GAD-SPLAT: GEOMETRY-APPEARANCE DECOUPLED RECONSTRUCTION FOR DISTRACTOR-FREE 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

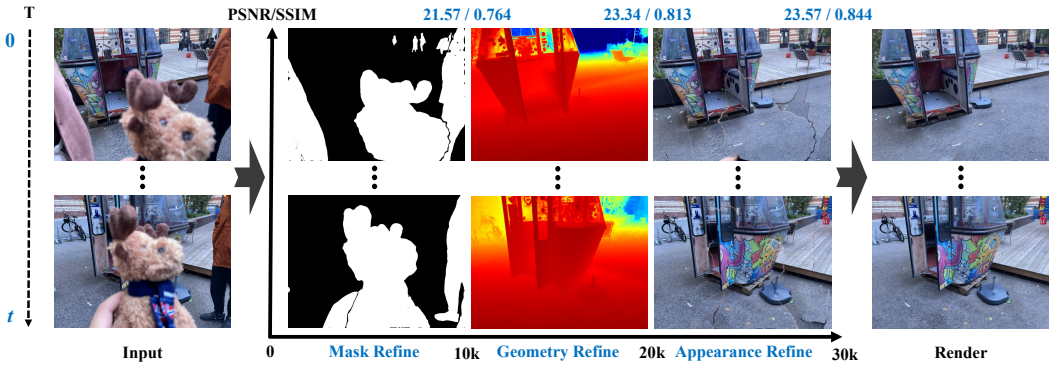

Figure 1: **Three-stage refinement in GAD-Splat.** Mask refinement combines motion cues and rendering residuals to estimate transient masks. Geometry refinement uses *Cross-view Depth Regularization* (CDR) to constrain Gaussian geometry using reference depths from a shared truncated signed distance field (TSDF). With geometry fixed, *Cross-view Appearance Transfer* (CAT) uses unmasked source pixels that satisfy geometric consistency checks to construct auxiliary color targets for masked target pixels. The reported PSNR and SSIM values show gains in reconstruction quality across the three stages.

ABSTRACT

Transient objects disrupt multi-view consistency and limit observations of static scenes, leading to ghosting and blurring artifacts in 3D Gaussian Splatting (3DGS) reconstructions. Existing approaches primarily suppress unreliable supervision or separate static and transient scene components. However, minimizing photometric residuals can still favor incorrect Gaussian configurations that fit observed colors but misrepresent static surfaces, particularly where unmasked observations are limited. We introduce GAD-Splat, which estimates masking probabilities from motion cues and rendering residuals, then selects candidate masks from a segmentation model according to the likelihood of their union. *Cross-view Depth Regularization* (CDR) fuses filtered depth predictions into a shared truncated signed distance field (TSDF). Reference depths from this volume constrain rendered expected depth through surface and free-space losses. With geometry fixed, *Cross-view Appearance Transfer* (CAT) transfers color residuals computed from unmasked source observations and running averages of their renderings. These residuals complement color estimates for the target viewing direction to provide auxiliary supervision for masked pixels. This staged optimization aims to remove transient artifacts while preserving static scene content. Experiments on public benchmarks and self-captured datasets show effective transient-object removal and improved rendered image quality.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Recent advances in 3D Gaussian Splatting (3DGS) have improved the fidelity and efficiency of novel-view synthesis through enhanced Gaussian representations, efficient rasterization, and improved optimization (Kerbl et al., 2023; Yu et al., 2024; Lu et al., 2024; Kheradmand et al., 2024). Its standard formulation assumes that the training images depict a static scene. In casual captures, pedestrians, vehicles, and other transient objects violate this assumption. Fitting these inconsistent observations can introduce spurious Gaussians, leaving ghosting and blurring artifacts in the reconstructed scene (Fu et al., 2025).

Existing methods reduce the influence of transient pixels through robust losses or learned masks (Sabour et al., 2023; Ren et al., 2024; Kulhanek et al., 2024; Sabour et al., 2025), or represent the static scene and transient objects separately (Martin-Brualla et al., 2021; Wang et al., 2025b; Lin et al., 2025). Recent approaches also delay Gaussian growth or refine masks over successive reconstructions (Fu et al., 2025; Seo et al., 2026; Wang et al., 2026). These methods improve distractor removal, but accurate masks alone cannot provide observations of the surfaces hidden behind the distractors. A surface occluded in many images may remain visible in only a few views or from a narrow range of directions.

With limited observations, fitting the visible colors does not necessarily recover the correct geometry (Kulhanek et al., 2024; Li et al., 2024). Photometric optimization jointly updates Gaussian positions, shapes, opacities, and colors, allowing changes in geometry to compensate for errors in appearance. This ambiguity also affects the use of other views to supervise masked regions: scene geometry determines which source pixels correspond to a target pixel. If geometry changes during joint optimization, the correspondences used for supervision can change as well. This motivates constraining the geometry first and then holding it fixed while refining appearance.

We introduce GAD-Splat, which follows this sequence to reconstruct static scenes from images containing transient objects (Figure 1). It first uses motion cues and rendering residuals to estimate which pixels should be masked, then uses these estimates to select candidate masks from a segmentation model. The resulting masks remain fixed during reconstruction.

To constrain geometry, *Cross-view Depth Regularization* (CDR) uses depth predictions from multiple views. Because depth predictions can reflect transient occluders, CDR retains only unmasked depth samples supported by consistent, unmasked samples in another view and fuses them into a shared truncated signed distance field (TSDF). Depths obtained from this volume supervise the rendered expected depth at unmasked pixels, adding geometric constraints to RGB supervision.

Depth constraints, however, do not supply the missing colors in masked regions. With geometry, opacity, and the set of Gaussians fixed, *Cross-view Appearance Transfer* (CAT) uses unmasked source pixels that satisfy geometric consistency checks to supervise masked target regions. It precomputes color estimates for the target viewing direction and constructs separate targets by adding transferred source residuals to running averages of the target renderings. Since Gaussian appearance parameters are shared across views, updates from these targets can also change rendered colors at unmasked pixels. We therefore apply the auxiliary gradients only to zero-order spherical harmonic coefficients and constrain the combined auxiliary gradient using the gradient from unmasked RGB supervision. We make the following contributions:

- We present a framework that reconstructs static scenes from images containing transient objects by first constraining geometry and then refining appearance with geometry fixed.
- We introduce CDR, which filters and fuses depth predictions from multiple views to constrain Gaussian geometry by supervising rendered expected depth.
- We introduce CAT, which constructs complementary color targets from source images and constrains the resulting auxiliary gradients to refine appearance in masked regions while keeping geometry fixed.

2 RELATED WORK

Reconstruction with Transient Objects. Existing methods handle transient objects by modeling them separately or masking and down-weighting inconsistent pixels. NeRF-W adds a transient component to a static radiance field (Martin-Brualla et al., 2021); DeSplat, HybridGS, and De-

Gauss use separate Gaussian representations for static and transient content (Wang et al., 2025b; Lin et al., 2025; Wang et al., 2025a). RobustNeRF uses robust losses, while NeRF On-the-go learns uncertainty maps to weight RGB supervision (Sabour et al., 2023; Ren et al., 2024). WildGaussians and SpotLessSplats use pretrained features to identify transient pixels (Kulhanek et al., 2024; Sabour et al., 2025), and 3DGS-HPC combines photometric and perceptual errors for patch classification (Chen et al., 2026). To account for errors in poorly observed regions, OCSplats adjusts classification using the number and angular diversity of views observing each region (Ling et al., 2025), while Hui et al. (2026) jointly model uncertainty from inconsistent observations and limited view coverage. Distractor identification also depends on how the scene is optimized. MemE exploits the tendency to fit static content before distractors by modeling rendering residuals at multiple scales (Wang et al., 2024), while RobustSplat delays Gaussian growth to limit early fitting of transient content (Fu et al., 2025). PDF-GS progressively updates masks from preceding reconstructions and reinitializes Gaussians during filtering (Seo et al., 2026). DualSplat combines reconstruction residuals, feature discrepancies, and candidate masks from a segmentation model to generate masks for individual objects, then refines them during a second reconstruction (Wang et al., 2026). AsymGS trains two Gaussian scene models with different masks and an asymmetric consistency loss (Li et al., 2025). We use motion cues and rendering residuals to select candidate masks from a segmentation model, then keep the resulting masks fixed for geometry and appearance refinement.

Geometric Regularization. Depth and normal priors provide constraints where RGB observations alone leave geometry ambiguous. FSGS and DNGaussian use monocular depth predictions to regularize reconstruction from sparse views (Zhu et al., 2024; Li et al., 2024), while DN-Splatter combines depth and normal priors (Turkulainen et al., 2025). SuGaR regularizes Gaussians to align with scene surfaces for mesh extraction (Guédon & Lepetit, 2024); 2DGS uses oriented Gaussian surfels with depth distortion and normal consistency losses (Huang et al., 2024). In our setting, depth predictions can reflect transient occluders. CDR therefore filters depth samples using masks and multi-view consistency checks before fusing them into a shared TSDF. The resulting depth maps supervise rendered expected depth at unmasked pixels during geometry refinement. Appearance is then refined with this geometry fixed.

Appearance Modeling and Cross-view Supervision. WildGaussians, SWAG, and GS-W model appearance changes across images through learned color transformations or appearance features (Kulhanek et al., 2024; Dahmani et al., 2024; Zhang et al., 2024). Image-based rendering draws appearance information from source views: IBRNet learns to aggregate source colors and features, while NeuRay estimates visibility to reduce the influence of occluded source features (Wang et al., 2021; Liu et al., 2022). Cross-view supervision is also used in distractor removal. Within a generalizable Gaussian reconstruction framework, DGGS adds reference-view reconstruction losses for regions occluded in the query view but visible in the reference views (Bao et al., 2026). With Gaussian geometry, opacity, and the set of Gaussians fixed, CAT estimates colors for the target viewing direction and transfers color residuals from unmasked source pixels. The resulting color targets provide auxiliary supervision for masked target pixels. The combined auxiliary gradient is restricted to zero-order spherical harmonic coefficients and constrained by the unmasked RGB gradient. Keeping the geometry fixed preserves the correspondences used for appearance supervision.

3 METHOD

GAD-Splat reconstructs a static scene from posed images containing transient objects (Figure 1). We first use motion cues and rendering residuals to select candidate masks from a segmentation model (Section 3.2). With these masks fixed, we refine Gaussian geometry using unmasked RGB supervision and Cross-view Depth Regularization (CDR; Section 3.3). With geometry, opacity, and the set of Gaussians fixed, Cross-view Appearance Transfer (CAT; Section 3.4) constructs auxiliary color targets for masked regions from unmasked source pixels through fixed cross-view correspondences.

3.1 PRELIMINARIES

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) represents a scene as a set of anisotropic 3D Gaussians. Each Gaussian has a center, a covariance matrix parameterized by scales and rotation, an opacity, and spherical harmonic (SH) coefficients for view-dependent color. For a fixed view v

and pixel p, alpha compositing the contributing Gaussians gives

$$C_v(p) = \sum_i w_i \mathbf{f}_i(\boldsymbol{\omega}_i), \qquad w_i = \alpha_i \prod_{j \prec i} (1 - \alpha_j), \tag{1}$$

where α_i is the effective pixel opacity determined by Gaussian i's opacity and projected footprint, and $j \prec i$ indicates that Gaussian j precedes Gaussian i in front-to-back depth order. The SH coefficients define the color function $\mathbf{f}_i$, evaluated in the unit direction $\boldsymbol{\omega}_i$ from the camera to Gaussian i's center.

3.2 Object-level Transient Observation Filtering

We identify transient regions using motion cues and rendering errors from an auxiliary Gaussian reconstruction. These cues determine per-pixel masking probabilities, which guide the selection of whole candidate masks from SAM2 (Ravi et al., 2025) (Figure 2). For each training view, the selected masks form a mask that remains fixed throughout geometry and appearance refinement.

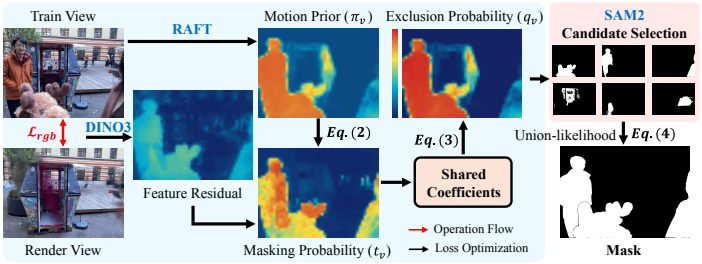

Figure 2: **Transient mask estimation in GAD-Splat.** Motion cues and reconstruction errors determine masking probabilities through parameters shared across views. These probabilities guide the selection of whole SAM2 candidate masks.

Motion and reconstruction errors. For view v, RGB residuals and DINOv3 (Siméoni et al., 2026) feature differences between the observed image and the current rendering are combined into r_v. To combine these residuals with the fixed motion prior π_v from GeoMotion (He et al., 2026), we assign density $S(r; b_v)$ to residuals of static observations and a uniform density to those of transient observations. Bayes' rule then gives the masking probability

$$t_v = \sigma(\operatorname{logit} \pi_v - \log S(r_v; b_v)), \tag{2}$$

where σ is the sigmoid function and logit its inverse. The parameter b_v controls the decay of S and is estimated from the median residual. Large residuals can inflate this estimate and reduce the masking probabilities assigned to the corresponding pixels. We fit a mixture of two Gaussians to the residuals and evaluate t_v using this estimate. The smaller mean is used to estimate b_v only if the fit supports two distinct peaks and the weighted average of t_v for the Gaussian with the larger mean favors retaining those observations.

Sharing information across views. To share information across views, we select K DINO descriptors from the training images as fixed anchors $\{\mathbf{c}_k\}_{k=1}^K$, using farthest-point sampling with cosine distance. Similarities between each pixel's descriptor and these anchors determine the weights for combining coefficients $\{\beta_k\}$ shared across views into the exclusion probability:

$$q_v(x) = \sigma\left(\sum_{k=1}^{K} w_{vk}(x)\beta_k\right), \qquad w_{vk}(x) = \operatorname{softmax}_k\left(\frac{\cos(\mathbf{d}_v(x), \mathbf{c}_k)}{\tau}\right). \tag{3}$$

Here $\mathbf{d}_v(x)$ is the normalized DINO descriptor at location x, and τ is the temperature. We fit q_v to t_v by minimizing the expected KL divergence between their Bernoulli distributions, $\mathbb{E}_{v,x} D_{\mathrm{KL}}[\mathrm{Bern}(t_v(x)) \| \mathrm{Bern}(q_v(x))]$. Only the shared coefficients $\{\beta_k\}$ are optimized, with t_v held fixed during each update and refreshed as the auxiliary reconstruction progresses.

Selecting whole masks. The logits $\operatorname{logit} q_v$ are bilinearly interpolated onto the SAM2 output grid, yielding $\ell_v(y)$ at pixel y. Whole candidate masks $\{\Gamma_{v,k}\}$ from SAM2 are selected using the following objective, where $\mathcal{T}$ denotes a subset of candidate indices:

$$\mathcal{U}_v^{\mathrm{mask}}(\mathcal{T}) = \bigcup_{k \in \mathcal{T}} \Gamma_{v,k}, \qquad F_v(\mathcal{T}) = \sum_{y \in \mathcal{U}_v^{\mathrm{mask}}(\mathcal{T})} \ell_v(y), \qquad \mathcal{T}_v^\star \approx \arg\max_{\mathcal{T}} F_v(\mathcal{T}). \tag{4}$$

Assuming independent pixel masking decisions following Bernoulli distributions, $F_v(\mathcal{T})$ is the log-likelihood gain over retaining all pixels, with each covered pixel counted once. A bounded local

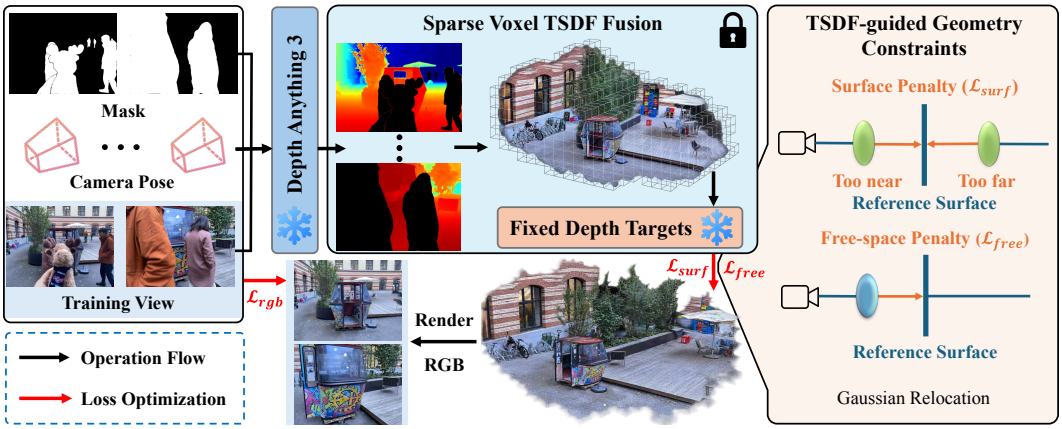

Figure 3: **Cross-view Depth Regularization (CDR).** Unmasked depth samples that agree with unmasked depth samples in another view are fused into a shared TSDF. The resulting reference depths supervise rendered expected depth through surface and free-space losses.

search adds, removes, or replaces whole masks. With q_v fixed, selected masks are refined and additional candidates are generated before the final selection. For direct supervision in the subsequent stages, the binary mask M_v^{keep} retains unmasked pixels with value one and excludes masked pixels with value zero. Further details are provided in Appendix A.2.

3.3 Cross-view Depth Regularization

With the masks fixed, *Cross-view Depth Regularization* (CDR) adds depth constraints to RGB supervision during geometry refinement (Figure 3). Depth Anything 3 (Lin et al., 2026) provides depth predictions, which are scaled to match the SfM reconstruction. These predictions can reflect transient occluders. A depth sample is therefore retained only if its pixel is unmasked and its projection into at least one other view falls on an unmasked pixel with a consistent depth. The retained samples are fused into a shared truncated signed distance field (TSDF) in world coordinates, followed by local smoothing near the reconstructed surfaces. Raycasting this volume yields reference depth maps Z_v for the training views. These maps remain fixed during geometry refinement.

Surface and free-space constraints. For a view v and pixel p, the rendered expected depth is

$$D_v^{\text{render}}(p) = \frac{\sum_i w_i z_i}{\max(\sum_i w_i, \epsilon_{\text{d}})}, \tag{5}$$

where w_i are the compositing weights in Equation 1, z_i is Gaussian i's center depth along the camera z-axis, and $\epsilon_{\text{d}} > 0$ is a numerical stabilizer.

Both depth losses are restricted to the unmasked pixels $\mathcal{K}_v = \{p \in \Omega_v : M_v^{\text{keep}}(p) = 1\}$, where Ω_v is the valid image domain. The free-space loss uses the subset $\mathcal{F}_v$ with valid reference depths, excluding pixels near mask boundaries. The surface loss uses a subset of $\mathcal{F}_v$, denoted $\mathcal{S}_v$: the reference depth must also agree with the Depth Anything 3 prediction at the same pixel, and pixels near reference depth discontinuities are excluded. Thus, $\mathcal{S}_v \subseteq \mathcal{F}_v \subseteq \mathcal{K}_v$. These subsets remain fixed during geometry refinement, and both losses use Z_v as the depth target.

Depth prediction errors and TSDF smoothing can shift the reference surface, making unrestricted depth fitting unreliable near boundaries. We therefore restrict the surface loss to $\mathcal{S}_v$, where the additional checks support penalizing deviations on either side of Z_v. Over the broader $\mathcal{F}_v$, we use the visible reference surface to identify free space in front of it and apply a one-sided depth penalty. The loss remains inactive for depths behind the reference, so it does not pull them toward a potentially inaccurate surface estimate. Averaging over valid pixels gives

$$\begin{aligned} \mathcal{L}_{\text{surf}} &= \mathbb{E}_{p\in\mathcal{S}_v}\left[h_\delta\left(D_v^{\text{render}}(p) - Z_v(p)\right)\right], \\ \mathcal{L}_{\text{free}} &= \mathbb{E}_{p\in\mathcal{F}_v}\left[h_\delta\left([Z_v(p) - D_v^{\text{render}}(p) - \tau_{\text{f}}]_+\right)\right]. \end{aligned} \tag{6}$$

We use the Smooth L1 penalty h_δ, with transition width δ, to reduce sensitivity to large depth errors. The positive part $[x]_+ = \max(x, 0)$ penalizes a rendered depth only when it lies more than τ_{f} in

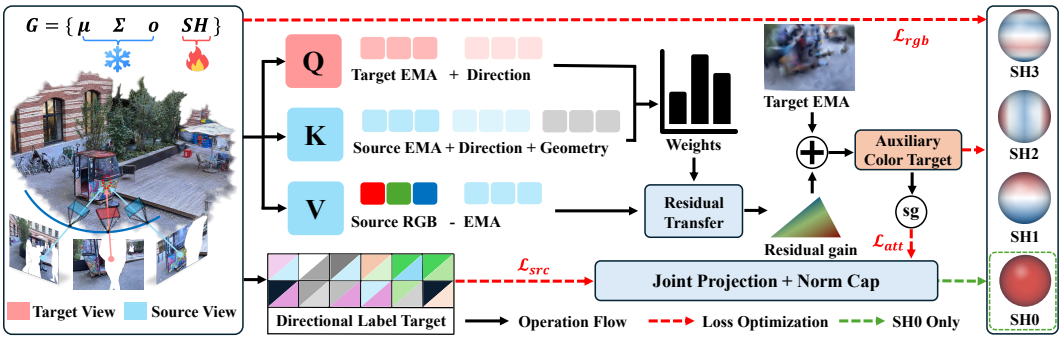

Figure 4: **Cross-view Appearance Transfer (CAT).** With geometry fixed, color estimates and residuals from unmasked source pixels provide auxiliary targets for masked regions. The auxiliary gradients update only zero-order SH coefficients and are constrained by the gradient from unmasked RGB supervision.

front of the reference, allowing small discrepancies. Gradients through Gaussian center depths and compositing weights update positions, covariances, and opacities. These losses constrain rendered expected depth; they do not enforce empty space along the entire ray. Further details are provided in Appendix A.3.

3.4 Cross-view Appearance Transfer

After geometry refinement, Gaussian geometry, opacity, and the set of Gaussians remain fixed. The resulting rendered depth maps D_v^{render} define fixed correspondences between target and source pixels. *Cross-view Appearance Transfer* (CAT; Figure 4) uses these correspondences to construct auxiliary color targets for masked pixels. A source observation is used only if the projection is valid, the source pixel is unmasked, and the rendered source depth agrees with the projected depth.

Estimating color for the target view. For each view, a fixed affine color transformation fitted on unmasked pixels aligns the observed colors with the rendering at the end of geometry refinement. The transformed RGB values are used in both direct and auxiliary supervision. For pixel p in target view t, let $\mathcal{D}_t(p)$ contain the selected source observations, with transformed colors $\mathbf{J}_s$ and confidences ξ_s determined by depth agreement. For each target pixel, we fit a regularized linear function of viewing direction to the source colors. Evaluating this function in the target direction gives $\mathbf{P}_t(p)$:

$$\begin{aligned} \mathbf{U}^\star &= \arg\min_{\mathbf{U}} \sum_{s\in\mathcal{D}_t(p)} \xi_s \left\|\mathbf{U}^\top \mathbf{y}(\boldsymbol{\omega}_s) - \mathbf{J}_s\right\|_2^2 + \lambda_{\mathrm{r}} \|\mathbf{U}\|_F^2, \\ \mathbf{P}_t(p) &= \mathrm{clip}_{[0,1]}\left(\mathbf{U}^{\star\top} \mathbf{y}(\boldsymbol{\omega}_t)\right). \end{aligned} \tag{7}$$

Here $\mathbf{y}(\boldsymbol{\omega}) = [1; \boldsymbol{\omega}]$, $\boldsymbol{\omega}$ is the unit direction from the camera to the corresponding 3D point, and $\lambda_{\mathrm{r}} > 0$ is the regularization weight. The fit is used when enough source observations are available and the system is well conditioned; otherwise, the estimate is a confidence-weighted average of the source colors. These color targets are precomputed and remain fixed during appearance refinement.

Residual transfer. At a fixed set of target pixels, a second color target is constructed by transferring residuals from source views. Let $\mathbf{A}_t, \mathbf{A}_s$ denote exponential moving averages (EMA) of the rendered colors at the target pixel and its corresponding source pixels. The source residual $\mathbf{J}_s - \mathbf{A}_s$ is the difference between the transformed observed color and the source rendering average. Attention combines these residuals over a separately selected source set $\mathcal{J}_t(p)$ with positive confidences ξ_s. A learned linear projection maps $\mathbf{A}_t$ and the target viewing direction to the query $\mathbf{q} \in \mathbb{R}^d$. A separate learned linear projection, shared across source views, computes each key $\mathbf{k}_s \in \mathbb{R}^d$ from $\mathbf{A}_s$, the source viewing direction, depth confidence, and angular similarity to the target. The averages are updated from renderings during appearance refinement and treated as fixed inputs during each parameter update. For $\mathcal{J}_t(p) \neq \emptyset$, the attention weights and resulting color target are

$$\begin{aligned} \pi_s &= \mathrm{softmax}_{s\in\mathcal{J}_t(p)}\left(\mathbf{q}^\top \mathbf{k}_s/\sqrt{d} + \log \xi_s\right), \\ \mathbf{B}_t(p) &= \mathrm{clip}_{[0,1]}\Big(\mathbf{A}_t + \mathbf{g} \odot \sum\nolimits_{s\in\mathcal{J}_t(p)} \pi_s(\mathbf{J}_s - \mathbf{A}_s)\Big). \end{aligned} \tag{8}$$

Here $\mathbf{g} \in [0,1]^3$ scales the transferred residual in each RGB channel and is shared across queries. The query and key projections and $\mathbf{g}$ are optimized using unmasked target pixels by comparing $\mathbf{B}_t$ with their transformed observed RGB. The target $\mathbf{B}_t$ changes as the attention parameters and running averages are updated, while $\mathbf{P}_t$ remains fixed.

Constrained appearance updates. Two auxiliary losses supervise rendered colors at masked pixels: an L1 loss $\mathcal{L}_{\text{src}}$ against $\mathbf{P}_t$, averaged over all masked pixels, and a Charbonnier loss $\mathcal{L}_{\text{att}}$ against $\mathbf{B}_t$, averaged over the fixed set of masked queries. Each pixel's loss is weighted by the reliability of its color target. The color targets and reliability weights are held constant when updating Gaussian parameters. The observed RGB at masked target pixels is excluded from both losses. Auxiliary gradients update only the view-independent, zero-order SH coefficients $\mathbf{a}$, while RGB supervision at unmasked pixels updates all SH coefficients. Since $\mathbf{a}$ is shared across views, auxiliary updates can also change rendered colors at unmasked pixels. We therefore constrain the combined auxiliary gradient using the gradient from unmasked RGB supervision:

$$\begin{aligned} &\boldsymbol{\gamma} = \nabla_{\mathbf{a}} \mathcal{L}_{\text{rgb}}, \qquad \boldsymbol{\delta} = \lambda_{\text{src}} \nabla_{\mathbf{a}} \mathcal{L}_{\text{src}} + \nabla_{\mathbf{a}} \mathcal{L}_{\text{att}}, \\ &\boldsymbol{\nu} = \arg\min_{\mathbf{v}} \tfrac{1}{2} \|\mathbf{v} - \boldsymbol{\delta}\|_2^2, \quad \text{s.t.} \quad \langle \boldsymbol{\gamma}, \mathbf{v} \rangle \geq 0, \quad \|\mathbf{v}\|_2 \leq \|\boldsymbol{\gamma}\|_2. \end{aligned} \tag{9}$$

Here λ_{src} weights the loss against $\mathbf{P}_t$; $\mathcal{L}_{\text{att}}$ includes its own coefficient. The projection is applied jointly to the gradients of all zero-order SH coefficients, before the optimizer uses $\boldsymbol{\gamma} + \boldsymbol{\nu}$ to update them. Source selection, reliability weights, and further details are provided in Appendix A.4.

Our three-stage refinement aims to recover the static scene while limiting the influence of transient observations. After identifying observations to exclude, we focus on constraining the scene geometry needed to establish correspondences with unmasked pixels in other views. These correspondences allow the final stage to supply color supervision at masked target pixels from the available source observations. This separation allows appearance refinement to use cross-view color supervision without changing the geometry on which the correspondences depend.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate GAD-Splat on six scenes from NeRF On-the-go (Ren et al., 2024) and nine self-captured scenes (Section 4.3). The six evaluated scenes are grouped by occlusion level: low (Mountain and Fountain), medium (Corner and Patio), and high (Spot and Patio-High). Separate clean test images serve as references for evaluating novel-view renderings using PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). Scores are averaged over test views within each scene, with equal weight assigned to scenes when computing group and dataset means.

Table 1: Image quality on NeRF On-the-go (Ren et al., 2024). The best, second, and third results are highlighted. PSNR and SSIM are ranked across methods; Baseline scores are taken from the respective papers; * marks scores aggregated from later reports. DualSplat's LPIPS scores are omitted (–) due to a different evaluation protocol.

Method	Low			Medium			High			Mean		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
3DGS* (SIGGRAPH 2023)	20.06	0.681	0.192	19.79	0.747	0.192	18.78	0.677	0.312	19.54	0.702	0.232
WildGaussians (NeurIPS 2024)	20.62	0.669	0.235	22.80	0.811	0.092	23.03	0.771	0.172	22.15	0.750	0.166
SpotLessSplats* (ACM TOG 2025)	21.11	0.676	0.231	23.60	0.838	0.130	22.80	0.781	0.189	22.50	0.765	0.183
DeSplat (CVPR 2025)	19.93	0.695	0.170	23.47	0.845	0.100	24.33	0.870	0.105	22.58	0.803	0.125
HybridGS (CVPR 2025)	21.42	0.684	0.268	23.51	0.830	0.160	23.05	0.768	0.204	22.66	0.760	0.211
RobustSplat (ICCV 2025)	21.08	0.719	0.200	24.03	0.862	0.122	24.54	0.872	0.124	23.22	0.818	0.149
PDF-GS (CVPR 2026)	21.51	0.725	0.190	23.98	0.865	0.120	24.47	0.875	0.125	23.32	0.822	0.145
DualSplat (CVPR 2026)	21.57	0.727	–	24.09	0.863	–	24.61	0.872	–	23.42	0.820	–
GAD-Splat (Ours)	22.04	0.748	0.157	24.26	0.879	0.074	24.76	0.880	0.092	23.68	0.836	0.108

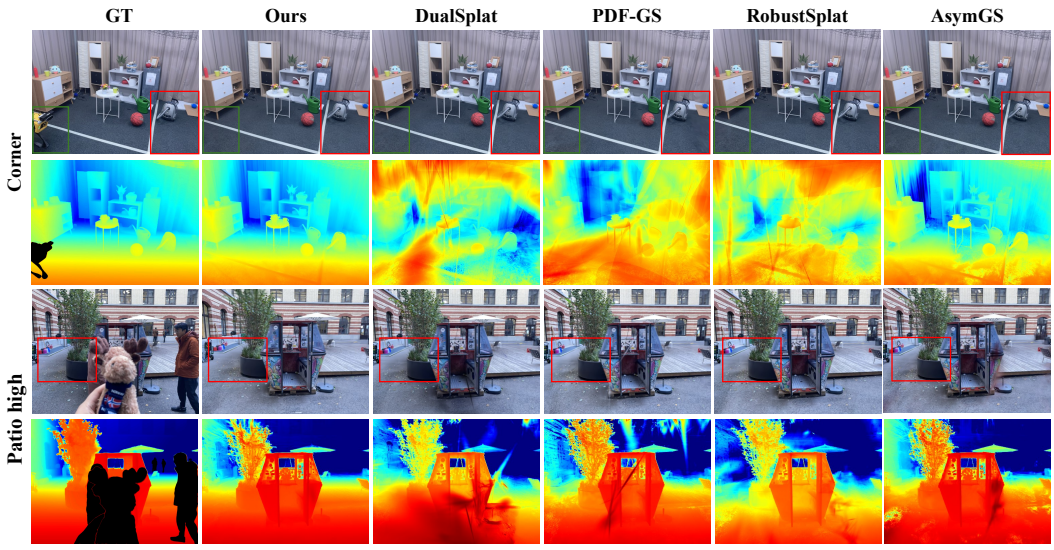

Figure 5: **Qualitative comparisons on NeRF On-the-go.** RGB images and rendered depth maps for Corner (top) and Patio-High (bottom). GAD-Splat removes transient objects while preserving carpet and ground textures in the highlighted regions. GT denotes captured RGB images containing transient objects.

4.2 Comparison on the NeRF On-the-go Dataset

Table 1 compares GAD-Splat with 3DGS (Kerbl et al., 2023), WildGaussians (Kulhanek et al., 2024), SpotLessSplats (Sabour et al., 2025), DeSplat (Wang et al., 2025b), HybridGS (Lin et al., 2025), RobustSplat (Fu et al., 2025), PDF-GS (Seo et al., 2026), and DualSplat (Wang et al., 2026). Among the compared methods, GAD-Splat achieves the highest mean PSNR (23.68 dB) and SSIM (0.836), exceeding the strongest reported baseline for each metric by 0.26 dB (DualSplat) and 0.014 (PDF-GS), respectively. The mean PSNR and SSIM are also highest in each occlusion group, showing improved image quality across the evaluated levels of transient occlusion. Detailed results for individual NeRF On-the-go scenes are provided in Appendix B.1.

The comparisons in Figure 5 show that methods removing the same transient objects can still differ in the texture and rendered depth of the static background. GAD-Splat preserves fine background textures in the highlighted regions and shows clearer object boundaries in the Corner depth rendering. After masking, different scene geometries can fit the remaining RGB observations, while masked pixels receive no direct color supervision. Our staged refinement addresses this ambiguity by constraining geometry before transferring color supervision from unmasked source pixels to masked regions through fixed cross-view correspondences.

4.3 Results on Self-Captured Data

Table 2 shows higher mean PSNR and lower LPIPS for GAD-Splat on our self-captured scenes. Figure 6 highlights differences in removing foreground people. In Scene 1, several baselines retain silhouettes, and DeSplat leaves white fragments on the pavement beside the statues. GAD-Splat removes both people and retains the statues. In Scene 2, the person obscures vegetation and part of the display. Our rendering preserves these elements without the pale remnants visible in DeSplat's result.

Table 2: Image quality on our self-captured dataset of nine scenes. Scores are averaged with equal weight assigned to each scene. The best, second, and third results in each column are highlighted.

Method	PSNR↑	SSIM↑	LPIPS↓
WildGaussians (NeurIPS 2024)	22.66	0.634	0.528
DeSplat (CVPR 2025)	26.62	0.867	0.083
RobustSplat (ICCV 2025)	26.41	0.832	0.097
PDF-GS (CVPR 2026)	25.63	0.824	0.105
DualSplat (CVPR 2026)	26.43	0.833	0.096
GAD-Splat (Ours)	27.27	0.856	0.078

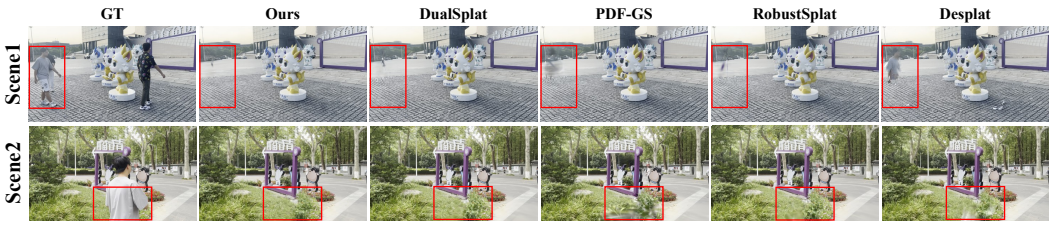

Figure 6: **Qualitative comparisons on our self-captured data.** Red boxes highlight regions for detailed comparison.

4.4 Ablation and Diagnostic Analysis

We examine the contributions of the three stages of GAD-Splat through mask estimation and reconstruction ablations.

Object-level Transient Observation Filtering. Removing the motion prior incorrectly masks static background regions (Figure 7), illustrating the ambiguity of reconstruction errors alone. Removing reconstruction evidence leaves the person's legs incompletely masked. Combining the two cues covers the person while avoiding the isolated background masks. Whole-mask selection follows the person's outline more closely than pixelwise thresholding, retaining boundary details through the candidate masks.

Cross-view Depth Regularization. Using the same predicted masks and training budget, we compare CDR and CAT on the six NeRF On-the-go scenes (Table 3). Removing CDR reduces PSNR by 0.20 dB. Removing the surface loss produces noisy rendered depth, whereas removing the free-space loss leaves less distinct object boundaries (Figure 8). These observations are consistent with their roles in constraining surface depth and penalizing rendered depths in front of the reference surface.

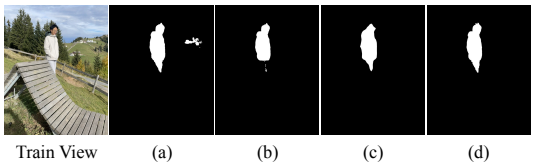

Figure 7: **Transient observation filtering ablations.** (a) w/o motion prior; (b) w/o reconstruction evidence; (c) pixelwise thresholding; (d) full object-level transient observation filtering.

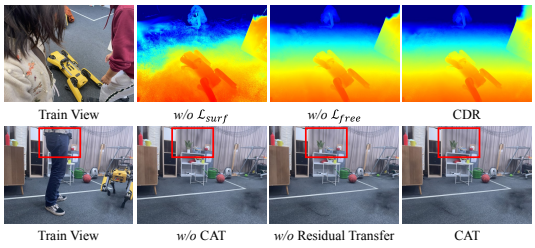

Figure 8: **Geometry and appearance ablations.**

Table 3: Ablation of CDR and CAT.

Configuration	PSNR↑	SSIM↑	LPIPS↓
w/o CDR and CAT	22.60	0.818	0.125
w/o CDR	23.48	0.827	0.116
w/o CAT	23.53	0.826	0.110
Full method	23.68	0.836	0.108

Cross-view Appearance Transfer. Removing CAT reduces PSNR by 0.16 dB (Table 3). Removing CAT or residual transfer leaves less distinct curtain folds in the highlighted region of Figure 8, although the person is removed in all three configurations. Residual transfer complements the fixed color targets with supervision from source rendering residuals. Both CDR and CAT improve all three metrics individually, and their combination gives the best scores, with a PSNR gain of 1.08 dB over the configuration without either component.

5 Conclusion and Limitations

We presented GAD-Splat, a robust framework for transient-object removal and static scene reconstruction. It decouples geometry and appearance refinement, combining depth regularization with cross-view appearance transfer. Experiments on NeRF On-the-go and our self-captured scenes show improved image quality and fewer transient artifacts while preserving static scene detail. However, insufficient unmasked observations or errors in the fixed geometry can limit the reconstruction of occluded static regions.

AI USE STATEMENT

Generative AI tools assisted with code generation and editing, literature search and summarization, English translation and editing, and manuscript and figure formatting. The authors reviewed the AI-assisted material, inspected and tested the code, verified the reported experiments, and take responsibility for the work. AI was not used to generate synthetic data or experimental measurements.

REPRODUCIBILITY STATEMENT

Section 3 describes the three-stage refinement and its training objectives. Appendix A details mask estimation, reference depth construction, and cross-view appearance transfer, including source selection and auxiliary losses. The datasets, evaluation metrics, and score aggregation are described in Section 4, with per-scene results and additional qualitative comparisons in Appendix B. We will release our code upon publication.

REFERENCES

Yanqi Bao, Jing Liao, Jing Huo, and Yang Gao. Distractor-free generalizable 3D Gaussian Splatting. In *International Conference on Learning Representations (ICLR)*, pp. 33344–33365, 2026.

Jiahao Chen, Yipeng Qin, Ganlong Zhao, Xin Li, Wenping Wang, and Guanbin Li. 3DGS-HPC: Distractor-free 3D Gaussian Splatting with hybrid patch-wise classification. In *International Conference on Machine Learning (ICML)*, 2026.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pp. 303–312, 1996. doi: 10.1145/237170.237269.

Hiba Dahmani, Moussab Bennehar, Nathan Piasco, Luis Roldão, and Dzmitry Tsishkou. SWAG: Splatting in the wild images with appearance-conditioned Gaussians. In *Computer Vision – ECCV 2024*, pp. 325–340, 2024. doi: 10.1007/978-3-031-73116-7_19.

Chuanyu Fu, Yuqi Zhang, Kunbin Yao, Guanying Chen, Yuan Xiong, Chuan Huang, Shuguang Cui, and Xiaochun Cao. RobustSplat: Decoupling densification and dynamics for transient-free 3DGS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27126–27136, 2025.

Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned Gaussian Splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5354–5363, 2024.

Xiankang He, Peile Lin, Ying Cui, Dongyan Guo, Chunhua Shen, and Xiaoqin Zhang. GeoMotion: Rethinking motion segmentation via latent 4D geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28145–28155, 2026.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428.

Zeji Hui, Amirali Khodadadian Gostar, WeiQin Chuah, Alireza Bab-Hadiashar, and Ruwan Tennakoon. Joint modeling of corruption-driven and information-limited uncertainty for robust 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 688–697, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D Gaussian Splatting as Markov Chain Monte Carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 80965–80986, 2024. doi: 10.52202/079017-2573.

Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. WildGaussians: 3D Gaussian Splatting in the wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 21271–21288, 2024. doi: 10.52202/079017-0670.

Chengqi Li, Zhihao Shi, Yangdi Lu, Wenbo He, and Xiangyu Xu. Robust neural rendering in the wild with asymmetric dual 3D Gaussian Splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 50166–50191, 2025. doi: 10.52202/085713-1672.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. DNGaussian: Optimizing sparse-view 3D Gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20775–20785, 2024.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, pp. 141261–141285, 2026.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 788–797, 2025.

Han Ling, Xian Xu, Yinghui Sun, and Quansen Sun. OCSplats: Observation completeness quantification and label noise separation in 3DGS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25680–25689, 2025. doi: 10.1109/ICCV51701.2025.02382.

Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7824–7833, 2022.

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 (CVPR)*, pp. 20654–20664, 2024.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. 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 (CVPR)*, pp. 7210–7219, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, pp. 28085–28128, 2025.

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 (CVPR)*, pp. 8931–8940, 2024.

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 (CVPR)*, pp. 20626–20636, 2023.

Sara Sabour, Lily Goli, George Kopanas, Mark Matthews, Dmitry Lagun, Leonidas Guibas, Alec Jacobson, David J. Fleet, and Andrea Tagliasacchi. SpotLessSplats: Ignoring distractors in 3D Gaussian Splatting. *ACM Transactions on Graphics*, 44(2):17:1–17:11, 2025. doi: 10.1145/3727143.

Kangmin Seo, MinKyu Lee, Tae-Young Kim, ByeongCheol Lee, JoonSeoung An, and Jae-Pil Heo. PDF-GS: Progressive distractor filtering for robust 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 468–477, 2026.

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. *Transactions on Machine Learning Research*, 2026.

Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. DN-Splatter: Depth and normal priors for Gaussian Splatting and meshing. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 2421–2431, 2025.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4690–4699, 2021.

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 (ICCV)*, pp. 6294–6303, 2025a.

Xu Wang, Zhiru Wang, Shiyun Xie, Chengwei Pan, and Yisong Chen. DualSplat: Robust 3D Gaussian Splatting via pseudo-mask bootstrapping from reconstruction failures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4912–4921, 2026.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 722–732, 2025b.

Yukun Wang, Kunhong Li, Minglin Chen, Longguang Wang, Shunbo Zhou, Kaiwen Xue, and Yulan Guo. Distractor-free novel view synthesis via exploiting memorization effect in optimization. In *Computer Vision – ECCV 2024*, pp. 477–493, 2024. doi: 10.1007/978-3-031-72949-2_27.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

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 (CVPR)*, pp. 19447–19456, 2024.

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 *Computer Vision – ECCV 2024*, pp. 341–359, 2024. doi: 10.1007/978-3-031-73116-7_20.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: Real-time few-shot view synthesis using Gaussian Splatting. In *Computer Vision – ECCV 2024*, pp. 145–163, 2024. doi: 10.1007/978-3-031-72933-1_9.

A Additional Method Details

A.1 Rendering and coordinate conventions

We use the RGB rendering C_v and rendered expected depth D_v^{render} defined in Equations 1 and 5. RGB values are clipped to $[0, 1]$ before image comparisons, color calibration, and updates of the running averages. For view v, $\mathbf{K}_v$ denotes the camera intrinsics and $(\mathbf{R}_v, \mathbf{t}_v)$ the world-to-camera transformation. The camera center is $\mathbf{c}_v^{\mathrm{cam}} = -\mathbf{R}_v^\top \mathbf{t}_v$, and the viewing direction at a world point $\mathbf{X}$ is $\boldsymbol{\omega}_v(\mathbf{X}) = (\mathbf{X} - \mathbf{c}_v^{\mathrm{cam}})/\|\mathbf{X} - \mathbf{c}_v^{\mathrm{cam}}\|_2$. All depths measure the camera-space z-coordinate. Intrinsics are adjusted to the image grid used in each projection.

A.2 Object-level Transient Observation Filtering

Motion and reconstruction errors. For the observed image and auxiliary rendering at the training resolution, e_v^{rgb} is the absolute RGB residual averaged over channels and pooled onto the DINO feature grid. The corresponding feature difference $e_v^{\mathrm{dino}} = 1 - \cos(\mathbf{d}_v^{\mathrm{obs}}, \mathbf{d}_v^{\mathrm{ren}})$ uses DINOv3 descriptors extracted from these two images. The residual and motion prior in Section 3.2 are

$$r_v = \tfrac{1}{2} e_v^{\mathrm{rgb}} + \tfrac{1}{4} e_v^{\mathrm{dino}}, \qquad \pi_v = 0.1 + 0.8 p_v^{\mathrm{mot}}, \tag{10}$$

where p_v^{mot} is the fixed motion prediction from GeoMotion. The rescaling keeps π_v away from zero and one, allowing reconstruction errors to change the masking probability even when the motion prediction is near zero or one. For $r \in [0, 1]$, the assumed static residual density is

$$S(r; b_v) = 0.05 + 0.95 \frac{\exp(-r/b_v)}{b_v(1 - \exp(-1/b_v))}. \tag{11}$$

The truncated exponential assigns greater density to small residuals, while the uniform component allows larger errors in observations of the static scene. Transient residuals are assigned unit uniform density on the same interval, so their likelihood ratio relative to static residuals is $1/S$. Bayes' rule gives $t_v = \pi_v/[\pi_v + (1 - \pi_v)S(r_v; b_v)]$, whose equivalent sigmoid form is Equation 2.

The median estimate $b_{0,v}$ is obtained from $\mathrm{median}(r_v)/\ln 2$ and clipped to a fixed positive interval. A mixture of two Gaussians is fitted to the residuals. The fit is accepted only if its Bayesian information criterion (BIC) is lower than those of both a single Gaussian and a single exponential, its density has a valley between the two means, and each Gaussian has sufficient effective samples. The t_v values computed using $b_{0,v}$ are averaged with membership probabilities for the Gaussian with the larger mean as weights. If the fit is accepted and this average is below 0.5, the estimate used for b_v is the smaller Gaussian mean, subject to the same bounds; otherwise, it is $b_{0,v}$. The mixture only determines this estimate; S retains the form in Equation 11. The resulting estimates are smoothed across updates by an exponential moving average to obtain b_v.

Sharing information across views. The descriptors $\mathbf{d}_v$ used in Equation 3 are extracted from the original training images, bilinearly interpolated onto the residual grid, and normalized. They remain fixed, as do the anchors selected by farthest-point sampling with cosine distance.

Selecting whole masks. SAM2 produces candidate masks $\{\Gamma_{v,k}\}$ from three sets of point prompts: a uniform grid, locations guided by q_v and distributed across feature and image space, and points along the image boundary. Selection approximately maximizes $F_v(\mathcal{T})$ in Equation 4 through a bounded local search that adds, removes, or replaces whole masks.

With q_v fixed, the selected masks are refined once using positive point prompts inside them and negative point prompts with $q_v < 0.5$ outside their bounding boxes. Additional prompts target uncovered regions with high q_v. The selected masks, their refinements, and the additional candidates are considered together in the final selection under the same objective.

Each selected mask's SAM2 logits are bilinearly interpolated onto the original image grid and thresholded at zero. The union of these binary masks is dilated to form the masked region; its complement within the valid image domain gives M_v^{keep} as defined in Section 3.2.

A.3 Cross-view Depth Regularization

This section describes the construction of the reference depths Z_v and pixel subsets $\mathcal{F}_v$ and $\mathcal{S}_v$ used in Section 3.3.

Depth predictions and scale alignment. We use the unmasked pixel set $\mathcal{K}_v$ defined in the main text and denote the masked pixels by $\mathcal{M}_v = \Omega_v \setminus \mathcal{K}_v$. Depth Anything 3 provides the predicted depth maps D_v. Fusion uses samples on the prediction grid, avoiding repeated contributions from interpolated depths that add no new observations. Scale estimation and the selection of pixels for surface supervision use depths resampled onto the original image grid. A prediction is excluded if its corresponding area in the original image intersects a masked region.

Reference depths are constructed in the SfM coordinate system. The predictions are first aligned to the supplied camera scale. If views are processed in separate groups, residual scale differences are estimated from SfM tracks observed at unmasked pixels. For group g, let d_ℓ and z_ℓ^{SfM} denote a predicted depth and the corresponding sparse point's camera depth. Iteratively reweighted least squares estimates a positive scale κ_g using

$$\kappa_g \leftarrow \frac{\sum_\ell \eta_\ell d_\ell z_\ell^{\mathrm{SfM}}}{\sum_\ell \eta_\ell d_\ell^2}, \tag{12}$$

where η_ℓ combines weights based on track length and reprojection error with robust weights derived from the residual median and median absolute deviation. The estimate is initialized from the weighted median of depth ratios, with track-length weights capped and weights normalized within each view. The same multiplicative scale is applied to both depth grids. An additive depth offset would displace points along different viewing rays without representing a uniform change of scene scale, so the alignment estimates only a positive scale.

Cross-view depth consistency. Hereafter, D_v denotes the aligned prediction and D'_v its counterpart on the original image grid. An unmasked prediction with finite positive depth defines the world point and its projection into view s:

$$\begin{aligned} \mathbf{X}_v(p) &= \mathbf{R}_v^\top \big(D_v(p)\mathbf{K}_v^{-1}\widetilde{\mathbf{p}} - \mathbf{t}_v\big), \\ \mathbf{x}_s &= \mathbf{R}_s\mathbf{X}_v(p) + \mathbf{t}_s, \qquad z_s = [\mathbf{x}_s]_3, \qquad p_s = \mathrm{round}\big(\Pi_s(\mathbf{x}_s)\big), \end{aligned} \tag{13}$$

where $s \neq v$, $\widetilde{\mathbf{p}}$ is the homogeneous pixel coordinate, and Π_s denotes camera projection. A sample is retained for fusion only if at least one other view has a valid projection inside the image with $z_s > 0$, an unmasked pixel p_s, and a finite positive depth satisfying

$$|z_s - D_s(p_s)| \le \tau_{\mathrm{d}}\big(z_s, D_s(p_s)\big), \qquad \tau_{\mathrm{d}}(z,d) = \tau_{\mathrm{a}} + \tau_{\mathrm{r}}\max(z,d), \tag{14}$$

where τ_{a} and τ_{r} are the absolute and relative depth tolerances, respectively, with τ_{a} expressed in reconstruction units. Depths in the other view are sampled at the nearest pixel. This check excludes masked or inconsistent observations before they contribute to the shared reference.

Reference depth construction. The retained samples are fused into a TSDF (Curless & Levoy, 1996) with values $F(\mathbf{X})$ and accumulated weights $W(\mathbf{X})$. Fusion combines the accepted observations in world coordinates, so reference depths for different views are derived from a common surface estimate. The voxel size h is estimated from their median spacing:

$$s_v^{\mathrm{pix}}(p) = D_v(p)\min\{f_{x,v}^{-1}, f_{y,v}^{-1}\}, \qquad h \simeq \mathrm{median}_{(v,p)}\, s_v^{\mathrm{pix}}(p), \qquad \tau_{\mathrm{t}} = 4h, \tag{15}$$

where $f_{x,v}, f_{y,v}$ are focal lengths on the prediction grid and τ_{t} is the truncation distance.

For a voxel projected to pixel $p_v = \lfloor \Pi_v(\mathbf{R}_v\mathbf{X} + \mathbf{t}_v)\rfloor$, let $z_v(\mathbf{X}) = [\mathbf{R}_v\mathbf{X} + \mathbf{t}_v]_3$. A retained observation contributes the projective signed distance

$$\varphi_v(\mathbf{X}) = \min\left\{1, \frac{D_v(p_v) - z_v(\mathbf{X})}{\tau_{\mathrm{t}}}\right\}, \qquad D_v(p_v) - z_v(\mathbf{X}) \ge -\tau_{\mathrm{t}}. \tag{16}$$

Voxels farther behind the observed surface are not updated. Each retained observation contributes unit weight:

$$F(\mathbf{X}) \leftarrow \frac{W(\mathbf{X})F(\mathbf{X}) + \varphi_v(\mathbf{X})}{W(\mathbf{X}) + 1}, \qquad W(\mathbf{X}) \leftarrow W(\mathbf{X}) + 1. \tag{17}$$

The accumulated weights are used for TSDF smoothing and raycasting; the depth losses in Equation 6 average over pixels without these weights.

Volumetric smoothing and raycasting. Before raycasting, local smoothing reduces variation near the reconstructed surfaces. A voxel is smoothed only if $W \ge 1$ and $|F| \le 0.75$. Its six axis-aligned

neighbors contribute if they satisfy the same conditions. For the set of such neighbors $\mathcal{N}(\mathbf{X})$, the update is

$$b_{\mathbf{XY}} = W(\mathbf{Y}) \exp\left[-\frac{(F(\mathbf{Y}) - F(\mathbf{X}))^2}{2\sigma_F^2}\right] \left(1 - n_{r(\mathbf{Y})}(\mathbf{X})^2\right),$$
$$F(\mathbf{X}) \leftarrow \frac{W(\mathbf{X})F(\mathbf{X}) + \sum_{\mathbf{Y}\in\mathcal{N}(\mathbf{X})} b_{\mathbf{XY}} F(\mathbf{Y})}{W(\mathbf{X}) + \sum_{\mathbf{Y}\in\mathcal{N}(\mathbf{X})} b_{\mathbf{XY}}}, \tag{18}$$

where σ_F controls sensitivity to differences in signed distance, $r(\mathbf{Y})$ identifies the neighbor's axis, and $\mathbf{n}$ is the spatial gradient of F, estimated by central differences and normalized with a positive numerical stabilizer. Neighbors with no observations use the center value when estimating this gradient. The weights favor similar signed distances and reduce smoothing along the estimated surface normal. The accumulated weights W remain unchanged.

Along each camera ray $\mathbf{X}_{v,p}(z) = \mathbf{c}_v^{\mathrm{cam}} + z\mathbf{R}_v^\top \mathbf{K}_v^{-1}\widetilde{\mathbf{p}}$, raycasting stops at the first detected crossing from $F_a > 0$ to $F_b \le 0$ with weight at least one at the latter sample. Linear interpolation between their depths z_a, z_b gives

$$\bar{Z}_v(p) \simeq z_a + \frac{F_a}{F_a - F_b}(z_b - z_a). \tag{19}$$

Here $\bar{Z}_v$ is the reference depth in reconstruction units; rays without a valid crossing have no reference depth. Raycasting on the training grid and applying the scene normalization scale $s_{\mathrm{scene}} > 0$ gives the target in Equation 6, $Z_v(p) = s_{\mathrm{scene}}\bar{Z}_v(p)$. Keeping Z_v fixed provides a geometric reference that does not change with the Gaussian parameters being optimized.

Pixels used for depth regularization. Unprimed pixel sets below use the training grid, and primed sets use the original image grid. Let $\mathcal{V}_v$ contain pixels with finite positive reference depths. Excluding mask boundaries gives the free-space subset

$$\mathcal{F}_v = \left(\mathcal{K}_v \setminus \mathrm{Dilate}(\mathcal{M}_v, r_{\mathrm{b}})\right) \cap \mathcal{V}_v. \tag{20}$$

Here dilation uses a disk of radius r_{b}. The same construction gives $\mathcal{F}'_v$ on the original grid. The surface subset additionally requires agreement with the Depth Anything 3 prediction and excludes reference depth discontinuities:

$$\mathcal{S}'_v = \left\{p \in \mathcal{F}'_v :\ 0 < D'_v(p) < \infty,\quad |D'_v(p) - \bar{Z}'_v(p)| \le \tau_{\mathrm{d}}(D'_v(p), \bar{Z}'_v(p))\right\} \setminus \mathcal{E}'_v. \tag{21}$$

Here $\bar{Z}'_v$ is the reference depth on the original grid, and $\mathcal{E}_v, \mathcal{E}'_v$ mark depth discontinuities on the two grids. When adjacent horizontal or vertical reference depths are both finite and positive and differ by more than τ_{d} from Equation 14, both pixels are marked and the marked region is dilated by one pixel. These checks use reconstruction units.

Membership in $\mathcal{S}'_v$ is transferred to the training grid along the same camera ray by nearest-neighbor sampling, denoted $\mathcal{P}_v$. With intrinsics $\mathbf{K}'_v$ for the original images and $\mathbf{K}_v$ for the training grid,

$$\widetilde{\mathbf{p}}' \sim \mathbf{K}'_v \mathbf{K}_v^{-1}\widetilde{\mathbf{p}}, \qquad \mathcal{S}_v = \left(\mathcal{F}_v \cap \mathcal{P}_v(\mathcal{S}'_v)\right) \setminus \mathcal{E}_v. \tag{22}$$

The mask, reference depth validity, and depth discontinuities are checked again on the training grid, preserving $\mathcal{S}_v \subseteq \mathcal{F}_v \subseteq \mathcal{K}_v$. Both losses use Z_v; the predictions D'_v determine where the surface loss is applied.

Geometry refinement objective. Each average in Equation 6 includes only pixels with finite rendered depth D_v^{render} and finite positive reference depth Z_v, and is zero if no valid pixels remain. The free-space average includes pixels with zero violation, so the loss is not amplified when only a few pixels violate the constraint. The transition width δ and tolerance τ_{f} are expressed in training units, whereas τ_{a} is used in reconstruction units.

The RGB and depth terms for a sampled view are combined as

$$\mathcal{L}_{\mathrm{gs}} = \mathcal{L}_{\mathrm{rgb}} + \lambda_{\mathrm{surf}}\mathcal{L}_{\mathrm{surf}} + \lambda_{\mathrm{free}}\mathcal{L}_{\mathrm{free}},$$
$$\mathcal{L}_{\mathrm{rgb}} = 0.8\mathcal{L}_1 + 0.2\mathcal{L}_{\mathrm{ssim}}, \qquad \mathcal{L}_1 = \frac{\sum_{p\in\mathcal{K}_v} \|\,\mathrm{clip}_{[0,1]}(C_v(p)) - I_v(p)\|_1}{3\max(1, |\mathcal{K}_v|)}. \tag{23}$$

Here I_v is the observed RGB image, and λ_{surf} and λ_{free} weight the surface and free-space losses. The SSIM loss averages $1 - \mathrm{SSIM}$, with local statistics computed only from unmasked pixels and

Gaussian window weights renormalized over those pixels. This prevents excluded colors or artificial zero values from affecting supervision near mask boundaries. Windows with an effective sample size below two are excluded, and the remaining losses are averaged using their unmasked window weights.

A.4 Cross-view Appearance Transfer

Although views share Gaussian parameters, their rendered colors depend on different compositing weights and viewing directions. Reducing an error in a source view therefore need not remove the corresponding error at a masked target pixel. Applying the estimated color through the target rendering constrains the Gaussians according to their contributions in that view.

Depth agreement supports a geometric correspondence, while exposure differences, view-dependent appearance, and compositing across multiple Gaussian layers can still bias the estimated target color. If these color targets also update geometry, optimization can reduce their errors by moving Gaussian centers or changing scales and opacities, altering surface placement and visibility. Keeping geometry, opacity, and the set of Gaussians fixed confines these updates to appearance and preserves the correspondences used to construct the targets.

Correspondences and source selection. Using the fixed rendered depth maps D_v^{render}, a target pixel p is unprojected to $\mathbf{X}_t(p) = \mathbf{R}_t^\top \left(D_t^{\mathrm{render}}(p)\mathbf{K}_t^{-1}\widetilde{\mathbf{p}} - \mathbf{t}_t\right)$. Its pixel location and camera depth in source view $s \neq t$ are

$$p_s = \Pi_s(\mathbf{R}_s\mathbf{X}_t(p) + \mathbf{t}_s)\,, \qquad z_s = [\mathbf{R}_s\mathbf{X}_t(p) + \mathbf{t}_s]_3\,. \tag{24}$$

A source observation is used only if the target and source depths are finite and positive, the projection lies inside the image with a one-pixel border margin, all four pixels used for bilinear RGB interpolation are valid and unmasked, and

$$|z_s - D_s^{\mathrm{render}}(p_s)| \le \tau_s, \qquad \tau_s = 0.01 + 0.05\max\{z_s, D_s^{\mathrm{render}}(p_s)\}. \tag{25}$$

Here $z_s > 0$, $D_s^{\mathrm{render}}(p_s)$ is sampled at the nearest pixel, and the absolute tolerance is in training units. Depth agreement determines the confidence $\xi_s = [1 - |z_s - D_s^{\mathrm{render}}(p_s)|/\tau_s]_+$. The source sets for color estimation and residual transfer are selected separately. Observations for $\mathcal{D}_t(p)$ are ranked by $\xi_s(0.875 + 0.125\boldsymbol{\omega}_t^\top\boldsymbol{\omega}_s)$, favoring depth agreement and similar viewing directions. Observations for $\mathcal{J}_t(p)$ require $\xi_s > 0$ and are ranked by ξ_s. The same validity checks apply to masked pixels and the unmasked queries used to fit the attention parameters. Checking all four interpolation pixels prevents a source color from mixing unmasked observations with masked source pixels near a mask boundary.

Estimating color for the target view. The affine color transformation in Section 3.4 uses a gain $\boldsymbol{\lambda}_v$ and offset $\boldsymbol{\beta}_v$ for each RGB channel:

$$\mathbf{J}_v(p) = \mathrm{clip}_{[0,1]}\left(\boldsymbol{\lambda}_v \odot I_v(p) + \boldsymbol{\beta}_v\right). \tag{26}$$

A fixed subset of unmasked queries is reserved for diagnostics and excluded as target samples when fitting this transformation and the attention parameters. The transformation is fitted against the rendering after geometry refinement and accepted only with sufficient samples and low fitting error; otherwise it is set to the identity. At a subpixel source location, observed RGB is bilinearly sampled before transformation and clipping. Rendered colors and their running averages remain in the rendering's color space. The same transformation is used for direct RGB supervision, source colors, and attention calibration, so all three use the same rendering color space. Otherwise, differences in exposure or white balance could produce conflicting color updates even at geometrically consistent correspondences.

The directional fit accounts for color differences between source and target viewing directions using a low-order function suited to the limited source observations. For the source set $\mathcal{D}_t(p)$, let the rows of $\mathbf{Y}$ and $\mathbf{J}$ be $\mathbf{y}(\boldsymbol{\omega}_s)^\top$ and $\mathbf{J}_s^\top$, and let $\mathbf{W} = \mathrm{diag}(\xi_s)$. Using the basis $\mathbf{y}$ and coefficients $\mathbf{U}^\star$ from Equation 7, the regularized linear fit satisfies

$$\left(\mathbf{Y}^\top\mathbf{W}\mathbf{Y} + \lambda_{\mathrm{r}}\mathbf{I}_4\right)\mathbf{U}^\star = \mathbf{Y}^\top\mathbf{W}\mathbf{J}. \tag{27}$$

The confidences are not normalized before fitting. At least four source observations are required, and the weighted design matrix must have full column rank and be well conditioned. These checks

use $\mathbf{W}_{\text{check}}^{1/2}\mathbf{Y}$, with $\mathbf{W}_{\text{check}} = \operatorname{diag}(\max(\xi_s, \epsilon_w))$, where $\epsilon_w > 0$ is a numerical stabilizer. The checks precede regularization because several observations from nearly identical directions may not constrain a directional fit; making the normal matrix invertible does not supply the missing angular information. If these conditions are not met or the solve fails, the target is the confidence-weighted average

$$\mathbf{P}_t(p) = \operatorname{clip}_{[0,1]}\left(\frac{\sum_{s\in\mathcal{D}_t(p)}\xi_s\mathbf{J}_s}{\max(\sum_{s\in\mathcal{D}_t(p)}\xi_s, \epsilon_w)}\right).$$

The target receives less weight when there are few source observations or their colors are poorly explained by the estimate. Let $\mathbf{f}_t(\boldsymbol{\omega}) = \mathbf{U}^{\star\top}\mathbf{y}(\boldsymbol{\omega})$ for the linear fit and $\mathbf{f}_t(\boldsymbol{\omega}) = \mathbf{P}_t(p)$ for the weighted average. The source color error and target reliability are

$$\begin{aligned}
\sigma_t^2(p) &= \frac{\sum_{s\in\mathcal{D}_t(p)}\xi_s\|\mathbf{J}_s - \mathbf{f}_t(\boldsymbol{\omega}_s)\|_2^2}{3\max(\sum_{s\in\mathcal{D}_t(p)}\xi_s, \epsilon_w)},\\
\rho_t(p) &= \eta_t(p)\min\left(\frac{|\mathcal{D}_t(p)|}{4}, 1\right)e^{-4\sigma_t(p)},\\
\eta_t(p) &= \begin{cases}1, & \text{linear fit,}\\ 0.25, & \text{weighted average.}\end{cases}
\end{aligned} \tag{28}$$

Both $\mathbf{P}_t$ and ρ_t are precomputed and remain fixed. The source loss in the main text is

$$\mathcal{L}_{\text{src}} = \frac{1}{3\max(1, |\mathcal{M}_t|)}\sum_{p\in\mathcal{M}_t}\rho_t(p)\left\|\operatorname{clip}_{[0,1]}(C_t(p)) - \mathbf{P}_t(p)\right\|_1. \tag{29}$$

The average includes all masked pixels $\mathcal{M}_t$. Pixels without valid source observations have $\rho_t = 0$ but remain in the denominator, so limited source coverage does not amplify the loss. The fixed denominator also allows lower reliability weights to reduce the overall strength of supervision.

Residual transfer. The fixed targets $\mathbf{P}_t$ provide supervision wherever sources are available across $\mathcal{M}_t$. Residual transfer uses a smaller, fixed query set to limit the cost of maintaining running averages and evaluating attention, while adapting its targets to changes in rendered appearance. At the fixed query pixels, the running averages $\mathbf{A}_t, \mathbf{A}_s$ from Section 3.4 supply the inputs to the learned query and key projections:

$$\mathbf{q} = \mathbf{W}_Q[\mathbf{A}_t; \boldsymbol{\omega}_t], \quad \mathbf{k}_s = \mathbf{W}_K[\mathbf{A}_s; \boldsymbol{\omega}_s; \xi_s; c_{ts}], \quad \mathbf{v}_s = \mathbf{J}_s - \mathbf{A}_s. \tag{30}$$

Here brackets denote concatenation and $c_{ts} = \boldsymbol{\omega}_t^\top\boldsymbol{\omega}_s$. The combined residual is $\mathbf{r}_t(p) = \sum_{s\in\mathcal{J}_t(p)}\pi_s\mathbf{v}_s$, which gives $\mathbf{B}_t$ through Equation 8. For an empty source set, $\mathbf{r}_t = 0$ and $\mathbf{B}_t = \mathbf{A}_t$. Subtracting $\mathbf{A}_s$ expresses the remaining discrepancy between each source observation and its rendering average. Adding the combined residual to $\mathbf{A}_t$ makes the transferred target a correction to the target appearance. As the rendering averages are updated, this correction evolves with the shared appearance, complementing the fixed estimate $\mathbf{P}_t$.

Initializing $\mathbf{W}_Q$ to zero makes the initial attention weights proportional to ξ_s, avoiding arbitrary source preferences before calibration. A nonzero initialization of $\mathbf{W}_K$ allows the query projection to receive gradients. The bounded gain $\mathbf{g}$ can attenuate an unhelpful residual without reversing or amplifying it. Thus, the learned parameters control the combination of observed residuals rather than freely predicting RGB values.

Attention weights express relative preference among sources: a single source receives unit attention even when its depth confidence is low. Depth agreement therefore determines a separate reliability for fitting the attention parameters:

$$\zeta_t(p) = \frac{\sum_{s\in\mathcal{J}_t(p)}\xi_s^2}{\max(\sum_{s\in\mathcal{J}_t(p)}\xi_s, \epsilon_{\text{tiny}})}. \tag{31}$$

Here $\epsilon_{\text{tiny}} > 0$ stabilizes the denominator.

Let $\mathcal{C}_t$ contain the fixed unmasked queries outside the diagnostic subset. The projections $\mathbf{W}_Q, \mathbf{W}_K$ and channelwise gain $\mathbf{g}$ are fitted by minimizing

$$\mathcal{L}_{\text{cal}} = \frac{1}{\max(1, |\mathcal{C}_t|)}\sum_{p\in\mathcal{C}_t}\operatorname{sg}(\zeta_t(p))\frac{\|\mathbf{B}_t(p) - \mathbf{J}_t(p)\|_2^2}{3}. \tag{32}$$

Here sg holds its argument constant during differentiation. Queries without valid sources have $\zeta_t = 0$ but remain in the denominator. The observed colors, running averages, and ζ_t are held fixed, so this loss updates only $\mathbf{W}_Q, \mathbf{W}_K$, and $\mathbf{g}$. Observed target colors at unmasked queries provide direct supervision for learning which source residuals to combine and how strongly to apply them.

For masked pixels, reliability also depends on whether the source residuals agree and whether residual transfer reduces color error at unmasked queries. The first factor compares the combined residual with the variation across sources:

$$d_t(p) = \frac{1}{3} \sum_{s \in \mathcal{J}_t(p)} \pi_s \|\mathbf{v}_s - \mathbf{r}_t\|_2^2, \qquad \chi_t(p) = \frac{\|\mathbf{r}_t\|_2^2/3}{\|\mathbf{r}_t\|_2^2/3 + d_t(p) + (1/255)^2}. \tag{33}$$

These quantities use the residual before applying the gain $\mathbf{g}$. An empty source set gives $\zeta_t = \chi_t = 0$. When source residuals suggest opposing color corrections, their dispersion increases relative to the combined residual, reducing χ_t. The noise term also down-weights corrections whose magnitude is small relative to color quantization. The second factor averages positive relative reductions in color error on $\mathcal{C}_t$. With $e_t^A(p) = \|\mathbf{A}_t(p) - \mathbf{J}_t(p)\|_2^2/3$ and $e_t^B(p) = \|\mathbf{B}_t(p) - \mathbf{J}_t(p)\|_2^2/3$, it is

$$\upsilon_t = \frac{1}{\max(1, |\mathcal{C}_t|)} \sum_{p \in \mathcal{C}_t} \min\left(1, \frac{[e_t^A(p) - e_t^B(p)]_+}{\max(e_t^A(p), \epsilon_{\text{mach}})}\right). \tag{34}$$

The positive stabilizer ϵ_{mach} avoids division by zero, and υ_t is shared by all masked queries in the view. The reliability $\kappa_t(p) = \text{sg}(\zeta_t(p)\chi_t(p)\upsilon_t)$ weights the auxiliary loss

$$\mathcal{L}_{\text{att}} = \frac{0.25}{\max(1, |\mathcal{B}_t|)} \sum_{p \in \mathcal{B}_t} \kappa_t(p) \left(\sqrt{\frac{\|\text{clip}_{[0,1]}(C_t(p)) - \text{sg}(\mathbf{B}_t(p))\|_2^2}{3} + (1/255)^2} - 1/255 \right), \tag{35}$$

where $\mathcal{B}_t$ contains the masked queries in the fixed query set, including those without valid sources. Thus, $\mathcal{L}_{\text{src}}$ averages over all masked pixels $\mathcal{M}_t$, whereas $\mathcal{L}_{\text{att}}$ averages over $\mathcal{B}_t$. During Gaussian updates, $\mathbf{B}_t$ and its reliability are held fixed; their construction uses observed target colors only at unmasked queries. The factor υ_t controls the use of transferred colors but does not weight $\mathcal{L}_{\text{cal}}$: doing so would suppress learning before residual transfer has improved on the running average. Keeping the targets and weights fixed during Gaussian updates also prevents the auxiliary loss from being reduced by changing its own supervision.

Constrained appearance updates. The directional fit estimates a pixel color from a limited set of source directions. Each pixel can combine several Gaussians with different contributions across views, so these observations do not uniquely determine the higher-order SH coefficients of each Gaussian. Updating those coefficients from estimated colors could fit the available directions while introducing unsupported color variation in other directions. Auxiliary gradients therefore update only the zero-order coefficients $\mathbf{a}$, while unmasked RGB supervision updates all SH coefficients. This restricts the directional variation learned from estimated colors while allowing spatially distinct Gaussians to recover local image detail through their zero-order coefficients. The direct RGB loss uses $\mathbf{J}_t$ in place of I_t in Equation 23, with the same computation over unmasked pixels.

The two auxiliary losses act on the same Gaussian parameters and can draw on the same source observations. Their gradients are therefore combined before projection to constrain their total effect. Fixed loss weights do not bound the combined auxiliary gradient relative to the gradient from unmasked RGB supervision, because the compositing weights determine how pixel errors contribute to parameter gradients. Removing only the opposing component would leave arbitrarily large orthogonal updates, which motivates the additional norm bound. The projection first removes any component of the auxiliary gradient opposing $\boldsymbol{\gamma}$, then limits its norm. For $\|\boldsymbol{\gamma}\|_2 > 0$, its closed form is

$$\begin{aligned} \boldsymbol{\delta}' &= \boldsymbol{\delta} - \frac{\min(\langle \boldsymbol{\gamma}, \boldsymbol{\delta} \rangle, 0)}{\|\boldsymbol{\gamma}\|_2^2} \boldsymbol{\gamma}, \\ \boldsymbol{\nu} &= \min\left(1, \frac{\|\boldsymbol{\gamma}\|_2}{\|\boldsymbol{\delta}'\|_2}\right) \boldsymbol{\delta}'. \end{aligned} \tag{36}$$

If either norm is zero, $\boldsymbol{\nu} = 0$. Inner products and norms are taken jointly over all zero-order SH coefficients $\mathbf{a}$. The optimizer receives $\boldsymbol{\gamma} + \boldsymbol{\nu}$ for these coefficients and the direct RGB gradient for

higher-order coefficients. For an infinitesimal gradient step, $\langle \boldsymbol{\gamma}, \boldsymbol{\nu} \rangle \geq 0$ ensures that the auxiliary term does not increase $\mathcal{L}_{\mathrm{rgb}}$ to first order. This local property does not guarantee decreasing image error under finite Adam updates.

Updating the running averages. The running averages are initialized from the renderings after geometry refinement and held fixed during each parameter update. The colors rendered before that update are then incorporated as

$$\mathbf{A}_v(p) \leftarrow \beta_{\mathrm{ema}}^{\Delta k} \mathbf{A}_v(p) + (1 - \beta_{\mathrm{ema}}^{\Delta k})\, \mathrm{sg}\Big(\mathrm{clip}_{[0,1]}(C_v(p))\Big), \tag{37}$$

where β_{ema} is the decay factor and Δk is the number of training steps since that pixel's average was last updated. Views are sampled at different intervals, so a fixed decay per visit would retain old colors for longer in rarely visited views. The factor $\beta_{\mathrm{ema}}^{\Delta k}$ instead discounts the previous average according to elapsed training steps.

B Detailed Experimental Results

Tables 4 and 5 report image-quality scores for the six NeRF On-the-go scenes and nine self-captured scenes, respectively. Figures 9, 10, and 11 supplement the RGB and rendered-depth comparisons in the main text.

B.1 NeRF On-the-go

The scores in Table 4 give the per-scene results summarized in Table 1. Baseline scores are taken from the cited reports (Kulhanek et al., 2024; Wang et al., 2025b; Lin et al., 2025; Fu et al., 2025; Seo et al., 2026; Wang et al., 2026); starred scores for 3DGS (Kerbl et al., 2023) and SpotLessSplats (Sabour et al., 2025) are aggregated from later reports. LPIPS is ranked separately by backbone: RobustSplat and PDF-GS use VGG, while GAD-Splat and the other baselines use AlexNet. DualSplat's LPIPS values are omitted because its evaluation protocol differs. GAD-Splat achieves the highest SSIM in all six scenes, while the PSNR ranking varies by scene. On Mountain, its PSNR exceeds DualSplat's by 0.62 dB. On Spot, RobustSplat has a higher PSNR (26.21 versus 25.94 dB), while GAD-Splat has higher SSIM (0.915 versus 0.907).

The RGB comparisons in Figure 9 complement the Corner and Patio-High examples in Figure 5 with Fountain, Patio, Mountain, and Spot. In Fountain, the compared RGB renderings remove the foreground pedestrian and retain the curved pipe and tree, with relatively small visible differences. In Mountain, GAD-Splat preserves finer detail in the vegetation around the wooden chair, whereas DualSplat and PDF-GS produce visibly smeared leaves and grass in the enlarged crop. In Spot, GAD-Splat retains the carpet texture in the marked region, while PDF-GS produces a blurred patch. In the rendered depth, the central structure in Patio has a clearer outline in our result than in DualSplat and PDF-GS. In Spot, our floor depth varies smoothly, while DualSplat shows diagonal bands and PDF-GS shows patchy variation beneath the robot.

B.2 Self-Captured Data

The results in Table 5 vary across scenes and metrics. GAD-Splat leads on all three metrics in Scenes 3, 5, and 8, whereas DeSplat leads on all three in Scene 7. In Scene 9, GAD-Splat improves PSNR and LPIPS over DeSplat (20.95 versus 19.73 dB and 0.269 versus 0.300), while DeSplat has higher SSIM (0.681 versus 0.553).

The RGB comparisons in Figure 10 extend Figure 6 to Scenes 3–9. In Scene 4, PDF-GS and DeSplat retain faint remnants of the person against the wall, whereas GAD-Splat, DualSplat, and RobustSplat remove the person and preserve the piano. In Scene 5, DeSplat leaves white fragments on the highlighted pavement, while our rendering retains the paving lines. The stone in Scene 3 and the sculpture in Scene 8 also remain visible after the foreground people are removed. The rendered-depth comparisons across all nine scenes in Figure 11 reveal differences that are less apparent in RGB. In Scene 4, our rendering shows smoother depth variation across the wall and floor, with distinct outlines of the piano and stool; DualSplat and RobustSplat show abrupt depth variations around these regions despite removing the person in RGB. In Scene 9, the ground around the scooter has less speckled depth variation in our rendering than in the compared methods.

Table 4: Per-scene image-quality scores on NeRF On-the-go, grouped by occlusion level. The top three ranks in each column are shaded first, second, and third (PSNR/SSIM ↑, LPIPS ↓); ties share ranks, with subsequent ranks skipped. LPIPS is ranked separately within the Alex and VGG groups; cross-report LPIPS values are not necessarily directly comparable.

Method	LPIPS	Low occlusion						Medium occlusion						High occlusion					
		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
3DGS*	Alex	19.78	0.676	0.200	20.33	0.685	0.183	22.27	0.788	0.181	17.31	0.706	0.203	19.98	0.721	0.304	17.58	0.633	0.319
WildGaussians	Alex	20.43	0.653	0.255	20.81	0.685	0.215	24.16	0.822	0.045	21.44	0.800	0.138	23.82	0.816	0.138	22.23	0.725	0.206
SpotLessSplats*	Alex	21.03	0.671	0.246	21.19	0.680	0.215	25.46	0.854	0.152	21.74	0.822	0.107	23.94	0.812	0.169	21.65	0.750	0.209
DeSplat	Alex	19.59	0.710	0.170	20.27	0.680	0.170	26.05	0.880	0.090	20.89	0.810	0.110	26.07	0.900	0.090	22.59	0.840	0.120
HybridGS	Alex	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
RobustSplat	VGG	21.15	0.737	0.201	21.01	0.701	0.199	26.42	0.897	0.104	21.63	0.827	0.139	26.21	0.907	0.102	22.87	0.837	0.146
PDF-GS	VGG	21.82	0.750	0.180	21.19	0.700	0.200	26.41	0.900	0.100	21.54	0.830	0.140	25.94	0.910	0.100	23.00	0.840	0.150
DualSplat	Alex	21.86	0.747	–	21.28	0.706	–	26.53	0.897	–	21.64	0.828	–	25.96	0.908	–	23.26	0.836	–
GAD-Splat	Alex	22.48	0.779	0.157	21.60	0.716	0.157	26.68	0.912	0.061	21.83	0.847	0.087	25.94	0.915	0.064	23.57	0.844	0.120

Table 5: Per-scene image-quality scores on our self-captured data, expanding the means in Table 2. Test views are held out from training, and all methods are evaluated on the same views. PSNR is shown to two decimal places and SSIM and LPIPS to three; means are computed before rounding. The best, second, and third results are highlighted for each scene and metric; ties share ranks.

Scene	WildGaussians			DeSplat			RobustSplat			PDF-GS			DualSplat			GAD-Splat		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Scene 1	23.62	0.736	0.304	26.55	0.904	0.043	27.33	0.917	0.035	27.11	0.912	0.040	27.48	0.916	0.036	28.14	0.926	0.036
Scene 2	19.15	0.541	0.635	22.92	0.822	0.082	22.92	0.823	0.079	22.69	0.818	0.083	23.04	0.826	0.078	23.66	0.835	0.081
Scene 3	22.72	0.508	0.726	28.55	0.882	0.052	28.75	0.876	0.056	28.03	0.866	0.064	28.46	0.874	0.057	29.17	0.898	0.043
Scene 4	29.10	0.963	0.092	31.05	0.955	0.059	31.39	0.957	0.040	27.91	0.947	0.077	31.63	0.959	0.040	31.07	0.965	0.048
Scene 5	23.71	0.676	0.550	27.96	0.885	0.066	28.19	0.890	0.069	27.58	0.881	0.081	28.31	0.892	0.068	29.07	0.908	0.051
Scene 6	21.29	0.529	0.680	26.71	0.884	0.051	25.81	0.835	0.074	24.92	0.823	0.083	25.96	0.843	0.073	26.79	0.868	0.063
Scene 7	22.08	0.506	0.769	27.82	0.879	0.057	26.37	0.799	0.110	25.60	0.788	0.113	26.30	0.799	0.109	27.16	0.831	0.081
Scene 8	23.36	0.593	0.637	28.33	0.907	0.041	28.88	0.904	0.038	28.46	0.897	0.041	28.64	0.903	0.038	29.39	0.919	0.033
Scene 9	18.90	0.651	0.356	19.73	0.681	0.300	18.04	0.485	0.366	18.33	0.487	0.361	18.04	0.485	0.369	20.95	0.553	0.269
Mean	22.66	0.634	0.528	26.62	0.867	0.083	26.41	0.832	0.097	25.63	0.824	0.105	26.43	0.833	0.096	27.27	0.856	0.078

B.3 Conclusion and Limitations

The results support using geometry and appearance constraints after transient observations are excluded. With the same predicted masks, CDR and CAT improve image quality individually, and their combination gives the best scores in Table 3. The per-scene comparisons show that the advantages of the full method still vary across scenes and metrics.

The framework nevertheless depends on the observations retained by its masks. Incorrect exclusion can remove useful static information, while missed transient regions can retain contaminated RGB supervision. For a background surface hidden in all source views, cross-view appearance transfer cannot provide color targets derived from unmasked observations.

Appearance refinement also inherits the geometry established by the preceding stage. Errors in this geometry can produce incorrect correspondences that remain fixed, and color updates cannot correct misplaced surfaces or create missing ones. In addition, the directional fit uses a limited set of source views, and auxiliary gradients update only zero-order SH coefficients. Strongly view-dependent appearance may therefore be difficult to recover from these auxiliary targets.

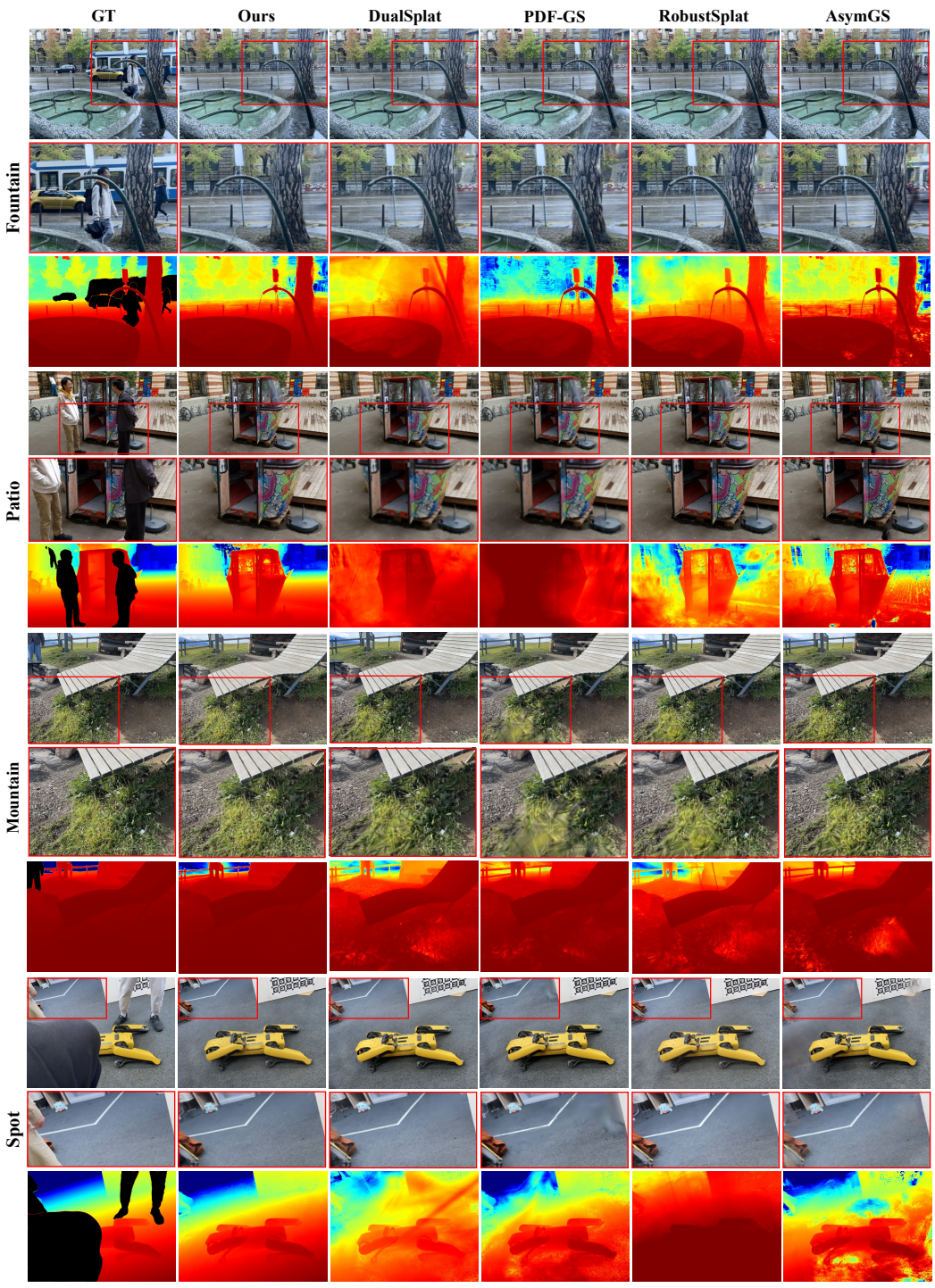

Figure 9: **Additional qualitative comparisons on NeRF On-the-go.** Fountain, Patio, Mountain, and Spot (top to bottom), showing RGB images, enlarged crops of the regions marked by red boxes, and rendered depth maps. The crops highlight differences in background texture after transient-object removal. GT denotes captured RGB images containing transient objects.

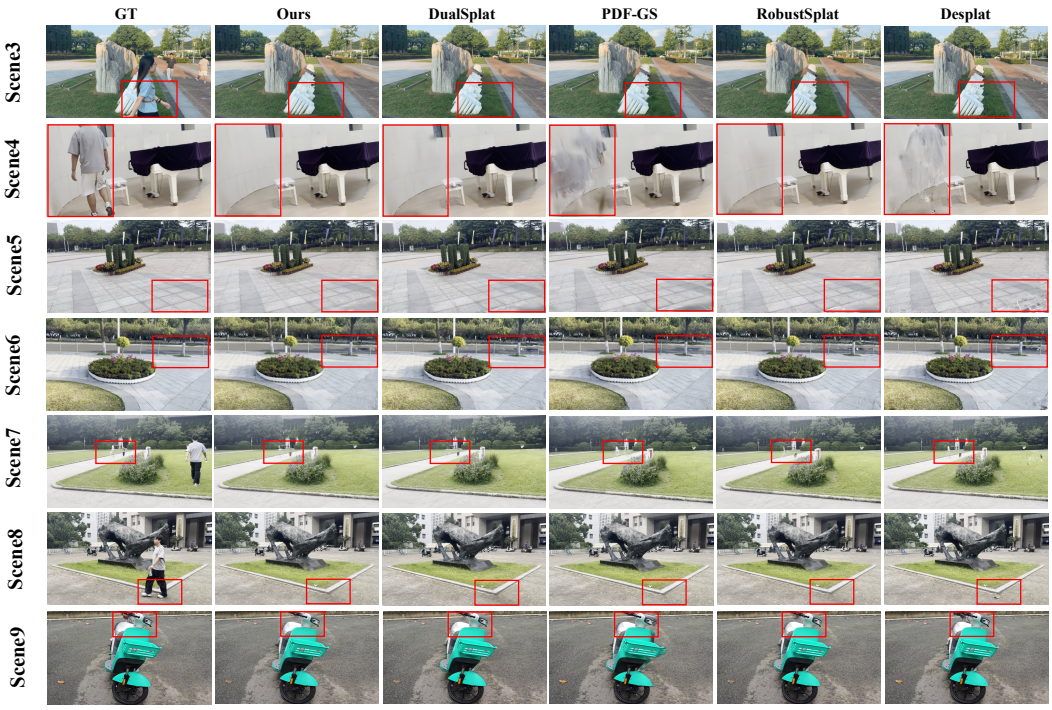

Figure 10: **Additional RGB comparisons on our self-captured data.** Scenes 3–9 are shown from top to bottom. Red boxes highlight regions for detailed comparison. GT denotes captured images that may contain transient objects.

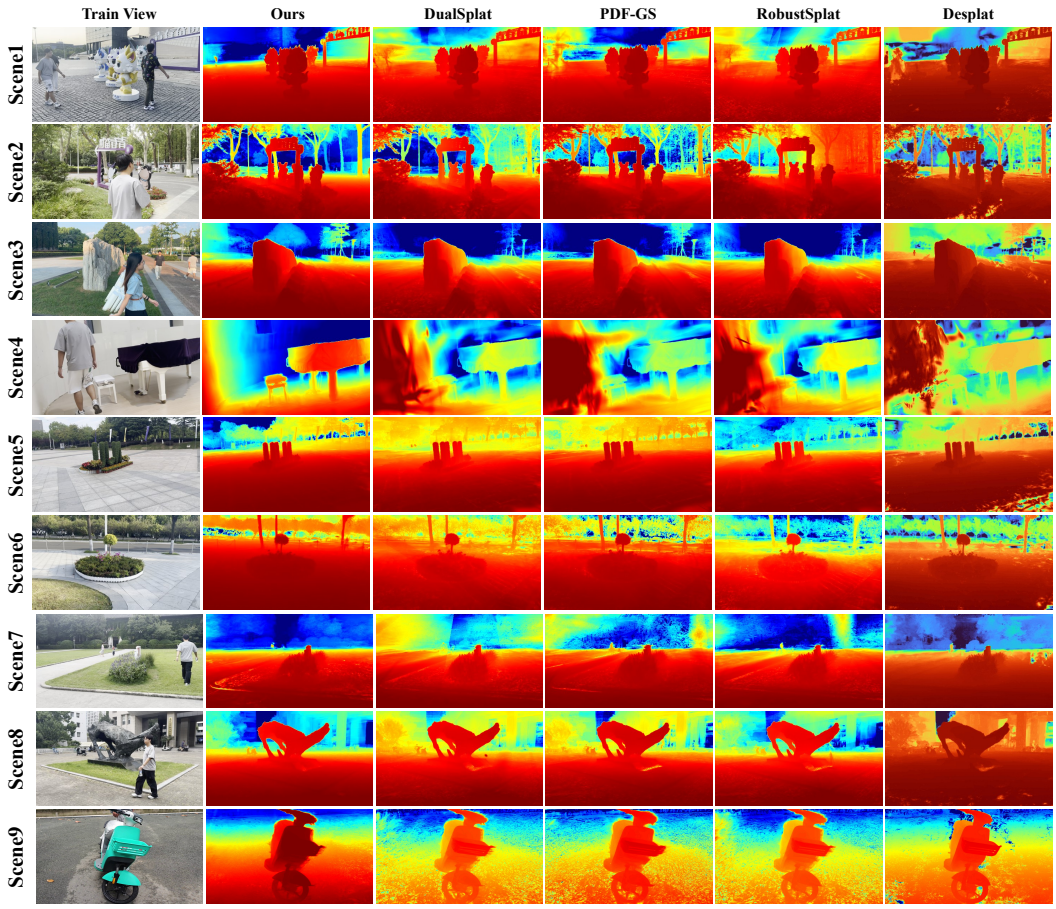

Figure 11: **Rendered depth comparisons on our self-captured data.** Scenes 1–9 are shown from top to bottom. The first column shows the corresponding training images.