beta > 0$. DINOv2 features provide semantically consistent and illumination-robust representations compared with raw RGB inputs. Following NeRF *On-the-go* [40], we adopt an NLL-inspired uncertainty objective with feature-similarity regularization:

$$\mathcal{L}_{\text{uncer}} = \frac{\mathcal{L}_{\text{SSIM}}(\hat{C}, C)}{2\beta^2} + \lambda_{\text{prior}} \log \beta + \lambda_{\text{reg_v}} \operatorname{Var}(\beta) \tag{5}$$

Here, β denotes predicted per-pixel observation uncertainty. A larger $\beta_i(x)$ indicates that the RGB residual at pixel x provides less reliable photometric supervision for the static 3DGS scene due to transient objects, occlusions, or pose-induced misalignment. The regularization term $\log \beta$ prevents degenerate solutions in which the network trivially predicts high uncertainty across all pixels, thereby incorrectly masking the entire scene as distractors. The variance term $\text{Var}(\beta)$ is computed over minibatch neighbors whose DINOv2 features exceed a cosine-similarity threshold, encouraging similar pixels across images to have consistent uncertainty estimates. Note that the rendered image $\hat{C}$ is detached during gradient computation of $\mathcal{L}_{\text{uncer}}$, so that gradients from the uncertainty objective do not affect the optimization of geometry and camera poses.

Dynamic masking for 3DGS. Directly reweighting the main rendering loss by $\frac{1}{\beta^2}$ is incompatible with 3DGS density control, as it alters the gradient statistics that govern adaptive Gaussian splitting. Consider a photometric loss of the form $\mathcal{L} = \frac{1}{2\beta_i^2}||C_i - \hat{C}_i||$. Since 3DGS performs adaptive densification based on the magnitude of these projected 2D gradients $|\nabla_{2D}\mathcal{L}|$, these gradient statistics become directly scaled by $\frac{1}{\beta_i^2}$. Consequently, small predicted uncertainties ($\beta_i \to 0$) artificially inflate gradient magnitudes, leading to premature Gaussian splitting and destabilizing density control by decoupling it from true geometric reconstruction error. Therefore, we convert the uncertainty β into a binary static mask $M = \mathbb{1}(\beta \leq \tau_{\text{mask}})$ using a threshold τ_{mask}. Here, $M_i(x) = 1$ means that pixel x in image i is treated as a reliable static observation and is kept for optimization, while $M_i(x) = 0$ means that the pixel is considered unreliable or dynamic and is ignored. Larger $\beta_i(x)$ therefore indicates lower observation reliability. Thus, $M_i(x) = 0$ should be interpreted as suppressing unreliable photometric supervision, not as assigning a semantic dynamic label. This static mask M is applied to our main photometric loss:

$$\begin{aligned}\mathcal{L}_{\text{render}} = M \odot \Big(&(1 - \lambda_{\text{SSIM}})\mathcal{L}_1(\hat{C}, C) \\ &+ \lambda_{\text{SSIM}}\mathcal{L}_{\text{SSIM}}(\hat{C}, C) \Big).\end{aligned} \tag{6}$$

Here, $\odot$ is the element-wise product. This forces gradients for $\mathcal{G}$ and $\mathcal{P}$ to come only from stable, static pixels, ensuring a clean reconstruction and accurate poses.

Coarse-to-fine uncertainty regularization. Early in training, inaccurate poses cause high $\mathcal{L}_{\text{SSIM}}$ errors even for static pixels (Fig. 4). As a result, U_θ cannot distinguish pose errors from dynamic distractors, leading to a degenerate masking equilibrium where most pixels are suppressed and optimization collapses. To prevent this collapse, we introduce a coarse-to-fine uncertainty regularization schedule that enforces global supervision during early pose refinement. Unlike prior methods that use a fixed uncertainty

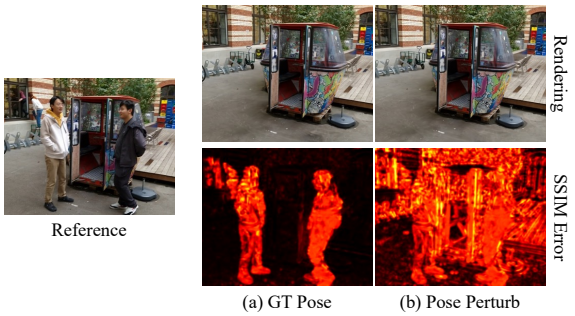


Figure 4. **SSIM loss under pose perturbations.** Comparing SSIM under ground truth and perturbed poses shows that even small pose errors prevent it from separating static components from distractors.

prior, we replace the static weight λ_{prior} in $\mathcal{L}_{\text{uncer}}$ (Eq. (5)) with a linearly decaying schedule $\lambda(n)$:

$$\lambda(n) = \max\left(\lambda_{\text{end}},\, \lambda_{\text{start}} - (\lambda_{\text{start}} - \lambda_{\text{end}})\frac{n}{N}\right). \quad (7)$$

A high λ_{start} initially suppresses uncertainty ($M_i \approx 1$), enforcing dense photometric supervision and enabling coarse pose alignment. As poses stabilize, $\lambda(n)$ gradually decays, relaxing the data-fidelity constraint and allowing uncertainty to emerge only where photometric supervision remains unreliable. U_θ can then mask persistent unreliable observations, such as dynamic distractors, without compromising the overall reconstruction.

Transient-aware matching prior. The photometric loss $\mathcal{L}_{\text{render}}$ alone provides weak and often ambiguous geometric supervision for camera poses. To enforce multi-view geometric consistency, we introduce a transient-aware matching prior. We leverage sparse correspondences from MASt3R initialization. This provides a scene graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ contains images and edges $\mathcal{E}$ connect image pairs (I_n, I_m) with sufficient overlap and pre-computed matches $\mathcal{M}^{n,m}$. We formulate a reprojection loss based on these matches. For a match $(x_n, x_m) \in \mathcal{M}^{n,m}$, we compute the 3D point X_n by unprojecting x_n using the rendered depth $\hat{D}_n(x_n)$ and pose $\mathcal{P}_n$. We then map this point to view I_m with the relative pose $\mathcal{P}_m\mathcal{P}_n^{-1}$ and reproject it to obtain the projected pixel $\hat{x}_m$:

$$\hat{x}_m = \pi(\mathcal{P}_m\mathcal{P}_n^{-1}, \hat{D}_n(x_n), x_n). \quad (8)$$

Crucially, we enforce this constraint only on static pixels. To filter unreliable matches, we use our static mask M_n and denote each edge as $e = (n, m)$:

$$\mathcal{L}_{\text{match}} = \sum_{e\in\mathcal{E}} \sum_{(x_n,x_m)\in\mathcal{M}^e} M_n(x_n)\,|\hat{x}_m - x_m|_1. \quad (9)$$

This matching prior provides robust, geometrically sound gradients for the camera poses $\mathcal{P}$ by ignoring correspondences on dynamic distractors.

3.4. Outlier Filtering and Relocalization

Although MASt3R initialization is generally robust, it can degrade when large dynamic distractors dominate the scene. Such pose outliers bias joint optimization and corrupt scene geometry. To mitigate this issue, we adopt a two-stage strategy: (1) geometry-based outlier filtering and (2) post-optimization relocalization.

Outlier filtering. We first validate the geometric consistency of the initial pose graph $G = (\mathcal{V}, \mathcal{E})$. For each image $I_n \in \mathcal{V}$, we check its consistency against its neighbors I_m, where $(n, m) \in \mathcal{E}$, using the pre-computed matches $\mathcal{M}^{n,m}$. We compute the fundamental matrix F from the initial poses $\mathcal{P}_n, \mathcal{P}_m$ and measure the epipolar error for all correspondences $(x_n, x_m) \in \mathcal{M}^{n,m}$ using the Sampson distance $S_\varepsilon = \varepsilon(x_n, x_m)$:

$$S_\varepsilon = \frac{(x_m^\top F x_n)^2}{(Fx_n)_1^2 + (Fx_n)_2^2 + (F^\top x_m)_1^2 + (F^\top x_m)_2^2}. \quad (10)$$

We aggregate the Sampson distances into a match-normalized image-level score:

$$S_{\varepsilon,\min}(I_n) = \min_{I_m} \frac{1}{|\mathcal{M}^{n,m}|} \sum_{(x_n,x_m)\in\mathcal{M}^{n,m}} S_\varepsilon(x_n, x_m, F). \quad (11)$$

The minimum is taken over neighboring images I_m of I_n.

An image I_n whose camera pose $\mathcal{P}_n$ is not well aligned with its neighboring images yields a high $S_{\varepsilon,\min}(I_n)$. When $S_{\varepsilon,\min}(I_n)$ exceeds the threshold τ_{filter}, the image is considered an outlier. Because the score averages Sampson distances over matches, it is independent of the number of matches in each image pair. These outlier images are excluded from the main joint optimization process, meaning they do not contribute to $\mathcal{L}_{\text{render}}$ (Eq. (6)), $\mathcal{L}_{\text{uncer}}$ (Eq. (5)), or $\mathcal{L}_{\text{match}}$ (Eq. (9)). This ensures that our initial scene representation $\mathcal{G}$ is built only from a set of geometrically consistent camera poses.

Post-optimization relocalization. After convergence on the filtered inlier set, we obtain a refined scene $\mathcal{G}^*$. We then revisit the previously excluded outlier images I_{outlier}, whose initial poses were unreliable due to distractors or insufficient geometric constraints. For each outlier I_{outlier}, we identify its "inlier" neighbors I_m from the original graph G. Using the optimized poses $\mathcal{P}_m$ of these neighbors, we render depth maps $\hat{D}_m$ from the scene $\mathcal{G}$. The stabilized geometry provides a reliable 3D structure for correspondence. We unproject the depth maps and localize each outlier via 3D-2D matching using the original MASt3R correspondences. Additional implementation details are provided in the supplement Sec. A.2.

3.5. Training Details

Initialization. We obtain the initial camera poses and a dense point cloud from MASt3R [22]. This point cloud

	Scenes	COLMAP Initialization									MASt3R Initialization								
		T-3DGS [29]			SpotLessSplats [42]			AsymGS [23]			T-3DGS† [29]			AsymGS† [23]			Ours		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Low	`Mountain`	19.21	0.633	0.283	18.68	0.456	0.510	19.36	0.567	0.472	17.67	0.451	**0.360**	18.69	**0.511**	0.370	**18.77**	0.486	0.424
	`Fountain`	20.45	0.691	0.237	18.92	0.487	0.419	19.38	0.603	0.364	17.65	0.443	**0.362**	**18.10**	**0.452**	0.393	17.81	0.450	0.403
Mid	`Corner`	24.52	0.826	0.204	23.65	0.733	0.273	24.36	0.810	0.227	22.88	0.742	0.231	**24.72**	**0.809**	**0.192**	24.55	0.777	0.269
	`Patio`	20.71	0.775	0.211	21.17	0.741	0.233	15.97	0.587	0.407	20.42	**0.754**	**0.196**	17.88	0.715	0.240	**20.67**	0.734	0.265
High	`Spot`	19.59	0.742	0.285	20.70	0.544	0.499	24.41	0.807	0.213	18.52	0.496	0.381	21.44	**0.688**	**0.353**	**22.43**	0.665	0.393
	`Patio-High`	20.12	0.716	0.268	20.16	0.585	0.357	19.62	0.652	0.346	15.97	0.397	0.461	18.89	0.537	0.390	**20.08**	**0.585**	**0.362**
	Mean	20.77	0.730	0.248	20.55	0.591	0.382	20.52	0.671	0.338	18.85	0.547	0.332	19.95	**0.619**	**0.323**	**20.72**	0.616	0.353

Table 1. **Comparison with distractor-aware approaches.** Under our training-only protocol, all methods in this table use only training images for initialization and scene optimization. Test views are evaluated after the same $\mathrm{Sim}(3)$ registration and camera-only test-time pose optimization with the reconstructed scene fixed. Our method achieves comparable view synthesis performance across multiple scenes. Methods are grouped by their pose and point-cloud initialization. † denotes MASt3R initialization with frozen training poses.

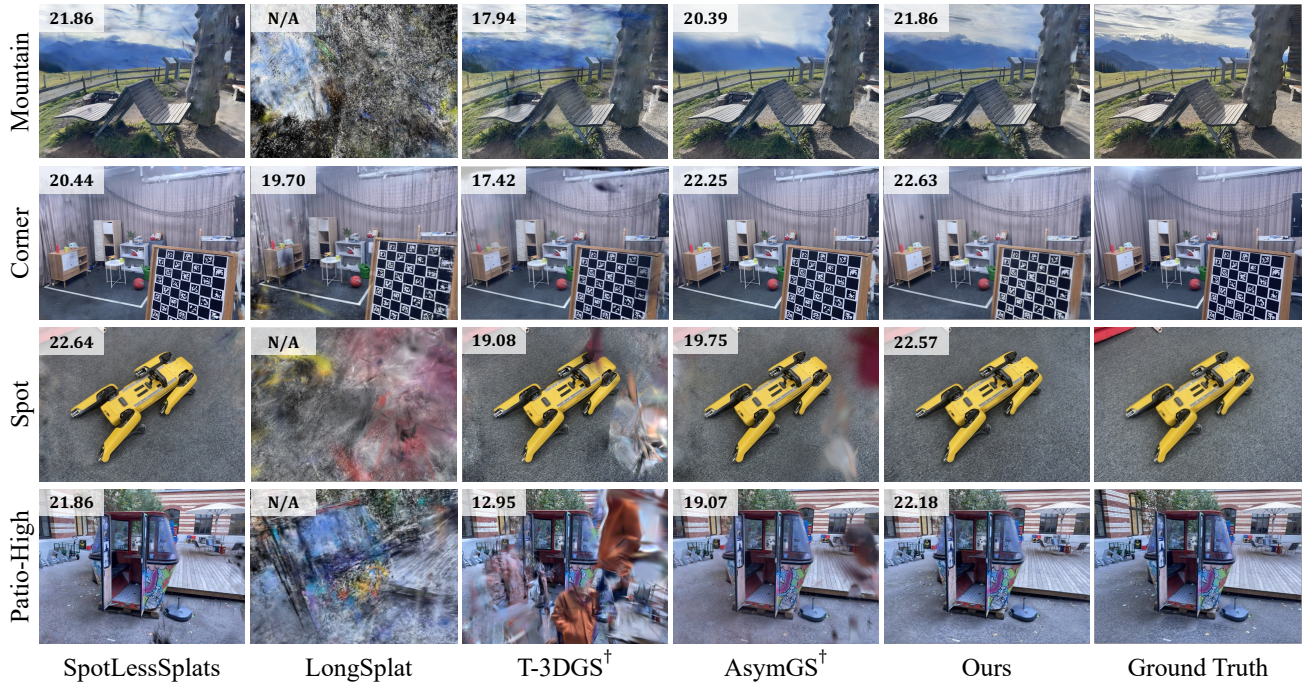


Figure 5. **Rendering of recovered scenes on the NeRF *On-the-go* dataset.** Our method effectively separates distractors from the scene (`Spot`, `Patio-High`), resulting in clean renderings of the static background. PSNR values for each image are shown to the left of the corresponding images.

is used to initialize our anchor Gaussians, following the method proposed in [28]. We initialize the camera intrinsics using the focal lengths estimated by MASt3R.

Scene objective. The scene objective $\mathcal{L}_{\text{scene}}$ used to optimize the Gaussians and camera poses is defined as:

$$\mathcal{L}_{\text{scene}} = \mathcal{L}_{\text{render}} + \lambda_{\text{match}}\mathcal{L}_{\text{match}} + \lambda_{\text{vol}}\mathcal{L}_{\text{vol}}, \qquad (12)$$

where $\mathcal{L}_{\text{vol}}$ is a variant of the volume regularization term proposed in [28], defined as $\mathcal{L}_{\text{vol}} = \sum_{i=1}^{N_{ng}} (\text{Prod}(\mathbf{s}_i) + \text{Mean}(\mathbf{s}_i))$. Here, N_{ng} denotes the number of neural Gaussians in the scene, $\mathbf{s}_i$ is the scaling vector of the i-th Gaussian, and $\text{Prod}(\cdot)$ and $\text{Mean}(\cdot)$ compute the product and mean of the vector's elements. This volume regularization encourages reduced overlap between neural Gaussians. The uncertainty objective $\mathcal{L}_{\text{uncer}}$ in Eq. (5) is optimized separately to update only U_θ and does not contribute gradients to the Gaussians or camera poses. Conversely, $\mathcal{L}_{\text{scene}}$ updates only the scene representation and camera poses.

Refinement. After relocalization, we freeze the camera poses and refine the Gaussians for N_{refine} iterations. The training procedure remains identical to the main optimization, but we incorporate additional signals from the relocalized outlier images to consolidate geometry and appearance. Refinement helps the Gaussians adapt to the improved camera alignment and recover finer structural details.

	Scenes	LongSplat [26]		T-3DGS† [29]		T-3DGS†† [29]		Ours	
		Rot.(°)↓	Trans.↓	Rot.(°)↓	Trans.↓	Rot.(°)↓	Trans.↓	Rot.(°)↓	Trans.↓
Low	Mountain	N/A	N/A	1.168	0.157	1.125	0.147	**0.789**	**0.110**
Low	Fountain	N/A	N/A	0.807	0.120	0.746	0.112	**0.567**	**0.094**
Mid	Corner	**0.043**	**0.024**	0.448	0.044	0.305	0.128	0.104	0.037
Mid	Patio	2.011	1.939	0.186	0.038	0.189	0.036	**0.073**	**0.025**
High	Spot	N/A	N/A	0.777	0.228	0.548	**0.218**	**0.341**	0.241
High	Patio-High	N/A	N/A	1.703	0.110	1.504	0.093	**0.946**	**0.084**
	Mean	N/A	N/A	0.848	0.116	0.736	0.122	**0.470**	**0.098**

Table 2. **Pose accuracy on the NeRF *On-the-go* dataset.** Rot. is rotation error in degrees, and Trans. is absolute trajectory error (ATE) scaled by 100. N/A denotes LongSplat [26] failures. † and †† denote T-3DGS initialized with MASt3R poses and T-3DGS using our pose refinement, respectively. Bold and underline indicate the best and second-best results.

Methods	Input Type	PSNR↑	SSIM↑	LPIPS↓	RPE_t↓	RPE_r↓	ATE↓
3DGS [17]	Unordered	30.20	0.92	0.10	*Considered as GT*		
CF-3DGS [11]	Ordered	31.28	0.93	0.09	0.041	0.069	0.004
HT-3DGS [15]	Ordered	33.53	0.96	0.07	0.041	0.070	0.004
LongSplat [26]	Ordered	32.83	0.94	0.08	0.032	0.068	0.003
Ours	*Unordered*	31.48	0.93	0.09	0.092	0.134	0.001

Table 3. **Quantitative comparison on the Tanks and Temples dataset.** Our unordered-input setting is compared with baselines that assume ordered sequences. 3DGS [17] uses COLMAP [44] poses and is treated as the pose reference.

Scene	LongSplat [26]			Ours		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Corner	19.70	0.675	0.271	**24.55**	**0.777**	**0.269**
Patio	15.24	0.494	0.421	**20.67**	**0.734**	**0.265**
Others	N/A	N/A	N/A	19.77	0.546	0.396

Table 4. **Comparison with LongSplat on the NeRF *On-the-go* dataset.** "Others" groups the scenes where LongSplat *fails* to reconstruct (Mountain, Fountain, Spot, and Patio-High).

4. Experiments

4.1. Experimental Setup

Datasets and Metrics. We evaluate our approach on three benchmarks: NeRF *On-the-go* [40], Tanks and Temples [20], and PhotoTourism [47]. NeRF *On-the-go* contains six casually captured indoor and outdoor sequences with dynamic distractors, 5∼30% occlusion, minimal illumination changes, and clutter-free test views. We resize Patio to 960×540 and the remaining scenes to 960×720. Following Nope-NeRF [2], we use eight Tanks and Temples scenes processed into forward-facing clips and hold out every 8th frame for testing. For PhotoTourism, we use Brandenburg Gate and Trevi Fountain as unordered Internet photo collections, sampling 200 training images per scene and reserving unobstructed views for testing.

We evaluate novel view synthesis quality using PSNR, SSIM [52], and LPIPS [62], and assess training poses against the provided reference poses. For NeRF *On-the-go*, we report rotation error and Absolute Trajectory Error (ATE) scaled by 100, while for Tanks and Temples, we report Relative Pose Error (RPE) and ATE.

Evaluation Protocol. Distractor-aware baselines [29, 42] conventionally use COLMAP [44] poses estimated from both training (cluttered) and test (clutter-free) images of NeRF *On-the-go*. This lets COLMAP exploit clean anchor views that are often unavailable in practice, giving pose-dependent baselines an overly favorable initialization.

Our protocol instead reconstructs each scene only from the cluttered training split, including pose/point-cloud initialization, Gaussian optimization, and any training-pose optimization. Clean test images are reserved for novel-view evaluation and are never used for SfM initialization or scene reconstruction. For evaluation, we estimate a $\mathrm{Sim}(3)$ transform between each method's reconstructed training poses and the provided reference training poses, apply it to the provided test reference poses, and then perform test-time pose optimization following [53]. This optimization minimizes $\mathcal{L}_{\text{render}}$ in Eq. (6) on the clean test image with $M_t = \mathbf{1}$, while the reconstructed Gaussian scene $\mathcal{G}^*$ is frozen and only the test camera pose is optimized. We apply this same test-time pose optimization to all baselines and our method. This protocol evaluates robustness in pose-free and cluttered settings without clean anchor images or ordering assumptions.

Baselines. We compare with distractor-aware methods (T-3DGS [29], SpotLessSplats [42]), pose-free methods (CF-3DGS [11], HT-3DGS [15], LongSplat [26]), and the recent in-the-wild 3DGS baseline AsymGS [23]. Pose-dependent distractor-aware baselines use COLMAP initialization under our protocol. We also evaluate MASt3R-initialized T-3DGS and AsymGS variants, where † denotes MASt3R [22] point clouds and poses with frozen training poses. The supplement reports †† variants that add our joint training-pose optimization. All MASt3R-initialized variants share the same learned initialization and optimization settings as our method. Given that CF-3DGS and HT-3DGS struggle with unordered image collections, such as the NeRF *On-the-go* dataset, their evaluation is restricted to the Tanks and Temples dataset. We note that 3R-GS [12] and DeepGfM [6] are not included in the evaluation, as their preprocessing code or full implementation is not publicly available.

Implementation Details. Each anchor emits $k = 10$ Gaussians from a lightweight two-layer MLP, and the initial sparse voxel size is $\epsilon_0 = 0.1$. We set $\lambda_{\text{SSIM}} = 0.2$, $\lambda_{\text{vol}} = 0.01$, $\lambda_{\text{reg_V}} = 1.0$, $\lambda_{\text{match}} = 1.0$, and $\tau_{\text{mask}} = \tau_{\text{filter}} = 0.1$. Sampson distance is measured in normalized pixel coordinates. We use 30K optimization iterations, 10K refinement iterations, and uncertainty-schedule weights $\lambda_{\text{start}} = 5.0$ and $\lambda_{\text{end}} = 1.0$ for all experiments.

Method	Pose opt.	Brandenburg			Trevi		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
AsymGS†	✗	17.72	0.768	**0.230**	15.83	0.485	**0.321**
AsymGS††	✓	17.76	0.759	0.237	15.77	0.485	0.323
T-3DGS†	✗	15.24	0.648	0.429	14.34	0.402	0.540
T-3DGS††	✓	17.89	0.728	0.256	15.20	0.500	0.339
Ours	✓	**19.16**	**0.789**	0.253	**16.54**	**0.548**	0.366

Table 5. **Novel view synthesis results on PhotoTourism.** All methods are evaluated without appearance modeling. All methods use MASt3R-based camera-pose and point-cloud initialization. Pose opt. indicates joint training-pose optimization.

Models	PSNR↑	SSIM↑	LPIPS↓	Trans.↓	Rot.(°)↓
w/o Coarse-to-fine anchor growing	20.27	0.606	0.354	0.220	0.503
w/o Coarse-to-fine uncertainty reg.	13.84	0.379	0.614	5.713	7.750
w/o Feature-sim. reg. ($\lambda_{\text{reg_V}} = 0$)	20.48	0.610	0.355	0.232	0.526
w/o Joint pose optimization	18.86	0.522	0.426	0.116	0.848
w/o Outlier filtering	20.61	0.614	0.353	0.182	0.591
Ours	**20.72**	**0.616**	**0.353**	**0.098**	**0.470**

Table 6. **Ablation study of training components on the NeRF *On-the-go* dataset.** We evaluate the impact of removing key components.

4.2. Comparisons

NeRF *On-the-go*. Our method achieves competitive NVS quality under training-only pose estimation on NeRF *On-the-go* (Tab. 1). It also achieves the best mean pose accuracy (Tab. 2), and Tab. 4 summarizes its comparison with LongSplat. As shown in Fig. 5, our approach isolates distractors and removes floaters even with coarse pose initialization. In contrast, MASt3R-initialized distractor-aware baselines, especially T-3DGS†, remain sensitive to coarse initial poses. LongSplat fails to reconstruct four of six scenes and produces severe floaters in unordered settings. The full scene-wise NeRF *On-the-go* comparison, including training-pose optimization variants, is provided in Sec. B.1.

PhotoTourism. As shown in Tab. 5, our method remains competitive and improves over MASt3R-initialized AsymGS and T-3DGS variants on both scenes. All PhotoTourism comparisons omit appearance modeling to focus on robustness under coarse initialization and distractor-aware pose-scene optimization. The supplement additionally reports a COLMAP-initialized AsymGS baseline under the same setting in Sec. B.2.

Tanks and Temples. As reported in Tab. 3, our unordered-input setting achieves rendering quality and pose accuracy comparable to sequence-dependent, pose-free methods such as CF-3DGS and HT-3DGS. This result shows that our framework does not require ordered image streams, while remaining competitive on a benchmark commonly used by sequence-based methods.



(a) w/o Coarse-to-Fine Regularization (b) w/ Coarse-to-Fine Regularization

Figure 6. **Effect of coarse-to-fine uncertainty regularization.** Without the schedule, inaccurate poses cause the uncertainty MLP to assign high uncertainty even to static regions, leading to reconstruction failure (a). Our coarse-to-fine regularization stabilizes the joint optimization of the scene and poses (b).

4.3. Ablation Studies

Core Training Components. Table 6 details the quantitative impact of each proposed component. The coarse-to-fine uncertainty regularization plays the most critical role, with its removal causing the largest drop in NVS performance. The feature-similarity regularizer improves multi-view uncertainty consistency, while joint pose optimization improves both rendering and pose accuracy. The supplementary masking-strategy ablation in Tab. G further shows that binary masking is more effective than soft reweighting or no masking, because it blocks unreliable distractor gradients from density control and pose optimization. Additional analyses are provided in the supplement Secs. B.3 to B.6, C.2 and C.3.

Preventing Degeneracy via Uncertainty Regularization. The effect of the coarse-to-fine regularization is most pronounced in distractor-heavy scenes such as `Patio-High` (Fig. 6). Without the schedule, the model assigns high uncertainty even to static regions, suppressing reliable pixels and hindering scene reconstruction and pose refinement. Applying the regularization suppresses transient distractors and promotes stable, coherent joint optimization.

5. Conclusion

We introduced a framework for jointly optimizing camera poses and 3D Gaussians in real-world scenes with many distractors, which often lead to inaccurate initializations such as those from MASt3R. Our method improves pose accuracy while maintaining competitive novel view synthesis quality over prior work.

Limitations. While finer anchors recover details, the coarse-to-fine hierarchy biases optimization toward stable low-frequency structure. This improves robustness to pose and distractor ambiguity but can make fine texture fitting less aggressive than rendering-focused baselines, lowering SSIM and LPIPS. We analyze this behavior in Sec. C.1. Our method is not real-time, and its performance on large-scale scenes remains untested.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 2

[2] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, pages 4160–4169, 2023. 2, 7

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pages 9630–9640. IEEE, 2021. 13

[4] Jiahao Chen, Yipeng Qin, Lingjie Liu, Jiangbo Lu, and Guanbin Li. Nerf-hugs: Improved neural radiance fields in non-static scenes using heuristics-guided segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19436–19446, 2024. 2

[5] Yue Chen, Xingyu Chen, Xuan Wang, Qi Zhang, Yu Guo, Ying Shan, and Fei Wang. Local-to-global registration for bundle-adjusting neural radiance fields. In *ICCV*, pages 8264–8273, 2023. 2

[6] Yu Chen, Rolandos Alexandros Potamias, Evangelos Ververas, Jifei Song, Jiankang Deng, and Gim Hee Lee. Deep gaussian from motion: Exploring 3d geometric foundation models for gaussian splatting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2, 7

[7] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *ECCV*, pages 264–280. Springer, 2022. 2

[8] Hiba Dahmani, Moussab Bennehar, Nathan Piasco, Luis Roldao, and Dzmitry Tsishkou. Swag: Splatting in the wild images with appearance-conditioned gaussians. In *European Conference on Computer Vision*, pages 325–340. Springer, 2024. 2

[9] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2(3):4, 2024. 2

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022. 2

[11] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. COLMAP-Free 3D Gaussian Splatting. In *CVPR*, pages 20796–20805, 2024. 2, 7

[12] Zhisheng Huang, Peng Wang, Jingdong Zhang, Yuan Liu, Xin Li, and Wenping Wang. 3R-GS: Best Practice in Optimizing Camera Poses Along with 3DGS. *arXiv preprint arXiv:2504.04294*, 2025. 2, 7

[13] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5885–5894, 2021. 2

[14] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, pages 5846–5854, 2021. 2

[15] Bo Ji and Angela Yao. SfM-Free 3D Gaussian Splatting via Hierarchical Training. *arXiv preprint arXiv:2412.01553*, 2024. 2, 7

[16] Kaiwen Jiang, Yang Fu, Mukund Varma T, Yash Belhe, Xiaolong Wang, Hao Su, and Ravi Ramamoorthi. A construct-optimize approach to sparse view synthesis without camera pose. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139–1, 2023. 1, 2, 7

[18] Hyunjin Kim, Haebeom Jung, and Jaesik Park. Metropolis-hastings sampling for 3d gaussian reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2

[19] Injae Kim, Minhyuk Choi, and Hyunwoo J Kim. Up-nerf: Unconstrained pose prior-free neural radiance field. *Advances in Neural Information Processing Systems*, 36: 68184–68196, 2023. 2

[20] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM TOG*, 36(4), 2017. 7

[21] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *Advances in Neural Information Processing Systems*, 37:21271–21288, 2024. 2

[22] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, pages 71–91. Springer, 2024. 2, 3, 5, 7, 14

[23] Chengqi Li, Zhihao Shi, Yangdi Lu, Wenbo He, and Xiangyu Xu. Robust neural rendering in the wild with asymmetric dual 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 38:50166–50191, 2026. 2, 6, 7, 13

[24] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pages 163–179. Springer, 2024. 2

[25] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, pages 5741–5751, 2021. 2, 3

[26] Chin-Yang Lin, Cheng Sun, Fu-En Yang, Min-Hung Chen, Yen-Yu Lin, and Yu-Lun Liu. Longsplat: Robust unposed 3d gaussian splatting for casual long videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27412–27422, 2025. 2, 7

[27] Jingyu Lin, Jiaqi Gu, Lubin Fan, Bojian Wu, Yujing Lou, Renjie Chen, Ligang Liu, and Jieping Ye. Hybridgs: Decoupling transients and statics with 2d and 3d gaussian splatting.

In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 788–797, 2025. 2

[28] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2, 3, 6, 12

[29] Alexander Markin, Vadim Pryadilshchikov, Artem Komarichev, Ruslan Rakhimov, Peter Wonka, and Evgeny Burnaev. T-3dgs: Removing transient objects for 3d scene reconstruction. *arXiv preprint arXiv:2412.00155*, 2024. 2, 6, 7, 13, 14

[30] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021. 2

[31] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024. 2, 12

[32] Andreas Meuleman, Ishaan Shah, Alexandre Lanvin, Bernhard Kerbl, and George Drettakis. On-the-fly reconstruction for large-scale novel view synthesis from unposed images. *ACM Transactions on Graphics (TOG)*, 44(4):1–14, 2025. 2

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421. Springer, 2020. 2

[34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 4, 13

[36] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024. 2

[37] Keunhong Park, Philipp Henzler, Ben Mildenhall, Jonathan T Barron, and Ricardo Martin-Brualla. Camp: Camera preconditioning for neural radiance fields. *ACM TOG*, 42(6):1–11, 2023. 2

[38] Jizong Peng, Tze Ho Elden Tse, Kai Xu, Wenchao Gao, and Angela Yao. A constrained optimization approach for gaussian splatting from coarsely-posed images and noisy lidar point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2961–2970, 2025. 2

[39] Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[40] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8931–8940, 2024. 2, 4, 7, 16

[41] Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20626–20636, 2023. 2

[42] Sara Sabour, Lily Goli, George Kopanas, Mark Matthews, Dmitry Lagun, Leonidas Guibas, Alec Jacobson, David Fleet, and Andrea Tagliasacchi. Spotlesssplats: Ignoring distractors in 3d gaussian splatting. *ACM Transactions on Graphics*, 44(2):1–11, 2025. 2, 6, 7, 13

[43] Erik Sandström, Ganlin Zhang, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Youmin Zhang, Manthan Patel, Luc Van Gool, Martin Oswald, and Federico Tombari. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1680–1691, 2025. 2

[44] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 1, 2, 7

[45] Dongbo Shi, Shen Cao, Lubin Fan, Bojian Wu, Jinhui Guo, Renjie Chen, Ligang Liu, and Jieping Ye. Trackgs: Optimizing colmap-free 3d gaussian splatting with global track constraints. *arXiv preprint arXiv:2502.19800*, 2025. 2

[46] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. 13

[47] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 7

[48] Rui Wang, Quentin Lohmeyer, Mirko Meboldt, and Siyu Tang. Degauss: Dynamic-static decomposition with gaussian splatting for distractor-free 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6294–6303, 2025. 2

[49] Weiyao Wang, Pierre Gleize, Hao Tang, Xingyu Chen, Kevin J Liang, and Matt Feiszli. Icon: Incremental confidence for joint pose and radiance field optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5406–5417, 2024. 2

[50] Yihao Wang, Marcus Klasson, Matias Turkulainen, Shuzhe Wang, Juho Kannala, and Arno Solin. Desplat: Decomposed gaussian splatting for distractor-free rendering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 722–732, 2025. 2

[51] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, pages 10481–10497, 2026. 13

[52] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 7

[53] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural Radiance Fields Without Known Camera Parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2, 7

[54] Yu Wei, Jiahui Zhang, Xiaoqin Zhang, Ling Shao, and Shijian Lu. Pcr-gs: Colmap-free 3d gaussian splatting via pose co-regularizations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26499–26508, 2025. 2

[55] Yitong Xia, Hao Tang, Radu Timofte, and Luc Van Gool. SiNeRF: Sinusoidal Neural Radiance Fields for Joint Pose Estimation and Scene Reconstruction. In *BMVC*. BMVA Press, 2022. 2

[56] Jiacong Xu, Yiqun Mei, and Vishal Patel. Wild-gs: Real-time novel view synthesis from unconstrained photo collections. *Advances in Neural Information Processing Systems*, 37:103334–103355, 2024. 2

[57] Yueming Xu, Haochen Jiang, Zhongyang Xiao, Jianfeng Feng, and Li Zhang. Dg-slam: Robust dynamic gaussian splatting slam with hybrid pose optimization. *Advances in Neural Information Processing Systems*, 37:51577–51596, 2024. 2

[58] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024. 2

[59] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 2

[60] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19447–19456, 2024. 2

[61] Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. In *European Conference on Computer Vision*, pages 341–359. Springer, 2024. 2

[62] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[63] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11461–11471, 2025. 2

Distractor-Aware COLMAP-Free 3D Gaussian Splatting

Supplementary Material

A. Implementation Details

We implement our method using PyTorch. Our rendering and 3D Gaussian updates are accelerated with CUDA and cuDNN. All experiments are conducted on NVIDIA RTX 4090 GPUs.

A.1. Pipeline Details and Hyperparameters

Learning rates. We use the Adam optimizer for both camera poses and 3D Gaussian parameters. For camera poses, we set the learning rate to 0.001 for both rotation and translation. For the 3D Gaussians, we adopt the default optimizer settings from the original Scaffold-GS [28] implementation.

Joint optimization. To prevent pose updates from being corrupted by noisy gradients when the scene has not yet been sufficiently reconstructed, we freeze camera pose optimization for the first 1,000 iterations. Similarly, because the uncertainty MLP may receive unstable gradients immediately after anchor growing, we freeze the uncertainty MLP for 20 iterations following each anchor-growing step. For pose updates, we use a modified version of the rasterizer introduced in [31].

Outlier filtering and relocalization. To compute the Sampson error, we use Kornia's epipolar geometry module. For 3D-2D matching, we leverage OpenCV's PnP-RANSAC solver with the SOLVEPNP_ITERATIVE option, using a confidence of 0.99 and a reprojection error threshold of 5.0 pixels. After obtaining the inliers and camera pose from the PnP-RANSAC, we further refine the pose using SciPy's least-squares optimizer with respect to the reprojection error, providing a robust initialization for subsequent optimization. Views classified as outliers are excluded from the main joint optimization so that erroneous pose gradients do not corrupt the reconstructed geometry. After the inlier scene converges, these views are relocalized against the reconstructed scene and used during the final Gaussian refinement.

Refinement. To improve novel-view synthesis quality, we perform a refinement stage in which camera poses are frozen and only the Gaussian parameters are further optimized. This stage is distinct from the test-time pose optimization used for evaluation. During refinement, training camera poses remain fixed and only Gaussian parameters are optimized.

A.2. Algorithm: Relocalization Pipeline

To achieve robust scene reconstruction and camera pose estimation, we employ an outlier filtering and relocalization strategy. Algorithm 1 identifies geometrically inconsistent input views, and Algorithm 2 relocalizes the filtered views after joint optimization. For outlier filtering, we compute a pairwise score as the mean Sampson distance over all MASt3R correspondences between neighboring images. The image-level score is the minimum pairwise score over the neighbors of each image. This match normalization makes the threshold τ_{filter} independent of the number of correspondences in each image pair.

Algorithm 1: Outlier Filtering

Input: Pose graph $G = (\mathcal{V}, \mathcal{E})$, initial poses $\{\mathcal{P}_n\}$, intrinsic K, match sets $\mathcal{M}^{n,m}$, threshold τ_{filter}
Output: Inlier images $\{V_I\}$ and Outlier images $\{V_O\}$

for *each image* $I_n \in \mathcal{V}$ **do**
  $\varepsilon_n \leftarrow \infty$;
  for *each neighbor* I_m *with* $(n, m) \in \mathcal{E}$ **do**
    $F_{n,m} \leftarrow \texttt{Fundamental}(\mathcal{P}_n, \mathcal{P}_m, K)$;
    $E_{n,m} \leftarrow \texttt{MeanSampson}(F_{n,m}, \mathcal{M}^{n,m})$;
    $\varepsilon_n \leftarrow \min(\varepsilon_n, E_{n,m})$;
  if $\varepsilon_n > \tau_{\textit{filter}}$ **then**
    $V_O \leftarrow V_O \cup \{I_n\}$;
  else
    $V_I \leftarrow V_I \cup \{I_n\}$;

Algorithm 2: Relocalization

Input: Pose graph $G = (\mathcal{V}, \mathcal{E})$, optimized scene $\mathcal{G}^*$, inlier poses $\{\mathcal{P}_m\}$, intrinsic K, match sets $\mathcal{M}^{o,m}$, outlier images $\{V_O\}$
Output: Outlier image poses $\{\mathcal{P}_O\}$

for *each image* $I_o \in \{V_O\}$ **do**
  $\{p_{3d}\}, \{p_{2d}\} \leftarrow \emptyset, \emptyset$;
  for *each inlier neighbor* I_m *with* $(o, m) \in \mathcal{E}$ **do**
    $D_m \leftarrow \texttt{RenderDepth}(\mathcal{G}^*, \mathcal{P}_m, K)$;
    for $(x_o, x_m) \in \mathcal{M}^{o,m}$ **do**
      $\{p_{3d}\} \leftarrow \{p_{3d}\} \cup \texttt{Unproject}(D_m, K, \mathcal{P}_m, x_m)$;
      $\{p_{2d}\} \leftarrow \{p_{2d}\} \cup \{x_o\}$;
  $\mathcal{P}_o \leftarrow \texttt{PnP}(K, \{p_{3d}\}, \{p_{2d}\})$;

Method	Init.	Pose opt.	Mountain			Fountain			Corner			Patio			Spot			Patio-H			Mean		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
T-3DGS [29]	C	✗	19.21	0.633	0.283	20.45	0.691	0.237	24.52	0.826	0.204	20.71	0.775	0.211	19.59	0.742	0.285	20.12	0.716	0.268	20.77	0.730	0.248
SpotLessSplats [42]	C	✗	18.68	0.456	0.510	18.92	0.487	0.419	23.65	0.733	0.273	21.17	0.741	0.233	20.70	0.544	0.499	20.16	0.585	0.357	20.55	0.591	0.382
T-3DGS† [29]	M	✗	17.67	0.451	0.360	17.65	0.443	0.362	22.88	0.742	0.231	20.42	0.754	0.196	18.52	0.496	0.381	15.97	0.397	0.461	18.85	0.547	0.332
T-3DGS†† [29]	M	✓	18.18	0.508	0.318	18.10	0.486	0.327	24.32	0.805	0.180	20.62	0.765	0.187	19.45	0.613	0.274	16.75	0.467	0.403	19.57	0.608	0.281
AsymGS [23]	C	✗	19.36	0.567	0.472	19.38	0.603	0.364	24.36	0.810	0.227	15.97	0.587	0.407	24.41	0.807	0.213	19.62	0.652	0.346	20.52	0.671	0.338
AsymGS† [23]	M	✗	18.69	0.511	0.370	18.10	0.452	0.393	24.72	0.809	0.192	17.88	0.715	0.240	21.44	0.688	0.353	18.89	0.537	0.390	19.95	0.619	0.323
AsymGS†† [23]	M	✓	18.79	0.531	0.368	18.14	0.480	0.359	24.85	0.829	0.173	17.69	0.715	0.245	21.59	0.738	0.311	19.25	0.581	0.340	20.05	0.646	0.299
Ours	M	✓	18.77	0.486	0.424	17.81	0.450	0.403	24.55	0.777	0.269	20.67	0.734	0.265	22.43	0.665	0.393	20.08	0.585	0.362	20.72	0.616	0.353

Table A. **Scene-wise NVS comparison on NeRF *On-the-go*.** All methods follow the training-only protocol and share the same $\mathrm{Sim}(3)$ registration and camera-only test-time pose optimization. Init. denotes pose and point-cloud initialization, where C and M indicate COLMAP and MASt3R, respectively. Pose opt. indicates joint training-pose optimization.

B. Additional Experiments

B.1. Full NeRF *On-the-go* Comparison

We provide the full scene-wise NeRF *On-the-go* comparison in Tab. A. This table includes the training-pose optimization variants omitted from the compact main table.

B.2. Additional PhotoTourism Baseline

Table B adds a COLMAP-initialized AsymGS baseline to the MASt3R-initialized comparisons in the main paper. All results in this table omit appearance modeling.

B.3. Feature and Geometry Backbone Ablation

To examine whether DCF-3DGS depends on a specific foundation-model component, we compare DINO variants [3, 35, 46] for uncertainty prediction and replace MASt3R initialization with Pi3 [51] while keeping the rest of our optimization pipeline unchanged. As shown in Tab. C, DINOv2 performs best among the tested DINO variants, while the Pi3-initialized variant further improves the mean rendering metrics. These results suggest that the proposed optimization framework is not tied to a single feature extractor or initialization backbone, although stronger initialization can further improve reconstruction quality.

B.4. Sensitivity to Initialization Quality

We evaluate how our method responds to degraded MASt3R initialization by perturbing initial camera poses and sparsifying the initial point cloud. For pose perturbation, we compose each initial pose with $\exp(\xi_i)$, where ξ_i is Gaussian noise sampled in $\mathfrak{se}(3)$. The Easy, Medium, and Hard settings use rotation noise of $1°$, $3°$, and $5°$, respectively, and translation noise of 0.01, 0.03, and 0.05 normalized by the scene scale. As shown in Tab. D and Fig. A, performance remains stable under mild pose perturbations and degrades gradually as the initialization becomes less accurate. Table E and Fig. A further shows that our method is relatively insensitive to point-cloud sparsification, even when only 10% of the initial points are retained. This robustness comes from our anchor-based Gaussian representation, which voxelizes the MASt3R point cloud into spatial anchors rather than using each input point as an independent Gaussian.

Method	Init.	Pose opt.	texttttBrandenburg			Trevi		
			PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
AsymGS	C	✗	19.56	0.877	0.184	17.14	0.659	0.297
AsymGS†	M	✗	17.72	0.768	0.230	15.83	0.485	0.321
AsymGS††	M	✓	17.76	0.759	0.237	15.77	0.485	0.323
T-3DGS†	M	✗	15.24	0.648	0.429	14.34	0.402	0.540
T-3DGS††	M	✓	17.89	0.728	0.256	15.20	0.500	0.339
Ours	M	✓	19.16	0.789	0.253	16.54	0.548	0.366

Table B. **Additional PhotoTourism comparison.** This table adds a COLMAP-initialized AsymGS baseline to the appearance-free main comparison. Init. denotes pose and point-cloud initialization, where C and M indicate COLMAP and MASt3R, respectively. Pose opt. indicates joint training-pose optimization.

Variant	PSNR↑	SSIM↑	LPIPS↓
Default (MASt3R + DINOv2)	20.720	0.6160	0.3530
MASt3R + DINOv1	20.505	0.6019	0.3630
MASt3R + DINOv3	20.044	0.5817	0.3752
Pi3 + DINOv2	**21.040**	**0.6365**	**0.3425**

Table C. **Feature and geometry backbone ablation.** We report mean novel-view synthesis results on NeRF *On-the-go*. All variants use the same optimization pipeline and evaluation protocol.

Pose Perturbation	PSNR↑	SSIM↑	LPIPS↓
None	**20.695**	**0.6170**	**0.3506**
Easy	20.672	0.6072	0.3583
Medium	19.582	0.5585	0.4133
Hard	18.022	0.4654	0.5054

Table D. **Pose perturbation ablation.** We evaluate robustness to different levels of initial pose perturbation on NeRF *On-the-go*.

Keep Ratio	PSNR↑	SSIM↑	LPIPS↓
100%	**20.695**	**0.6170**	**0.3506**
70%	20.536	0.6020	0.3569
50%	20.427	0.6029	0.3595
30%	19.807	0.5813	0.3708
10%	20.294	0.5905	0.3628

Table E. **Point-cloud sparsification ablation.** We evaluate robustness to different retained ratios of the initial MASt3R point cloud on NeRF *On-the-go*.

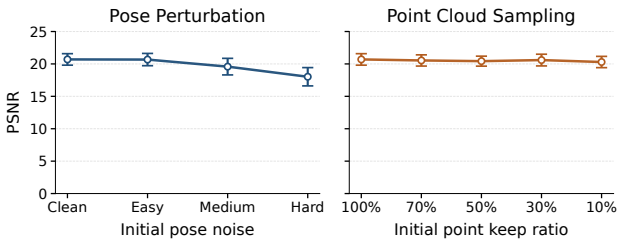


Figure A. **Sensitivity to initialization quality.** We visualize the NVS degradation under pose perturbation and point-cloud sparsification.

B.5. Uncertainty Schedule Ablation

We evaluate different schedules for the uncertainty prior in Eq. (5), with quantitative results reported in Tab. F. Early regularization preserves pose supervision, while gradual relaxation enables distractor masking after geometry stabilizes. Without scheduling, uncertainty is allowed to grow before camera poses stabilize, which causes most pixels to be suppressed and leads to optimization collapse. Fixed schedules either over-regularize uncertainty or allow unstable masking too early, while gradual schedules remain stable. Among them, the linear schedule used in our main model achieves the best PSNR, SSIM, and LPIPS among the tested schedules. As shown in Fig. B, the mask ratio remains conservative early in training and increases only after coarse pose alignment emerges, supporting the role of scheduled uncertainty regularization.

Variant	PSNR↑	SSIM↑	LPIPS↓	Description
None	7.95	0.108	0.712	all-masked collapse
Fixed high	19.33	0.567	0.385	over-regularized
Fixed low	13.12	0.372	0.617	unstable masking
Cosine	20.31	0.597	0.371	stable
Exponential	20.08	0.589	0.381	stable
Linear (Ours)	20.72	0.616	0.353	best metrics

Table F. **Ablation of uncertainty schedules on NeRF *On-the-go*.** We compare schedules for the uncertainty prior in Eq. (5).

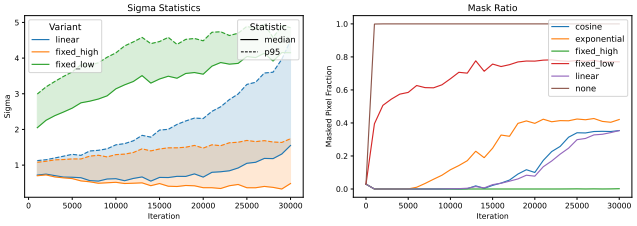


Figure B. **Mask-ratio and uncertainty trajectories on `Patio-High`.** Under the linear schedule, masking remains conservative early in training and becomes more selective after pose alignment stabilizes. The late-stage gap between median and 95th-percentile uncertainty shows that the model assigns high uncertainty mainly to dynamic distractors while keeping most stable regions low-uncertainty.

B.6. Masking Strategy Analysis

We further analyze the binary masking design used in Eq. (6). As reported in Tab. G, soft uncertainty reweighting performs worse than binary masking even though both use the same predicted uncertainty map. The reason is that soft weights still pass gradients from unreliable distractor regions to density control and pose optimization. In contrast, binary masking removes these pixels from the rendering loss, which prevents transient distractors from creating spurious Gaussians or pulling camera poses toward inconsistent image evidence. The no-mask variant shows the opposite failure mode, where all residuals are treated as reliable supervision and reconstruction quality degrades under dynamic clutter.

Variant	PSNR↑	SSIM↑	LPIPS↓	#Gauss.↓	Trans.↓	Rot.(°)↓
Soft reweighting	19.78	0.582	0.341	1.68M	0.393	0.788
No mask	18.90	0.580	0.378	702.2K	0.186	0.490
Binary mask (Ours)	20.72	0.616	0.353	344.3K	0.098	0.470

Table G. **Ablation of masking strategies on NeRF *On-the-go*.** We compare binary masking with soft reweighting and no masking.

B.7. Time Analysis

We compare COLMAP preprocessing with MASt3R-based initialization in Tab. H. Standard MASt3R initialization takes 891.54s. The MASt3R (Fast) variant uses the same initialization model but parallelizes forward passes over 24 image pairs, reducing runtime to 455.18s, below the fastest COLMAP setting of 765.77s. Table I shows that our pipeline reduces total runtime and peak VRAM compared with T-3DGS.

Method	**Initialization**	**Time (s)**
T-3DGS [29]	COLMAP (8 threads)	1102.28
T-3DGS [29]	COLMAP (16 threads)	905.86
T-3DGS [29]	COLMAP (32 threads)	765.77
Ours	MASt3R	891.54
Ours	MASt3R (Fast)	455.18

Table H. **Initialization-time comparison.** We compare with T-3DGS on the NeRF *On-the-go* `Patio-High` scene. T-3DGS uses COLMAP poses, while ours uses MASt3R [22]-based initialization.

Method	Main (min)	Refine (min)	Total (min)	Peak VRAM (GiB)
T-3DGS [29]	62.1	–	73.6	7.56
Ours	22.2	5.5	33.0	2.06

Table I. **Runtime and memory comparison.** We report end-to-end runtime and peak-memory usage on NeRF *On-the-go*. The total runtime includes COLMAP initialization for T-3DGS and MASt3R (Fast) initialization for our method.

C. Additional Visual Comparisons

C.1. Metric Sensitivity to High-Frequency Details

From an optimization perspective, our coarse-to-fine anchor hierarchy stabilizes low-frequency scene structure and camera poses before modeling high-frequency details. This prevents early overfitting to pose- or distractor-induced residuals, but can make fine texture fitting less aggressive than rendering-focused baselines in clean static regions. Figure C compares the ground truth, T-3DGS††, and our method on full images and local patches, with metric scores shown below each rendering. Both methods remove the distractor in the patch and obtain similar reconstruction error, yet structural and perceptual scores differ due to local texture sharpness. On the full image, our method more cleanly separates the distractor from the static scene. This example shows that full-reference metrics can understate distractor separation when differences are dominated by clean-region texture detail.

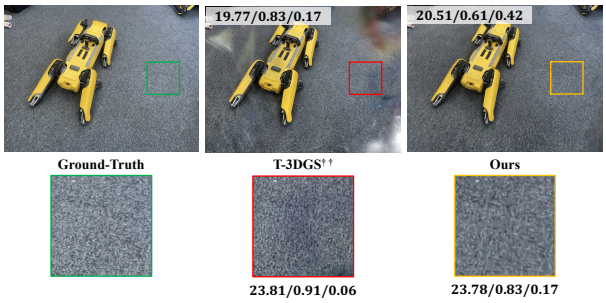


Figure C. **Metric sensitivity to high-frequency details.** Values below images and patches denote PSNR/SSIM/LPIPS. Patch and full-image views show different behavior between metric scores and distractor separation.

C.2. Feature-Similarity Regularization

We visualize the effect of the feature-similarity variance regularizer in Fig. D. Rather than assuming independently extracted DINOv2 features are inherently view-consistent, this term explicitly encourages pixels with similar DINOv2 features across images to have similar uncertainty estimates. Without this regularizer, the predicted uncertainty becomes less consistent across views, which weakens distractor suppression and matches the quantitative drop reported in Tab. 6.

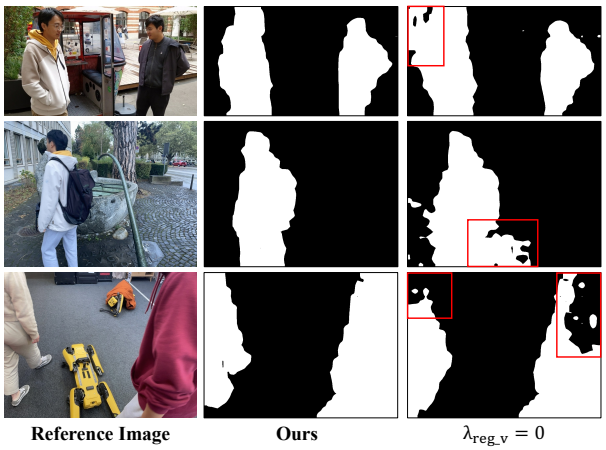


Figure D. **Effect of feature-similarity uncertainty regularization.** Removing $\lambda_{\text{reg_v}}$ weakens multi-view consistency in the predicted uncertainty map.

C.3. Outlier Filtering and Relocalization

Figure E illustrates the importance of outlier filtering and relocalization. Without this step, incorrect initial poses can distort the global geometry and destabilize scene optimization. Filtering unreliable poses prevents such failures and yields a more coherent reconstruction.

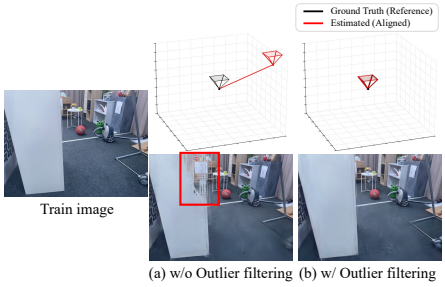


Figure E. **Effect of Outlier Filtering and Relocalization.** Inaccurate poses can corrupt the entire scene, as shown in (a), while applying outlier filtering, as in (b), prevents such failures and stabilizes the reconstruction.

C.4. NeRF *On-the-go* Dataset

Qualitative comparisons on the NeRF *On-the-go* dataset are presented in Fig. F using training viewpoints. Our method clearly distinguishes dynamic distractors while maintaining high reconstruction quality.

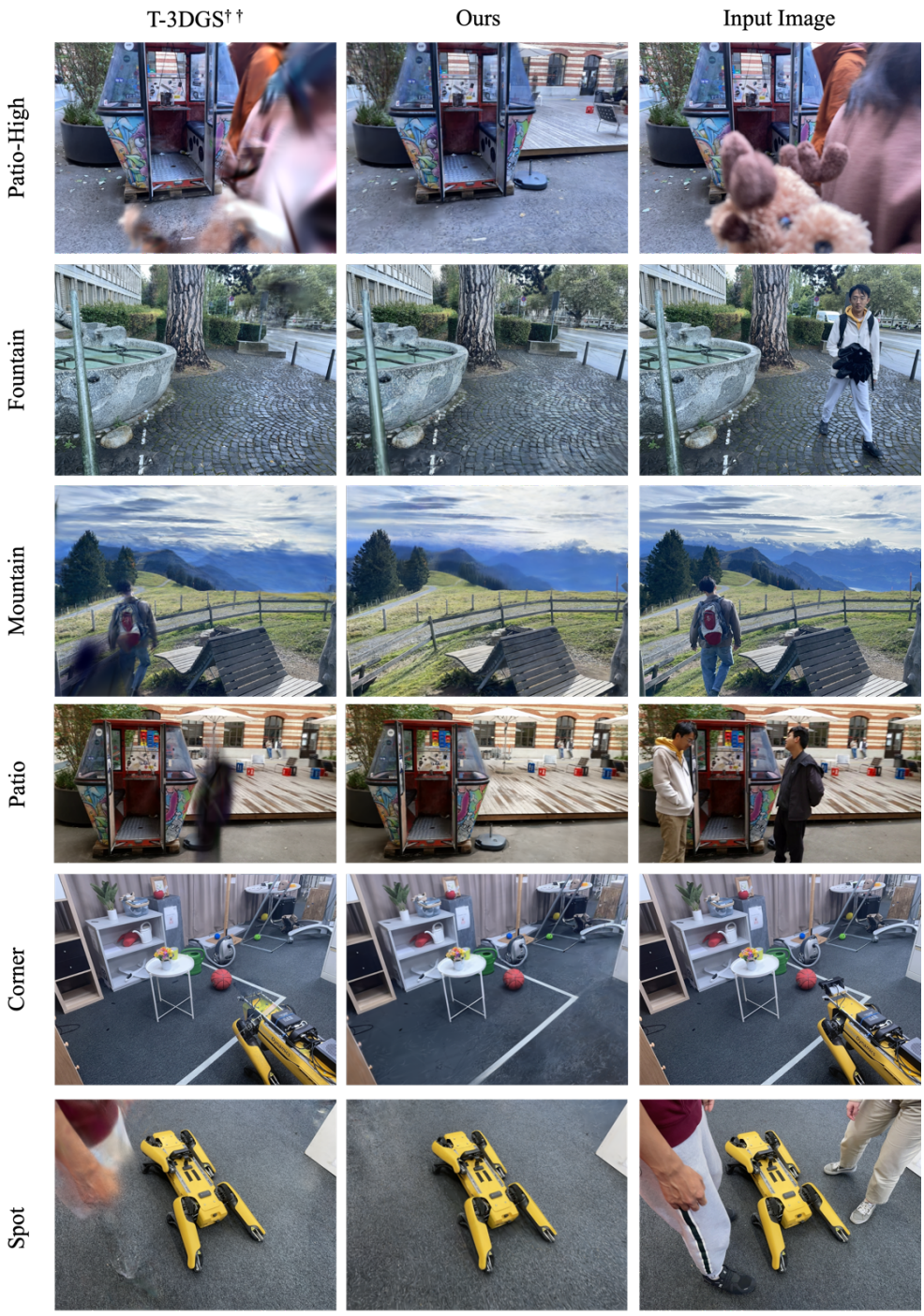


Figure F. **Training-view qualitative comparison.** Qualitative comparison on the NeRF *On-the-go* dataset [40], rendered from the training cameras. The examples show that our method separates transient distractors from the static scene even under noisy training poses.