ReorgGS: Equivalent Distribution Reorganization for 3D Gaussian Splatting

Luchao Wang
Moore Threads
luchao.wang@mthreads.com

Kaimin Liao
Moore Threads
kaimin.liao@mthreads.com

Hua Wang
Moore Threads
wh@mthreads.com

Qian Ren
Moore Threads
renq2019@pku.edu.cn

Zhi Chen
Moore Threads
cswccz@gmail.com

Yaohua Tang
Moore Threads
tangyaohua28@gmail.com

Abstract

A converged 3D Gaussian Splatting (3DGS) model may approximate the target scene while remaining poorly parameterized for further optimization. We identify this failure mode as *parameterization degeneration*: high-opacity floaters attenuate gradients to true surfaces through alpha compositing, and redundant overlapping clusters create strongly coupled parameter blocks with nearly collinear Jacobian responses. These effects explain why continued optimization can plateau even when the model still contains removable artifacts. We propose ReorgGS, an equivalent distribution reorganization method for converged 3DGS models. ReorgGS treats the existing Gaussian set as an empirical probability field, resamples centers from it, estimates local anisotropic covariances with kNN, initializes low opacity, and continues optimization with the original 3DGS renderer and loss. Unlike opacity reset, which only rescales opacity on the old overlap graph, ReorgGS rebuilds centers, covariances, and visibility structure, thereby changing the graph itself. Our analysis shows that distributional equivalence is not optimization equivalence. The reorganized model preserves scene support while improving gradient accessibility under alpha compositing and reducing opacity-weighted overlap, thereby weakening local parameter coupling during subsequent optimization. Under the same additional optimization budget, ReorgGS improves fitting quality at a fixed Gaussian count, suppresses persistent floaters, and reduces rendering overhead from redundant overlap.

1 Introduction

3D Gaussian Splatting (3DGS) [Kerbl et al., 2023] represents scenes with anisotropic 3D Gaussians and renders them via differentiable alpha compositing, achieving strong reconstruction quality, fast training, and efficient rendering. The standard 3DGS pipeline builds this representation through local, history-dependent heuristic operations—such as gradient-based densification, cloning, splitting, and pruning, alongside periodic opacity resets. While these operations rapidly expand scene coverage, they also inevitably rigidify the occlusion order, overlap relations, and parameter-coupling structure of the Gaussian set.

This trajectory dependence induces a late-stage optimization bottleneck—one rooted not in representation capacity, but in parameter topology. In many converged models, training views are rendered reasonably well, yet high-opacity floaters persist in free space, redundant Gaussians repeatedly explain the same local residuals, and directly continuing optimization yields negligible gains. The root cause lies in the non-uniqueness of the 3DGS representation: different Gaussian sets can project to

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

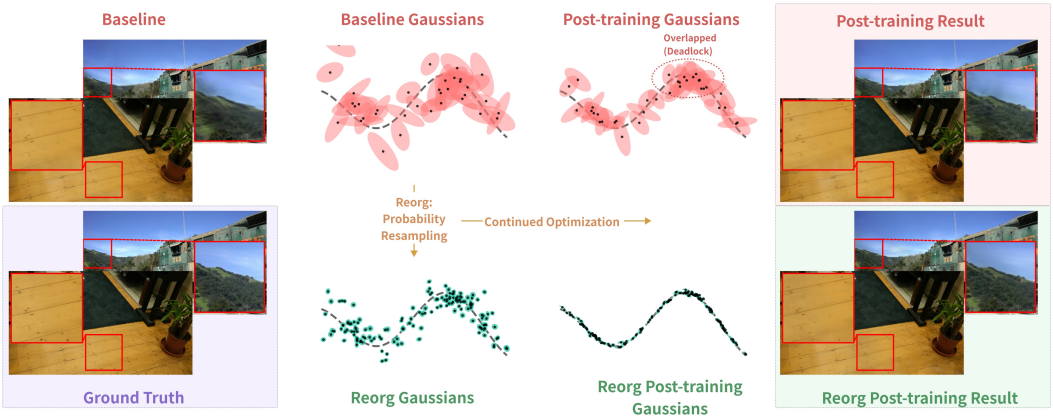

Figure 1: Reorganization unlocks the full potential of post-training. Given a suboptimal baseline (e.g., LiteGS, top-left), directly applying post-training methods (top row) fails to resolve the underlying thick and messy geometric topology. This overlapped configuration causes gradient deadlock during optimization, leaving fine details persistently blurred in the final rendering (top-right). In contrast, our ReorgGS (bottom row) first resamples the baseline's chaotic probability field into discrete, unoccluded primitives. This fundamental topological reset smooths the optimization landscape, allowing the same post-training process to explicitly align the points to the local manifold and easily recover high-fidelity details (bottom-right) that closely match the Ground Truth (bottom-left).

nearly identical images, yet possess drastically different visibility structures and optimization landscapes. Through alpha compositing, a high-opacity floater can act as an opaque barrier, attenuating gradients to true surfaces behind it. Likewise, a redundant overlapping cluster forces multiple parameter blocks to respond to the same pixel residual, causing severe collinearity. We term this state *degenerate parameterization*: the model is distributionally adequate, but optimization-degenerate. This perspective elegantly unifies three traditionally disparate symptoms: floaters correspond to ray-level gradient truncation, redundant overlap induces strong local coupling, and optimization plateauing is the inevitable consequence of these structural barriers. This leads to the central question of our work:

> If a converged 3DGS model already approximately represents a scene, can we replace it—without increasing the Gaussian budget—with a distributionally equivalent but strictly easier-to-optimize parameterization?

We propose ReorgGS to answer this question. Given a converged Gaussian set $\mathcal{G}^p$, ReorgGS interprets it as an empirical Gaussian-mixture probability field. We resample centers from this field, estimate local anisotropic covariances using k-nearest neighbors, and initialize a new, low-opacity Gaussian set $\mathcal{G}^q$. This approach fundamentally distinguishes ReorgGS from standard opacity resets. A standard reset merely applies $\alpha_i \leftarrow \alpha_0$ while preserving the original centers, covariances, and old overlap graph; it mitigates weights on a flawed topology, but cannot escape the topology itself. ReorgGS, instead, applies low opacity to newly sampled centers with geometrically re-estimated shapes, thereby explicitly dismantling and rebuilding the visibility structure that dictates occlusion and parameter coupling. Our contributions are summarized as follows:

1. **We identify parameterization degeneration as a critical late-stage bottleneck in 3DGS.** We demonstrate that persistent floaters, redundant overlap, and optimization plateaus are symptoms of an unfavorable optimization coordinate system, rather than simply a lack of capacity or training time.
2. **We introduce ReorgGS, an equivalent distribution reorganization method.** ReorgGS preserves the learned scene support while resampling centers, re-estimating anisotropic covariances, and resetting opacity. This fundamentally improves visibility structure and reduces local coupling without altering the renderer, loss function, or Gaussian budget.
3. **We analyze and validate the optimization dynamics of reorganization.** We mathematically and empirically show that distributional equivalence does not imply optimization

equivalence. ReorgGS improves gradient accessibility and reduces parameter coupling, delivering up to 0.5 dB higher training PSNR and 0.14 dB higher test PSNR over direct continuation, along with a 12.8% rendering speedup in representative settings.

2 Related Work

Representational Variants of 3DGS. 3DGS parameterizes scenes with depth-sorted anisotropic primitives [Kerbl et al., 2023]. To improve geometric fidelity, numerous variants alter the fundamental representation via neural anchors [Lu et al., 2024], planar constraints [Huang et al., 2024], geometry extraction [Yu et al., 2024b], or anti-aliasing filters [Yu et al., 2024a]. While these architectural shifts confirm the importance of primitive organization, they achieve it by modifying the underlying mathematical formulation. ReorgGS takes an orthogonal path: we demonstrate the standard 3DGS primitive is strictly sufficient. Instead of altering the representation, we reorganize a degenerate Gaussian set into a well-conditioned parameterization to unlock superior optimization without changing the renderer.

Heuristic Density Control. Standard 3DGS relies on heuristic densification and opacity resets to shape the Gaussian distribution [Kerbl et al., 2023]. Subsequent works refine these micro-level operations via advanced gradient signals, steepest-descent, localized point management, or dedicated floater-free frameworks [Cheng et al., 2024, Zhang et al., 2024, Ye et al., 2024, Yang et al., 2024, Wang et al., 2025b, Chen et al., 2025, Wang et al., 2025a]. However, these methods primarily govern *incremental growth* and remain greedy. They struggle to undo structural damage (e.g., redundant overlap, deadlocks) accumulated over long training trajectories. Rather than patching local growth heuristics, ReorgGS executes a global topological reset to dismantle unfavorable occlusion orders and parameter collinearity in the post-plateau regime.

Compact 3DGS and Probabilistic Sampling. Extensive research reduces 3DGS overhead via pruning, quantization, distillation, or entropy coding [Niedermayr et al., 2024, Fan et al., 2024, Girish et al., 2024, Lee et al., 2024, Fang and Wang, 2024, Mallick et al., 2024, Wang et al., 2024, Dai et al., 2025, Lee et al., 2025]. These methods universally treat a converged Gaussian set as a static object to be compressed. In contrast, ReorgGS treats it as a *flawed optimization coordinate system*, actively improving gradient accessibility to break fitting ceilings under restricted point budgets. Closest to our probabilistic perspective is 3DGS-MCMC [Kheradmand et al., 2024], which injects stochastic exploration into the training loop. While MCMC performs continuous micro-exploration *during* optimization, ReorgGS intervenes at the macro level: we halt optimization, resample the degenerate probability field into a clean, unoccluded state, and provide a drastically flattened loss landscape for the optimizer to resume.

3 Method

3.1 From 3DGS Parameters to a Reorganized Distribution

A 3DGS model represents a scene by a set of Gaussians $\mathcal{G} = \{g_i\}_{i=1}^N$. Each primitive has a center μ_i, covariance Σ_i, opacity α_i, and color parameters C_i, with spatial response $G_i(x) = \exp\left(-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right)$. ReorgGS starts from a model that has converged or reached a plateau. Rather than treating its primitives as fixed optimization variables, we interpret their spatial support as an unnormalized empirical Gaussian-mixture field: $P(x) \approx \sum_{i=1}^N \alpha_i \mathcal{N}(x \mid \mu_i, \Sigma_i)$.

This field is not introduced to change the rendering model. It is a compact way to separate what the current model has learned from how it is parameterized. Many Gaussian sets can induce nearly the same spatial support while having very different opacity patterns, overlap relations, and visibility orders. Standard densification, pruning, and opacity reset gradually construct one such parameterization, but the final arrangement need not be a good coordinate system for further optimization. The importance of this distinction comes from alpha compositing. For a ray or pixel r, 3DGS renders $C(r) = \sum_{i=1}^N T_i(r)a_i(r)C_i(r), T_i(r) = \prod_{j<i}(1-a_j(r))$, where $a_i(r)$ is the effective projected opacity and $T_i(r)$ is the accumulated transmittance before the i-th Gaussian. A high-opacity floater or a redundant foreground cluster can make $T_i(r)$ small, starving the true surface behind it of gradient. Thus, continued optimization may fail not because the scene support is absent, but because the existing support is organized with an unfavorable visibility and overlap structure.

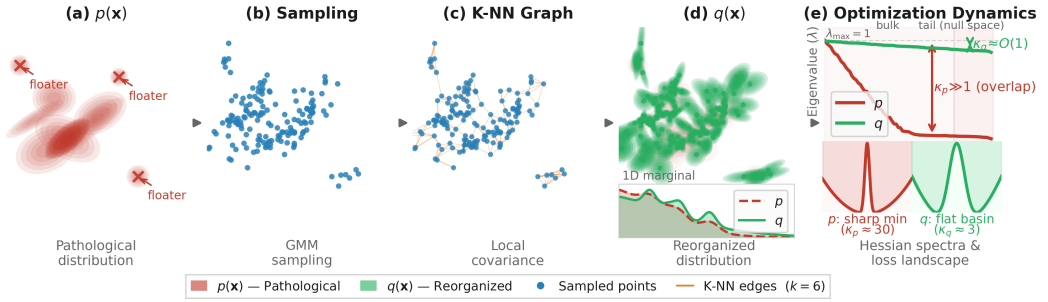

Figure 2: Method pipeline and optimization dynamics of ReorgGS. (a) Vanilla 3DGS often degenerates into a pathological distribution $p(x)$ with severe spatial overlap. (b) Our α-driven equivalent resampling discards the spatial volume term, naturally filtering out large-scale floating artifacts. (c, d) Via kNN local manifold alignment, we reconstruct the covariance to form a healthy continuous field $q(x)$ with bounded overlap. (e) Schematic illustration of Hessian Spectrum Analysis: In the vanilla distribution p, severe physical overlap causes heavy occlusion, forcing a massive amount of eigenvalues to collapse from the bulk into the tail (null space), leading to a catastrophic condition number ($\kappa_p \gg 1$) and sharp deadlocks. Conversely, ReorgGS bounded overlap effectively eliminates the null space, keeping the tail eigenvalues stable ($\kappa_q \approx O(1)$). This preconditions the loss landscape, smoothly transforming sharp deadlocks into optimizable flat minima.

Direct continuation keeps this structure and updates the old variables in place, $\mathcal{G}_i \leftarrow \mathcal{G}_i - \eta \nabla_{\mathcal{G}_i} \mathcal{L}$. ReorgGS instead constructs a new Gaussian set $\mathcal{G}' = \{g'_j\}_{j=1}^M$, with parameters $\{\mu'_j, \Sigma'_j, \alpha'_j, C'_j\}_{j=1}^M$, whose induced field $q(x) = \sum_{j=1}^M \alpha'_j \mathcal{N}(x \mid \mu'_j, \Sigma'_j)$ approximately preserves the support of $P(x)$ while changing the centers, covariances, opacity initialization, and overlap graph. The goal is therefore not only distributional similarity, $q \approx P$, but optimization improvement: the reorganized set should expose gradients more evenly and reduce local parameter coupling during the subsequent 3DGS optimization.

3.2 Algorithm

Given a 3DGS model $\{\mu_i, \Sigma_i, \alpha_i, C_i\}_{i=1}^N$ that has reached the middle-to-late stage of training, converged, or entered a plateau, ReorgGS constructs a new Gaussian set as follows.

Step 1. GMM modeling. We view the current Gaussian set as an empirical Gaussian-mixture field and define its normalized form as $\hat{p}(x) = \frac{P(x)}{Z}$, where $Z = \int P(x)dx$. In implementation, the continuous integral Z does not need to be computed explicitly. Sampling only requires the discrete mixture weights of the components.

Step 2. Categorical sampling. We first sample a latent variable $z_j \in \{1, \ldots, N\}$ from a categorical distribution $P(z_j = k) = \frac{\alpha_k}{\sum_{l=1}^N \alpha_l}$. Thus, Gaussian components with higher opacity are more likely to be selected. We use only α_k as the categorical weight and do not multiply it by the Gaussian volume. The reason is that, in degenerate 3DGS optimization, abnormally large Gaussians often correspond to blurred regions or artifacts. Volume-weighted sampling could further amplify such large-scale abnormal structures. We therefore ignore volume in the categorical probability to reduce this sampling bias.

Step 3. Gaussian coordinate sampling. After choosing category $z_j = k$, we sample a new center from the corresponding 3D Gaussian $x_j \sim \mathcal{N}(\mu_k, \Sigma_k)$. In practice, we use the reparameterized form $x_j = \mu_k + L_k \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and L_k is the Cholesky factor of Σ_k. Since 3DGS parameterizes $\Sigma_k = R_k S_k S_k^T R_k^T$, one can also directly use $R_k S_k$ as the sampling transform. Through this procedure, ReorgGS regenerates centers from the continuous spatial distribution induced by the old model, rather than simply copying old Gaussian positions.

Step 4. Anisotropic initialization. Sampled centers alone are not sufficient for an effective 3DGS initialization. To make the new Gaussians quickly adapt to local geometry, we perform a kNN query for each sampled point x_j within the new point set $\{x_j\}_{j=1}^{M}$, obtaining its k nearest neighbors $\{x_{j_1}, \ldots, x_{j_k}\}$. We then estimate the new covariance using the local second moment: $\Sigma'_j = \frac{1}{k}\sum_{l=1}^{k}(x_{j_l} - x_j)(x_{j_l} - x_j)^T$. The kNN covariance provides a natural anisotropic initialization.

Step 5. Constructing the reorganized distribution $q(x)$. Using sampled point x_j as the center and local covariance Σ'_j as the shape, we construct the new Gaussian set. We set $\mu'_j = x_j, \alpha'_j = \alpha_0$, where α_0 is a small opacity value (default 0.1). The spatial scale of each reorganized Gaussian is naturally determined by Σ'_j, with support-scale calibration. Subsequent optimization continues to adjust scale, rotation, and opacity to fit the image reconstruction objective. Only the zeroth-order spherical-harmonic coefficient (DC) is inherited from the sampled source component z_j: $C'_{j,0} = C_{z_j,0}$. All higher-order SH coefficients start at zero and are relearned during continuation.

Step 6. Recovering scale and rotation. To write Σ'_j back into the standard 3DGS parameterization, we perform eigendecomposition $\Sigma'_j = U_j \Lambda_j U_j^T$, set $R'_j = U_j$ and $S'_j = \sqrt{\Lambda_j}$, so that $\Sigma'_j = R'_j S'_j (S'_j)^T (R'_j)^T$. After obtaining $(\mu'_j, S'_j, R'_j, \alpha'_j, C'_j)$, we continue training with the original 3DGS renderer, loss function, and optimization strategy. Since the Gaussian set has been rebuilt, the optimizer state is reinitialized for the new parameters.

3.3 Optimization Properties of the Reorganized Distribution

ReorgGS is not only a distribution approximation. Preserving the scene support, i.e., making $q(x)$ close to $P(x)$, does not imply that the two Gaussian sets have the same visibility, gradients, or local curvature. The benefit of ReorgGS comes from reorganizing four coupled structures: opacity-density allocation, local overlap, alpha-compositing transmittance, and covariance alignment. Appendix A provides the detailed analysis.

1. From opacity values to spatial density. Let $p(x)$ be the normalized spatial distribution of the original model, and let $q(x)$ be the mixture induced by resampled centers and kNN covariances. Appendix A.1 analyzes two complementary regimes. Under standard kNN density-estimation assumptions, the asymptotic analysis gives $\mathbb{E}[q(x)] \to p(x), \mathrm{Var}[q(x)] = O(1/k)$. The finite-budget analysis quantifies sampling accuracy and local smoothing for the main setting $M = N$. With constant low opacity α_0 (set to 0.1, see Sec. 3.2), the reorganized occupancy field satisfies $\mathbb{E}[\hat{F}(x)] \approx M\alpha_0 p(x)$. Thus, ReorgGS transfers scene occupancy from large per-primitive opacity to spatial sample density: dense regions receive more primitives, sparse regions fewer, while each primitive starts with low opacity. This preserves support while reducing absolute ray occlusion.

2. Spatial demixing and bounded overlap. Standard continued optimization keeps the old overlap graph created by densification. ReorgGS instead sets covariance scales from kNN neighborhoods. If the local point density is $\rho(x) = Mp(x)$, the effective number of local overlaps satisfies (Appendix A.2): $K_q(x) \approx \rho(x) \cdot \frac{k}{\rho(x)} = O(k)$. Hence local coverage is controlled by the neighborhood size rather than by the number of redundant clones in the original cluster.

3. Gradient accessibility. Under alpha compositing, a primitive behind foreground components receives gradients scaled by $T_i = \prod_{j<i}(1-\alpha_j)$. High-opacity floaters or many redundant foreground primitives can drive $T_i^p \to 0$, starving true surfaces of gradient. ReorgGS initializes low opacity and bounds foreground overlap; if $m_q = O(k)$, then $T_i^q \geq (1-\alpha_0)^{m_q} \approx 1 - m_q\alpha_0 > 0$. The improvement therefore comes from a new visibility structure, not from merely lowering opacity on the old one (Appendix A.3).

4. Local manifold alignment and improved conditioning. Redundant Gaussians with inconsistent scales and orientations can explain the same residual with nearly collinear Jacobian responses, increasing off-diagonal coupling in the Gauss–Newton Hessian or Fisher Information Matrix. kNN covariance initialization aligns nearby primitives with the local tangent-normal structure. If $\Sigma_i \approx \Sigma_{\mathrm{manifold}}$, then $H_{\mathrm{local}} \propto \sum_i \Sigma_i^{-1} \approx K U \Lambda^{-1} U^T$, where U is the local eigenbasis. ReorgGS effectively eliminates the null space and transforms sharp deadlocks into optimizable flat minima (Appendix A.4).

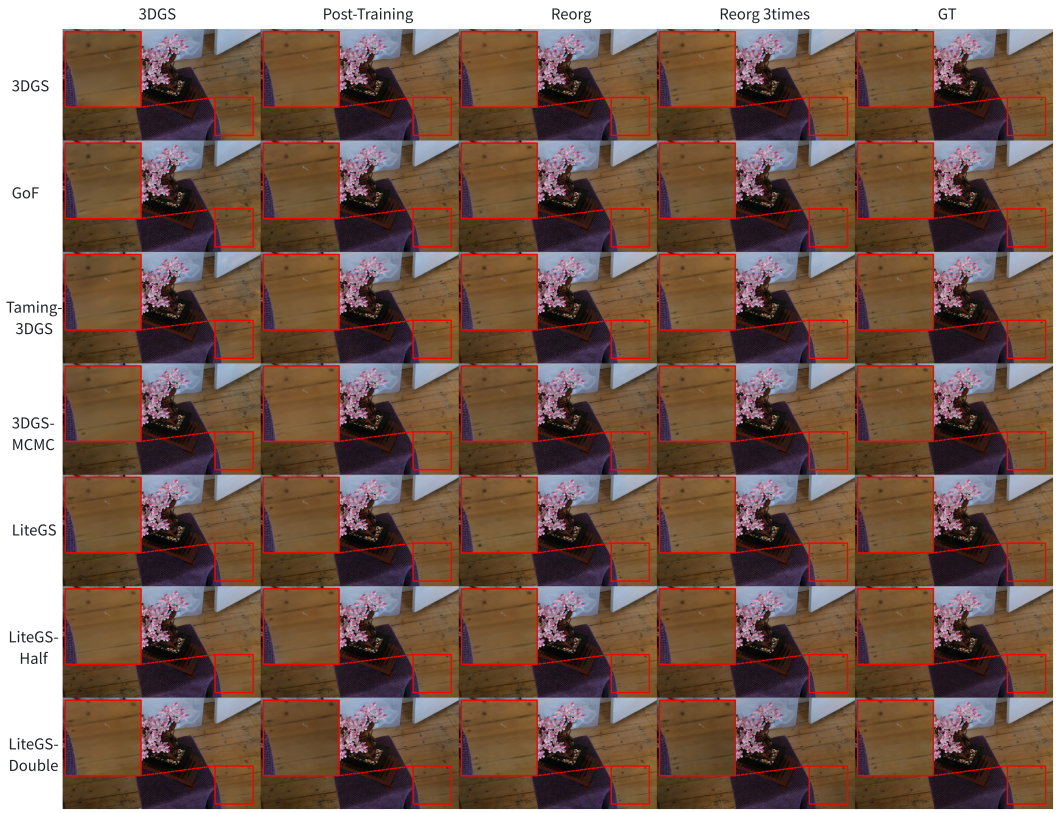

Figure 3: Plug-and-play capability and qualitative comparisons across diverse 3DGS baselines. We evaluate the universality of our Reorg mechanism on various initializations (rows), including standard 3DGS, GoF, Taming-3DGS, and LiteGS variants. Columns from left to right: Original baseline, standard Post-Training, our Reorg (single pass), Reorg applied iteratively (Reorg $\times 3$), and Ground Truth (GT). As highlighted in the red insets (wood grain textures), standard post-training struggles to resolve the blurry artifacts caused by suboptimal topology. In contrast, our Reorg consistently unlocks sharp, high-fidelity fine details across all tested baselines. Furthermore, applying Reorg iteratively further refines the geometry, achieving visual quality highly consistent with the GT.

4 Experiments

4.1 Experimental Setup

Our experiments examine whether ReorgGS can convert a converged 3DGS model into an equivalent parameterization that is easier to optimize, without increasing the number of Gaussians. Each backbone first completes its original training schedule. Here, convergence refers to completion of that schedule. Direct continuation and ReorgGS both use LiteGS for a further 30k iterations, with restarted optimizers and the same densification/pruning and opacity-reset schedules. The dedicated ablation disables densification and opacity reset. We then compare three settings under the same additional optimization budget:

- **Baseline**: the converged original model, with no further modification.
- **Direct continuation**: continued optimization of the original Gaussian set.
- **ReorgGS**: equivalent distribution reorganization followed by continued optimization.

Unless otherwise stated, ReorgGS sets $M = N$ at reorganization. Both continuation branches share scene-wise target budgets, although subsequent densification/pruning may change actual counts. The experiments therefore compare parameterization structure, rather than model capacity or training budget.

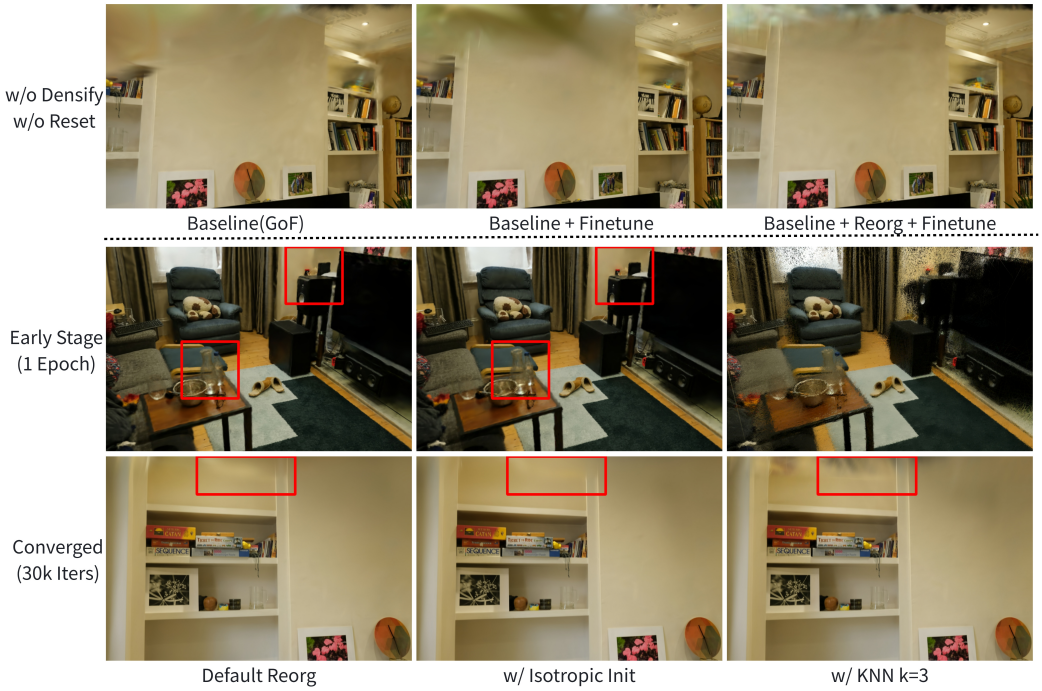

Figure 4: Ablation studies of ReorgGS. Top (Effectiveness of Reorg): We demonstrate the necessity of topological reorganization. We continue training the GoF baseline for 30k iterations without densification or opacity reset. Under this fixed topology, naive finetuning (middle) fails to recover sharp details (e.g., bookshelf) because the Gaussians are trapped in a deadlocked, suboptimal distribution. In contrast, applying our Reorg prior to the same finetuning (right) resolves the geometric occlusion, enabling pure gradient descent to successfully restore high-frequency structures. Middle & Bottom (Reorg Strategies): Ablation on Reorg specific designs across different stages (1 epoch vs. 30k iters). Discarding the covariance prior via Isotropic Initialization ("Iso-init") struggles to capture complex geometry early on (glass, middle row) and leaves splotchy noise on flat surfaces (wall, bottom row). Reducing the kNN neighborhood size to $k = 3$ leads to slower structural recovery. Our default Reorg strategy maintains optimal convergence and artifact-free rendering.

Backbones. We evaluate ReorgGS on vanilla 3DGS [Kerbl et al., 2023], GOF [Yu et al., 2024b], Taming-3DGS [Mallick et al., 2024], 3DGS-MCMC [Kheradmand et al., 2024], and LiteGS [Liao et al., 2025]. These methods use different training mechanisms and Gaussian organizations, allowing us to test whether ReorgGS improves continued optimization across backbones.

Datasets. We evaluate seven scenes from Mip-NeRF 360 [Barron et al., 2022], two from Tanks&Temples [Knapitsch et al., 2017], and two from Deep Blending [Hedman et al., 2018], using PSNR, SSIM, and LPIPS [Zhang et al., 2018].

Gaussian-budget control. To control model capacity, we first train standard 3DGS to convergence and record the target point count. Other backbones reuse this scene-wise budget unless specified otherwise in point-budget ablations. We additionally consider LiteGS-half ($\times 0.5$ budget) and LiteGS-double ($\times 2$ budget) to test robustness. Further analyses on experimental setups, statistical variance, and computational costs are deferred to Appendix B.

4.2 Main Results: Equal-budget Comparison across Backbones

Table 1 compares the baseline, direct continuation, and ReorgGS under the same additional optimization budget. ReorgGS improves training PSNR for every backbone and improves test PSNR in most settings, with gains over direct continuation of 0.5 dB on training views and 0.14 dB on test views. This result shows that the plateau of a converged model is not caused only by insufficient it-

erations: under the same capacity and the same additional budget, reorganizing the parameterization reaches optimization regions that direct continuation cannot reliably access.

Table 1: Equal-budget comparison across backbones. Each cell reports train / test. ReorgGS uses the input count at reorganization, and both continuation branches share scene-wise target budgets.

	PSNR			SSIM			LPIPS		
Backbone	Baseline	Direct	ReorgGS	Baseline	Direct	ReorgGS	Baseline	Direct	ReorgGS
3DGS	31.37 / 28.26	32.61 / 28.72	33.16 / 28.89	0.918 / 0.872	0.936 / 0.880	0.940 / 0.883	0.169 / 0.198	0.137 / 0.174	0.130 / 0.169
GOF	30.65 / 27.99	32.76 / 28.80	33.21 / 28.93	0.916 / 0.873	0.940 / 0.882	0.941 / 0.884	0.167 / 0.194	0.131 / 0.170	0.130 / 0.169
Taming-3DGS	31.04 / 28.44	32.72 / 28.84	33.38 / 29.08	0.911 / 0.874	0.937 / 0.881	0.942 / 0.886	0.178 / 0.194	0.135 / 0.171	0.129 / 0.166
3DGS-MCMC	31.69 / 28.71	33.09 / 28.99	33.63 / 29.06	0.928 / 0.886	0.944 / 0.883	0.946 / 0.885	0.151 / 0.177	0.125 / 0.168	0.123 / 0.166
LiteGS	32.91 / 29.09	33.15 / 28.95	33.67 / 29.03	0.938 / 0.888	0.945 / 0.884	0.947 / 0.885	0.135 / 0.167	0.123 / 0.165	0.121 / 0.165
LiteGS-half	32.12 / 28.87	32.32 / 28.86	32.77 / 28.96	0.925 / 0.884	0.932 / 0.882	0.934 / 0.883	0.156 / 0.180	0.144 / 0.174	0.144 / 0.176
LiteGS-double	33.63 / 29.07	33.91 / 28.95	34.47 / 28.97	0.948 / 0.888	0.954 / 0.882	0.956 / 0.883	0.117 / 0.160	0.105 / 0.160	0.103 / 0.159

The training-view gains are consistently larger than the test-view gains, which is meaningful for scene-specific 3DGS because the training views are the observations used to reconstruct each scene. Under a fixed Gaussian count and the same additional optimization budget, better training-view fitting indicates that ReorgGS uses capacity previously wasted by occlusion and coupling, while test-view metrics verify that the improvement does not collapse into view-specific artifacts.

Fig. 3 qualitatively supports the same trend: ReorgGS reduces topology-induced blur across backbones, and iterative reorganization further refines fine details. For more comprehensive qualitative and quantitative results across different datasets, please refer to Appendix C.

The reorganized spatial structure also reduces redundant alpha compositing; in representative settings, ReorgGS provides up to 12.8% rendering speedup.

Prevalence of improvements. Across all seven configurations, including LiteGS-half and LiteGS-double, and 11 scenes, test PSNR improves in 59 of 77 paired comparisons, with a median gain of 0.1002 dB. Averaging the seven configurations within each scene gives improved test PSNR in 8 of 11 scenes and improved test SSIM and LPIPS in all 11 scenes. These paired and scene-level summaries show the breadth of the gains (Appendix B.4).

Sensitivity to initialization quality. We also train from 100k randomly initialized Gaussians for 30k iterations to obtain a shared checkpoint, then continue Direct and ReorgGS for a further 30k iterations each. ReorgGS increases mean test PSNR from 27.3075 to 27.7009 dB, improves SSIM from 0.8324 to 0.8443, and reduces LPIPS from 0.2133 to 0.2002. This setting demonstrates the benefit of reorganization for a weaker initial representation (Appendix B.5).

4.3 Robustness under Different Gaussian Budgets

The LiteGS-half, LiteGS, and LiteGS-double rows in Table 1 further vary the Gaussian budget while keeping the baseline, direct continuation, and ReorgGS under the same target-budget protocol within each row. ReorgGS remains beneficial under both reduced and increased budgets. The improvement is larger under the half budget, where each Gaussian carries more representational pressure and a degenerate overlap pattern wastes a larger fraction of the available capacity. In this half-budget setting, Direct continuation slightly outperforms ReorgGS on LPIPS, likely due to the perceptual penalty of low-opacity initialization under severe capacity constraints, though ReorgGS still dominates in PSNR and SSIM. Under the double budget, the model has more redundancy and the test-view gain becomes smaller, but ReorgGS still improves training PSNR. This suggests that parameterization degeneration is not unique to compact models; it also appears when the Gaussian budget is larger, although its effect on novel-view metrics may be partly absorbed by the extra capacity.

4.4 Cascaded Reorganization

We further evaluate repeated Reorg operations. As shown in Table 2, Reorg x2, x3, and x4 continue to improve training PSNR, but the marginal gain decreases. This trend agrees with the degenerate-parameterization view: the first Reorg usually removes the strongest occlusion barriers and redundant coupling, while later reorganizations face fewer remaining structural obstacles.

To rule out the effect of total training budget, we compare two equal-budget trajectories. Direct continuation optimizes the original parameterization for the same total number of iterations, while

Reorg $\times N$ allocates the same additional optimization budget after each reorganization. Table 2 reports PSNR only and lists each backbone and point budget separately. Reorg remains better overall under the same total budget, showing that the cascaded gain comes from repeatedly replacing degenerate parameterizations rather than simply using more optimization steps. Training PSNR gains saturate as the number of reorganizations increases, while test-view improvements remain small but generally positive. This is consistent with our analysis: ReorgGS first improves optimization accessibility and local parameter decoupling, which is most directly reflected in training-view fitting. Test-view improvement then depends on whether the better fitting becomes a more consistent geometric and appearance representation.

Table 2: Cascaded reorganization under equal total optimization budget. Each cell reports train / test PSNR. For ReorgGS columns, darker color indicates a larger number of reorganization stages.

	$2\times$ iters		$3\times$ iters		$4\times$ iters	
Backbone	Direct	ReorgGS	Direct	ReorgGS	Direct	ReorgGS
3DGS	33.07 / 28.77	33.46 / 28.90	33.26 / 28.80	33.57 / 28.93	33.42 / 28.79	33.62 / 28.93
GOF	33.10 / 28.89	33.54 / 28.93	33.29 / 28.92	33.59 / 28.97	33.41 / 28.93	33.66 / 28.97
Taming-3DGS	33.25 / 28.89	33.71 / 29.04	33.53 / 28.89	33.79 / 29.04	33.60 / 28.88	33.78 / 29.04
3DGS-MCMC	33.33 / 29.01	33.74 / 29.06	33.50 / 29.02	33.76 / 29.06	33.61 / 28.97	33.76 / 29.05
LiteGS	33.35 / 28.91	33.76 / 29.04	33.52 / 28.89	33.80 / 28.97	33.66 / 28.89	33.78 / 29.00
LiteGS-half	32.52 / 28.89	32.82 / 28.94	32.68 / 28.92	32.81 / 28.97	32.76 / 28.90	32.80 / 29.00
LiteGS-double	34.17 / 28.93	34.63 / 28.93	34.37 / 28.86	34.69 / 28.90	34.50 / 28.83	34.68 / 28.89

4.5 Necessity of Reorganization (w/o Densification & Reset)

To demonstrate that persistent floaters and blurry details are symptoms of topological deadlocks rather than mere lack of training, we evaluate finetuning under a strict constraint: disabling both heuristic densification and opacity reset.

As shown in the top row of Fig. 4, directly finetuning the baseline under this fixed topology (Baseline + Finetune) fails to resolve the blurry artifacts on the bookshelf. The existing high-opacity floaters maintain their occlusion advantage, blocking gradients to the true surfaces and trapping the optimization in a local minimum. In contrast, applying ReorgGS before finetuning explicitly breaks these old visibility structures. By resampling the geometry with low opacity, ReorgGS smoothly eliminates gradient deadlocks, enabling pure gradient descent to successfully restore sharp, high-fidelity structures without relying on heuristic cleanup.

4.6 Ablation on Reorg Configurations

We further ablate the internal design choices of Reorg across different training stages, as visualized in the bottom two rows of Fig. 4. All variants use the same sampling distribution, differing only in the initialized state.

Isotropic vs. Anisotropic Initialization. We compare our default kNN-based anisotropic initialization against an isotropic variant (which serves as a minimalist implementation of Reorg without geometric priors). As observed in the early stage (1 epoch, left vs. middle), the isotropic variant struggles to capture structures like the glass bottle and the wall corner. While it eventually converges to a comparable visual quality at 30k iterations without statistically significant degradation, our default anisotropic initialization effectively aligns primitives to local manifolds much earlier, leading to fundamentally faster and more stable structural recovery.

kNN Neighborhood Size (k). We set $k = 20$ as our default configuration to provide robust local geometric estimation. Shrinking the neighborhood size aggressively (e.g., "w/ kNN $k = 3$") makes the covariance estimation highly susceptible to local noise. As shown in the right column of Fig. 4, this introduces severe needle-like noise artifacts during the early optimization stage. Although most of these spikes are eliminated upon convergence, noticeable artifacts still persist in poorly observed regions (e.g., the upper wall edge at 30k iterations). The default $k = 20$ smoothly avoids these instabilities.

5 Limitations and Broader Impact

ReorgGS is a reorganization method rather than a geometry completion method. It assumes that the converged input model already contains reasonable scene support, and therefore cannot recover structures that are completely missing or never observed in the original Gaussian set. Its effectiveness may also decrease when the input distribution is dominated by severe reconstruction errors, since resampling from such a distribution can preserve part of the wrong support. In addition, ReorgGS requires additional optimization after reorganization. ReorgGS can reduce persistent floaters and redundant overlap, which may improve the reliability and efficiency of scene-specific 3D reconstruction systems. At the same time, higher-quality reconstruction can increase privacy risks when applied to people, private spaces, or restricted environments. As with other high-fidelity 3D reconstruction methods, ReorgGS should be used only with appropriate consent, data-use permission, and care when releasing reconstructed scenes or trained Gaussian models.

6 Conclusion

We identify parameterization degeneration as a late-stage bottleneck in 3DGS: a converged model may represent the scene reasonably well while retaining floaters, redundant overlap, and poor gradient accessibility. ReorgGS addresses this by rebuilding the converged Gaussian set as an equivalent low-opacity, locally aligned parameterization, without changing the renderer or increasing capacity. Across backbones and Gaussian budgets, this reorganization improves fixed-budget optimization, suppresses persistent floaters, and reduces rendering redundancy, suggesting that the organization of Gaussian density, opacity, covariance, and visibility is a central factor in late-stage 3DGS performance.

Acknowledgements and Disclosures

This work was supported by Moore Threads.

References

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

Youyu Chen, Junjun Jiang, Kui Jiang, Xiao Tang, Zhihao Li, Xianming Liu, and Yinyu Nie. Dashgaussian: Optimizing 3d gaussian splatting in 200 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11146–11155, 2025.

Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning*, 2024.

Zhenqi Dai, Ting Liu, and Yanning Zhang. Efficient decoupled feature 3d gaussian splatting via hierarchical compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11156–11166, 2025.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37:140138–140158, 2024.

Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European conference on computer vision*, pages 165–181. Springer, 2024.

Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, pages 54–71. Springer, 2024.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.

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

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. *Advances in Neural Information Processing Systems*, 37:80965–80986, 2024.

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.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21719–21728, 2024.

Soonbin Lee, Fangwen Shu, Yago Sanchez, Thomas Schierl, and Cornelius Hellge. Compression of 3d gaussian splatting with optimized feature planes and standard video codecs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25496–25505, 2025.

Kaimin Liao, Hua Wang, Zhi Chen, Luchao Wang, and Yaohua Tang. Litegs: A high-performance framework to train 3dgs in subminutes via system and algorithm codesign. *arXiv preprint arXiv:2503.01199*, 2025.

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

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10349–10358, 2024.

Luchao Wang, Qian Ren, Kaimin Liao, Hua Wang, Zhi Chen, and Yaohua Tang. Stablegs: A floater-free framework for 3d gaussian splatting. *arXiv preprint arXiv:2503.18458*, 2025a.

Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, et al. Steepest descent density control for compact 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26663–26672, 2025b.

Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex C Kot, and Bihan Wen. Contextgs: Compact 3d gaussian splatting with anchor level context model. *Advances in neural information processing systems*, 37:51532–51551, 2024.

Haosen Yang, Chenhao Zhang, Wenqing Wang, Marco Volino, Adrian Hilton, Li Zhang, and Xiatian Zhu. Gaussian splatting with localized points management. *arXiv preprint arXiv:2406.04251*, 2024.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1053–1061, 2024.

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

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics (TOG)*, 43(6):1–13, 2024b.

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

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 326–342. Springer, 2024.

A Theoretical proofs

A.1 Distributional Equivalence and Constant-opacity Occupancy

Let the converged 3DGS model induce the normalized scene occupancy distribution

$$p(x) = \frac{1}{Z}\sum_{i=1}^{N}\alpha_i\mathcal{N}(x \mid \mu_i, \Sigma_i), \qquad Z = \int \sum_{i=1}^{N}\alpha_i\mathcal{N}(x \mid \mu_i, \Sigma_i)dx.$$

ReorgGS samples new centers from this distribution. Equivalently, it first selects an original Gaussian according to

$$P(z_j = k) = \frac{\alpha_k}{\sum_{l=1}^{N}\alpha_l},$$

and then draws a coordinate from

$$x_j \sim \mathcal{N}(\mu_k, \Sigma_k).$$

Given the resampled centers, ReorgGS estimates a local covariance Σ_j^q from the kNN neighborhood and constructs

$$q_M(x) = \frac{1}{M}\sum_{j=1}^{M}\mathcal{N}(x \mid x_j, \Sigma_j^q).$$

Proposition (distributional consistency). Assume that $p(x)$ is twice continuously differentiable in a local neighborhood, and that the kNN bandwidth satisfies $k \to \infty$ and $k/M \to 0$. Then, as $M \to \infty$,

$$\mathbb{E}[q_M(x)] \to p(x), \qquad \mathrm{Var}[q_M(x)] = O(1/k).$$

Proof. For samples drawn from p, the local 3D volume that contains k nearest neighbors satisfies

$$V_k(x) \approx \frac{k}{Mp(x)}.$$

The corresponding local covariance scale is therefore

$$\|\Sigma^q(x)\| = O\left((k/M)^{2/3}\right).$$

Treating the kNN local estimator as a kernel density estimator with bandwidth $\Sigma^q(x)$, applying local Gaussian smoothing to p, and using a second-order Taylor expansion gives

$$\mathbb{E}[q_M(x)] = p(x) + \frac{1}{2}\mathrm{Tr}(\Sigma^q(x)\nabla^2 p(x)) + o(\|\Sigma^q(x)\|).$$

Since $k/M \to 0$, the bandwidth vanishes and the bias term goes to zero. Hence $\mathbb{E}[q_M(x)] \to p(x)$. For the variance of the local kernel estimator,

$$\mathrm{Var}[q_M(x)] \approx \frac{1}{M}\int \mathcal{N}(x \mid z, \Sigma^q(z))^2 p(z)dz.$$

Using the squared Gaussian integral, the leading scale is

$$\mathrm{Var}[q_M(x)] \propto \frac{p(x)}{M|\Sigma^q(x)|^{1/2}}.$$

For a 3D kNN bandwidth, $|\Sigma^q(x)|^{1/2} \propto k/M$. Thus,

$$\mathrm{Var}[q_M(x)] \propto \frac{1}{k}.$$

The variance therefore vanishes as $k \to \infty$. This completes the proof.

This result shows that resampling and kNN covariance estimation preserve the original scene support in an asymptotic sense.

Proposition (finite-budget approximation). Let $x_1, \ldots, x_M$ be independent samples from p. For any measurable region A with probability mass $p_A = \int_A p(x)dx$, the center count $N_A = \sum_j \mathbf{1}\{x_j \in A\}$ satisfies

$$\mathbb{E}\left[\frac{N_A}{M}\right] = p_A, \qquad \mathrm{Var}\left[\frac{N_A}{M}\right] = \frac{p_A(1-p_A)}{M}, \qquad \Pr(N_A = 0) = (1-p_A)^M \le e^{-Mp_A}.$$

For any fixed L-Lipschitz function $f : \mathbb{R}^3 \to [0,1]$ and $0 < \delta < 1$, the normalized mixture q_M defined above satisfies, with probability at least $1-\delta$,

$$\left| \int f(x) q_M(x) dx - \int f(x) p(x) dx \right| \le \frac{1}{2\sqrt{M\delta}} + \frac{L}{M} \sum_{j=1}^{M} \sqrt{\mathrm{tr}(\Sigma_j^q)}.$$

Proof. Independence gives $N_A \sim \mathrm{Binomial}(M, p_A)$, yielding its expectation, variance, and empty-region probability. The last inequality follows from $1 - p_A \le e^{-p_A}$. For the mixture bound, let $F_M = M^{-1} \sum_j f(x_j)$. Since $0 \le f \le 1$, Chebyshev's inequality gives

$$\Pr\left(\left| F_M - \int f(x) p(x) dx \right| > \frac{1}{2\sqrt{M\delta}} \right) \le \delta.$$

Conditioned on the sampled centers and covariances, let $\varepsilon_j \sim \mathcal{N}(0, \Sigma_j^q)$. The Lipschitz property and Cauchy–Schwarz inequality give

$$\left| \int f(x) q_M(x) dx - F_M \right| \le \frac{L}{M} \sum_j \mathbb{E}\|\varepsilon_j\|_2 \le \frac{L}{M} \sum_j \sqrt{\mathrm{tr}(\Sigma_j^q)}.$$

Combining the two bounds proves the result.

These estimates characterize finite-budget reorganization through center-sampling concentration and covariance-controlled local smoothing. With $M = N$, regions of larger learned probability mass receive proportionally more centers, and the local covariance scales determine the smoothing term. The asymptotic and finite-budget analyses together describe support preservation across sampling regimes.

ReorgGS further converts the original opacity weights into spatial sampling density and initializes the new primitives with a constant low opacity. Let the constant-opacity occupancy field after reorganization be

$$\hat{F}(x) = \sum_{j=1}^{M} \alpha_0 K(x - x_j),$$

where K is a local kernel. Since $x_j \sim p(x)$,

$$\mathbb{E}[\hat{F}(x)] = M\alpha_0 \int K(x-z) p(z) dz.$$

When the local kernel bandwidth is small,

$$\mathbb{E}[\hat{F}(x)] \approx M\alpha_0 p(x).$$

Thus, the occupancy strength originally represented by α_i is represented after reorganization mainly by the sample density $Mp(x)$, while each primitive starts from the same low opacity α_0. This separation is essential for the subsequent optimization analysis: ReorgGS preserves the distributional support while changing the parameterization among opacity, spatial density, and local covariance.

A.2 Bounded Overlap under kNN Reorganization

Distributional equivalence does not imply optimization equivalence. The original converged model may stack many high-opacity Gaussians in the same local region. These primitives make similar rendering contributions, but they also create strong overlap and strong parameter coupling. The kNN covariance in ReorgGS ties the support scale of each new Gaussian to the local sampling density, which controls the number of effective overlaps under standard local regularity assumptions.

Let the local density of reorganized centers be

$$\rho(x) = Mp(x).$$

Near x, the volume that contains k nearest neighbors satisfies

$$V_k(x) \approx \frac{k}{\rho(x)}.$$

The kNN covariance makes the effective support volume of each new Gaussian proportional to $V_k(x)$. Let $K_q(x)$ denote the number of reorganized Gaussians that have a significant density or rendering response near x. If the local density is smooth, the samples are sufficiently dense, and the kernel support is controlled by the kNN scale, then

$$K_q(x) \approx \rho(x) \cdot V_k(x) \approx \rho(x) \cdot \frac{k}{\rho(x)} = O(k).$$

This statement does not mean that the reorganized Gaussians do not overlap. Rather, it states that the number of effective local overlaps no longer grows without bound with the number of redundant Gaussians in a degenerate original cluster. Dense regions produce more centers but smaller covariances; sparse regions produce fewer centers but larger covariances. These two effects compensate each other, keeping the effective local coverage at a scale controlled by k.

In contrast, standard densification clones or splits existing Gaussians near their current locations, and can preserve or amplify an existing local pile-up. If a degenerate cluster contains s strongly overlapping Gaussians, its local effective overlap can grow with s. ReorgGS resamples centers from the global empirical distribution and re-estimates covariances, rewriting such local redundancy into a density-controlled representation with adaptive support. This geometric property underlies the reduction of alpha-compositing occlusion and Hessian / FIM coupling.

A.3 Gradient Accessibility under Alpha Compositing

For a single ray r, the reconstruction loss is

$$\mathcal{L}_r = \|C(r) - C^*(r)\|_2^2.$$

For a responding primitive g, ignoring higher-order occlusion interaction terms, the leading gradient term is scaled by its foreground transmittance:

$$\|\nabla \mathcal{L}_r(g)\| \propto T_g(r) a(g, r) \|J_g(r)\|,$$

where $a(g, r)$ is the effective opacity of this primitive on the ray and $J_g(r)$ is its local rendering Jacobian. When the effective foreground opacities are small,

$$\log T_g(r) = \sum_{h<g} \log(1 - a_h(r)) \approx -\sum_{h<g} a_h(r).$$

Defining

$$A_g(r) = \sum_{h<g} a_h(r),$$

we obtain

$$T_g(r) \approx \exp(-A_g(r)).$$

This gives the opacity barrier induced by alpha compositing: the larger the accumulated effective opacity in front of a target surface, the smaller the gradient received by parameters near that surface. For high-opacity floaters in a degenerate distribution p, the true behavior is more severe than the small-opacity approximation. If a foreground Gaussian satisfies

$$a_j^p(r) \to 1,$$

then

$$\log(1 - a_j^p(r)) \to -\infty, \qquad T_g^p(r) = \prod_{h<g} (1 - a_h^p(r)) \to 0.$$

Thus, a high-opacity floater can approximately cut off the optimization path to the true surface behind it.

Since ReorgGS changes the Gaussian set, there is no strict index-wise correspondence between the primitives before and after reorganization. We therefore consider the same target physical surface near depth d along ray r, and compare the responding primitives g_p and g_q under the original and reorganized parameterizations. Let $A_d^p(r)$ be the accumulated effective opacity in front of the target surface in the original distribution. In the reorganized distribution, if there are at most $m_q(r, d)$ effective responding primitives in front of the surface and each satisfies $a_j^q(r) \le \alpha_0$, then

$$A_d^q(r) = \sum_{j<d} a_j^q(r) \le m_q(r, d)\alpha_0,$$

and

$$T_d^q(r) = \prod_{j<d} (1 - a_j^q(r)) \ge (1 - \alpha_0)^{m_q(r,d)} \approx \exp(-m_q(r, d)\alpha_0).$$

Proposition (gradient accessibility). If $A_d^q(r) \le m_q\alpha_0$ and the original degenerate distribution satisfies $A_d^p(r) > m_q\alpha_0$, then

$$T_d^q(r) > T_d^p(r).$$

Furthermore, for the responding primitives g_q and g_p near the same target physical surface,

$$\frac{\|\nabla \mathcal{L}_r(g_q)\|}{\|\nabla \mathcal{L}_r(g_p)\|} \approx \exp(A_d^p(r) - A_d^q(r)) \cdot \frac{a(g_q, r)}{a(g_p, r)}.$$

Proof. By alpha compositing,

$$T_d^p(r) \approx \exp(-A_d^p(r)), \qquad T_d^q(r) \approx \exp(-A_d^q(r)).$$

Together with $A_d^q(r) \le m_q\alpha_0 < A_d^p(r)$, this gives

$$T_d^q(r) \ge \exp(-m_q\alpha_0) > \exp(-A_d^p(r)) \approx T_d^p(r).$$

For any responding primitive g, the leading gradient term satisfies

$$\|\nabla \mathcal{L}_r(g)\| \propto T_d(r)a(g, r)\|J_g(r)\|.$$

Since g_p and g_q respond to the same local physical surface, their Jacobian norms are of the same order. Hence,

$$\frac{\|\nabla \mathcal{L}_r(g_q)\|}{\|\nabla \mathcal{L}_r(g_p)\|} \approx \frac{T_d^q(r)a(g_q,r)}{T_d^p(r)a(g_p,r)} \approx \exp(A_d^p(r) - A_d^q(r)) \cdot \frac{a(g_q,r)}{a(g_p,r)}.$$

This completes the proof.

The expression also explains the trade-off introduced by low-opacity initialization. Because α_0 is small, the local factor $a(g_q, r)/a(g_p, r)$ may be below one. However, $A_d^p(r)$ is a macroscopic occlusion quantity accumulated over all foreground floaters and redundant Gaussians along the ray, and it affects transmittance exponentially. When the original parameterization contains a strong opacity barrier, removing that barrier can yield an exponential recovery of gradients that dominates the local linear reduction caused by low opacity.

A.4 Hessian / FIM Coupling and Local Manifold Alignment

Gradient accessibility explains how ReorgGS reopens optimization paths to occluded surfaces. A second degeneracy comes from parameter coupling. If multiple Gaussians strongly overlap and explain the same pixel residuals, their Jacobian columns become highly correlated, making it difficult for the optimizer to determine which parameter block should be updated.

Define the opacity-weighted overlap energy as

$$\mathcal{O}(\mathcal{G}) = \sum_{i \neq j} \alpha_i \alpha_j \int G_i(x) G_j(x) dx.$$

Assuming normalized Gaussians,

$$\int \mathcal{N}(x \mid \mu_i, \Sigma_i)\mathcal{N}(x \mid \mu_j, \Sigma_j)dx = \mathcal{N}(\mu_i \mid \mu_j, \Sigma_i + \Sigma_j).$$

Thus, $\mathcal{O}(\mathcal{G})$ measures the opacity-weighted strength of spatial overlap. It can also be viewed as a geometric proxy for local optimization coupling. Let all parameters be $\Theta = \{\theta_i\}$. Under the Gauss–Newton approximation,

$$H \approx J^T J,$$

where $J = \partial C / \partial \Theta$. The off-diagonal block for parameter blocks i, j is

$$H_{ij} \approx \left(\frac{\partial C}{\partial \theta_i}\right)^T \left(\frac{\partial C}{\partial \theta_j}\right).$$

The off-diagonal blocks of the Fisher Information Matrix are similarly determined by inner products between the effects of different parameters on the observation distribution. If two Gaussians strongly overlap in space, they affect many of the same pixels and rays, making their Jacobian columns highly correlated. Therefore, $\int G_i G_j$ is not only a measure of geometric redundancy, but also a proxy for off-diagonal coupling in the Hessian / FIM.

Proposition (overlap energy reduction). In a degenerate distribution p, if a local cluster contains s strongly overlapping Gaussians with average opacity $\bar{\alpha}$, the overlap energy grows approximately quadratically:

$$\mathcal{O}_p = \Omega(s^2 \bar{\alpha}^2).$$

This quadratic growth corresponds to severe parameter multicollinearity. If only a standard opacity reset is applied, then

$$\mathcal{O}_{\text{reset}} = \alpha_0^2 \sum_{i \neq j} \int G_i^p(x) G_j^p(x) dx.$$

It reduces the opacity weights, but preserves the geometric overlap term of the old parameterization. After ReorgGS, we instead have

$$\mathcal{O}_{\text{Reorg}} = \alpha_0^2 \sum_{i \neq j} \int G_i^q(x) G_j^q(x) dx.$$

By the bounded-overlap analysis in Section A.2, if the effective local overlap after reorganization is controlled by m_q, then

$$\mathcal{O}_q = O(M m_q \alpha_0^2),$$

whereas the original overlap in a degenerate cluster can reach $\Omega(s^2 \bar{\alpha}^2)$. ReorgGS therefore reduces not only the opacity weights, but also the geometric overlap induced by the parameterization. This weakens off-diagonal Hessian / FIM coupling and makes the local curvature closer to a block-diagonal structure.

Finally, the kNN covariance provides a local manifold-alignment effect. If the scene is locally approximated by a low-dimensional manifold, let U denote the principal directions of the local neighborhood and Λ its variance spectrum. The kNN covariance estimate gives

$$\Sigma_j^q \approx U \Lambda U^T.$$

Since the local curvature of Gaussian position and scale parameters is related to $(\Sigma_j^q)^{-1}$, the aggregate local curvature of several reorganized Gaussians can be approximated as

$$H_{\text{local}} \propto \sum_j (\Sigma_j^q)^{-1} \approx K\, U \Lambda^{-1} U^T.$$

where K denotes the number of reorganized Gaussians in the local neighborhood. This does not mean that ReorgGS exactly aligns with the global eigenvectors of the Hessian, nor that it constructs an optimal preconditioner. It means that kNN covariance estimation makes the principal directions of local Gaussians consistent with the sampled geometric neighborhood, reducing conflicting directional updates. Together with low-opacity initialization and bounded overlap, ReorgGS converts a degenerate parameterization caused by high opacity, redundant stacking, and inconsistent orientations into a distributionally similar but more gradient-accessible and less coupled one.

B Additional Experimental Details and Discussions

In this section, we provide further context on our experimental design, addressing potential questions regarding the opacity reset baseline, statistical variance in our sampling process, and a detailed breakdown of computational costs. Our implementation will be released as open-source code. All experiments are run on NVIDIA A100 GPUs.

B.1 Computational Cost: Training Overhead and Rendering Speedup

While ReorgGS explicitly improves the rendering speed, we also transparently detail its computational overhead during the post-training phase.

Training & Module Overhead. As a post-training method, ReorgGS inherently increases the total training time due to the additional gradient descent steps required. For instance, each post-training stage adds approximately 5 minutes for the LiteGS architecture on an NVIDIA A100 GPU. However, the ReorgGS algorithmic module itself is extremely lightweight. Leveraging efficient GPU kNN

implementations, the complete topological reorganization (resampling and covariance re-estimation) of millions of Gaussians takes only a few seconds. Since this module is invoked very infrequently (e.g., 1 to 3 times per scene), its core algorithmic overhead is completely negligible compared to the rendering backward pass.

Rendering Speedup. Crucially, although the Gaussian budget is strictly matched to the baseline count ($M = N$), the reorganized topology inherently reduces rendering redundancy. By explicitly bounding local geometric overlap via K-NN initialization, fewer effective Gaussians participate in the depth-sorting and alpha-compositing operations per pixel tile. Validated on the LiteGS architecture, ReorgGS successfully reduces the total inference time from 10.75 seconds down to 9.37 seconds, yielding a concrete 12.8% acceleration without compromising geometric fidelity.

B.2 On the Insufficiency of Pure Opacity Reset

A natural question is why we do not isolate "pure opacity reset" as a separate comparative baseline to prove the effectiveness of ReorgGS. Importantly, in our experimental setup, the **Direct Continuation baseline already incorporates standard periodic opacity resets** (as per the default 3DGS training schedule). The fact that direct continuation still plateaus with blurry artifacts proves that merely resetting opacity on a degenerate topology is fundamentally insufficient.

The operations are simply not on the same level. Opacity reset only scales down the edge weights of the existing overlap graph ($\alpha_i \leftarrow \alpha_0$). The overlapping primitives maintain the exact same centers μ and covariance matrices Σ, meaning their Jacobians remain highly collinear. To rapidly minimize the photometric loss, the optimizer quickly scales the opacities of these exact same redundant Gaussians back up, returning the model to its original topological deadlocks. In contrast, ReorgGS operates on a fundamentally deeper structural level. By probabilistically resampling centers and re-estimating kNN covariances, it physically rebuilds the overlap graph and fundamentally breaks the parameter coupling.

B.3 Robustness to Random Seeds (Statistical Stability)

Because ReorgGS involves categorical sampling and Gaussian coordinate sampling, one might ask whether the choice of random seed introduces high variance into the final reconstruction, and whether averaging results across multiple seeds per scene is necessary.

In practice, this statistical variance is entirely marginalized out by two macro-level factors. First, at the *intra-scene* level, a typical 3DGS model contains millions of primitives ($M \approx 10^5 \sim 10^6$). Drawing such a massive number of independent samples ensures that the macroscopic occupancy distribution tightly converges to the empirical field $p(x)$ (as formalized in Proposition A.1, $\mathrm{Var}[q_M(x)] \propto 1/k \to 0$). Second, at the *inter-scene* level, our quantitative tables report metrics averaged across dozens of diverse scenes from multiple datasets. This combination of massive large-scale sampling within each scene and macro-averaging across multiple datasets completely flattens any microscopic statistical fluctuations caused by individual random seeds, making the reported performance highly stable and reproducible.

B.4 Paired and Scene-level Improvement Statistics

We summarize paired differences $\Delta = \mathrm{ReorgGS} - \mathrm{Direct}$ across seven configurations and 11 scenes. Table 3 reports their mean, median, interquartile range (IQR), and favorable-sign count. PSNR and SSIM use positive differences, while LPIPS uses negative differences. After averaging configurations within each scene, train PSNR improves in 10/11 scenes and train SSIM and LPIPS improve in 11/11. Test PSNR improves in 8/11 scenes, with test SSIM and LPIPS improving in 11/11. The IQR describes variation across configuration–scene pairs.

B.5 Continuation from a Weaker Initialization

For each of the 11 scenes, we initialize 100k random Gaussians and train for 30k iterations to obtain a shared baseline checkpoint. Direct continuation and ReorgGS then each receive 30k additional iterations from that checkpoint under the shared continuation protocol. Table 4 reports scene-averaged results. The 100k count specifies the initial point set, and density control follows the continuation protocol.

Table 3: Paired gains across seven configurations and 11 scenes.

Split	Metric	Mean	Median	IQR	Improved
Train	PSNR	+0.5338	+0.2535	[+0.0836, +1.0911]	67/77
Train	SSIM	+0.0025	+0.0018	[+0.0002, +0.0039]	60/77
Train	LPIPS	-0.0030	-0.0025	[-0.0063, +0.0007]	53/77
Test	PSNR	+0.1149	+0.1002	[+0.0138, +0.2473]	59/77
Test	SSIM	+0.0021	+0.0015	[+0.0002, +0.0026]	62/77
Test	LPIPS	-0.0019	-0.0014	[-0.0036, +0.0005]	51/77

Table 4: Continuation from a shared, weak-initialization checkpoint.

Setting	Train PSNR	Test PSNR	Test SSIM	Test LPIPS
Baseline	30.2305	25.9415	0.7914	0.2679
Direct	32.1188	27.3075	0.8324	0.2133
ReorgGS	33.2255	27.7009	0.8443	0.2002

C Qualitative Comparisons

As visualized in our qualitative results, ReorgGS consistently recovers finer geometric and textural details that are otherwise lost in baseline methods due to optimization deadlocks. We highlight specific challenging regions across different scenarios:

KITCHEN (High-Frequency Textures): For indoor scenes rich in fine-grained patterns, as shown in Fig. 5, such as the floor tiles in the KITCHEN dataset, direct finetuning often yields over-smoothed results because redundant Gaussians blur the structural boundaries. ReorgGS redistributes the primitives to align with local manifolds, faithfully restoring the sharp, intricate textural details of the floor.

PLAYROOM (Ghosting Artifacts): As shown in Fig. 6, baseline methods typically struggle with severe topological entanglement, which manifests as noticeable ghosting artifacts (double images) around the wall switches. By explicitly breaking these fixed occlusion structures, ReorgGS successfully converges to a single, crisp geometry for the switch.

BICYCLE (Complex Background Geometry): In outdoor environments like the BIKE scene, as shown in Fig. 7, recovering complex, depth-varying background structures is highly challenging. While baselines tend to merge the background leaves and branches into cloudy, semi-transparent artifacts, ReorgGS excels at resolving the crisp silhouettes and natural gaps of the intricate foliage, demonstrating its superior capability in modeling high-frequency geometric boundaries.

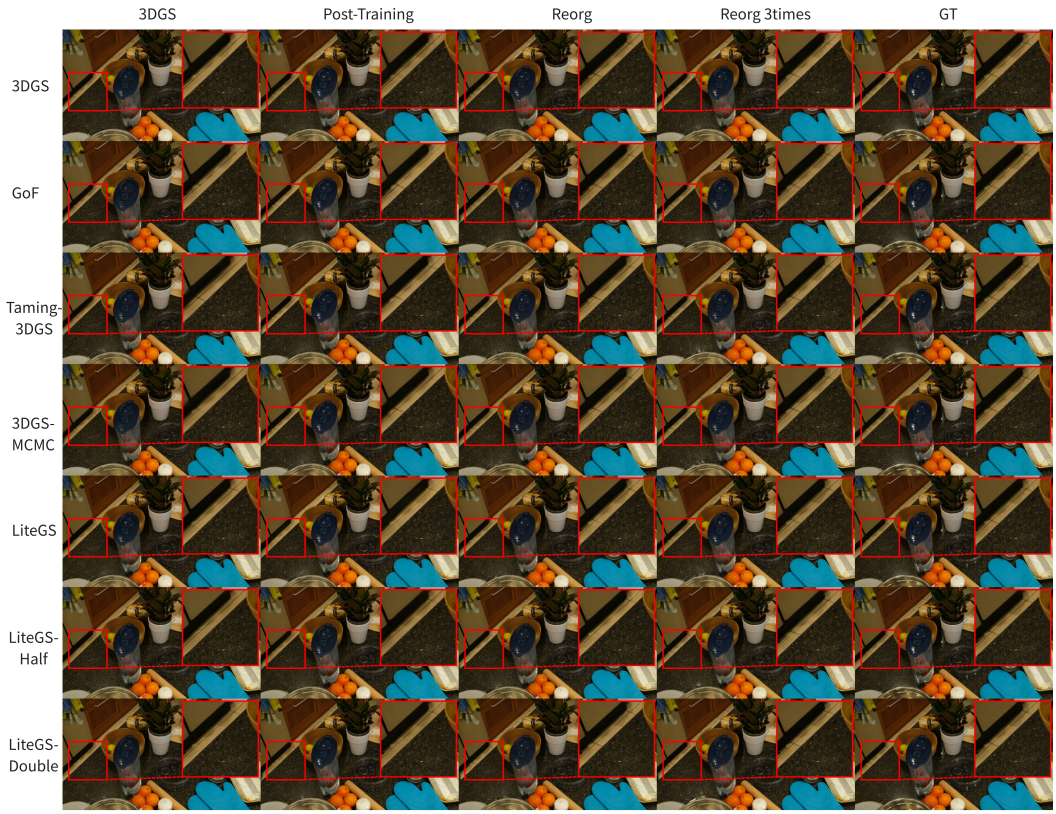

Figure 5: **Qualitative comparisons on the Kitchen scene.** We evaluate Reorg across six baselines (rows). Columns show the original baseline, standard post-training, Reorg (1 pass), Reorg (3 passes), and Ground Truth. As highlighted in the red insets, standard post-training struggles with the dark bottle boundaries and countertop reflections, whereas Reorg effectively breaks the optimization deadlocks to restore sharp, high-fidelity details.

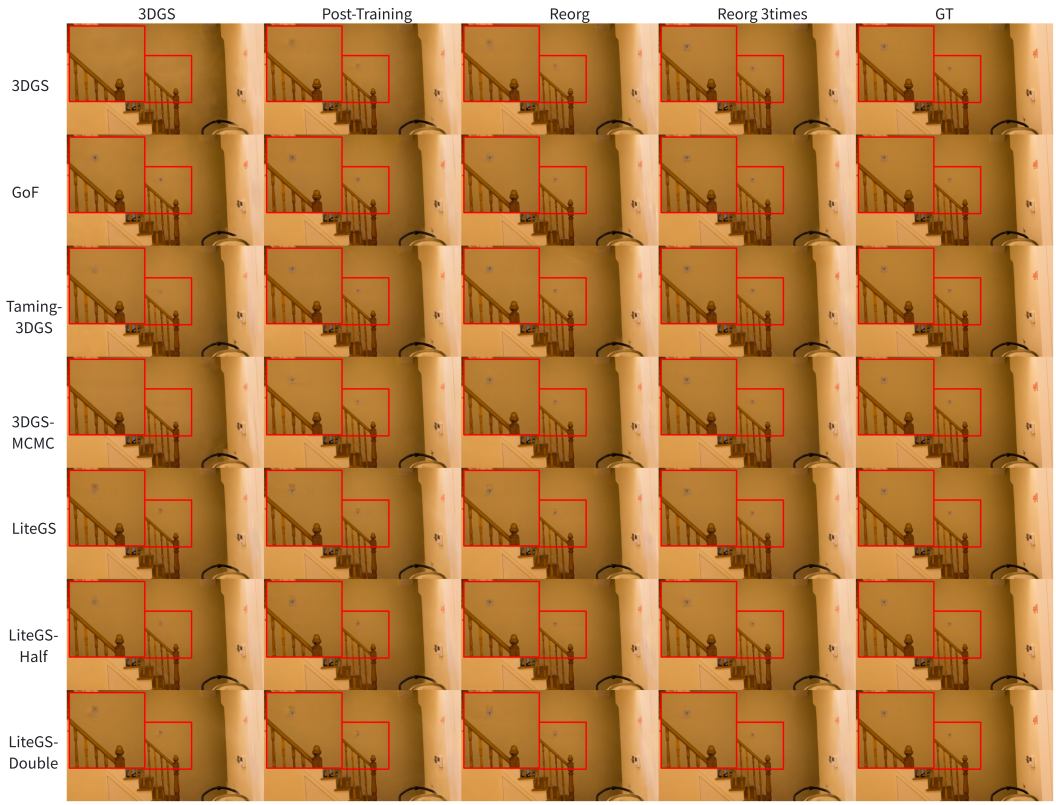

Figure 6: **Qualitative comparisons on the Playroom scene.** This scene challenges the parameterization of thin geometric structures (wooden railings) against a plain background. While standard post-training often leaves floating artifacts or broken geometry, Reorg successfully restores continuous and clean foreground structures without polluting the background.

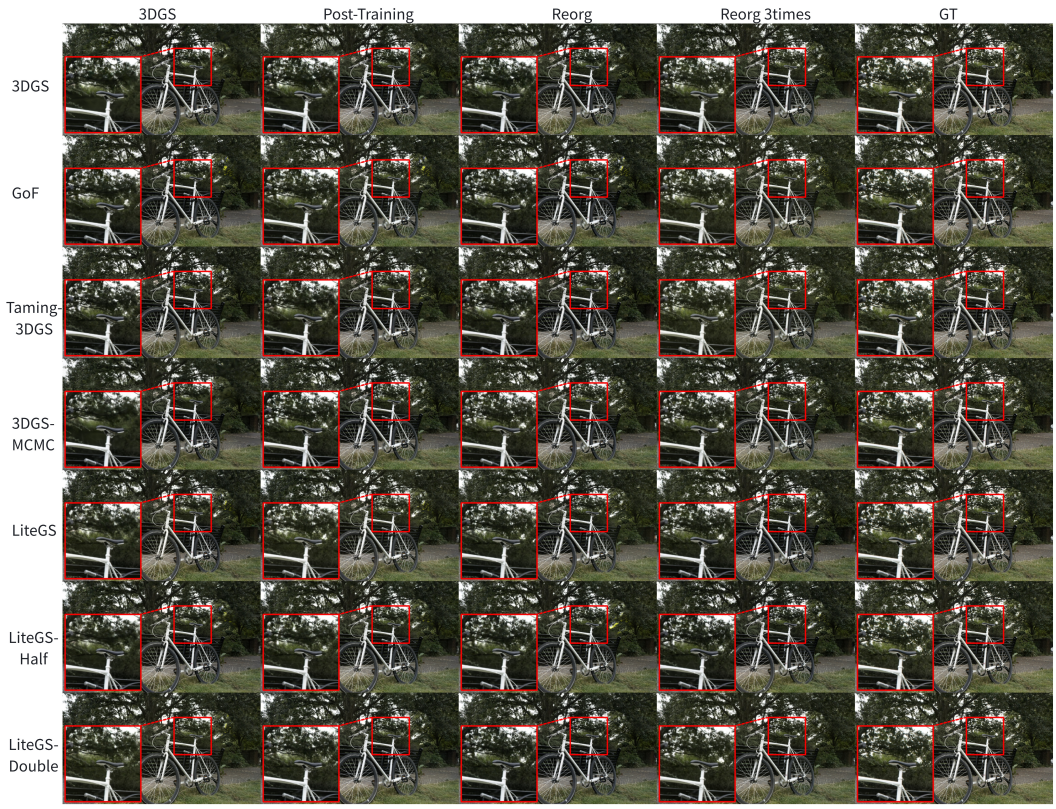

Figure 7: **Qualitative comparisons on the Bicycle scene.** This scene presents extreme depth occlusions between thin metallic cables and a high-frequency foliage background. Reorg accurately disentangles the foreground objects from the complex background, significantly alleviating the blurring artifacts seen in standard gradient-based finetuning.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the paper's main scope: parameterization degeneration in converged 3DGS models, the ReorgGS reorganization procedure, and its optimization effects. These claims are supported by the method description, theoretical analysis, and controlled experiments across multiple backbones and Gaussian budgets.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The paper includes a concise limitations discussion before the conclusion. It states that ReorgGS assumes the converged Gaussian set already contains reasonable scene support and that the method requires additional optimization after reorganization.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The paper states the assumptions used in the optimization analysis and provides detailed derivations in the appendix, including the links between opacity-density allocation, bounded overlap, gradient accessibility, and local curvature. The main text also gives proof sketches and intuition for how these results explain the behavior of ReorgGS.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The experiments describe the evaluated backbones, datasets, Gaussian-budget control, metrics, and comparison protocol, including baseline, direct continuation, and ReorgGS under the same additional optimization budget. The method section specifies the reorganization steps needed to reproduce the algorithm: opacity-weighted sampling, Gaussian coordinate sampling, kNN covariance estimation, low-opacity initialization, appearance inheritance, and continued optimization with the unchanged renderer and loss.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The authors will release the implementation as open-source code, together with instructions for reproducing the main experiments. The paper also uses standard public novel-view synthesis datasets and established 3DGS backbones.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The experimental section specifies the datasets, standard 3DGS train/test protocol, evaluated backbones, fixed Gaussian-budget control, metrics, and equal-budget comparison settings. It also describes the main ReorgGS hyperparameters and ablation factors, including covariance initialization and kNN neighborhood size.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No].

Justification: The paper reports aggregate metrics across standard scenes and controlled comparisons across several backbones and budgets, but it does not currently report error bars, confidence intervals, or statistical significance tests. This is common in 3DGS scene-level evaluations where each scene is optimized separately.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: The experimental setup states that all experiments are run on NVIDIA A100 GPUs under the same additional optimization budgets specified for each comparison. Per-run wall-clock times will be provided with the released code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work studies a rendering and optimization method for 3D Gaussian Splatting using standard research datasets and does not involve human-subject experiments, sensitive personal data collection, or harmful deployment. The submission should preserve anonymity and follow the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The paper includes a concise broader-impact discussion. It notes potential benefits for efficient 3D reconstruction and possible misuse in privacy-sensitive 3D capture or synthetic visual content.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release high-risk foundation models, generative image models, scraped datasets, or assets with substantial dual-use risk. It studies an optimization method for standard 3DGS scene representations evaluated on common novel-view synthesis benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The source methods and datasets are credited through citations in Sections 2 and 4.1. The code release will include third-party license notices, asset-source links, and reproduction instructions.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper does not introduce a new dataset, benchmark, pretrained model, or other new asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects. All experiments are computational evaluations on standard 3D reconstruction and novel-view synthesis datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject research, so IRB approval or equivalent review is not applicable. The work consists of algorithmic development and computational experiments on existing datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method, theoretical analysis, and experiments do not use LLMs as an important, original, or non-standard research component. Any possible use of LLMs for writing, editing, or formatting would not affect the method or scientific claims and therefore does not require declaration under the checklist guidance.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.