GRE-GS: GEOMETRY REPAIR AND GAUSSIAN EXTRAPOLATION FOR SPARSE-VIEW 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) achieves real-time, high-quality novel-view synthesis through explicit Gaussian primitives. However, under sparse-view settings, limited observations provide insufficient geometric support, yielding unreliable geometry and incomplete scene coverage. To address this issue, we present GRE-GS (Geometry Repair and Gaussian Extrapolation for Sparse-View Gaussian Splatting), a reconstruction framework that explicitly separates the two failure modes of sparse-view 3DGS—unreliable geometry and incomplete coverage—and addresses them with geometry repair and Gaussian extrapolation, respectively. For regions covered by training views, GRE-GS repairs the initial point cloud generated by a pretrained geometric model. It completes missing surfaces via locally aligned depth propagation and corrects depth errors via structural conflict correction. For uncovered regions, GRE-GS extrapolates new Gaussians from the trained Gaussian representation—extending their positions using projected DA3 depths and synthesizing colors with an MLP conditioned on neighboring Gaussian appearance. We also introduce a training-view safety mechanism that constrains and filters newly added Gaussians to keep them from disrupting the existing representation. Extensive experiments on Mip-NeRF360, Tanks and Temples, and DTU demonstrate that our method substantially improves novel-view rendering quality across different sparse-view settings.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) combines high-quality novel-view synthesis with real-time rendering (Kerbl et al., 2023), but sparse observations leave weak geometric constraints and make reconstruction sensitive to initialization and camera coverage. Existing methods use regularization, depth priors, multi-view constraints, densification, or stronger initialization (Niemeyer et al., 2022; Yang et al., 2023; Wang et al., 2023; Li et al., 2024; Zhu et al., 2024; Patle et al., 2025; Kim et al., 2026), yet the remaining errors do not all have the same observational support.

These reconstruction errors reflect two distinct failure modes: unreliable geometry in regions covered by training views and incomplete scene coverage outside the training field of view (FoV). In observed regions, visible surfaces may be missing or represented at inconsistent depths despite available image evidence. In unobserved regions, no direct photometric evidence constrains the missing geometry. Because these cases differ in observational support, GRE-GS repairs observed-region geometry using local image and depth evidence, while accepting extrapolated geometry only after checking consistency with every real training view.

GRE-GS first repairs observed-region geometry. MASt3R provides metric anchors, while Depth Anything v3 (DA3) provides local depth ordering. Local-Anchor Geometry Completion aligns DA3 depth to neighboring geometry before filling uncovered surfaces; Structural Conflict Correction relocates points inconsistent with local structure. After Gaussian training, virtual cameras expose low-coverage boundaries and candidates inherit nearby frozen geometry. A scene-specific neighbor-weight multilayer perceptron (MLP), trained before extrapolation using pseudo-holes in the Gaussian cloud, transfers appearance. The training-FoV safety guard then suppresses candidates that interfere with any training view.

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.

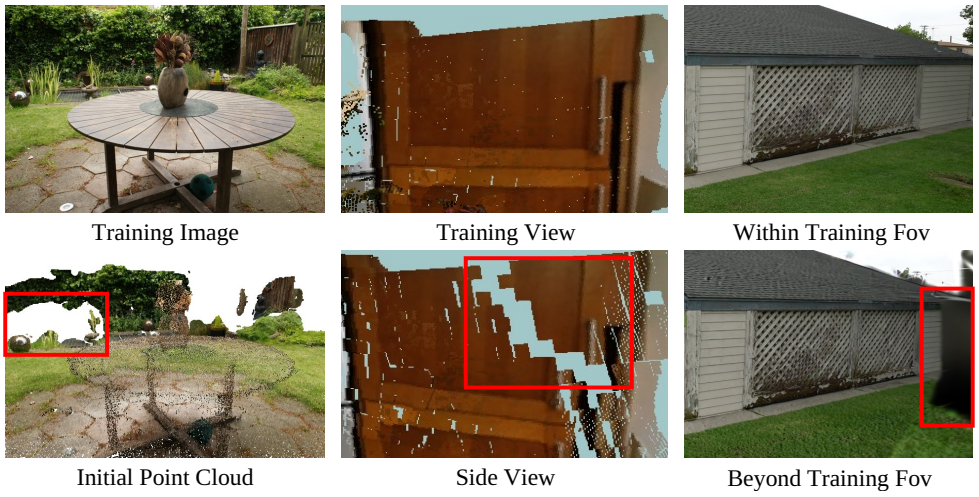

Figure 1: Sparse-view failure modes. Observed surfaces can be missing or depth-inconsistent inside the training FoV, while camera motion exposes unsupported regions beyond it.

We evaluate GRE-GS on Mip-NeRF360, Tanks and Temples, and DTU under 12-, 24-, and 3-view settings. Geometry repair improves the common initialization across all settings, and completion and correction provide complementary gains. Extrapolation further improves held-out views, while the safety guard reduces the worst measured training-view degradation from 9.71–40.19 dB without the guard to at most 0.01 dB.

Our contributions are:

- **Observation-aware formulation:** observed defects are repaired from local support, while unsupported coverage is handled conservatively.
- **Observed-Region Geometry Repair:** aligned completion adds supported surfaces and structural correction relocates inconsistent geometry rather than merely increasing density.
- **Training-FoV-Safe Gaussian Extrapolation:** training-derived candidates extend a frozen model through pseudo-hole-trained appearance fusion and the training-FoV safety guard.

2 Related Work

2.1 Sparse-View Neural Rendering and Gaussian Splatting

Neural radiance fields established the implicit scene-representation line for view synthesis (Mildenhall et al., 2021). Sparse-view neural rendering then uses optimization regularization or geometric priors: RegNeRF regularizes unobserved viewpoints, FreeNeRF controls frequency learning, and SparseNeRF imposes relative depth constraints (Niemeyer et al., 2022; Yang et al., 2023; Wang et al., 2023). Sparse-view 3DGS methods use normalized depth supervision (DNGaussian), unpooling (FSGS), co-regularization (CoR-GS), stochastic removal (DropGaussian), or controlled densification (AD-GS) (Li et al., 2024; Zhu et al., 2024; Zhang et al., 2024; Park et al., 2025; Patle et al., 2025). Other approaches emphasize structured initialization or coverage under sparse views, including CoherentGS, InstantSplat, SparseGS, and MVPGS (Paliwal et al., 2024; Fan et al., 2024; Xiong et al., 2025; Xu et al., 2024). Hole completion and repair have also been explored through scene reuse and diffusion priors, as in ShareGS and Ri3D (Hong et al., 2025; Paliwal et al., 2025). These lines of work address complementary failure modes, but do not use the same observation-conditioned split between supported repair and unsupported extrapolation as GRE-GS. GRE-GS repairs observed regions by adding locally aligned samples and relocating structurally inconsistent points, while extending coverage beyond the training FoV with newly extrapolated Gaussians; this is distinct from revising the densification rule itself (Rota Bulò et al., 2024).

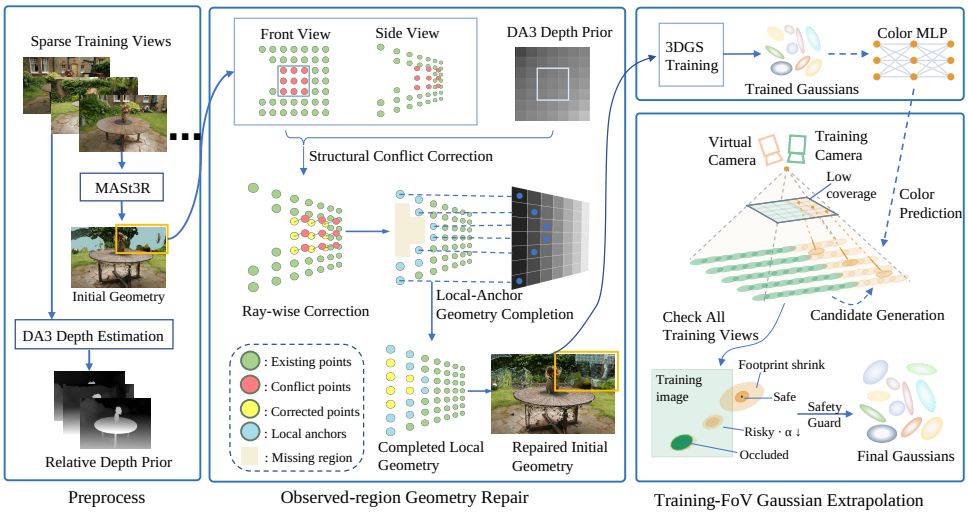

Figure 2: GRE-GS overview. Structural Conflict Correction and Local-Anchor Geometry Completion repair observed-region geometry before 3DGS optimization. Training-derived virtual cameras then expose low-coverage boundaries; candidates inherit frozen geometry and appearance and pass the training-FoV safety guard.

2.2 Geometric Priors for Sparse Reconstruction

Monocular depth aids sparse reconstruction despite uncertain scale: SparseNeRF uses depth ordering, while DNGaussian and FSGS use normalized or regularized guidance (Wang et al., 2023; Li et al., 2024; Zhu et al., 2024). Recent Gaussian methods also propagate high-confidence depth and normal priors or impose depth-regularized optimization (Xiang et al., 2026; Han et al., 2026; Chung et al., 2024). DUSt3R predicts dense pointmaps, MASt3R strengthens matching, and DA3 provides monocular structure (Wang et al., 2024; Leroy et al., 2024; Lin et al., 2026). Point-cloud initialization is another source of sparse-view failure; Dust-GS uses dense point-cloud construction, while TWINGS combines fused priors for stronger initialization (Chen et al., 2025; Kim et al., 2026). GRE-GS treats initialization as a scaffold: MASt3R supplies local metric anchors, and DA3 is aligned independently in each supported region before completion or ray-wise relocation.

2.3 Scene Completion and View Extrapolation

Virtual cameras can regularize radiance fields, while FSGS grows Gaussians in underconstrained regions (Niemeyer et al., 2022; Zhu et al., 2024). VEGS and generative approaches add learned priors beyond the training-camera distribution, including scene inpainting and diffusion-based completion (Hwang et al., 2024; Weber et al., 2024; Wu et al., 2024; Zhou & Tulsiani, 2023; Zhong et al., 2025; Wu et al., 2025). GRE-GS instead uses virtual cameras only as coverage probes, with no target RGB or base-model update. Candidate colors remain convex fusions of observed neighboring colors learned from same-scene pseudo-holes. Unlike visibility pruning, our guard checks every candidate's footprint and depth relation in every training view, attenuates or rejects unsafe candidates, and verifies the final tolerance. This design extends coverage conservatively.

3 Method

3.1 Problem Formulation and Overview

Given calibrated training views $\mathcal{V}_{\text{tr}} = \{(I_i, K_i, T_i)\}_{i=1}^{N_{\text{tr}}}$, MASt3R produces a colored point cloud with confidence scores and DA3 predicts a monocular depth map D_i^A for each image. We remove low-confidence points, voxel-subsample the cloud, and suppress small detached components to obtain a scaffold $\mathcal{P}_0$. MASt3R supplies scene-consistent metric anchors; DA3 contributes local depth ordering and surface variation and is never assumed to share a global reconstruction scale.

As shown in Fig. 2, GRE-GS first repairs defects supported by training observations. Structural Conflict Correction relocates inconsistent points, after which missing-region masks are recomputed and Local-Anchor Geometry Completion adds supported surface samples. The resulting cloud $\mathcal{P}_3$ initializes a base Gaussian model $\mathcal{G}_{\text{base}}$. Training-side virtual views then probe low-coverage boundaries, and candidate Gaussians are extrapolated from nearby frozen geometry. A scene-specific model transfers appearance learned from pseudo-holes in $\mathcal{G}_{\text{base}}$, and the training-FoV safety guard rejects or attenuates candidates according to their projected footprints and depth relations, producing $\mathcal{G}_{\text{final}}$.

3.2 Observed-Region Geometry Repair

The cleaned scaffold $\mathcal{P}_0$ may leave an observed surface uncovered or place existing points at structurally inconsistent depths. Structural Conflict Correction first relocates selected points at inconsistent depths without changing their number or color. Local-Anchor Geometry Completion then adds points only in supported missing regions.

3.2.1 Structural Conflict Correction

On pixels already covered by $\mathcal{P}_0$, we compare reconstructed and monocular log depth. Residual seeds must exceed 0.15 in log depth and grow only through spatially coherent residuals above 0.08. We protect a three-pixel band around DA3 discontinuities and reject components with fewer than 18 pixels, sign coherence below 0.82, or insufficient surrounding support. Within a retained region, robust inverse-depth alignment defines the target depth:

$$\frac{1}{z^M} = a_R \frac{1}{z^A} + b_R, \qquad z_{\text{tgt}}(u,v) = \left(\frac{a_R}{D_i^A(u,v)} + b_R \right)^{-1}.$$

The visible point at (u, v) is moved to z_{tgt} along its original camera ray. Multi-view proposals are ranked by their local support, and a group is rolled back when cross-view z-buffer checks repeatedly place its updated points in front of supported surfaces. Correction changes selected 3D positions but preserves point count and color. The complete component, ring, and rollback settings are listed in the supplement.

3.2.2 Local-Anchor Geometry Completion

For training view i, z-buffering a point cloud $\mathcal{P}$ yields visible depth $Z_i(\mathcal{P})$. Let Ω_i^A be the valid DA3 domain, $\mathcal{C}_i(\mathcal{P})$ the dilated projected coverage, and $\mathcal{B}_i$ a protected image-boundary band. Candidate missing pixels are

$$\mathcal{M}_i(\mathcal{P}) = \left(\Omega_i^A \setminus \mathcal{C}_i(\mathcal{P})\right) \setminus \mathcal{B}_i.$$

For each connected region R, progressively dilated rings collect pixels where both reconstructed and DA3 depths are valid. We robustly fit either $\hat{z}^M = a_R z^A + b_R$ or a scale-only map and reserve 20% of the anchors for validation. A region is accepted only with at least 80 fitting and 20 validation anchors, validation median relative error at most 0.15, and 90th-percentile error at most 0.50; otherwise it is skipped. Valid aligned depths are back-projected through the training camera at a three-pixel stride and colored from I_i, yielding

$$\mathcal{C}_{\text{comp}}(\mathcal{P}) = \mathcal{P} \cup \Delta\mathcal{P}_{\text{comp}}.$$

Completion therefore transfers local surface structure without moving existing geometry or globally back-projecting monocular depth.

3.2.3 Composition and Gaussian Optimization

The full repair applies correction first and then recomputes $\mathcal{M}_i$ before completion, because relocating points can expose or close projected gaps:

$$\mathcal{P}_3 = \mathcal{C}_{\text{comp}}(\mathcal{C}_{\text{corr}}(\mathcal{P}_0); \mathcal{M}(\mathcal{C}_{\text{corr}}(\mathcal{P}_0))).$$

We initialize Gaussian centers and colors from $\mathcal{P}_3$ and optimize the remaining attributes using the standard RGB objective plus a confidence-weighted, locally normalized log-depth loss. All repair ablations share the same Gaussian optimization settings and differ only in initialization geometry.

3.3 TRAINING-FOV-SAFE GAUSSIAN EXTRAPOLATION

After training $\mathcal{G}_{\text{base}}$ from the repaired cloud, we freeze it and perturb each training camera with scene-normalized outward translations and pitch/yaw rotations. These virtual cameras have no target images; they only render alpha, depth, and Gaussian footprints. We retain connected low-alpha regions adjacent to supported geometry and reject ordinary background or components without sufficient boundary anchors. Exact pose sets and component thresholds are given in the supplement.

3.3.1 CANDIDATE EXTRAPOLATION AND ATTRIBUTE PROPAGATION

DA3 samples from each real source view are projected into its virtual cameras. Within a retained region R, nearby rendered Gaussian depths establish a local scale s_R and reference depth z_R^G. Candidate depth is bounded by

$$z_{\text{new}}(p) = \text{clip}\left(s_R \widetilde{D}^A(p), \frac{z_R^G}{\rho}, \rho z_R^G\right).$$

Projected DA3 samples provide local depth values when available, while the inverse-depth plane continues the same boundary-supported trend when they are absent. Unsupported regions are rejected. Accepted depths are back-projected, consolidated across virtual cameras, and voxel-deduplicated.

Before extrapolation, we train a lightweight scene-specific neighbor-weight MLP on $\mathcal{G}_{\text{base}}$ itself. Compact Gaussian blocks in dense regions are removed as pseudo-holes; for each held-out Gaussian, the model scores its 16 remaining neighbors and reconstructs the held-out degree-zero spherical-harmonic (SH-DC) color as a softmax-weighted convex combination. Training and validation holes are spatially disjoint. At inference, the learned scorer supplies the candidate color,

$$c_g = \sum_{j \in \mathcal{N}_{16}(g)} w_{gj} c_j, \qquad w_{gj} = \text{softmax}_j\, f_\theta(\phi_{gj}),$$

while scale, rotation, and opacity are propagated from the same frozen neighborhood by distance weighting. Attribute caps are applied before safety filtering. The feature vector ϕ_{gj}, architecture, pseudo-hole construction, and optimization settings are detailed in the supplement.

3.3.2 FOOTPRINT-AWARE TRAINING-FOV SAFETY GUARD

We call this filtering module the training-FoV safety guard. A candidate center can lie outside a training image while its anisotropic footprint still overlaps observed pixels. For every real training camera, we evaluate the rendered footprint and classify the candidate as renderer-invisible, reliably occluded behind a supported base surface, or risky. For Tanks and Temples and DTU, renderer-invisible candidates retain their proposed opacity (factor 1.0); occluded candidates receive a class factor selected by the attenuation search; and risky candidates are first scaled by a footprint-derived transition weight and then by a separately searched factor. Candidates whose resulting opacity is below 0.01 are removed.

Let $\mathcal{E}$ denote the finite attenuation factors actually evaluated for a class. We retain the largest tested feasible value

$$\hat{\eta} = \max_{\eta \in \mathcal{E}} \left\{ \eta : \min_i \Delta_i(\eta) \geq -\varepsilon_{\text{safe}} \right\},$$

where Δ_i is the peak signal-to-noise ratio (PSNR) change on real training view i relative to $\mathcal{G}_{\text{base}}$. The implementation searches the attenuation factor with eight coarse-to-fine bisection-style probes. Safety is established by explicitly rendering the final accepted representation in every training view and requiring $\varepsilon_{\text{safe}} = 0.01$ dB; the stricter Mip-NeRF360 variant keeps only candidates culled by the renderer in every training view. Test images are loaded only after $\mathcal{G}_{\text{final}}$ is frozen.

4 EXPERIMENTS

Datasets and protocols. We evaluate seven Mip-NeRF360 scenes (Barron et al., 2022) with 12 or 24 training views, eight Tanks and Temples scenes (Knapitsch et al., 2017) with three views, and 15 DTU scans (Jensen et al., 2014) with three views. Our Mip-NeRF360 images use the standard $1/8$ resolution (approximately 618×411, with scene-dependent dimensions). The two Mip-NeRF360

Table 1: Quantitative comparison on Mip-NeRF360 at $1/8$ image resolution. Entries are PSNR ↑ / SSIM ↑ / LPIPS ↓. † indicates results reproduced by us using released code.

Method	12 views	24 views
3DGS	18.5200 / 0.5230 / 0.4150	22.8000 / 0.7080 / 0.2760
FSGS†	18.4693 / 0.4933 / 0.3947	23.4313 / 0.7413 / 0.2309
CoR-GS†	18.8696 / 0.5079 / 0.4047	23.3289 / 0.7363 / 0.2293
DropGaussian	19.7400 / 0.5770 / 0.3640	**24.1300** / 0.7620 / 0.2250
AD-GS†	19.0857 / 0.5176 / 0.3739	23.7933 / 0.7560 / 0.2068
Ours	**19.7913 / 0.5941 / 0.2981**	23.7786 / **0.7777 / 0.1704**

settings share 206 held-out views; Tanks and Temples and DTU contain 175 and 375 test views, respectively.

Metrics. We report PSNR, the structural similarity index measure (SSIM), and VGG-based learned perceptual image patch similarity (LPIPS) (Wang et al., 2004; Zhang et al., 2018). Mip-NeRF360 and Tanks and Temples use full images. DTU follows the sparse-view foreground-mask protocol: PSNR is evaluated on valid foreground pixels, while SSIM and LPIPS use white-background composites. Full mask processing and per-scene view counts are provided in the supplement.

Implementation. MASt3R initializes geometry and DA3 supplies structural depth. We train for 7,000 iterations on Mip-NeRF360, 900 on Tanks and Temples, and 10,000 on DTU. Within each dataset, repair variants share cameras, optimization, densification, and pruning settings and differ only in initialization geometry. Main repair and extrapolation results are three-run means; the matched safety-guard diagnostic is a single-run comparison using the same base model for all variants.

Baselines. We compare with 3DGS and representative sparse-view Gaussian methods, including DNGaussian, FSGS, CoR-GS, DropGaussian, and AD-GS (Kerbl et al., 2023; Li et al., 2024; Zhu et al., 2024; Zhang et al., 2024; Park et al., 2025; Patle et al., 2025). On Mip-NeRF360, we reproduced FSGS, CoR-GS, and AD-GS at $1/8$ resolution using the same scenes, splits, 206 test views, and VGG-LPIPS evaluation. We also reproduced DNGaussian on Tanks and Temples and AD-GS on DTU under the corresponding three-view evaluation protocols. For the remaining baseline entries, we report published results. Full baseline provenance is summarized in the supplement. On Mip-NeRF360 (Table 1), GRE-GS outperforms all matched reproductions at 12 views, improving over AD-GS by 0.7056 dB PSNR, 0.0765 SSIM, and 0.0758 LPIPS. At 24 views, AD-GS is 0.0147 dB higher in PSNR, while GRE-GS improves SSIM by 0.0217 and LPIPS by 0.0364. DropGaussian has the highest reported 24-view PSNR; GRE-GS has the highest SSIM and lowest LPIPS.

Under three input views (Table 2), GRE-GS reaches 21.5690 PSNR on Tanks and Temples, outperforming the best reported baseline by 1.8450 dB. On DTU, our AD-GS reproduction reaches 20.8018 PSNR, 0.8685 SSIM, and 0.1099 LPIPS, while GRE-GS obtains 23.6015, 0.9052, and 0.0761, respectively. Against the common-prior baseline using the same MASt3R initialization, DA3 supervision, training views, and optimization recipe, GRE-GS gains 0.8139 dB on Tanks and Temples and 0.4041 dB on DTU.

Completion gives the larger gain on Mip-NeRF360 and Tanks and Temples (+0.4304–0.7077 dB PSNR), while correction gives its largest gain on DTU (+0.2135 dB). Full repair adds 0.7854, 0.4539, 0.7414, and 0.2175 dB over the baseline, confirming complementary failure modes.

Qualitative analysis. Fig. 3 compares FSGS, CoR-GS, AD-GS, GRE-GS, and ground truth. GRE-GS produces sharper, more complete structures in these examples.

Component visualization. Fig. 4 separates correction on Bonsai (upper row) from completion on Garden (lower row), showing ground truth, pre/post-repair geometry, and pre/post-repair renders.

Table 2: Quantitative comparison under reported 3-view settings. Entries are PSNR ↑ / SSIM ↑ / LPIPS ↓. † indicates a result reproduced by us using released code.

Method	Tanks and Temples	DTU
3DGS	16.9920 / 0.5560 / 0.3520	17.6500 / 0.8160 / 0.1460
DNGaussian	18.3082 / 0.5700 / 0.4354†	18.9100 / 0.7900 / 0.1760
FSGS	19.6440 / 0.6370 / 0.3120	17.2400 / 0.8240 / 0.1580
CoR-GS	19.2460 / 0.6500 / 0.3420	19.2100 / 0.8530 / 0.1190
DropGaussian	19.4750 / 0.6520 / 0.3200	18.4100 / 0.8490 / 0.1360
AD-GS	19.7240 / 0.6780 / 0.2850	20.8018 / 0.8685 / 0.1099†
Ours-Base	20.7551 / 0.7243 / 0.2497	23.1974 / 0.9033 / 0.0780
Ours	**21.5690 / 0.7371 / 0.2283**	**23.6015 / 0.9052 / 0.0761**

Table 3: Component-wise ablation of Observed-Region Geometry Repair under identical Gaussian optimization. Values are three-run means reported as PSNR ↑ / SSIM ↑ / LPIPS ↓.

	Mip-NeRF360	
Configuration	24 views	12 views
Baseline	22.7900 / 0.7593 / 0.1830	18.9537 / 0.5772 / 0.3078
Completion Only	23.4444 / 0.7732 / 0.1732	19.3841 / 0.5890 / 0.2985
Correction Only	22.8741 / 0.7613 / 0.1819	18.9865 / 0.5790 / 0.3071
Full Geometry Repair	**23.5754 / 0.7766 / 0.1697**	**19.4076 / 0.5905 / 0.2977**

Configuration	Tanks 3-view	DTU 3-view
Baseline	20.7551 / 0.7243 / 0.2497	23.1974 / 0.9033 / 0.0780
Completion Only	21.4628 / 0.7348 / 0.2293	23.3544 / 0.9044 / 0.0768
Correction Only	20.7951 / 0.7250 / 0.2494	23.4109 / 0.9046 / 0.0766
Full Geometry Repair	**21.4965 / 0.7360 / 0.2280**	**23.4149 / 0.9049 / 0.0761**

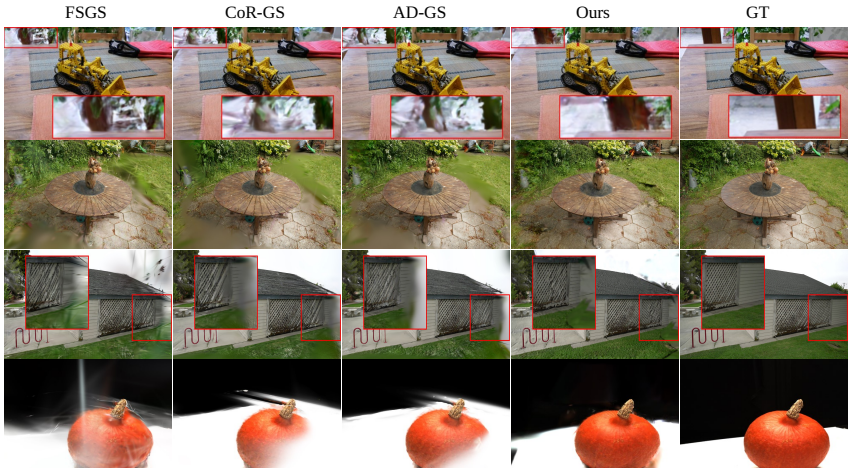

Figure 3: Qualitative comparison of observed-region repair. Rows show Kitchen, Garden, Barn, and DTU scan30; columns show FSGS, CoR-GS, AD-GS, GRE-GS, and ground truth. Red boxes mark representative artifacts or recovered regions.

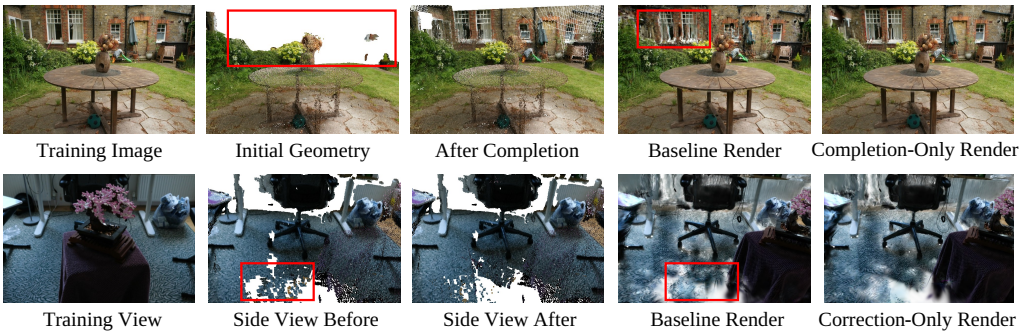

Figure 4: Correction and completion examples. Each row shows ground truth, pre/post-repair geometry, and pre/post-repair renders; the upper row is Bonsai correction and the lower row is Garden completion.

We freeze the repaired Gaussian model before extrapolation. Table 4 reports three-run novel-view means and the worst training-view PSNR change. Values are averaged over three runs.

Table 4: Effect of training-FoV-safe Gaussian extrapolation. Values are three-run means; triplets are PSNR / SSIM / LPIPS.

Dataset	Repair only	Full method	ΔPSNR
Mip-NeRF360 (24 views)	23.5754 / 0.7766 / **0.1697**	**23.7786** / **0.7777** / 0.1704	+0.2032
Mip-NeRF360 (12 views)	19.4076 / 0.5905 / **0.2977**	**19.7913** / **0.5941** / 0.2981	+0.3837
Tanks 3-view	21.4965 / 0.7360 / **0.2280**	**21.5690** / **0.7371** / 0.2283	+0.0725
DTU 3-view	23.4149 / 0.9049 / **0.0761**	**23.6015** / **0.9052** / **0.0761**	+0.1866

Dataset	ΔSSIM	ΔLPIPS	Worst train ΔPSNR
Mip-NeRF360 (24 views)	+0.0011	+0.0007	+0.0000
Mip-NeRF360 (12 views)	+0.0036	+0.0004	+0.0000
Tanks 3-view	+0.0011	+0.0003	−0.0100
DTU 3-view	+0.0003	0.0000	−0.0097

Extrapolation improves PSNR and SSIM in all settings, with PSNR gains of +0.2032, +0.3837, +0.0725, and +0.1866 dB; the worst training-view change is bounded by 0.01 dB.

Qualitative extrapolation. Fig. 5 shows Church (upper row) and Room (lower row) before and after extrapolation. Highlighted regions illustrate recovered or sharpened structure. Table 5 is a matched single-run diagnostic: within each dataset, repair-only, unguarded, and guarded extrapolation share the same frozen base. Its repair values therefore differ from the three-run means in Table 4.

The guard bounds worst training-view loss at 0.0100 dB.

On Mip-NeRF360 with 12 views, the unguarded variant obtains a higher novel-view PSNR than the guarded variant, but this gain comes with a severe -21.5562 dB degradation on the worst training view. The guard removes such unsafe candidates, prioritizing training-view fidelity over maximizing novel-view PSNR alone.



Figure 5: Training-FoV-safe extrapolation on Church (upper) and Room (lower). Columns show before extrapolation, after extrapolation, and ground truth.

Table 5: Matched single-run safety-guard ablation; higher is better. Full run details and candidate counts are provided in the supplement.

	Novel-view PSNR			Worst train ΔPSNR	
Dataset	Repair	No guard	Guarded	No guard	Guarded
Mip-NeRF360 (24 views)	23.5667	23.7222	**23.7903**	−13.0560	**+0.0000**
Mip-NeRF360 (12 views)	19.3817	**19.8935**	19.7899	−21.5562	**+0.0000**
Tanks	21.4982	20.6730	**21.5749**	−9.7083	**−0.0100**
DTU	23.4327	23.5969	**23.6220**	−40.1865	**−0.0097**

5 Conclusion

We present GRE-GS, an observation-conditioned framework for sparse-view 3DGS that separates observed-region geometry repair from training-FoV-safe extrapolation. Local completion and structural correction improve unreliable geometry, while virtual-view-driven extrapolation extends scene coverage with minimal disturbance to observed views. Experiments on Mip-NeRF360, Tanks and Temples, and DTU demonstrate consistent gains across sparse-view settings, supporting the effectiveness of explicitly treating these two failure modes separately.

AI use statement

In this work, we used generative AI tools to assist with research ideation and to provide feedback on experimental design and ablation settings. We also used generative AI tools for translation and language-related assistance in preparing the manuscript. We did not use generative AI tools to generate experimental results, fabricate data, or autonomously conduct experiments, and other required-disclosure uses that are not relevant to this work were not applicable.

Additionally, we used generative AI tools to aid writing and language polishing, improve the clarity and organization of the manuscript, and support literature retrieval and discovery by identifying potentially relevant prior work. All AI-assisted suggestions and retrieved references were independently reviewed and verified by the authors. Experimental results reported in this paper were obtained and validated by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469. IEEE, 2022.

Shen Chen, Jiale Zhou, and Lei Li. Dense point clouds matter: Dust-GS for scene reconstruction from sparse viewpoints. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3D gaussian splatting in few-shot images. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 811–820. IEEE, 2024.

Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. InstantSplat: Sparse-view gaussian splatting in seconds. *arXiv preprint arXiv:2403.20309*, 2024.

Liang Han, Bangcai Wei, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Sparse-view surface reconstruction using gaussian splatting through high-confidence depth propagation with normal priors. In *European Conference on Computer Vision*, pp. 491–511. Springer, 2026.

Yuhang Hong, Bo Tao, and Zeyu Gong. ShareGS: Hole completion with sparse inputs based on reusing selected scene information. *Pattern Recognition*, 167:111729, 2025.

Sungwon Hwang, Min-Jung Kim, Taewoong Kang, Jayeon Kang, and Jaegul Choo. VEGS: View extrapolation of urban scenes in 3D gaussian splatting using learned priors. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2024.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large-scale multi-view stereopsis evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 406–413, 2014.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Hyeseong Kim, Geonhui Son, Deukhee Lee, and Dosik Hwang. TWINGS: Thin plate splines warp-aligned initialization for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26062–26071, June 2026.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision*, pp. 71–91. Springer, 2024.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20775–20785. IEEE, 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*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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. *Communications of the ACM*, 65(1):99–106, 2021.

Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5470–5480. IEEE, 2022.

Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. CoherentGS: Sparse novel view synthesis with coherent 3D gaussians. In *European Conference on Computer Vision*, pp. 19–37. Springer, 2024.

Avinash Paliwal, Xilong Zhou, Wei Ye, Jinhui Xiong, Rakesh Ranjan, and Nima Khademi Kalantari. Ri3D: Few-shot gaussian splatting with repair and inpainting diffusion priors. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25094–25103. IEEE, 2025.

Hyunwoo Park, Gun Ryu, and Wonjun Kim. DropGaussian: Structural regularization for sparse-view gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21600–21609. IEEE, 2025.

Gurutva Patle, Nilay Girgaonkar, Nagabhushan Somraj, and Rajiv Soundararajan. AD-GS: Alternating densification for sparse-input 3D gaussian splatting. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in gaussian splatting. In *European Conference on Computer Vision*, pp. 347–362. Springer, 2024.

Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. SparseNeRF: Distilling depth ranking for few-shot novel view synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9031–9042. IEEE, 2023.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

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.

Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. NeRFiller: Completing scenes via generative 3D inpainting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20731–20741. IEEE, 2024.

Chung-Ho Wu, Yang-Jung Chen, Ying-Huan Chen, Jie-Ying Lee, Bo-Hsu Ke, Chun-Wei Tuan Mu, Yi-Chuan Huan, Chin-Yang Lin, Min-Hung Chen, Yen-Yu Lin, et al. AuraFusion360: Augmented unseen region alignment for reference-based 360 unbounded scene inpainting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16366–16376. IEEE, 2025.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. ReconFusion: 3D reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21551–21561, 2024.

Peng Xiang, Liang Han, Hui Zhang, Yu-Shen Liu, and Zhizhong Han. VGGS: VGGT-guided gaussian splatting for efficient and faithful sparse-view surface reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 10969–10977, 2026.

Haolin Xiong, Sairisheek Muttukuru, Hanyuan Xiao, Rishi Upadhyay, Pradyumna Chari, Yajie Zhao, and Achuta Kadambi. SparseGS: Sparse view synthesis using 3D gaussian splatting. In *2025 International Conference on 3D Vision (3DV)*, pp. 1032–1041. IEEE, 2025.

Wangze Xu, Huachen Gao, Shihe Shen, Rui Peng, Jianbo Jiao, and Ronggang Wang. MVPGS: Excavating multi-view priors for gaussian splatting from sparse input views. In *European Conference on Computer Vision*, pp. 203–220. Springer, 2024.

Jiawei Yang, Marco Pavone, and Yue Wang. FreeNeRF: Improving few-shot neural rendering with free frequency regularization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8254–8263. IEEE, 2023.

Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. CoR-GS: Sparse-view 3D gaussian splatting via co-regularization. In *European Conference on Computer Vision*, pp. 335–352. Springer, 2024.

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

Yingji Zhong, Zhihao Li, Dave Zhenyu Chen, Lanqing Hong, and Dan Xu. Taming video diffusion prior with scene-grounding guidance for 3D gaussian splatting from sparse inputs. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6133–6143. IEEE, 2025.

Zhizhuo Zhou and Shubham Tulsiani. SparseFusion: Distilling view-conditioned diffusion for 3D reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12588–12597. IEEE, 2023.

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pp. 145–163. Springer, 2024.

A Evaluation Protocol and Data Splits

Mip-NeRF360 uses seven scenes and 206 held-out views under both the 12-view and 24-view settings; the two settings share the same test images. Tanks and Temples uses eight scenes and 175 test views, and DTU uses 15 scans with 25 test views per scan, for 375 views. All variants within a setting share the same cameras. Exact per-scene lists are retained in the experiment directories. Geometry construction, optimization, virtual-camera construction, candidate generation, and safety filtering use training-side information only; test images are accessed after the representation is frozen.

Mip-NeRF360 is evaluated from `images_8`, with scene-dependent dimensions. Tanks and Temples uses 960×540 images. DTU uses 397×298 rectified images, approximately $1/4$ of the original resolution. We report PSNR, SSIM, and VGG-LPIPS. Mip-NeRF360 and Tanks and Temples use full-image evaluation. For DTU, the foreground mask is resized with bicubic interpolation; mask value 255 is retained for PSNR, while SSIM and LPIPS use white-background compositing.

Entries marked † in Tables 1 and 2 are our official-code reproductions: FSGS, CoR-GS, and AD-GS on Mip-NeRF360 at $1/8$ resolution, DNGaussian on Tanks and Temples, and AD-GS on DTU. The Tanks and Temples DNGaussian result uses the released LLFF configuration with random point-cloud initialization. Unmarked baseline entries report published results: the Mip-NeRF360 3DGS and DropGaussian rows are original-paper $1/8$ reports, the other Tanks and Temples baselines come from the unified AD-GS evaluation, and the remaining DTU values combine a CoR-GS 3DGS reimplementation, later reproductions, and reported sparse-view results. Table 6 summarizes this provenance.

B Implementation Details

B.1 Scaffold preprocessing and observed-region repair

Starting from MASt3R points with confidence q_j, we retain $q_j > 1.5$, voxelize with a scene-adaptive size, and keep the original sample nearest each occupied voxel center. Mip-NeRF360 targets approximately 600,000 points per scene; Tanks and Temples and DTU use frozen scene-specific targets selected from training-view coverage. Small detached components are removed only when they also fail the structural-body distance test.

Table 6: Provenance of numeric baseline rows in the main tables.

Benchmark	Main-table row(s)	Numeric source	Image scale	Evaluation protocol
Mip-NeRF360	GRE-GS 12/24	This work, three-run mean	$1/8$	GRE-GS protocol
Mip-NeRF360	3DGS 12/24	DropGaussian paper report	$1/8$	DropGaussian report
Mip-NeRF360	FSGS 12	Our official-code reproduction	$1/8$	GRE-GS protocol
Mip-NeRF360	CoR-GS 12/24	Our official-code reproduction	$1/8$	GRE-GS protocol
Mip-NeRF360	AD-GS 12/24	Our official-code reproduction	$1/8$	GRE-GS protocol
Mip-NeRF360	FSGS 24	Our official-code reproduction	$1/8$	GRE-GS protocol
Mip-NeRF360	DropGaussian 12/24	Original-paper report	$1/8$	Original-paper protocol
Tanks and Temples	DNGaussian	Our official-code reproduction	960×540	GRE-GS protocol
Tanks and Temples	Other baseline rows	AD-GS unified evaluation	Native	AD-GS protocol
DTU	AD-GS	Our official-code reproduction	397×298	GRE-GS evaluation protocol
DTU	3DGS	CoR-GS reimplementation	Approx. $1/4$	CoR-GS protocol
DTU	FSGS, DropGaussian	Later reproductions	Approx. $1/4$	Source reproduction protocol

For each missing region, rings of radius $4, 8, 12, 20$, and 32 pixels provide paired reconstructed and DA3 depths. Coverage is dilated by two pixels; boundary regions within four pixels are excluded; anchors are sampled every three pixels. A fit uses at least 80 fitting and 20 validation anchors, with 20% held out for validation. We compare affine and scale-only maps with five trimmed refitting iterations and an 85th-percentile inlier cutoff, rejecting regions whose validation median relative error exceeds 0.15 or whose 90th percentile exceeds 0.50.

Correction uses supported log-depth residual trends. Seeds require residual magnitude above 0.15 and grow through residuals above 0.08; DA3 edges and a three-pixel protection band are excluded. Components require at least 18 pixels, mean log error 0.12, sign coherence 0.82, and bounded local support. Multiple-view proposals retain the strongest supported update. A source-view/object group with at least 300 points is rolled back when more than 85% of its points fall in front of supported surfaces under cross-view z-buffer checks.

B.2 Gaussian optimization

Let $\widehat{d}_i = \log(\widehat{D}_i + \epsilon)$ and $d_i^A = \log(D_i^A + \epsilon)$. With local standardization $\mathcal{N}_W$ and valid-depth confidence w_i, we optimize

$$\mathcal{L}_{\text{rgb}} = (1 - \lambda_{\text{ssim}})\|\widehat{I}_i - I_i\|_1 + \lambda_{\text{ssim}}\big(1 - \text{SSIM}(\widehat{I}_i, I_i)\big), \tag{1}$$

$$\mathcal{L}_{\text{rel}} = \frac{\sum_p w_i(p)\left|\mathcal{N}_W(\widehat{d}_i)(p) - \mathcal{N}_W(d_i^A)(p)\right|}{\sum_p w_i(p) + \epsilon}, \tag{2}$$

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_d(t)\left(\mathcal{L}_{\text{rel}} + 0.02\mathcal{L}_{\text{abs}}\right). \tag{3}$$

Here $\mathcal{L}_{\text{abs}}$ is a weak absolute log-depth stabilization term. Local standardization uses a 31×31 patch and valid-depth ratio 0.25; the depth weight decreases from 0.03 to 0.005. We train Mip-NeRF360, Tanks and Temples, and DTU for 7,000, 900, and 10,000 iterations, respectively, with densification ending at iteration 7,000 and opacity resets every 3,000 iterations where applicable. Main comparisons report three independent runs.

B.3 VIRTUAL VIEWS, PROPOSALS, AND APPEARANCE TRANSFER

For Mip-NeRF360, each training camera generates pitch and yaw values from $\{-30, -15, -5, 0, 5, 15, 30\}^\circ$ with outward translation 0.12. Tanks and Temples and DTU use translation 0.060 and selected $5^\circ/10^\circ$ perturbations, including fan rotations. Virtual cameras render only alpha, depth, and Gaussian footprints from the frozen repaired model. Our low-coverage detector uses alpha thresholds 0.25/0.45 for holes/support, a minimum component size of 60 pixels, adjacency dilation of 11 pixels, and 0.25-voxel proposal deduplication with a 250,000-Gaussian cap. Proxy depths use factors $\{0.9, 1.0, 1.1\}$; local propagation uses a 16-pixel patch and maximum relative residual 0.08.

One color MLP is trained independently for each frozen repaired scene before any extrapolated candidate is added. We sample 128 compact pseudo-hole blocks with 192 target Gaussians per block and split complete blocks 80/20 into training and validation sets. Targets are removed from the neighbor-search pool, forcing SH-DC color reconstruction from 16 remaining neighbors. Three spatially separated 2,500-Gaussian blocks are held out for the internal inpainting check and never used for fitting. Each target–neighbor feature has 19 dimensions covering relative position/distance, standardized SH-DC color and contrast, scale and scale difference, quaternion alignment, opacity, and opacity difference. A shared $19 \to 64 \to 32 \to 1$ MLP scores each neighbor; softmax over 16 scores produces convex color-fusion weights. AdamW (Loshchilov & Hutter, 2019) uses learning rate 10^{-3}, weight decay 10^{-5}, batch size 4096, validation-based reduction, and early stopping after 12 stale epochs. Only SH-DC color uses learned weights; scale, rotation, and opacity use inverse-distance propagation.

B.4 TRAINING-FOV SAFETY GUARD

Candidates are classified over every real training camera as invisible, occluded, or risky. For Tanks and Temples and DTU, invisible candidates use opacity factor 1.0, occluded candidates use a searched class factor, and risky candidates use a footprint-derived transition weight followed by a separately searched factor. The occluded and risky searches each use eight coarse-to-fine probes to find the largest factor satisfying the per-view 0.01-dB PSNR tolerance, after which candidates below opacity 0.01 are removed. Mip-NeRF360 retains only candidates with renderer radius zero in every training camera, yielding measured training-view change exactly zero. The final representation is rendered again in all training views.

C KNOWN FAILURE MODES

Observed-region repair can inherit incorrect monocular structure in weakly anchored components; validation and cross-view rollback mitigate this failure mode. Extrapolation can add few or no candidates when every training camera projects a large footprint, as in Ballroom and Horse, while weak depth support can yield small regressions in individual DTU scans. The safety guard constrains measured training-view changes, while unobserved geometry remains underconstrained.

D COMPACT QUANTITATIVE SUPPLEMENT

D.1 PER-SCENE SAFE EXTRAPOLATION RESULTS

Table 7 reports per-scene PSNR changes from safe extrapolation. Scan45, Scan63, and Scan110 on DTU show small PSNR decreases.

Table 7: Per-scene PSNR changes from safe extrapolation.

Setting	Scene	Repair only	Full method	ΔPSNR
Mip-NeRF360 (24 views)	bicycle	21.802	21.856	+0.054
Mip-NeRF360 (24 views)	bonsai	23.371	23.995	+0.624
Mip-NeRF360 (24 views)	counter	23.322	23.419	+0.097
Mip-NeRF360 (24 views)	garden	24.597	24.827	+0.230

Setting	Scene	Repair only	Full method	ΔPSNR
Mip-NeRF360 (24 views)	kitchen	23.917	24.188	+0.271
Mip-NeRF360 (24 views)	room	24.972	25.109	+0.137
Mip-NeRF360 (24 views)	stump	22.903	23.056	+0.153
Mip-NeRF360 (12 views)	bicycle	18.897	18.981	+0.084
Mip-NeRF360 (12 views)	bonsai	17.903	18.830	+0.927
Mip-NeRF360 (12 views)	counter	18.690	19.139	+0.449
Mip-NeRF360 (12 views)	garden	20.458	20.955	+0.497
Mip-NeRF360 (12 views)	kitchen	21.717	22.078	+0.361
Mip-NeRF360 (12 views)	room	20.866	21.288	+0.422
Mip-NeRF360 (12 views)	stump	17.229	17.268	+0.039
Tanks	Ballroom	24.115	24.115	+0.000
Tanks	Barn	20.000	20.138	+0.138
Tanks	Church	18.679	18.998	+0.319
Tanks	Family	22.592	22.606	+0.014
Tanks	Francis	21.906	22.004	+0.098
Tanks	Horse	20.004	20.004	+0.000
Tanks	Ignatius	21.291	21.333	+0.042
Tanks	Museum	23.291	23.354	+0.063
DTU	scan8	23.091	23.222	+0.131
DTU	scan21	18.970	19.012	+0.042
DTU	scan30	26.636	26.637	+0.001
DTU	scan31	18.175	18.545	+0.370
DTU	scan34	23.995	24.115	+0.120
DTU	scan38	22.315	22.430	+0.115
DTU	scan40	23.211	24.599	+1.388
DTU	scan41	23.050	23.075	+0.025
DTU	scan45	20.387	20.382	-0.005
DTU	scan55	26.130	26.698	+0.568
DTU	scan63	26.372	26.169	-0.203
DTU	scan82	24.796	24.797	+0.001
DTU	scan103	26.638	26.642	+0.004
DTU	scan110	22.601	22.594	-0.007
DTU	scan114	24.856	25.104	+0.248

D.2 Run-to-run variation

Table 8 reports variation over three runs.

Table 8: Run-to-run variation. Values are three-run mean ± standard deviation.

Setting	Configuration	PSNR	SSIM	LPIPS
Mip-NeRF360 (24 views)	Base initialization	22.7900 ± 0.0542	0.7593 ± 0.0012	0.1830 ± 0.0011
	Full geometry repair	23.5754 ± 0.0140	0.7766 ± 0.0002	0.1697 ± 0.0003
	Full method	23.7786 ± 0.0430	0.7777 ± 0.0019	0.1704 ± 0.0013
Mip-NeRF360 (12 views)	Base initialization	18.9537 ± 0.0086	0.5772 ± 0.0002	0.3078 ± 0.0002
	Full geometry repair	19.4076 ± 0.0233	0.5905 ± 0.0001	0.2977 ± 0.0002
	Full method	19.7913 ± 0.0219	0.5941 ± 0.0017	0.2981 ± 0.0013
Tanks	Base initialization	20.7551 ± 0.0532	0.7243 ± 0.0003	0.2497 ± 0.0007
Tanks	Full geometry repair	21.4965 ± 0.0046	0.7360 ± 0.0002	0.2280 ± 0.0001
Tanks	Full method	21.5690 ± 0.0133	0.7371 ± 0.0016	0.2283 ± 0.0015
DTU	Base initialization	23.1974 ± 0.0183	0.9033 ± 0.0002	0.0780 ± 0.0001
DTU	Full geometry repair	23.4149 ± 0.0217	0.9049 ± 0.0001	0.0761 ± 0.0001
DTU	Full method	23.6015 ± 0.0241	0.9052 ± 0.0001	0.0761 ± 0.0001

D.3 SAFETY-GUARD DIAGNOSTIC

Table 9 reports novel-view PSNR, worst training-view change, final additions, and retention rate for the safety-guard diagnostic. The guard limits the worst measured training-view PSNR loss to 0.01 dB. The changed-pixel analysis reports the pixels modified by extrapolation.

Table 9: Safety-guard diagnostic with a fixed repaired representation per setting.

Setting	Configuration	Novel PSNR	Worst train Δ	Added	Retention
Mip-NeRF360 (24 views)	Repair only	23.5667	+0.0000	–	–
	No guard	23.7222	-13.0560	1,390,643	1.000
	Guarded	23.7903	+0.0000	911,029	0.655
Mip-NeRF360 (12 views)	Repair only	19.3817	+0.0000	–	–
	No guard	19.8935	-21.5562	882,649	1.000
	Guarded	19.7899	+0.0000	407,354	0.462
Tanks (3 views)	Repair only	21.4982	+0.0000	–	–
Tanks (3 views)	No guard	20.6730	-9.7083	940,166	1.000
Tanks (3 views)	Guarded	21.5749	-0.0100	752,929	0.801
DTU (3 views)	Repair only	23.4327	+0.0000	–	–
DTU (3 views)	No guard	23.5969	-40.1865	855,480	0.920
DTU (3 views)	Guarded	23.6220	-0.0097	685,254	0.737