Supplementary Material for: GeoTrack-GS: Refining Depth Priors with Self-Supervised Geometric Constraints

Zhipeng Ye[a], Zihao Lu[a], Yuan Zhang[a], Wengjie Qin[a], Jiayi Hong[a],
, Yan Cao[b], Xuming Wu[a,*], Zhibin Shao[c,*]

[a] *Lingnan Normal University,Zhanjiang 524048, China*
[b] *Luoyang Normal University,Luoyang 471934, China*
[c] *Shenzhen Polytechnic University, Shenzhen 518055, China*

S1 Evaluation Protocol

Overview. All quantitative results in the main paper and this supplementary are obtained by re-evaluating *all* methods using the same metric implementations and the same input/output resolution. We do not mix quoted numbers from prior works in the main tables.

Photometric metrics (PSNR/SSIM). PSNR and SSIM are computed on rendered RGB images in **sRGB** space (8-bit images converted to float in $[0, 1]$) without linearization. We use the standard SSIM implementation with an 11×11 Gaussian window (default settings in common libraries).

Perceptual metric (LPIPS). LPIPS is computed using the **VGG backbone** with the official LPIPS implementation. We always report LPIPS(VGG) for consistency across datasets.

Resolution alignment and resizing. For all datasets, renderings are resized to the ground-truth resolution using **bilinear interpolation** before computing photometric metrics. No aspect-ratio change is performed (datasets provide fixed resolutions).

DTU masking/cropping for photometric metrics. For DTU, we apply the **provided foreground mask** when computing PSNR/SSIM/LPIPS to avoid background bias. We report masked photometric scores.

DTU geometric evaluation (CD-L1 and F-score). We compute CD-L1 and F-score using the **standard DTU evaluation code**. We follow the default settings in the DTU evaluation protocol, including point-cloud alignment and distance thresholds. We extract a reconstructed point cloud from the Gaussian model by (i) rendering depth maps for all training views, (ii) back-projecting valid pixels to 3D, and (iii) merging points across views with outlier filtering. Exact thresholds and preprocessing settings are listed in Sec. S1.1.

**Corresponding authors: wuxm@lingnan.edu.cn (X. Wu); zhibin_shao@szpt.edu.cn (Z. Shao)

Table S1. DTU evaluation parameters. Fixed settings used for point-cloud extraction, preprocessing, and DTU geometry evaluation.

Item	Symbol	Value
Depth validity criterion	–	valid depth pixels (no hole)
Opacity threshold (for valid points)	$\alpha_{\min}$	0.5
Voxel downsampling size	v	1.0 mm
Statistical outlier neighbors (SOR)	k	20
Outlier std multiplier (SOR)	σ_o	1.5
F-score distance threshold	τ	1.0 mm

Fair point-cloud extraction for all methods. For *every* method, we extract the reconstructed point cloud from its trained Gaussian model using the *same* renderer and the *same* extraction pipeline. Specifically, we render depth and opacity maps for all training views, keep pixels satisfying (i) finite depth ($d > 0$ and not NaN/Inf) and (ii) opacity $\alpha > \alpha_{\min}$, then back-project valid pixels to 3D and merge points across views. We apply identical voxel downsampling (v) and statistical outlier removal (SOR with (k, σ_o)) before running the official DTU evaluation code. This ensures a fair and implementation-consistent comparison across all baselines.

S1.1 Implementation Details and Parameters

DTU geometry evaluation settings (fixed). We use the official DTU evaluation code for computing CD-L1 and F-score, including its default alignment procedure. To avoid discrepancies caused by different point-cloud preprocessing choices, we fix the point-cloud extraction and filtering pipeline for *all* methods. Specifically, we extract a point cloud by rendering depth maps, back-projecting valid pixels, merging points across views, and applying the same voxel downsampling and outlier removal before DTU evaluation. All DTU geometry numbers reported in this paper are produced with the parameters in Table SS1.

Note on track filtering. Table S3 summarizes the *raw* tracks extracted from the matching graph. For optimization, we discard tracks with fewer than $N_{\min} = 3$ observations and with large reprojection residuals, and only the filtered set is used in $\mathcal{L}_{geo}$.

Software versions. We report the versions of key libraries (PyTorch, CUDA, LPIPS package) and the commit hashes of evaluation scripts in the released code.

DTU point-cloud extraction settings. We use a depth confidence threshold τ_d to filter invalid depth pixels and apply voxel downsampling with voxel size v before DTU evaluation. We remove outliers using a statistical filter with k neighbors and standard deviation multiplier σ_o.

DTU F-score thresholds. We compute F-score using a single distance threshold $\tau = 1.0$ mm (F-score@1mm), consistent with Table SS1 and used for all DTU results

Reproducibility. All tables are generated by the same script that loads per-scene metrics and computes mean/std. We will release the evaluation scripts and configuration files upon acceptance.

Randomness and reporting. Unless otherwise stated, all experiments are run with a fixed random seed. For analyses involving randomness (e.g., view sampling or track dropping), we report mean ± std. over scenes. Due to computational cost, we do not repeat full training with multiple seeds for every method; we will release scripts for reproducibility.

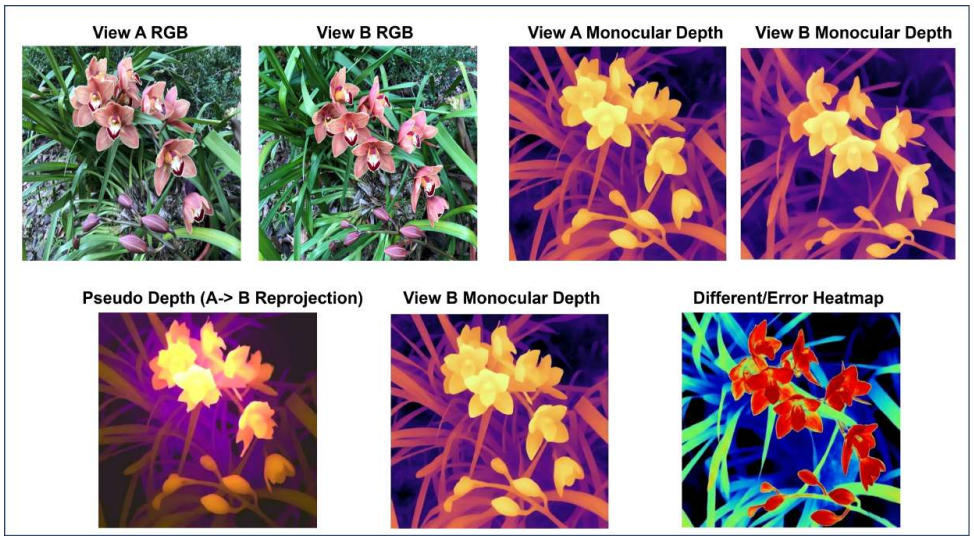


Figure S1. Monocular depth prior inconsistency on an LLFF scene. Top row: two input views and their monocular depth predictions. Bottom row: pseudo depth obtained by back-projecting the monocular depth of view A into 3D and reprojecting it into view B, the monocular depth of view B, and the corresponding error heatmap. Large discrepancies on flowers and thin leaves indicate that monocular depth provides view-dependent, multi-view inconsistent geometric cues, motivating our geometry-first correction.

S2 Diagnostic Analyses

To better understand how GeoTrack-GS behaves in sparse-view 3DGS settings, we visually inspect monocular depth prior inconsistency, geometric behavior, and geometry–appearance decoupling on representative scenes.

Monocular depth prior inconsistency. We quantify the multi-view inconsistency of monocular depth priors by reprojecting the predicted depth from one view into another and comparing it with the depth predicted directly on the target view. As illustrated in Fig. S1, even if each depth map looks plausible on its own, the cross-view reprojection error can be large and highly structured, especially in texture-less regions and near depth discontinuities. This explains why aggressively enforcing monocular priors may distort geometry in challenging areas, and motivates using them only as a weak cue combined with stronger, track-based constraints.

Geometry–appearance decoupling. Fig. S2 illustrates how GeoTrack-GS decouples geometry from view-dependent appearance on a specular tabletop scene. The baseline (3DGS/FSGS, Fig. S2a) explains highlights by deforming the surface: the rendered RGB is blurry and the normal map exhibits obvious "fake bumps" on the table. When we apply only our geometric constraints ($\mathcal{L}_{geo} + \mathcal{L}_{aniso}$, Fig. S2b), the table becomes clean and flat in the normal map, but the RGB appearance is overly matte and loses high-frequency specular structure. With the full GeoTrack-GS model including GT-DCA (Fig. S2c), sharp specular streaks are recovered while the table geometry remains flat and well-behaved. This confirms that, under our geometry-first design, GT-DCA successfully transfers high-frequency, view-dependent effects to the appearance branch instead of letting them leak into the geometry.

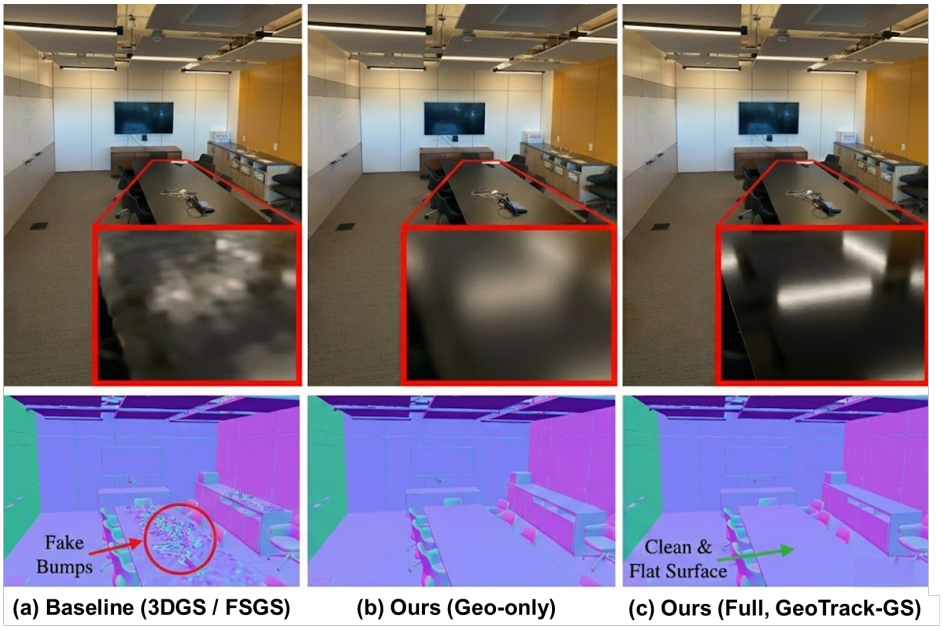
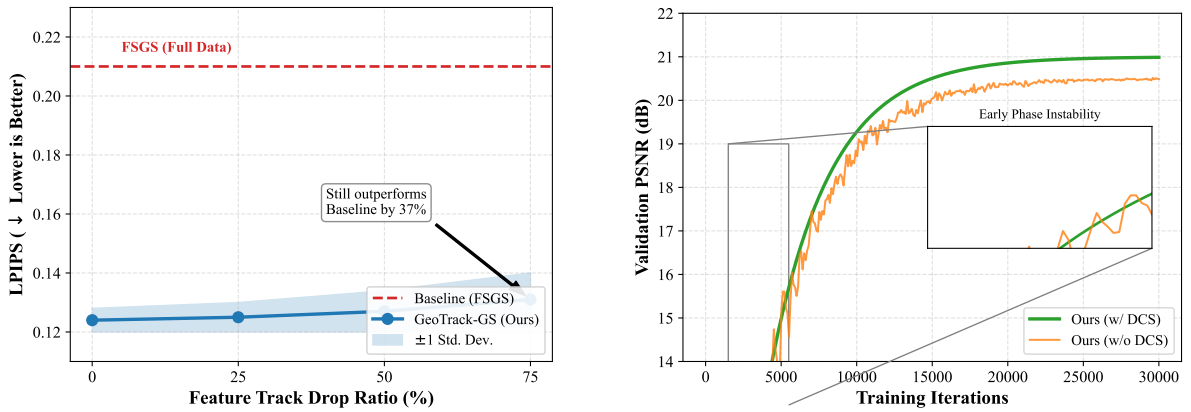


Figure S2. Geometry–appearance decoupling on a specular tabletop scene. (a) Baseline (3DGS/FSGS) tends to bake highlights into geometry, producing blurry reflections and "fake bumps" in the normal map. (b) Ours (Geo-only) with $\mathcal{L}_{geo} + \mathcal{L}_{aniso}$ yields a clean and flat surface but over-smooths specular appearance. (c) Ours (Full, GeoTrack-GS) combines geometric constraints with GT-DCA, preserving flat geometry while recovering sharp specular highlights.



(a) Robustness to feature track sparsity (LLFF, 3 views). **(b)** Training stability with/without DCS (LLFF *Fern*, 3 views).

Figure S3. Track sparsity robustness and training stability. **(a)** Mean LPIPS with shaded bands (±1 std.) across LLFF scenes when randomly dropping 25%/50%/75% of tracks. **(b)** Validation PSNR curves illustrating that DCS suppresses early oscillations and improves convergence stability.

S3 Robustness to Track Sparsity and Training Stability

This section provides detailed results for the analysis summarized in the main paper (Sec. 4.7), including (i) robustness to feature track sparsity and (ii) training stability with and without the proposed Decoupled Constraint Stabilization (DCS).

S3.1 Robustness to Feature Track Sparsity

We evaluate whether GeoTrack-GS requires dense track coverage for stable optimization. On LLFF under the 3-view setting, we randomly drop 25%, 50%, and 75% of the valid feature tracks used by the macro-level geometric loss $\mathcal{L}_{geo}$, and retrain the model with all other settings fixed. Fig. S3(a) reports the mean LPIPS over scenes with ±1 standard deviation.

Despite aggressive track removal, GeoTrack-GS remains robust: the LPIPS degradation is minor even at 75% track drop, indicating that a sparse but reliable geometric skeleton is sufficient to constrain global structure and correct multi-view inconsistent depth priors in our framework.

S3.2 Training Stability with and without DCS

We further analyze the effect of DCS on optimization dynamics. On a representative LLFF scene (*Fern*, 3 views), we track validation PSNR during training and compare: (i) *w/o DCS*, which uses fixed weights for $\mathcal{L}_{depth}$, $\mathcal{L}_{geo}$, and $\mathcal{L}_{pseudo}$, and (ii) *full GeoTrack-GS*, which applies quality-aware gating for $\mathcal{L}_{geo}$ and curriculum scheduling for $\mathcal{L}_{pseudo}$ (Sec. 3.3 in the main paper).

As shown in Fig. S3(b), the model without DCS exhibits noticeable oscillations in the early phase, while DCS yields smoother convergence and slightly better final validation performance. This supports our design choice of decoupling spatially sparse noise (track outliers) from temporally unstable supervision (pseudo-views) via tailored modulation.

S3.3 Reproducibility Details

Track dropping protocol. Tracks are dropped *after* triangulation and filtering, i.e.,from the final *filtered* track set used in $\mathcal{L}_{geo}$ (i.e., tracks with $|\mathcal{V}_k| \geq N_{\min}$ after reprojection-error filtering). For each drop ratio, we sample a new subset uniformly at random and keep the subset fixed throughout training. We repeat the experiment across scenes and report the mean and standard deviation in Fig. S3(a).

Table S2. Optional numeric summary of track dropping (LLFF, 3 views). LPIPS is averaged over scenes; values correspond to Fig. S3(a).

Track drop ratio	0%	25%	50%	75%
LPIPS (mean)	0.124	0.126	0.128	0.131

DCS settings. The outlier ratio $r_{\text{out}}(t)$ is computed as the fraction of tracks whose reprojection error exceeds a fixed pixel threshold (measured on the training resolution). The geometric weight follows $\lambda_{geo}(t) = \lambda_{geo}^{\max}(1 - r_{\text{out}}(t))^{\gamma}$ with $\gamma = 2$ (Table S4). The pseudo-view weight uses a linear ramp schedule starting at t_0 with ramp length t_{ramp}. All remaining hyperparameters are identical to the default configuration in Table S4.

S4 Feature Track Statistics and Hyperparameters

This section provides the comprehensive statistical data and detailed parameter settings referenced in the main paper (Section 4.1).

S4.1 Feature Track Statistics

Table S3 details the characteristics of the feature tracks extracted from the SfM process. These statistics demonstrate that our method operates effectively even with moderately sparse or noisy geometric skeletons.

Table S3. Feature Track Statistics. Average number of extracted 3D feature tracks per scene, average track length, and initial outlier ratio. *The average track length is computed before applying the $N_{\min} = 3$ observation filter.* Tracks used by the training loss $\mathcal{L}_{geo}$ always satisfy $|\mathcal{V}_k| \geq N_{\min}$.

Dataset	Avg. Tracks / Scene	Avg. Track Length	Init. Outlier Ratio
LLFF (3 views)	9,420	2.6	12.5%
LLFF (9 views)	38,150	5.4	8.2%
DTU (3 views)	4,850	2.5	9.1%
DTU (9 views)	22,600	4.8	6.5%
Mip-NeRF 360	65,300	3.9	14.8%

S4.2 Detailed Hyperparameter Settings

Table S4 lists the exact values for all key hyperparameters used in our experiments.

Table S4. Detailed Hyperparameter Settings Default values used for all experiments.

Category	Parameter	Symbol	Value
Optimization	Total Iterations	N_{iters}	30,000
	Position LR	η_{pos}	$2.0 \times 10^{-3} \rightarrow 2.0 \times 10^{-5}$
	Feature LR	η_{feat}	0.0025
	GT-DCA LR	η_{dca}	1×10^{-4}
Loss Weights	Photometric (L1)	λ_1	0.8
	Photometric (SSIM)	λ_{ssim}	0.2
	Monocular Depth	λ_{depth}	0.1
	Macro Geometry	λ_{geo}^{max}	1.0
	Micro Anisotropy	λ_{aniso}	1.0
Micro Geometry	PCA Neighbors	K	16
	Max Anisotropy	θ_{aniso}	5.0
	Scale Matching	ϵ	1×10^{-6}
Constraint	Gating Exponent	γ	2.0
	Warm-up Iters	T_{warm}	3,000

S5 Detailed Robustness Analysis to View Sparsity

In Section 4.3 of the main paper, we summarized the performance degradation trends. Here, we provide the full numerical results for the sparse-view analysis on both LLFF and DTU datasets under 3, 6, and 9 input views.

S5.1 Results on LLFF Dataset

Table S5 presents the detailed breakdown of PSNR, SSIM, and LPIPS metrics for the LLFF dataset. As discussed in the main paper, GeoTrack-GS maintains competitive performance even as the number of views drops to 3.

View selection protocol. For each scene, we sample the training views uniformly at random under a fixed random seed and keep the selected view indices *identical* for all methods to ensure a fair comparison. Unless otherwise stated, the reported numbers correspond to one fixed split per scene. We provide the exact view indices (or the sampling seed) in the released configuration files.

Table S5. Sparse-view analysis on LLFF (Full Data). Comparison under 3, 6, and 9 input views. **Bold** indicates best performance.

Method	3 views			6 views			9 views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
FSGS (Baseline)	20.31	0.652	0.288	24.55	0.795	0.177	25.89	0.845	0.143
DNGaussian	19.12	0.649	0.294	22.18	0.755	0.198	23.17	0.788	0.180
SparseGS	19.86	0.668	0.322	22.85	0.801	0.179	24.10	0.855	0.152
CoR-GS	20.45	0.712	0.196	24.49	0.837	0.115	26.06	0.874	0.089
FewViewGS	20.54	0.693	0.214	23.09	0.769	0.164	25.90	0.868	0.095
SE-GS	**20.79**	**0.724**	**0.183**	**24.78**	**0.839**	0.110	**26.36**	0.878	**0.084**
DropoutGS	19.35	0.622	0.282	23.35	0.791	0.177	24.33	0.825	0.160
SCGaussian	20.77	0.705	0.218	24.67	0.831	0.119	26.12	0.885	0.087
GeoTrack-GS	20.52	0.691	0.231	24.55	0.830	**0.108**	26.08	**0.879**	0.093

S5.2 Results on DTU Dataset

Table S6 provides the quantitative results on the DTU dataset. Our method demonstrates consistent superiority in geometric fidelity (CD-L1 and F-score).

Table S6. Sparse-view analysis on DTU (Full Data). Comparison of PSNR ↑, F-score ↑, and CD-L1 (mm) ↓.

Method	3 views			6 views			9 views		
	PSNR	F-score	CD-L1	PSNR	F-score	CD-L1	PSNR	F-score	CD-L1
FSGS (Baseline)	18.12	32.4	0.75	23.50	40.2	0.55	26.10	48.5	0.45
DNGaussian	18.91	38.5	0.62	24.30	47.2	0.45	26.85	54.5	0.37
SparseGS	18.89	31.2	0.82	24.15	41.5	0.60	26.60	49.0	0.48
CoR-GS	19.21	40.5	0.58	24.51	48.8	0.42	27.18	55.6	0.35
SE-GS	19.24	36.8	0.68	25.28	46.5	0.44	28.08	54.0	0.36
DropoutGS	**20.22**	28.5	0.95	**25.58**	38.0	0.72	**28.50**	46.5	0.58
SCGaussian	20.56	42.0	0.52	25.45	51.0	0.38	28.20	60.5	0.30
FewViewGS	19.74	39.0	0.65	24.75	49.5	0.41	27.31	57.2	0.34
Ours	19.36	**43.0**	**0.49**	25.30	**52.5**	**0.36**	28.32	**63.5**	**0.28**

S5.3 Per-scan DTU Geometry Results

Table SS7 reports per-scan CD-L1 and F-score on DTU under the 3-view setting. All numbers are computed using the official DTU evaluation code with the fixed parameters in Table SS1.

Table S7. Per-scan DTU geometry results (3 views). CD-L1 (mm) and F-score are computed with the official DTU evaluation code and fixed settings (Table SS1).

Scan	CD-L1 (mm) ↓	F-score ↑
Scan 8	0.55	40.5
Scan 24	0.45	45.2
Scan 37	0.51	41.8
Scan 40	0.48	43.5
Scan 55	0.46	44.0
Mean ± Std	**0.49** ± 0.04	**43.0** ± 1.9

Note. The above scans are the evaluated subset used in our experiments. We will include the complete per-scan table if additional scans are evaluated in later revisions.

S5.4 Performance Degradation Under Sparse Views

To further illustrate the robustness of GeoTrack-GS under extreme sparsity, Figure S4 visualizes the performance degradation trends as the number of input views decreases from 9 to 3.

As shown in Figure S4(a), while prior-guided methods tend to over-smooth geometry when views are scarce, our method maintains competitive perceptual quality on LLFF. More importantly, Figure S4(b) confirms that on DTU, GeoTrack-GS consistently achieves the lowest Chamfer Distance across all view settings, proving that our decoupled constraints effectively preserve accurate 3D structures even when supervision is severely limited.

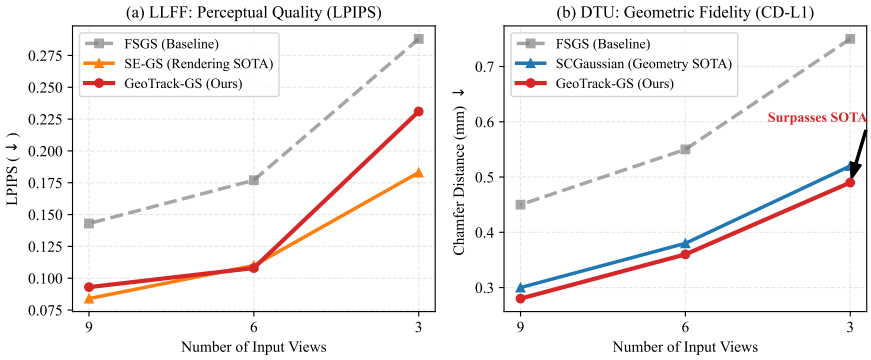


Figure S4. Performance degradation analysis under sparse views (9 → 6 → 3). (a) Perceptual Quality (LLFF): While all methods degrade as views decrease, GeoTrack-GS (red) remains competitive against strong rendering baselines and consistently improves over the depth-guided baseline (FSGS). **(b) Geometric Fidelity (DTU):** GeoTrack-GS demonstrates superior robustness. Specifically, in the extreme 3-view setting, it achieves the lowest Chamfer Distance (0.49 mm), surpassing both the baseline and strong geometry-focused methods.

S6 Additional Analysis

S6.1 Hyper-parameter Sensitivity

In Section 4.6 of the main paper, we discussed the robustness of our method to the weight of the geometric loss λ_{geo}. Table S8 confirms that performance is stable across a reasonable range (0.5× to 2.0×).

Table S8. Sensitivity analysis of λ_{geo}. Evaluated on DTU (3-view).

λ_{geo} scale	PSNR ↑	SSIM ↑	LPIPS (VGG) ↓
0.1×	18.55	0.765	0.195
0.5×	19.12	0.830	0.142
1.0× (Default)	**19.36**	**0.852**	**0.128**
2.0×	19.21	0.844	0.136
5.0×	18.85	0.815	0.155

S6.2 Computational Efficiency

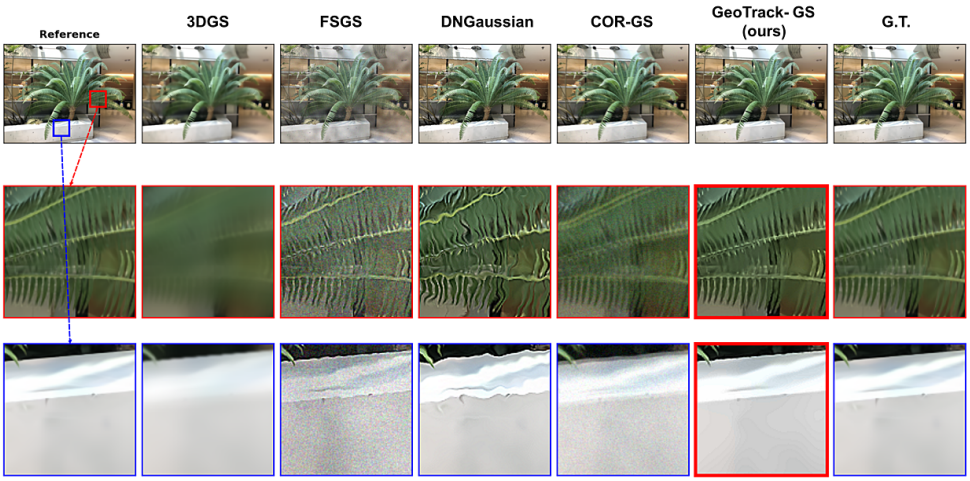
Table S9 compares the training efficiency. While GeoTrack-GS introduces additional computational cost, the overhead (approx. 38% increase) is moderate compared to CoR-GS.

Table S9. Efficiency Comparison. Training time and peak GPU memory usage on LLFF (Fern, 3-view, RTX 4090).

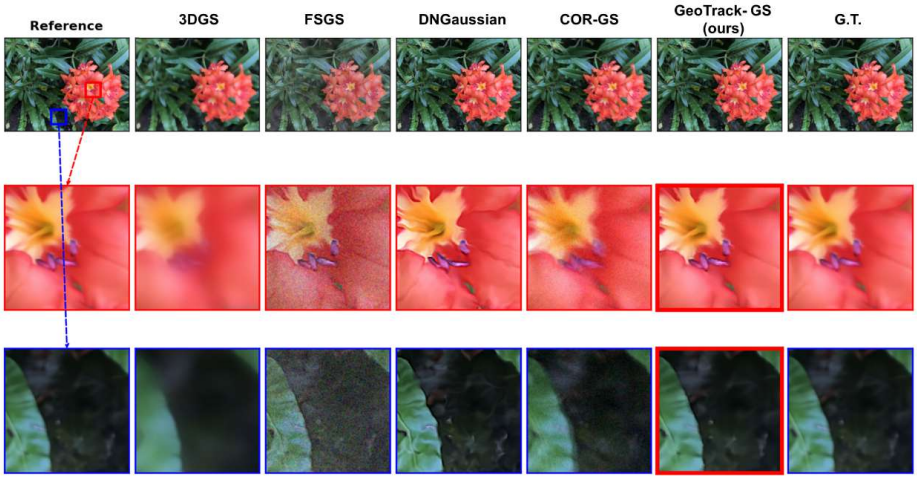
Method	Time / scene (min) ↓	Peak memory (GB) ↓
FSGS (Baseline)	18	6.5
CoR-GS	45	14.2
Ours	**25**	**7.8**

S7 Extensive Visual Comparisons

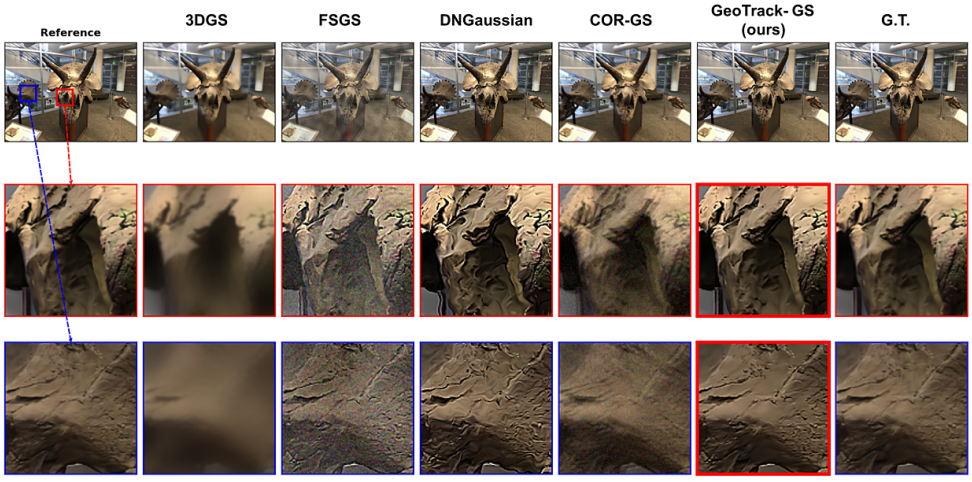
In this section, we provide a comprehensive visual assessment.



(a) Fern

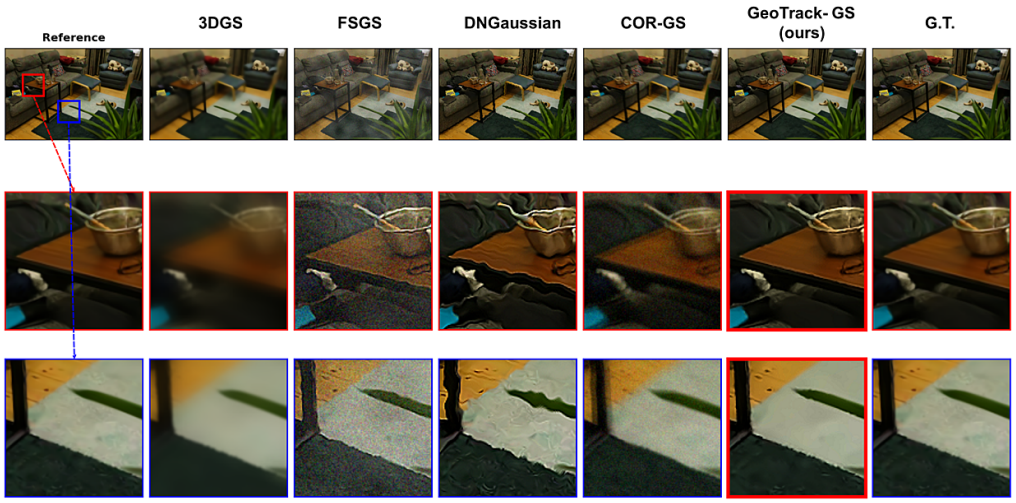


(b) Flower

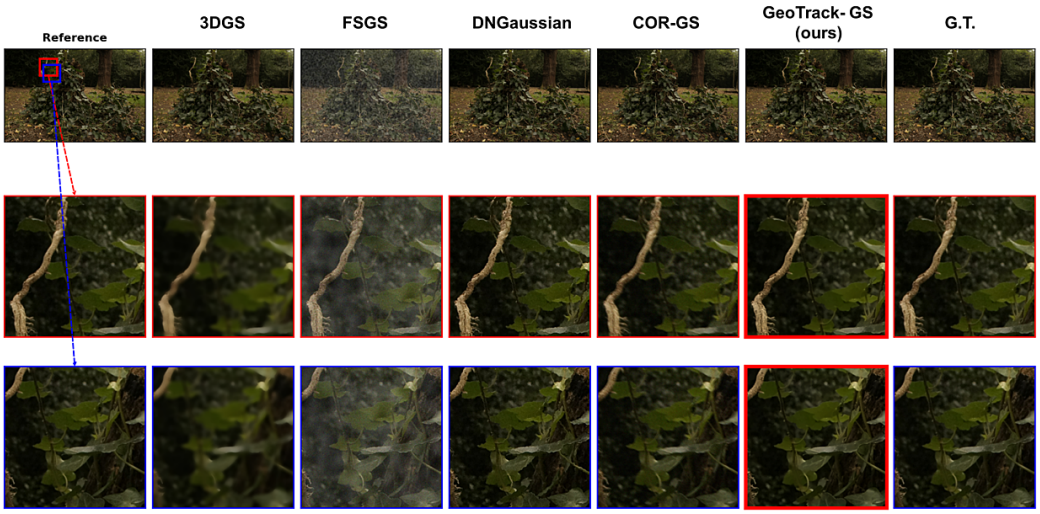


(c) Horns

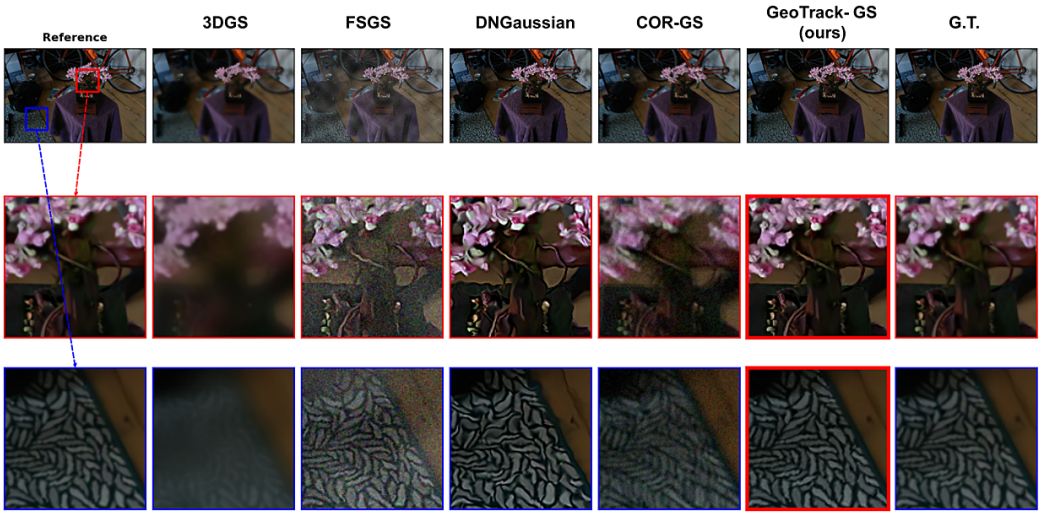
Figure S5. Qualitative comparisons on LLFF Dataset. GeoTrack-GS preserves fine high-frequency details across multiple scenes.



(a) Room



(b) Stump



(c) Bonsai

Figure S6. Qualitative comparisons on Mip-NeRF 360. GeoTrack-GS suppresses floating artifacts.

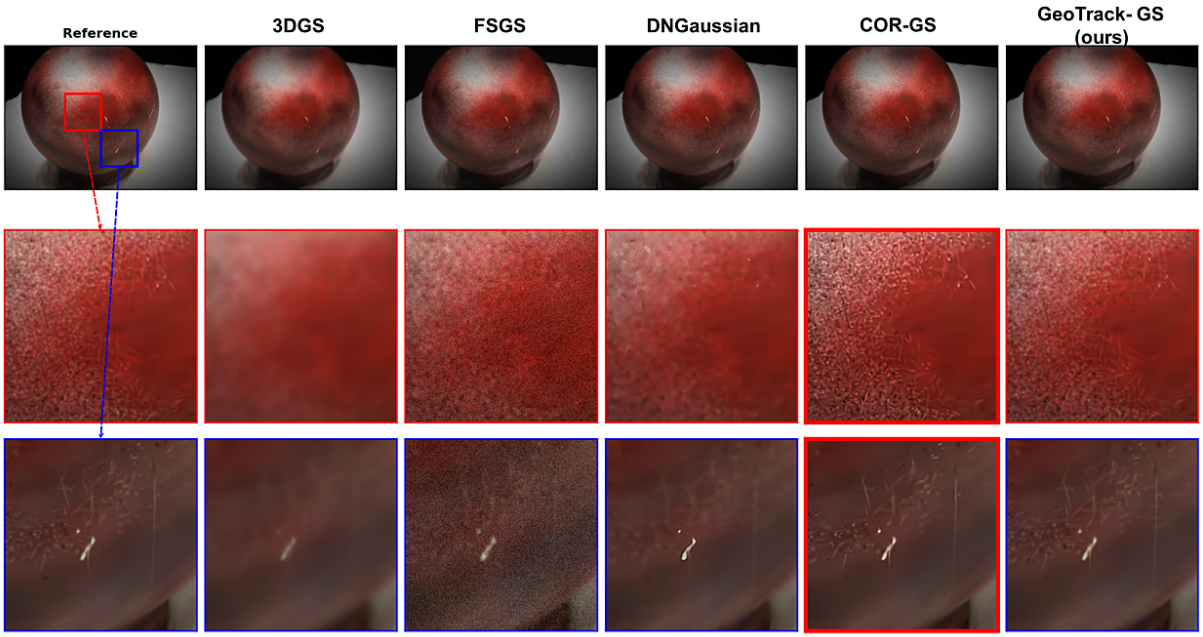


Figure S7. Qualitative comparison on DTU Scan 8. GeoTrack-GS reconstructs accurate geometry even with texture-less regions.