Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Visibility-Aware Texture Consensus and Local Structural Constraints for 3D Gaussian Reconstruction in Texture- Degraded Scenes

Mengmeng Xu
Suqian University
Liansuo Wei
weiliansuo@squ.edu.cn

Suqian University
Weiwei Shen
Suqian University

Article

Keywords: 3D Gaussian splatting, texture-degraded scenes, geometry–appearance decoupling, multi-view color consensus, local principal component analysis, robust optimization

Posted Date: September 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10727268/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Visibility-Aware Texture Consensus and Local Structural Constraints for 3D Gaussian Reconstruction in Texture-Degraded Scenes

Mengmeng Xu[1], **Liansuo Wei**[1]✉ & Weiwei Shen[1]

[1] School of Information Engineering, Suqian University, Suqian, Jiangsu 223800, China.

✉Corresponding author: Liansuo Wei (weiliansuo@squ.edu.cn)

Abstract

3D Gaussian Splatting (3DGS) can propagate unreliable photometric residuals into geometric parameters in weakly textured, repetitive-texture, low-light, and occlusion-boundary regions. We investigate Visibility-Aware Texture Consensus (VTC), parameter-level loss routing, and local-PCA structural constraints for texture-degraded reconstruction. Under a strict 80/20 path-isolated protocol on four NeRF Synthetic scenes, the simplified configuration improves the official-validation macro-average by 0.0661 dB in PSNR, 0.00246 in SSIM, 0.00136 in Alpha IoU, and reduces LPIPS by 0.00036 and Outside alpha by 6.33×10−5 relative to Control. In an additional four-scene, five-seed paired matrix at 300 iterations, PSNR improves for all five seeds in every scene; the exact two-sided Wilcoxon signed-rank test gives p=0.0625. Controllable texture removal, illumination reduction, and occlusion experiments show monotonic changes in degradation proxies, supporting the proposed identifiability analysis. Additional DTU and Mip-NeRF 360 experiments verify cross-dataset interfaces, while standard-densification tests delimit transfer behavior. Overall, the evidence supports improved relative robustness under a low-budget fixed-Gaussian protocol, while not supporting state-of-the-art claims on real-world data or performance gains from consensus inheritance.

Keywords: 3D Gaussian splatting; texture-degraded scenes; geometry–appearance decoupling; multi-view color consensus; local principal component analysis; robust optimization

Introduction

The core objective of multi-view 3D reconstruction is to recover scene geometry and surface appearance simultaneously, enabling downstream applications such as virtual reality, digital twins, film and television production, and robotic perception (see the survey in [1]). Neural Radiance Fields (NeRF), based on implicit neural fields, represent scenes as continuous radiance fields and achieve high-quality novel-view synthesis [2–4], but dense sampling along camera rays and per-sample network queries incur substantial training and rendering costs. In contrast, 3D Gaussian Splatting (3DGS) parameterizes a scene with explicit anisotropic 3D Gaussians and combines them with differentiable rasterization for high-quality real-time rendering [5]. Extensions of 3DGS have continued to advance in anti-aliasing, density control, dynamic reconstruction, and scene editing [6,7]. Nevertheless, vanilla 3DGS uses a single image-reconstruction objective to optimize geometry and appearance jointly, allowing unreliable photometric residuals in weakly textured regions to propagate into geometric variables and thereby limiting its applicability in degraded scenes.

3DGS updates geometric and appearance parameters using the same rendering error. When a scene contains weakly textured surfaces, repetitive textures, dark occlusions, or slender structures, cross-view color correspondences become unstable, and local color, exposure, and occlusion errors may be interpreted as geometric deviations. The resulting abnormal updates of Gaussian centers, scales, and opacities manifest as boundary drift, contour expansion, thin-rod fracture, alpha leakage outside the foreground mask, and floating dark Gaussians (Fig. 1). Accordingly, we define a texture-degraded scene as an observation condition in which the foreground contains a high proportion of low-texture pixels, large cross-view color dispersion, or insufficient foreground–background boundary support, such that single-view photometric supervision cannot stably constrain geometric parameters. This operational definition emphasizes

measurable criteria and the identifiability of supervision, thereby delimiting the method design and evaluation protocol that follow.

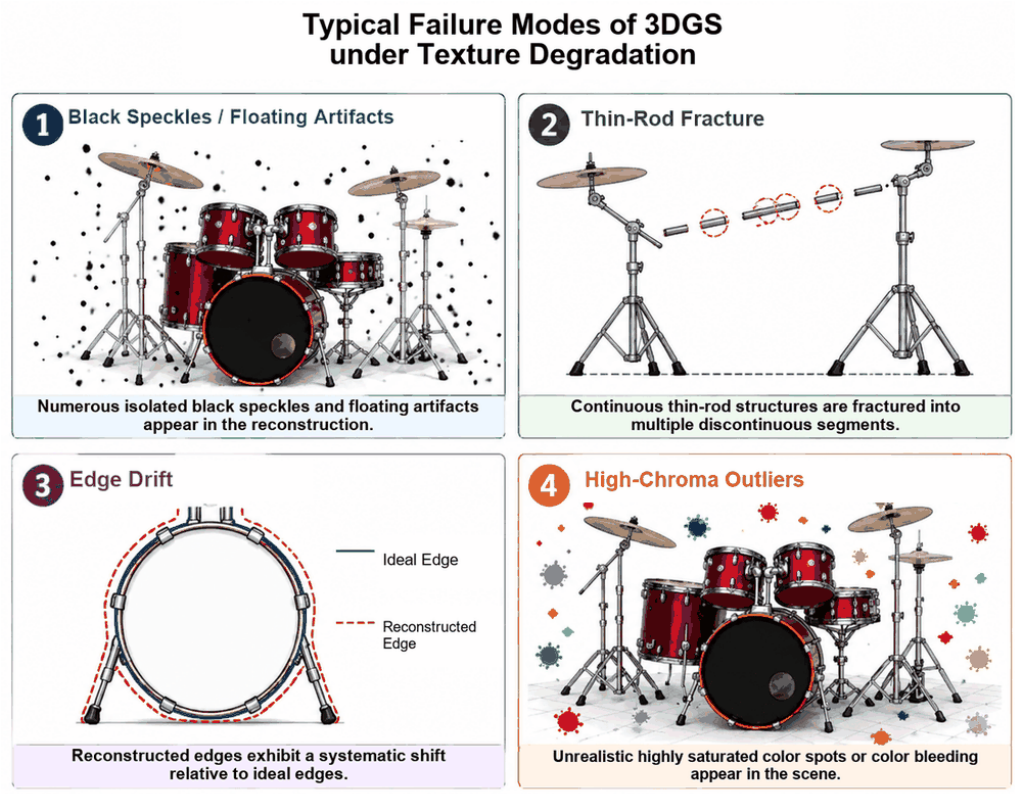


Fig. 1 Typical failure modes of 3DGS under texture degradation: floating dark speckles, thin-rod fracture, boundary drift, and high-chroma outliers

Existing approaches strengthen geometric reconstruction through depth supervision [8], normal and surface constraints [9–11], edge regularization, and adaptive density control [12,13]. However, three common limitations remain. First, visible observations are typically treated as equally reliable, without observation-level rejection of colors corrupted by occlusion, background mixing, or exposure differences. Second, local geometric and appearance relationships may be modeled jointly, so color errors can be back-propagated to geometric parameters through a shared rendering objective. Third, uniform geometric constraints are often applied to all regions, restricting the optimization freedom of normally textured areas. Texture-degraded reconstruction therefore requires separate treatment at three levels: filtering unreliable observations, disentangling geometry from appearance in the parameter pathways, and selectively constraining geometric updates according to local structure.

We first establish an identifiability analysis of geometric supervision under texture degradation (Sec. 3.2), and then design the method from three perspectives: observation reliability, parameter routing, and local structural statistics. Visibility-Aware Texture Consensus (VTC) filters observations using image bounds, foreground membership, projected visibility, and depth consistency, and constructs cross-view color consensus and consensus weights from the channel-wise median and median absolute deviation (MAD) (Sec. 3.3). Parameter routing restricts the parameter groups affected by each loss term, yielding optimization-level geometry–appearance decoupling (Sec. 3.4). Local principal component analysis (local PCA) estimates neighborhood structural directions and is used to construct structural confidence and a geometric anchoring regularizer (Sec. 3.5).

We evaluate performance gains and mechanism boundaries separately. Global reconstruction quality is measured with PSNR, SSIM, and LPIPS, whereas local diagnostics use Alpha IoU, alpha leakage outside the foreground mask, and counts of anomalous Gaussians. Experiments on directional projection and parameter-level gradient routing distinguish changes in optimization behavior from changes in reconstruction quality. Under the fixed-Gaussian protocol, small but paired improvements

are observed, whereas stronger directional intervention changes update directions without stable quality gains. Standard-densification results are reported separately in the supplementary validation experiments.

The main contributions of this work are as follows:

(1) We establish an identifiability analysis of geometric supervision under texture degradation. Starting from alpha-compositing rendering, we derive an explicit decomposition of the Gaussian-center gradient with respect to a single observation, and under assumptions of local color homogeneity and residual decomposability, we provide an identifiability criterion and drift-accumulation mechanism for weakly textured regions (Proposition 1).

(2) We propose VTC. Before training, VTC filters reliable observations using four criteria, constructs a color consensus using channel-wise medians, and determines consensus weights from MAD and the effective sample count. The consensus remains fixed during training, preventing the current rendering error from contaminating the supervision target through feedback (Propositions 2–3: convexity and MAP interpretations).

(3) We develop parameter-level geometry–appearance decoupling and local structural constraints. Parameter routing restricts the action domain of each loss so that VTC acts only on the appearance parameters Θ_a; local PCA provides neighborhood principal axes, normals, and structural confidence; and the geometric anchoring regularizer imposes an adaptive prior in the orthogonal subspace according to structural confidence (a heavy-tailed generalized Gaussian prior with exponent $p_a = 1.65$).

(4) We establish a strict evaluation and mechanism-analysis protocol. Training and held-out views are strictly isolated at the data-path level; center-direction projection, photometric-gradient routing, and candidate modules are evaluated through controlled-variable experiments; and cumulative displacement decomposition is used to test whether update directions are altered. This protocol clearly separates the scope of the evidence from its extrapolation limits.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the problem formulation, identifiability analysis, and proposed method. Section 4 reports the experimental protocol, comparisons, ablations, mechanism analyses, and supplementary validation. Section 5 summarizes the conclusions and evidence boundaries.

Related Work

Neural Radiance Fields and 3D Gaussian Scene Representations

Multi-view 3D reconstruction involves camera-pose estimation, geometric recovery, and appearance modeling. Classical structure-from-motion methods estimate camera parameters and sparse point clouds through feature matching and multi-view geometric constraints, with COLMAP becoming a standard preprocessing tool [14]. Building on this pipeline, NeRF represents a scene as a continuous radiance field conditioned on spatial position and viewing direction and performs high-quality novel-view synthesis through volume rendering [2]. Mip-NeRF 360 further introduces spatial parameterization and anti-aliasing mechanisms for unbounded scenes, improving rendering stability under large scale variations [3]. Recent surveys have systematically reviewed neural radiance fields and inverse-rendering-based reconstruction [1,4,6]. However, implicit representations typically require dense sampling along camera rays and repeated network queries, resulting in relatively high training and rendering costs.

3DGS represents a scene with a set of explicit anisotropic 3D Gaussians and uses differentiable rasterization for high-quality real-time rendering [5], avoiding extensive empty-space sampling and offering clear advantages in training efficiency and rendering speed. Subsequent work has focused on anti-aliasing, density control, and primitive management. Mip-Splatting reduces view-scaling aliasing through 3D smoothing and 2D Mip filtering [12]; GaussianPro uses progressive Gaussian propagation to improve representation in under-reconstructed regions [15]; Localized Points Management refines densification and pruning from the perspective of local Gaussian management [13]; and Rendering-Aware Bayesian 3DGS introduces uncertainty estimation and adaptive complexity control from a probabilistic perspective [16].

These methods mainly address representation efficiency, sampling stability, and control of Gaussian count. Vanilla 3DGS, however, still relies on image-space photometric residuals to optimize geometry and appearance simultaneously. In weakly textured, repetitive, or dark occluded regions, these residuals lack stable geometric correspondences and can therefore provide erroneous supervision to Gaussian position, scale, and opacity.

Geometry-Aware Gaussian Reconstruction and Surface Constraints

To address the insufficient geometric constraints of vanilla 3DGS, extensive research has explored depth priors, surface alignment, and explicit structural relationships. For depth priors, DNGaussian introduces monocular depth into Gaussian optimization through global-to-local depth normalization under sparse-view settings [8], while Sparse2DGS prioritizes geometric recovery in sparse views and enhances surface reconstruction of 2D Gaussians using geometric priors [17]. For surface alignment, SuGaR regularizes 3D Gaussians to align with scene surfaces and supports mesh extraction [9]; 2D Gaussian Splatting replaces 3D ellipsoidal Gaussians with 2D Gaussian disks having explicit tangent-plane structure, allowing primitives to better conform to actual surfaces [10]; Geometry Field Splatting represents local surfaces with Gaussian surfels and uses a geometry field to enhance surface continuity [18]; and SurfaceSplat unifies explicit surface reconstruction and Gaussian splatting in a collaborative optimization framework [11]. For explicit structural relationships, GeoGaussian introduces geometry-aware initialization and optimization for weakly textured planes and geometrically degenerate regions [19]; SAGS builds structure-aware relationships among Gaussians and constrains primitive motion using local and global structural information [20]; and MGSR mutually reinforces 2D and 3D representations to improve surface reconstruction under varying illumination [21]. These studies demonstrate the importance of depth, normals, surface structure, and Gaussian-neighborhood relationships for geometric recovery. Most methods, however, constrain all Gaussian parameters through additional geometric supervision or uniform structural regularization without separating local geometric from appearance relationships. When appearance variation is inconsistent with true structure, color errors may still propagate to geometric variables through the shared rendering objective. Section 3.4 complements this line of work by restricting, at the optimization level, the parameter groups affected by each loss term.

Robust Optimization under Sparse Observations and Outlier Interference

Under sparse-view settings, limited observations can cause Gaussian representations to overfit the training views, producing geometric drift and appearance artifacts in unobserved regions. SparseGS proposes Gaussian optimization for panoramic sparse-view synthesis and improves real-time novel-view quality with few input views [22]. CoR-GS jointly optimizes two Gaussian radiance fields with stochastic differences and uses point-level and rendering-level disagreements for co-regularization, thereby reducing sparse-view overfitting [23]. Transient distractors in real scenes, such as pedestrians, vehicles, and reflections, also break cross-view consistency. DeSplat decomposes static-scene Gaussians from view-dependent distractor components to prevent transient content from being baked into the scene representation [24]. These approaches primarily compensate for insufficient observations using model disagreement or additional priors, or remove anomalous components altogether. Texture-degraded regions, however, are part of the scene itself and cannot simply be removed; observation reliability must instead be assessed from visibility, boundary support, and cross-view consistency.

A closely related line of research concerns robust supervision and uncertainty modeling in neural radiance fields. NeRF-W uses appearance embeddings and a transient head to decompose illumination changes and temporary occlusions in unconstrained photo collections [28]. RobustNeRF models distractors as optimization noise and automatically suppresses abnormal observations with robust losses [29]. RawNeRF improves low-light reconstruction from raw images through noise modeling and exposure stabilization [30]. Uncertainty-quantification approaches such as Bayes' Rays use posterior uncertainty to characterize regional reliability [31]. These works address unreliable observations in implicit representations either by decomposing transient components or by globally down-weighting them through robust objectives. VTC differs in that it operates on explicit Gaussian representations, explicitly determines visibility and depth consistency at Gaussian granularity, and down-weights supervision using a fixed robust consensus target

computed before training rather than an uncertainty estimate updated online. Conceptually, both lines of work address suppression of the interference term characterized by Proposition 1 in Sec. 3.2. Gaussian-splatting studies have also explored scene segmentation [32] and dynamic reconstruction [6], but the identifiability of geometric supervision under texture degradation has not yet been systematically examined.

Geometry–Appearance Decoupled Modeling

Geometric structure and surface appearance have distinct physical meanings in Gaussian representations, yet vanilla 3DGS jointly optimizes Gaussian centers, covariances, opacities, and spherical-harmonic coefficients through a unified image-reconstruction loss. Although this joint parameterization facilitates end-to-end training, it may introduce undesirable coupling between geometry and appearance. GStex equips 2D Gaussians with per-primitive texture representations, separating surface texture from Gaussian geometric parameters [25]. Geometry Gaussians further pursues explicit appearance–geometry decoupling through independent representations, reducing the effect of view-dependent color on geometric recovery [26]. In implicit neural fields, NeRV jointly predicts volumetric density, surface normals, material parameters, and direction-dependent light visibility to decompose the physical imaging process [27]; related inverse-rendering literature has also surveyed surface reconstruction and material–illumination estimation [1]. Unlike these approaches, we introduce no new representation parameters and do not estimate BRDFs or illumination components. Instead, under fixed observation lighting, we delimit the reliable range of appearance supervision through pre-training visibility filtering and cross-view color statistics, and we constrain both the parameter groups acted on by each loss and the direction of geometric updates at the optimization level. In other words, existing decoupling methods primarily separate representation parameters or texture representations, whereas our method decouples supervision attribution, parameter pathways, and update directions. Even when appearance and geometry are separated in the representation, unreliable photometric residuals can still back-propagate into geometric variables through a joint rendering objective; this optimization-level coupling is the focus of our work.

Method

Problem Formulation and Overall Framework

Given N_v training views, the multi-view data consist of RGB images, foreground masks, and camera parameters. The observation set is

$$\mathcal{O} = \{(I_v, M_v, K_v, R_v, t_v) \mid v = 1, \dots, N_v\} \tag{1}$$

where I_v denotes the RGB image of the v-th view, M_v is the corresponding foreground mask, and K_v and (R_v, t_v) denote the camera intrinsics and extrinsics, respectively. The foreground mask distinguishes the target region from the background. As in standard 3DGS, the initial Gaussian centers can be obtained from a sparse point cloud reconstructed by structure from motion or from occupancy-silhouette reconstruction [14]. The scene is represented by N Gaussians:

$$\mathcal{G} = \{g_k = (\boldsymbol{\mu}_k, \mathbf{s}_k, \mathbf{q}_k, \alpha_k, \mathbf{h}_k) \mid k = 1, \dots, N\} \tag{2}$$

where $\boldsymbol{\mu}_k \in \mathbb{R}^3$ is the 3D center; $\mathbf{s}_k$ and $\mathbf{q}_k$ are the scale vector and rotation quaternion, respectively; $0 < \alpha_k \leq 1$ denotes opacity; and $\mathbf{h}_k$ denotes the spherical-harmonic appearance coefficients. The scale vector and rotation quaternion jointly determine the covariance matrix:

$$\boldsymbol{\Sigma}_k = \mathbf{R}_k \mathbf{S}_k \mathbf{S}_k^{\top} \mathbf{R}_k^{\top} \tag{3}$$

This matrix describes the orientation and shape of a Gaussian in 3D space. For pixel p, let $\mathcal{N}(v, p)$ denote the sequence of Gaussians projected to that pixel in view v and sorted by depth. The rendered color is

$$\hat{\mathbf{C}}(v, p) = \sum_{i \in \mathcal{N}(v,p)} \mathbf{c}_i(v)\, \tilde{\alpha}_i(v, p) \prod_{j<i} \left(1 - \tilde{\alpha}_j(v, p)\right) \tag{4}$$

where $c_i(v)$ is the color of Gaussian i evaluated from its spherical-harmonic coefficients according to the viewing direction of view v; $\tilde{\alpha}_i(v, p)$ is the effective opacity contribution of that Gaussian at pixel p, obtained by multiplying its opacity by the value of the projected 2D Gaussian kernel; and the product term is the accumulated transmittance produced by nearer Gaussians. The predicted alpha is

$$\hat{A}(v,p) = \sum_{i\in\mathcal{N}(v,p)} \tilde{\alpha}_i(v,p) \prod_{j<i} \left(1 - \tilde{\alpha}_j(v,p)\right) \tag{5}$$

The discrepancy between the predicted alpha Â(v,p) and the ground-truth foreground mask M(v,p) is used as the input to the foreground-mask loss in Sec. 3.4.

To distinguish the parameter subsets actually updated by different modules, the Gaussian parameters are partitioned, according to physical meaning, into two non-overlapping groups:

$$\Theta_g = \{\boldsymbol{\mu}_k, \mathbf{s}_k, \mathbf{q}_k, \alpha_k\}_{k=1}^{N}, \quad \Theta_a = \{\mathbf{h}_k\}_{k=1}^{N}, \quad \Theta_g \cap \Theta_a = \varnothing \tag{6}$$

where Θ_g controls Gaussian position, shape, orientation, and alpha compositing and is therefore referred to as the geometry-related parameter group; Θ_a consists of the spherical-harmonic color coefficients and is referred to as the appearance parameter group. All subsequent losses and gradient operations act on the corresponding parameter groups according to the parameter-routing rules.

Figure 2 shows the overall framework. Before training, VTC produces the reliable observation set, color consensus, MAD, and consensus weights (Sec. 3.3). Parameter routing then restricts the parameter groups affected by each loss term (Sec. 3.4). The local-PCA stage estimates neighborhood structure and constructs the geometric anchoring regularizer (Sec. 3.5). A scene-level coefficient scales the constraint strength according to the degree of degradation (Sec. 3.6). The default configuration enables VTC, parameter routing, local-PCA statistics, and geometric anchoring; directional projection, the structure–appearance dual-graph regularizer, and residual shrinkage are used only in mechanism experiments. The consensus-inheritance extension is described in Sec. 3.3.4.

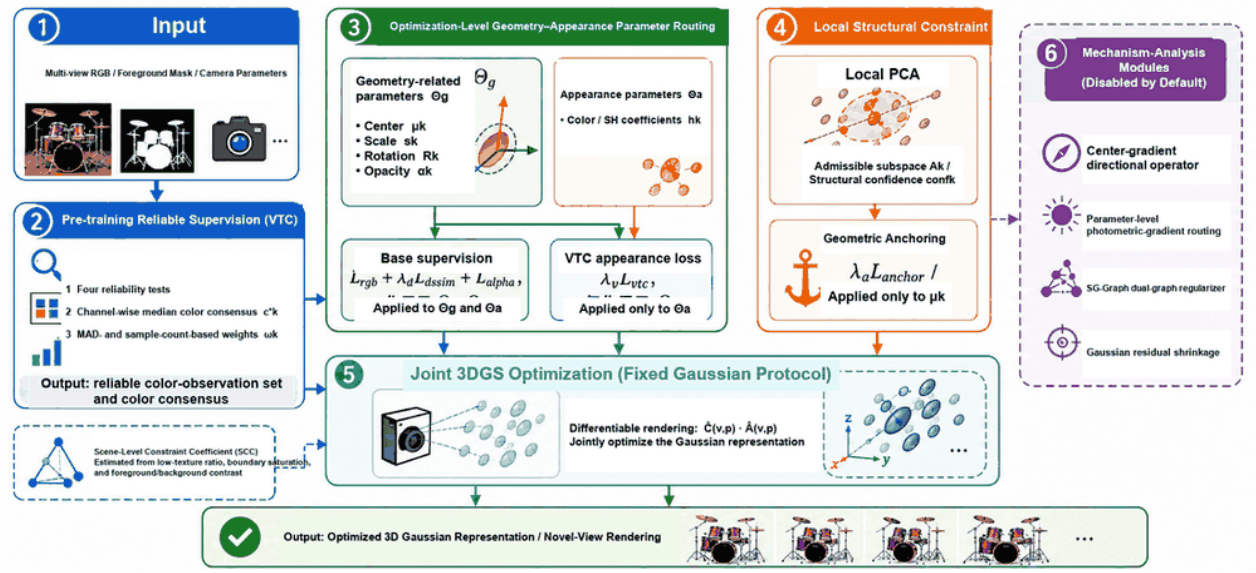


Fig. 2 Overall framework of the proposed method (solid lines: default configuration; dashed lines: candidate modules used only for mechanism analysis)

Our basic hypothesis regarding the failure mechanism is that weakly textured regions and foreground–background mixed boundaries reduce the identifiability of photometric gradients with respect to geometric parameters. In a local coordinate system, the Gaussian-center gradient can be decomposed into components along the principal structural axis or tangent plane and components along the orthogonal (normal) direction. The former correspond to plausible within-structure adjustments, whereas the latter are more susceptible to color ambiguity, occlusion ordering, and background mixing. We therefore distinguish two types of experimental evidence. System-level performance evidence on held-out views (PSNR, SSIM, LPIPS, Alpha IoU, and outside-mask leakage) is used to delimit performance claims, whereas directional mechanism evidence in the shared-initialization local-PCA coordinate system (allowed-direction and orthogonal components of cumulative center displacement) is used to delimit mechanism claims. These two evidence types are independent and jointly define the scope within which the method can be interpreted. A formal statement of the hypothesis, the identifiability criterion, and a proof sketch are given in Sec. 3.2.

Identifiability Analysis of Geometric Supervision under Texture Degradation

Decomposition of Geometric Gradients over Observations

To characterize the strength with which photometric supervision constrains geometric parameters, we start from the alpha-compositing model in Eq. (4) and derive an explicit form of the Gaussian-center gradient for a single observation. Let v denote the projected 2D kernel of Gaussian p at pixel k in view $G_k(v,p) = \exp\left[-\frac{1}{2}(p-\hat{p}_k)^\top {\Sigma'_k}^{-1}(p-\hat{p}_k)\right]$, with effective opacity $\tilde{\alpha}_k(v,p) = \alpha_k\, G_k(v,p)$. Splitting the summation in Eq. (4) according to the front-to-back order around Gaussian k into the foreground composite F_k, preceding transmittance $T_k = \prod_{j<k}\left(1-\tilde{\alpha}_j\right)$, and background composite $B_k = \sum_{i>k} c_i\, \tilde{\alpha}_i \prod_{k<j<i}\left(1-\tilde{\alpha}_j\right)$, the rendered color can be written as

$$\hat{\mathbf{C}}(v,p) = \mathbf{F}_k(v,p) + T_k(v,p)[\tilde{\alpha}_k(v,p)\mathbf{c}_k(v) + (1-\tilde{\alpha}_k(v,p))\mathbf{B}_k(v,p)] \tag{7}$$

In Eq. (7), only $\tilde{\alpha}_k$ depends on the projected center $\hat{p}_k$, whereas F_k, T_k, and B_k are independent of $\hat{p}_k$. Differentiating $\hat{p}_k$ gives

$$\frac{\partial \hat{\mathbf{C}}(v,p)}{\partial \hat{\mathbf{p}}_k} = \alpha_k T_k(v,p) G_k(v,p)\big(\mathbf{c}_k(v) - \mathbf{B}_k(v,p)\big)\boldsymbol{\Sigma}'^{-1}_k(p-\hat{\mathbf{p}}_k) \tag{8}$$

Expanding the L_1 photometric loss in Eq. (19) by the chain rule and denoting the channel-wise residual sign by $r(v,p) = b_p\, \mathrm{sgn}\left(\hat{C}(v,p) - C(v,p)\right)$, the projected-center gradient of Gaussian k over all training observations can be decomposed as

$$\frac{\partial \mathcal{L}_{rgb}}{\partial \hat{\mathbf{p}}_k} = \sum_{v,p} \mathbf{r}\,(v,p) \odot [\alpha_k T_k G_k(\mathbf{c}_k - \mathbf{B}_k)\boldsymbol{\Sigma}'^{-1}_k(p-\hat{\mathbf{p}}_k)] \triangleq \sum_{v,p} \boldsymbol{\Gamma}_k\,(v,p) \tag{9}$$

where $\odot$ denotes channel-wise multiplication. The projected-center gradient is back-propagated to the 3D center through the projection Jacobian $\frac{\partial \hat{p}_k}{\partial \mu_k}$, which does not alter the directional conclusions below.

Equation (9) shows that the direction of the single-observation contribution $\Gamma_k(v,p)$ to the geometric gradient is jointly determined by the color-contrast term $(c_k - B_k)$ and the kernel-offset direction ${\Sigma'_k}^{-1}$(p − $\hat{p}_k$), while its magnitude is modulated by the transmittance T_k, kernel response G_k, and color contrast $\|c_k - B_k\|$. The color-contrast term is therefore the signal carrier of geometric supervision: only when a Gaussian color is distinguishable from the background it occludes can moving the center change the rendered color, allowing the photometric residual to carry geometric information.

Identifiability Criterion under Weak Texture

Based on Eq. (9), we introduce the following assumptions and criterion. Assumption 1 (local color homogeneity): within the reliable observation support of Gaussian k, the color contrast is bounded by $\|c_k(v) - B_k(v,p)\| \le \varepsilon_c$, and ε_c approaches zero in weakly textured regions. Assumption 2 (residual decomposability): the residual sign can be decomposed as $r(v,p) = r_k^\star(v,p) + \eta(v,p)$, where $r_k^\star$ is related to the Gaussian's own geometric error (information term) and η is an interference term induced by external factors such as occlusion, mask-boundary quantization, exposure differences, and background mixing, satisfying $\mathbb{E}[\eta] = 0$ and $\| \eta \| \le 1$.

$$\kappa_k = \frac{\|\mathbf{g}_k^\star\|_2}{\sqrt{\mathbb{E}\left[\|\boldsymbol{\xi}_k\|_2^2\right]}} \tag{10}$$

Proposition 1 (non-identifiability of geometric gradients under weak texture). Under Assumptions 1 and 2, Eq. (9) decomposes as $g_k = g_k^\star + \xi_k$. The magnitude of the information term $\| g_k^\star \| = O\left(\alpha_k \varepsilon_c \sum T_k G_k \left\|{\Sigma'_k}^{-1}(p-\hat{p}_k)\right\|\right)$ decreases linearly to zero with ε_c, whereas the interference term ξ_k is a zero-mean weighted sum whose second moment is determined by the squared observation weights. Define the signal-to-noise ratio κ_k (Eq. (10)) and the interference-mass ratio $\rho_k = \frac{\sum_{(v,p)\notin\Omega_k}\|\Gamma_k(v,p)\|}{\sum_{(v,p)}\|\Gamma_k(v,p)\|}$. When $\varepsilon_c \to 0$, the geometric supervision signal inside the foreground vanishes and the residual gradient is dominated by boundaries and unreliable observations, with $\kappa_k < 1$ and $\rho_k \to 1$. The center-update direction is then determined by the interference field rather than by the Gaussian's own geometric error. Repeated iterations accumulate this bias as directional drift, which macroscopically appears as boundary drift, contour expansion, and floating dark Gaussians.

Proof sketch. From Eq. (9), every summand in $g_k^\star$ contains the factor $(c_k - B_k)$, whose magnitude is upper-bounded by Assumption 1 through $O(\varepsilon_c)$. Each ξ_k in η is zero-mean and bounded; hence, the weighted

sum has zero expectation and an RMS magnitude determined by the second moments of the observation-wise $\|\Gamma_k(v,p)\|$ terms. When $\kappa_k < 1$—that is, when $\|g_k^\star\|$ is smaller than the RMS of ξ_k—the expected angle between a single-step update and the true geometric descent direction is no better than that of a random direction, rendering the gradient direction non-identifiable. This criterion is a sufficient characterization of failure under weak texture: vanishing color contrast is sufficient but not necessary for loss of identifiability, and the failure mechanism in regions dominated by view-dependent appearance (e.g., specular reflection) is outside the scope of this assumption.

Proposition 1 also provides a theoretical explanation for the parameter-routing experiment in Sec. 4.6 (Fig. 10). Completely blocking RGB gradients to geometric parameters removes ξ_k but also discards the boundary information term c_k where $\|B_k - g_k^\star\|$ is large, causing a substantial performance drop. A more appropriate design is therefore to down-weight unreliable observations to reduce the second moment of ξ_k, rather than to sever the gradient pathway. This motivates the three design consequences below.

Consequences for Method Design

Corollary 1 (necessity of supervision filtering and down-weighting). The four VTC criteria remove the major sources of the interference-mass ratio ρ_k in Eq. (9)—out-of-bounds, non-foreground, invisible, and depth-inconsistent observations—thereby directly shrinking the support of ξ_k. The MAD-based consensus weight further down-weights highly dispersed observations, reducing the second moment of ξ_k without suppressing the information term of reliable observations. Corollary 2 (necessity of geometric anchoring). When neighborhood structural confidence is high and the allowed subspace A_k is well defined, Proposition 1 indicates that gradient components in the orthogonal subspace lack information support. Geometric anchoring imposes prior shrinkage in this subspace, effectively constraining the displacement variance induced by ξ_k to the prior scale while preserving information-driven updates in the allowed subspace. Corollary 3 (necessity of stop-gradient). If the consensus target is updated with the training state, $c_k^\star$ absorbs the current rendering error, creating a circular dependency between the residual in Eq. (9) and its supervision target and re-injecting interference into the supervision signal. Fixing the consensus breaks this loop and ensures that the VTC gradient direction is determined only by the cross-view prior.

Proposition 2 (convexity of the VTC appearance subproblem with fixed consensus). Once the consensus target is fixed, ĉ_k(v) is a linear mapping of the spherical-harmonic coefficients h_k (a linear combination of viewing-direction basis functions). Because the Huber function is convex, $\mathcal{L}_{\text{vtc}}$ is convex with respect to the appearance parameter group Θ_a. If the viewing-direction bases of the reliable observations span the corresponding spherical-harmonic subspace, the objective is strictly convex and has a unique minimizer; otherwise, it is flat along null-space directions without introducing additional local minima. Thus, VTC does not create new local minima in the appearance block of the representation, providing a basic property of the joint objective in Eq. (21).

Proposition 3 (MAP interpretation of the geometric anchoring regularizer). Let the displacement k of Gaussian $\Delta\mu_k = \mu_k - \mu_k^0$ be decomposed in the local-PCA basis, with orthogonal component $d_\perp = (I - A_k)\Delta\mu_k$. Assume the prior $d_\perp$ follows a zero-centered generalized Gaussian distribution with precision $w_a \cdot \widehat{\text{conf}}_k$ and exponent p_a, while an uninformative prior is used within the allowed subspace. Under a Gaussian-likelihood approximation, the prior term in the negative log-posterior is proportional to $w_a \widehat{\text{conf}}_k \parallel d_\perp \parallel^{p_a}$, which is precisely the geometric anchoring regularizer in Eq. (29). $p_a = 2$ corresponds to a Gaussian prior, whereas our choice $p_a = 1.65 < 2$ yields a heavy-tailed generalized Gaussian prior that tolerates occasional large but necessary orthogonal corrections and avoids prematurely freezing geometric updates. Structural confidence $\widehat{\text{conf}}_k$ acts as the prior precision: a more reliable neighborhood structure induces stronger shrinkage of orthogonal displacement, while structural uncertainty automatically weakens the prior. This interpretation gives the anchoring weight and exponent a prior-scale meaning rather than treating them as arbitrary hyperparameters.

Visibility-Aware Texture Consensus (VTC)

Reliable Observation Filtering

Before training, the reliable-supervision filtering module establishes a Gaussian-level multi-view observation consensus. For the k-th Gaussian, its initialized center μ_k^0 is projected into the v-th training view using the camera calibration parameters:

$$\mathbf{u}_{k,v} = \phi\left(\mathbf{K}_v(\mathbf{R}_v\boldsymbol{\mu}_k^0 + \mathbf{t}_v)\right) \tag{11}$$

where $\varphi(\cdot)$ denotes perspective division from homogeneous coordinates to 2D image coordinates. Candidate observations are then filtered sequentially using four binary criteria: whether the projected point lies inside the valid image domain (b_{rng}), whether the corresponding pixel belongs to the foreground (b_{fg}), whether the Gaussian has valid projected visibility in the current view (b_{vis}), and whether its projected depth is consistent with the depth-buffer result (b_{dep}). Projected visibility is determined by z-buffer rasterization of the initialization point cloud, and depth consistency requires the relative deviation between the projected depth and z-buffer depth to be no greater than 5%. The reliable-observation indicator for Gaussian v in view k is

$$b_{k,v} = b_{rng}(\mathbf{u}_{k,v}) \wedge b_{fg}(\mathbf{u}_{k,v}) \wedge b_{vis}(k,v) \wedge b_{dep}(k,v) \tag{12}$$

Only when all four conditions are satisfied is the corresponding color observation included in the reliable sample set:

$$\Omega_k = \{\mathbf{c}_{k,v} = I_v(\mathbf{u}_{k,v}) \mid b_{k,v} = 1\} \tag{13}$$

Importantly, reliable-observation determination is completed in advance using the initialized Gaussian centers and the training data and remains fixed throughout training. This design prevents the rendering state, which changes with model parameters, from being reintroduced into the consensus target and thereby preserves the independence and stability of the reliable-supervision statistics.

Multi-View Color Consensus and Consensus Weights

To reduce the influence of local occlusion, abnormal exposure, and erroneous color sampling, the median is computed independently over the RGB channels of the reliable samples, yielding the Gaussian-level color-consensus target:

$$\mathbf{c}_k^\star = \mathrm{median}(\Omega_k) \tag{14}$$

Compared with the arithmetic mean, the channel-wise median is more robust to outlying observations. The finite-sample breakdown point of the sample median is 50%; that is, as long as fewer than half of the observations in the reliable set Ω_k are contaminated by occlusion, incorrect foreground assignment, abnormal exposure, or similar factors, the consensus target $c_k^\star$ remains within the convex hull of the uncontaminated samples and has bounded influence. By contrast, the arithmetic mean has a breakdown point of $1/n_k$, so a single outlier can produce an unbounded shift. We further use the median absolute deviation to quantify the dispersion of multi-view color observations:

$$MAD_k = \mathrm{median}_{\mathbf{c}\in\Omega_k} \parallel \mathbf{c} - \mathbf{c}_k^\star \parallel_1 \tag{15}$$

A smaller MAD_k indicates that Gaussian k exhibits a stable color response across different training views, whereas a larger MAD_k suggests that the region may be affected by occlusion, boundary mixing, view-dependent reflection, or texture degradation. Let the number of effective observations be

$$n_k = |\Omega_k| \tag{16}$$

The Gaussian-level consensus weight is jointly determined by sample sufficiency and color stability. We use the following explicit form:

$$\omega_k = \min\left(1, \frac{n_k}{n_{min}}\right) \frac{1}{1+\left(\frac{MAD_k}{\tau_m}\right)^2} \tag{17}$$

where $n_{\min}$ is the minimum sample-count threshold and τ_m is the color-stability scale parameter after normalizing color channels to [0,1]. The weight is monotonically non-decreasing with the effective sample count and monotonically non-increasing with color dispersion. When n_k is below $n_{\min}$ or MAD_k exceeds τ_m, the corresponding Gaussian receives only a small weight and is excluded from strong appearance supervision. Hyperparameter values are given in Sec. 4.3. The formulation also guarantees $\omega_k \in [0,1]$; consequently, the influence of any single observation on the VTC gradient is upper-bounded by ω_k and the Huber saturation value. This bounded-influence property is consistent with Corollary 1 in Sec. 3.2.3, which seeks to reduce the second moment of the interference term.

VTC Appearance-Consistency Loss

Let ĉ_k denote the color estimate obtained from the current Gaussian appearance parameters during training. The VTC appearance-consistency loss is defined as

$$\mathcal{L}_{vtc} = \frac{1}{N}\sum_{k=1}^{N} \omega_k \, \rho\big(\hat{\mathbf{c}}_k - \mathrm{sg}(\mathbf{c}_k^{\star})\big) \tag{18}$$

where $\rho(\cdot)$ is the Huber robust penalty and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. Stop-gradient ensures that the precomputed consensus color is used only as a fixed supervision target and is not updated by the current model state during back-propagation. VTC therefore separates the stable color statistics obtained before training from the instantaneous photometric residuals produced during training, preventing current rendering errors from feeding back into the consensus target. The resulting Gaussian-level statistics are used, on the one hand, to weight the appearance-consistency loss, which acts only on the appearance parameter group θ_a, and, on the other hand, to contribute through support mixing to the structural-confidence computation in Sec. 3.5.2. The VTC data flow is illustrated in Fig. 3. As stated in Proposition 2 of Sec. 3.2.3, with a fixed consensus target this loss is convex with respect to the appearance parameter group and introduces no new local minima in the appearance block.

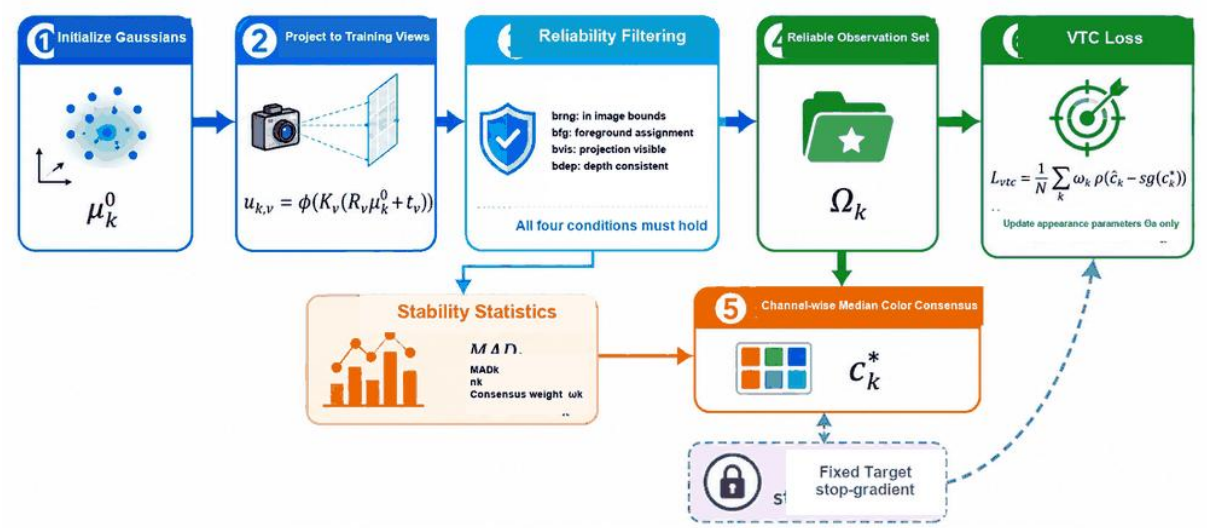


Fig. 3 Pre-training reliable-observation filtering, channel-wise median color consensus, and MAD-based stability estimation in VTC

Consensus-Inheritance Extension for Adaptive Densification Backends

The default configuration builds the consensus statistics once before training, which is consistent with the fixed-Gaussian protocol. Cloning and splitting in the standard 3DGS backend alter the Gaussian population, breaking the correspondence between a fixed consensus and newly created primitives. We therefore implement a candidate consensus-inheritance extension and calibrate ρ_{inh}, the refresh interval T_r, and the refresh threshold κ_r using a 3×3×3 full-factorial grid on Drums with seed 17 for 3,000 iterations. The extension is defined as follows.

(1) Split/clone inheritance: a newly created Gaussian inherits the parent consensus color $c^{\star}$ and a decayed weight $\omega_{\mathrm{child}} = \rho_{\mathrm{inh}}\,\omega_{\mathrm{parent}}$; $\rho_{\mathrm{inh}} \in (0,1)$ controls inter-generation decay of the prior, with 0.5 used as a candidate value. (2) Contamination-resistant periodic re-estimation: every T_r iterations (candidate value: 1,000), the current centers are reprojected and a candidate consensus $\hat{c}^{\star}_{\mathrm{new}}$ is recomputed. The update is accepted only when the channel-wise deviation satisfies $\|\hat{c}^{\star}_{\mathrm{new}} - c^{\star}\|_{\infty} \le \kappa_r$ MAD (candidate $\kappa_r = 3.0$); otherwise, the previous consensus is retained. This prevents current rendering errors from being written back into the supervision target through re-estimation and preserves the anti-circular-dependency property required by Corollary 3 in Sec. 3.2.3. (3) Opacity-reset synchronization: after an opacity-reset event, the consensus weights of affected Gaussians are decayed in proportion to the reset magnitude.

The purpose of this candidate extension is to allow consensus statistics to evolve with the Gaussian population while retaining the contamination-resistant structure of a fixed reliable prior coupled with dynamic parameter optimization. Section 4.7 reports the complete results of the 27-run grid and the training-topology audit. Because an equal-budget standard-densification Control and a formal four-scene × three-seed 30k matrix are currently unavailable, this experiment supports parameter calibration and implementation feasibility only and does not establish a performance gain from consensus inheritance.

Optimization-Level Geometry–Appearance Parameter Routing

In image space, we retain the basic supervision of standard 3DGS: the RGB term uses the L_1 photometric loss, while the foreground-alpha term uses pixel-wise mean-squared error. Define

$$\mathcal{L}_{rgb} = \frac{\sum_{v,p} b_p \|\hat{\mathbf{C}}(v,p) - \mathbf{C}(v,p)\|_1}{\sum_{v,p} b_p + \varepsilon} \tag{19}$$

$$\mathcal{L}_{msk} = \frac{1}{N_v P} \sum_{v,p} \left(\hat{A}(v,p) - M(v,p) \right)^2 \tag{20}$$

where $b_p \in \{0,1\}$ indicates whether pixel p belongs to the valid supervision region, ε is a stabilizing term that prevents division by zero, and P is the number of pixels in one image. The joint training objective is

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_d \mathcal{L}_{dssim} + \lambda_m \mathcal{L}_{msk} + \lambda_v \mathcal{L}_{vtc} + \lambda_a \mathcal{L}_{anchor} \tag{21}$$

where $\mathcal{L}_{\text{dssim}}$ is an optional structural-similarity loss ($\lambda_d = 0$ in the fixed-Gaussian prototype, while the standard 3DGS backend retains the original implementation with $\lambda_d = 0.2$), and $\mathcal{L}_{\text{anchor}}$ is the geometric anchoring regularizer defined in Sec. 3.5.3. The routing rules are as follows. The basic RGB loss, optional DSSIM term, and foreground-mask loss back-propagate through the standard rendering path to both parameter groups Θ_g and Θ_a. The VTC appearance-consistency loss acts only on Θ_a. Geometric anchoring and local structural regularization act only on Θ_g. The center-direction operator in Sec. 3.5.4 separately processes the total gradient of the center parameters after back-propagation. This routing leaves the Gaussian representation and alpha-compositing equations unchanged and realizes optimization-level geometry–appearance decoupling by restricting the target parameter groups of each loss and the location of gradient processing. The default configuration retains the full photometric gradient; changes in the strength of parameter-level gradient routing are used only as mechanism-analysis settings in Sec. 4.6.

Local Structural Statistics and Geometric Anchoring

Local-PCA Coordinate System

For each initialized center μ_k^0, let K denote the index set of its $\mathcal{N}_k$ nearest neighbors. All neighborhoods are constructed once from the shared initialized centers and remain fixed during training. To prevent VTC statistics from altering the local coordinate system, the PCA covariance is computed without weights; the VTC consensus weights are used only later for support mixing and regularization weighting. The local covariance matrix is defined as

$$\mathbf{C}_k = \frac{1}{|\mathcal{N}_k| - 1} \sum_{j \in \mathcal{N}_k} \left(\boldsymbol{\mu}_j^0 - \bar{\boldsymbol{\mu}}_k^0 \right) \left(\boldsymbol{\mu}_j^0 - \bar{\boldsymbol{\mu}}_k^0 \right)^{\mathsf{T}} \tag{22}$$

We eigendecompose C_k and sort the eigenvalues in ascending order:

$$\mathbf{C}_k \mathbf{v}_i = \lambda_i \mathbf{v}_i, \qquad 0 \le \lambda_1 \le \lambda_2 \le \lambda_3 \tag{23}$$

The corresponding unit eigenvectors v_1, v_2, and v_3 form the local-PCA coordinate system. The minimum-eigenvalue direction v_1 is treated as a local-normal candidate, and the maximum-eigenvalue direction v_3 as the principal-axis candidate for slender structures. Rather than hard-classifying a neighborhood as line-like or plane-like, we compute linearity and planarity scores from the eigenvalue spectrum:

$$lin_k = \frac{\lambda_3 - \lambda_2}{\lambda_3 + \varepsilon}, \qquad pla_k = \frac{\lambda_2 - \lambda_1}{\lambda_3 + \varepsilon} \tag{24}$$

The structural mixing coefficient β_k is determined by the relative magnitudes of these two scores:

$$\beta_k = \frac{lin_k}{lin_k + pla_k + \varepsilon} \tag{25}$$

As β_k approaches 1, the neighborhood becomes more line-like; as it approaches 0, the neighborhood becomes more plane-like. We therefore construct the allowed-direction projection matrix

$$\mathbf{A}_k = \beta_k \mathbf{v}_3 \mathbf{v}_3^{\mathsf{T}} + (1 - \beta_k)\left(\mathbf{v}_2 \mathbf{v}_2^{\mathsf{T}} + \mathbf{v}_3 \mathbf{v}_3^{\mathsf{T}} \right) \tag{26}$$

so that line-like neighborhoods primarily preserve the principal-axis direction, plane-like neighborhoods primarily preserve the tangent-plane directions, and intermediate values provide a continuous interpolation between the two projection types.

Structural Confidence

Structural confidence measures whether the current PCA basis can reliably represent the true local structure by combining directional saliency, eigenvalue anisotropy, and observation support:

$$conf_k = [\max(lin_k, pla_k)]^{\gamma} \frac{\lambda_3 - \lambda_1}{\lambda_3 + \varepsilon} \tag{27}$$

$$\widehat{conf}_k = (1-\eta) conf_k + \eta \omega_k \tag{28}$$

where γ is a confidence exponent controlling the shape of the confidence distribution, and η is a support-mixing coefficient that incorporates the VTC consensus weight ω_k into confidence modulation without changing the unweighted covariance or its eigenvectors. Neighborhoods with strong consensus and a clear structural spectrum therefore receive high $\widehat{\mathrm{conf}}_k$, whereas weakly supported or isotropic neighborhoods automatically reduce the constraint strength. When $\widehat{\mathrm{conf}}_k$ falls below the structural threshold τ_c, the associated constraints revert to the standard 3DGS form, preventing unreliable coordinate systems from imposing erroneous restrictions.

Geometric Anchoring Regularizer

Using the initialized center as a reference, geometric anchoring penalizes excessive center displacement in local orthogonal directions while preserving necessary positional adjustments within the allowed subspace:

$$\mathcal{L}_{anchor} = \frac{1}{N} \sum_{k=1}^{N} w_a \widehat{conf}_k \| (\mathbf{I} - \mathbf{A}_k)(\boldsymbol{\mu}_k - \boldsymbol{\mu}_k^0) \|_2^{p_a} \tag{29}$$

The anchoring strength adapts to structural confidence. When the neighborhood direction is well defined, orthogonal displacement is constrained more strongly; when local structure is uncertain, the constraint is automatically weakened, reducing the influence of erroneous PCA directions on geometric optimization. This design also has a probabilistic interpretation. As stated in Proposition 3 of Sec. 3.2.3, Eq. (29) is equivalent to MAP estimation with a generalized Gaussian prior of precision $w_a \cdot \widehat{\mathrm{conf}}_k$ and exponent p_a imposed on the orthogonal subspace, with structural confidence acting as the prior precision.

Center-Gradient Direction Operator (Mechanism-Analysis Tool)

After back-propagation of the joint loss, we read the total gradient of the center of the k-th Gaussian:

$$\mathbf{g}_k = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} \tag{30}$$

Before the optimizer update and before estimating first- and second-order moments, a directional operator can be applied:

$$\tilde{\mathbf{g}}_k = \mathbf{D}_k \mathbf{g}_k, \qquad \mathbf{D}_k = \mathbf{A}_k + \kappa\big(\widehat{conf}_k\big)(\mathbf{I} - \mathbf{A}_k) \tag{31}$$

$$\kappa\big(\widehat{conf}_k\big) = \begin{cases} 1, & \widehat{conf}_k < \tau_c, \\ \kappa_0, & \widehat{conf}_k \geq \tau_c, \end{cases} \tag{32}$$

where $0 \leq \kappa_0 < 1$ is the retention coefficient for the orthogonal direction. When $\widehat{\mathrm{conf}}_k < \tau_c$, D_k reduces to the identity matrix and the center gradient is fully retained. Otherwise, the component in the allowed subspace is unchanged, whereas the orthogonal component is scaled by κ_0. The lower bound κ_0 preserves sufficient correction capacity for initialization errors and prevents directional projection from freezing geometry too early. Because this operation is applied before the momentum updates of adaptive optimizers such as Adam, the direction-dependent retention ratios affect both first- and second-order statistics. Importantly, the mechanism experiments in Sec. 4.6 show that directional projection does not stably translate into reconstruction-quality gains under the current protocol. The directional operator is therefore disabled in the default configuration; local-PCA statistics are used only for geometric anchoring and confidence weighting, while the directional operator is retained solely as a mechanism-analysis tool for characterizing directional behavior of geometric updates.

Scene-Level Constraint Coefficient

The severity of texture degradation varies substantially across scenes, so a uniform constraint strength cannot accommodate all cases. We therefore define a scene-level constraint coefficient (SCC), which estimates scene degradation from statistics of the training views and uniformly scales the loss weights of VTC, geometric anchoring, and candidate modules. Let τ_g denote the proportion of low-texture pixels in the foreground whose gradient magnitude is below threshold r_{lt}, let r_{bd} denote the ratio of foreground-

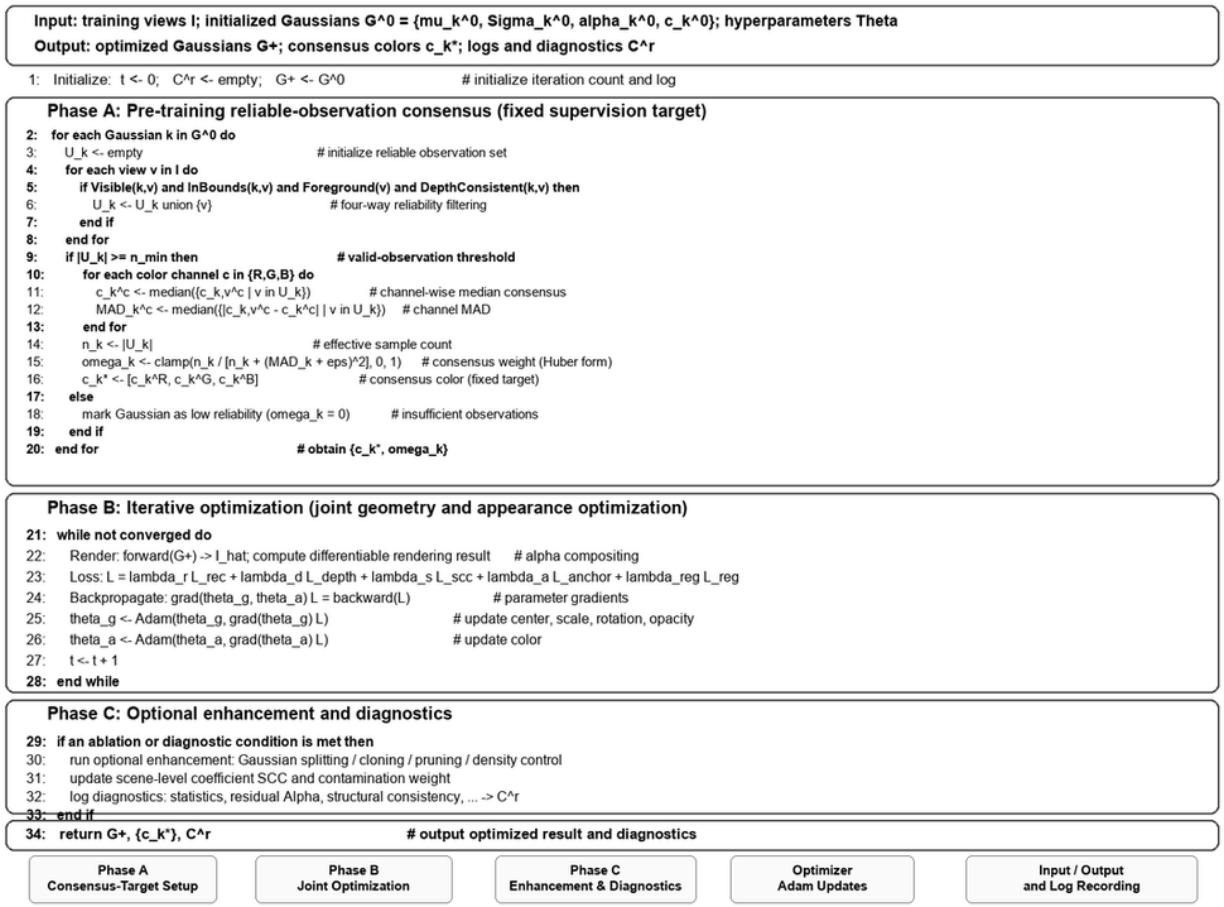
mask boundary pixels to all foreground pixels, and let c_{fb} denote the mean color contrast between the foreground and background regions. Then

$$s = \mathrm{clip}\left(a_1 \hat{r}_{lt} + a_2 \hat{r}_{bd} + a_3\left(1 - \hat{c}_{fb}\right),\, s_{min},\, 1\right) \tag{33}$$

where $\hat{r}_{\mathrm{lt}}$, $\hat{r}_{\mathrm{bd}}$, and $\hat{c}_{\mathrm{fb}}$ are the corresponding statistics normalized using fixed reference intervals. The actual configuration uses $a_1 : a_2 : a_3 = 0.65 : 0.20 : 0.15$ and $s_{\min} = 0.35$, and all statistics are computed only from the training split. The degradation score of Drums reaches the clipping upper bound, so scanning $s_{\min}$ does not change its constraint coefficient. Within the tested range, the PSNR span for Lego and Chair is no greater than 0.025 dB, and the weight-based scan remains below 0.033 dB.

Optimization Procedure

Algorithm 1 summarizes one joint-optimization iteration. Directional constraints act on center gradients before the optimizer update, whereas the candidate residual-shrinkage module handles remaining outside-mask leakage after the parameter update. VTC statistics and local-PCA statistics remain fixed throughout training; only Gaussian parameters and rendering states evolve with optimization, forming a closed loop of "fixed reliable prior–dynamic parameter optimization." On the standard densification backend, the consensus-inheritance extension in Sec. 3.3.4 is enabled to maintain correspondence between consensus statistics and the evolving Gaussian population, while the remaining procedure is unchanged.



VTC and local-PCA statistics are computed before training with complexities of $O(NN_v)$ and $O(NK)$, respectively. During training, the method performs per-Gaussian weight lookups and 3×3 matrix multiplications. Training-stage overhead is discussed in Sec. 4.7 together with the extended-budget experiments.

Experiments

Datasets and Evaluation Protocol

Experiments are conducted on four NeRF Synthetic scenes: Drums, Lego, Chair, and Hotdog. Drums contains representative low-texture drumheads, slender stands, and dark occlusions; Lego and Chair mainly contain weakly textured surfaces and occlusion boundaries; and Hotdog contains smooth surfaces, foreground boundaries, and local highlights. Together, these scenes cover the weak textures, similar or repetitive textures, dark occlusions, boundary mixing, and slender structures targeted in this work.

For each scene, one out of every five views from the original 100 training views is selected at fixed indices as heldout_strict, and the remaining 80 views form train_strict. Occupancy-silhouette initialization uses train_strict only and generates 29,148, 54,726, 35,496, and 42,014 initial Gaussian centers for Drums, Lego, Chair, and Hotdog, respectively, at voxel resolution 160. Held-out views are excluded from initialization, training, threshold selection, and model selection, eliminating evaluation leakage at the data-path level.

The main experiments use a fixed-Gaussian protocol without densification or pruning so that the effects of method modules can be isolated under the same initialization. The four-scene paired matrix at 300 iterations uses random seeds 17, 23, 41, 53, and 61. The official-validation lockbox and extended-budget checks separately use three seeds or a single seed and are reported according to their respective protocols. Extended-budget trends are shown in Fig. 11, and the 30,000-iteration check on the standard densification backend for Drums is reported in Table 14.

For the controllable-degradation benchmark, texture removal, illumination reduction, and occluding-panel perturbations are applied separately to 50% of the train_strict views in each of the four scenes, with low, medium, and high severity levels for each degradation type. Alpha, camera parameters, depth, and held-out images remain unchanged, and all leakage audits pass. The macro-average proxy distributions over the four scenes are reported in Table 1, and the level-wise paired results for Drums with seed 17 are reported in Table 2.

DTU scan1/scan6 under lighting condition 3 and Mip-NeRF 360 room/counter are used only for real-data pipeline validation. DTU uses a 42/7 train–held-out split with SIFT triangulation from training images and official calibration, whereas Mip-NeRF 360 uses an 80/20 split and the official COLMAP sparse points. The intersections between all data paths are zero. The 10-iteration results in Table 3 verify only the data, camera, and training interfaces and are not included in cross-dataset performance comparisons.

Table 1 $\hat{\varepsilon}_c$ proxy distributions under three types of controllable degradation at three severity levels (macro-average over four scenes)

Degradation type	Severity	Control parameter	Mean $\hat{\varepsilon}_c$	Median $\hat{\varepsilon}_c$	$\Pr(\hat{\varepsilon}_c < 0.02)$
Texture removal	Low	$\rho = 0.75$	0.0533	0.0288	0.445
Texture removal	Medium	$\rho = 0.5$	0.0408	0.0231	0.485
Texture removal	High	$\rho = 0.25$	0.0291	0.0175	0.548
Illumination reduction	Low	$a = 0.8$	0.0528	0.0276	0.458
Illumination reduction	Medium	$a = 0.6$	0.0397	0.0208	0.517
Illumination reduction	High	$a = 0.4$	0.0266	0.0139	0.607
Occluding panel	Low	$q = 0.08$	0.0611	0.0281	0.471
Occluding panel	Medium	$q = 0.16$	0.0534	0.0197	0.542
Occluding panel	High	$q = 0.28$	0.0415	0.0067	0.651

Table 2 Paired Drums results under three degradation types and three severity levels (300 iterations, seed 17)

Degradation	Severity	Control PSNR↑	Simplified PSNR↑	ΔPSNR ↑	ΔSSIM ↑	ΔLPIPS ↓	ΔIoU	ΔOutside
Texture removal	Low	15.119	15.141	+0.0220	+0.00201	-0.00084	+0.00107	-3.95e-05

Degradation	Severity	Control PSNR↑	Simplified PSNR↑	ΔPSNR ↑	ΔSSIM ↑	ΔLPIPS ↓	ΔIoU	ΔOutside
Texture removal	Medium	15.119	15.141	+0.0224	+0.00206	-0.00093	+0.00109	-4.05e-05
Texture removal	High	15.118	15.141	+0.0224	+0.00212	-0.00117	+0.00111	-4.03e-05
Illumination reduction	Low	15.107	15.128	+0.0210	+0.00246	-0.00083	+0.00133	-4.49e-05
Illumination reduction	Medium	15.013	15.029	+0.0155	+0.00240	-0.00053	+0.00137	-4.70e-05
Illumination reduction	High	14.950	14.959	+0.0094	+0.00253	-0.00068	+0.00127	-4.88e-05
Occluding panel	Low	15.127	15.148	+0.0205	+0.00190	-0.00003	+0.00103	-3.98e-05
Occluding panel	Medium	15.118	15.136	+0.0184	+0.00183	+0.00018	+0.00089	-4.02e-05
Occluding panel	High	15.085	15.100	+0.0149	+0.00188	+0.00036	+0.00078	-3.40e-05

Table 3 DTU and Mip-NeRF 360 data adaptation and interface validation

Dataset	Scene	Train/held-out	Initialization point cloud	Initial points	Interface validation (10 iter)
DTU	scan1, light3	42/7	SIFT triangulation from training images	11 957	PSNR↑ 8.6530 ; SSIM↑ 0.1903
DTU	scan6, light3	42/7	SIFT triangulation from training images	15 515	Adapter passed; no formal training result
Mip-NeRF 360	room	80/20	Official COLMAP sparse points	60 000	PSNR 13.6356 ; SSIM 0.2794
Mip-NeRF 360	counter	80/20	Official COLMAP sparse points	60 000	Adapter passed; no formal training result

Evaluation Metrics

Overall reconstruction quality is evaluated using PSNR, SSIM, and LPIPS, with LPIPS using the VGG backbone from the official 3DGS implementation. Contour quality is measured by Alpha IoU and Outside alpha. Gaussian-level counts of dark splats, very-dark splats, and high-chroma outliers are used to localize failure modes. Global metrics are used for performance comparison, whereas local metrics are used only for mechanism diagnosis; the two are not interchangeable.

(1) Alpha IoU. A soft intersection-over-union is used to measure the agreement between predicted foreground alpha and the ground-truth mask:

$$IoU\big(\hat{A}, M\big) = \frac{\sum_p \min\big(\hat{A}(v,p), M(v,p)\big)}{\sum_p \max\big(\hat{A}(v,p), M(v,p)\big)} \tag{34}$$

(2) Outside-mask alpha leakage (Outside alpha). This metric measures the amount of opacity mass leaking into the background region; lower is better:

$$Outside\big(\hat{A}, M\big) = \frac{1}{P}\sum_p \hat{A}\,(v,p)\big(1 - M(v,p)\big) \tag{35}$$

(3) Gaussian-level diagnostic metrics. Using the relative luminance $\ell(\mathrm{c})$ of the Gaussian consensus color (Rec. 601 luminance coefficients) and effective opacity as criteria, we define counts of dark splats, very-dark splats, and high-chroma outliers:

$$N_{dark} = |\{k \mid \ell(\mathbf{c}_k^\star) < 0.05,\ \bar{\alpha}_k > 0.5\}|, \quad N_{vdark} = |\{k \mid \ell(\mathbf{c}_k^\star) < 0.02,\ \bar{\alpha}_k > 0.5\}| \tag{36}$$

$$N_{chroma} = |\{k \mid \mathrm{sat}(\mathbf{c}_k^\star) > 0.8,\ \| \mathbf{c}_k^\star - \bar{\mathbf{c}}_k^\star(\mathcal{N}_k) \|_1 > 0.3\}| \tag{37}$$

where $\mathrm{sat}(\cdot)$ is HSV saturation, $c_k^\star(\mathcal{N}_k)$ is the mean neighborhood consensus color, and $\bar{\alpha}_k$ is the mean effective opacity over training views. Thresholds are fixed across all scenes and configurations. The 200 RGBA and depth views from the public Blender test split are used for depth-proxy evaluation. Table 4 reports static opacity-proxy counts from the exported PLY; this metric does not replace the view-averaged effective opacity.

Table 4 shows that the very-dark-splat count is zero for every scene and configuration. Dark-splat and high-chroma counts also do not decrease consistently across scenes. Therefore, the Gaussian-level diagnostics characterize scene-specific differences in anomaly distributions but do not support a claim that the overall number of anomalous Gaussians is reduced by 50%–65%.

Table 4 Gaussian-level diagnostic counts over four scenes (mean over three seeds)

Scene	Configuration	N_{dark}	N_{vdark}	N_{chroma}	Replication unit
Drums	Control	0.00	0.00	59.00	3 seeds
Drums	Simplified	0.00	0.00	59.33	3 seeds
Lego	Control	38.67	0.00	154.67	3 seeds
Lego	Simplified	44.67	0.00	154.00	3 seeds
Chair	Control	0.00	0.00	0.00	3 seeds
Chair	Simplified	0.00	0.00	0.00	3 seeds
Hotdog	Control	0.00	0.00	11.00	3 seeds
Hotdog	Simplified	0.00	0.00	12.33	3 seeds

Implementation Details and Hyperparameters

All configurations use the same initialization, data split, resolution, and optimizer. Table 5 lists the actual runtime parameters: $n_{\min} = 18$, $\tau_m = 0.138$, $K = 18$, and $\lambda_m/\lambda_v/\lambda_a = 0.20/0.070/0.020$; the SCC weights are 0.65:0.20:0.15, with $s_{\min} = 0.35$.

Table 6 reports ±50% endpoint scans of seven key hyperparameters. The experiments fix the Drums scene, 300 iterations, and the Gaussian count; each endpoint is evaluated with seeds 17, 23, and 41, for a total of 42 training runs. For each seed, the metrics are first averaged over the 20 heldout_strict views, after which the cross-seed mean and sample standard deviation are computed.

Table 5 Main hyperparameter settings

Category	Parameter	Value	Description
VTC	$n_{\min}$	18	Minimum effective sample count used in actual runs
VTC	τ_m	0.138	Color-stability threshold used in actual runs
VTC	Depth-consistency tolerance	0.030	Projected-depth consistency tolerance
VTC	λ_v	0.070	Texture-consensus loss weight
Local PCA	K	18	K-nearest-neighbor neighborhood size
Local PCA	$\gamma/\tau_c/\eta$	0.82 / 0.25 / 0.34	Confidence exponent, structural threshold, and support mixing
Local PCA	w_a/p_a	0.36 / 1.65	Normal weight and loss exponent
Loss weights	$\lambda_m/\lambda_v/\lambda_a$	0.20 / 0.070 / 0.020	Mask, VTC, and local-PCA weights
SCC	$\omega_b{:}\,\omega_t{:}\,\omega_c/s_{\min}$	0.65:0.20:0.15 / 0.35	Boundary, texture, and contrast weights and lower bound
SCC	Score normalization	clip((r−0.22)/0.14)	All statistics use train_strict only
Candidate inheritance	$\rho_{\mathrm{inh}}/T_r/\kappa_r$	0.5 / 1000 / 3.0	Design candidate; systematic validation not yet completed
Fixed Gaussian	Iteration budget	300 / 1200 / 3000	300 iterations for the four-scene main matrix; longer budgets only partially evaluated
Standard backend	Iteration budget	7000 / 30000	Single-seed VTC transfer check on Drums
Main experiment	Random seeds	17 / 23 / 41 / 53 / 61	Four-scene pairing at 300 iterations; scene × seed is the replication unit
Initialization	Voxel resolution	160	Constructed using train_strict only

Table 6 ±50% endpoint scan of seven key hyperparameters (Drums, 300 iterations; mean ± sample standard deviation over three seeds)

Parameter	Endpoint (value)	PSNR↑/dB	SSIM↑	LPIPS↓	Alpha IoU↑	Outside alpha↓
τ_m	0.5× (0.069)	15.1357±0.0029	0.66319±0.00058	0.37131±0.00038	0.62053±0.00031	7.4758e-04±1.55e-05
τ_m	1.5× (0.207)	15.1429±0.0041	0.66309±0.00070	0.37181±0.00039	0.62072±0.00031	7.3898e-04±1.81e-05
γ	0.5× (0.41)	15.1421±0.0045	0.66335±0.00073	0.37042±0.00041	0.62029±0.00033	7.4993e-04±1.82e-05
γ	1.5× (1.23)	15.1421±0.0045	0.66334±0.00073	0.37043±0.00042	0.62028±0.00034	7.4971e-04±1.82e-05
η	0.5× (0.17)	15.1421±0.0045	0.66334±0.00073	0.37043±0.00042	0.62029±0.00033	7.4990e-04±1.82e-05
η	1.5× (0.51)	15.1421±0.0045	0.66334±0.00073	0.37042±0.00042	0.62029±0.00033	7.4974e-04±1.82e-05
τ_c	0.5× (0.125)	15.1421±0.0045	0.66335±0.00073	0.37042±0.00041	0.62030±0.00033	7.4991e-04±1.82e-05
τ_c	1.5× (0.375)	15.1421±0.0045	0.66334±0.00073	0.37043±0.00042	0.62028±0.00034	7.4968e-04±1.82e-05
w_a	0.5× (0.18)	15.1421±0.0045	0.66334±0.00073	0.37043±0.00041	0.62029±0.00033	7.4994e-04±1.82e-05
w_a	1.5× (0.54)	15.1422±0.0045	0.66334±0.00074	0.37042±0.00041	0.62029±0.00034	7.4971e-04±1.82e-05
p_a	0.5× (0.825)	15.1421±0.0045	0.66334±0.00073	0.37043±0.00042	0.62029±0.00034	7.4991e-04±1.82e-05
p_a	1.5× (2.475)	15.1421±0.0045	0.66334±0.00073	0.37043±0.00041	0.62030±0.00034	7.4978e-04±1.82e-05
$n_{\min}$	0.5× (9)	15.1448±0.0043	0.66339±0.00074	0.37050±0.00046	0.62009±0.00033	7.4770e-04±1.80e-05
$n_{\min}$	1.5× (27)	15.1366±0.0042	0.66299±0.00068	0.37042±0.00028	0.62049±0.00031	7.5027e-04±1.79e-05

The scan shows that n_{min} has the largest endpoint PSNR span (0.00819 dB), followed by τ_m (0.00728 dB), while each of the other five parameters remains below 0.00005 dB. The maximum spans of SSIM, LPIPS, Alpha IoU, and Outside alpha are 0.000394, 0.000499, 0.000406, and 2.57×10^{-6}, respectively. These results indicate good local stability around the default settings under the 300-iteration fixed-Gaussian protocol on Drums.

Overall Performance Comparison

In the equal-budget three-seed comparison on Drums (Table 7), the simplified configuration improves PSNR from 15.120 to 15.142 dB and SSIM from 0.6613 to 0.6633, reduces LPIPS from 0.3711 to 0.3704, increases Alpha IoU from 0.6192 to 0.6203, and reduces Outside alpha from 0.000789 to 0.000750 relative to Control. Because 300 iterations are insufficient for full convergence, these results are used to evaluate relative changes under identical initialization rather than to represent the final reconstruction ceiling.

Table 7 Three-seed results for the fixed-Gaussian Drums prototype (mean ± sample standard deviation)

Configuration	PSNR↑/dB	SSIM↑	LPIPS↓	Alpha IoU↑	Outside↓
Control	15.120±0.004	0.6613±0.0007	0.3711±0.0004	0.6192±0.0003	0.000789±0.000018
Simplified default	15.142±0.004	0.6633±0.0007	0.3704±0.0004	0.6203±0.0003	0.000750±0.000018

The cross-scene results show the same overall direction (Table 8). PSNR increases by 0.0221, 0.0582, 0.0952, and 0.0909 dB on Drums, Lego, Chair, and Hotdog, respectively. SSIM, Alpha IoU, and Outside alpha improve on all four scenes, while LPIPS decreases on the first three scenes and is essentially unchanged on Hotdog.

Table 8 Same-seed paired differences for the fixed-Gaussian prototype over four scenes (Variant − Control)

Scene	PSNR ↑ C → S/dB	ΔPSNR ↑/dB	ΔSSIM ↑	ΔLPIPS ↓	ΔIoU	ΔOutside
Drums	15.120→15.142	0.0221	0.00200	−0.00062	0.00105	−0.0000391
Lego	19.182→19.241	0.0582	0.00229	−0.00189	0.00127	−0.0000618
Chair	20.496→20.591	0.0952	0.00262	−0.00020	0.00142	−0.0000398
Hotdog	20.515→20.606	0.0909	0.00234	0.00005	0.00173	−0.0001583

Table 9 Statistical levels and aggregation procedure for strict evaluation

Statistical level	Sample size	Computation	Reported quantity

Statistical level	Sample size	Computation	Reported quantity
View level	20 views per seed	Compute five metrics per view	PSNR↑、SSIM↑、LPIPS↓、Alpha IoU↑、Outside alpha↓
Seed level	5 seeds (17, 23, 41, 53, 61)	Average over 20 views for each seed first	Five seed means per configuration
Configuration summary	$n = 5$	Arithmetic mean and sample standard deviation over the five seed means	Mean ± sample standard deviation
Paired effect	$n = 5$ paired samples	Exact two-sided Wilcoxon signed-rank test within each scene	p value and paired rank-biserial correlation r_{rb}
Inference boundary	$n = 5$	Exploratory effect-size analysis only	No significance or generalization claims

Table 10 summarizes verifiable baselines under different backends and training budgets to clarify the experimental coverage. Standard 3DGS at 30k covers all four scenes with three seeds, the official 2DGS backend covers only Drums with a single seed at 3k, and Edge-fixed uses the 300-iteration fixed-Gaussian diagnostic protocol. Because the training budgets and backend settings differ, these results are not directly combined into a single ranking.

Table 10 Verifiable baseline results and protocol coverage

Method/protocol	Scene	PSNR↑	SSIM↑	LPIPS↓	Budget/seeds	Coverage note
Standard 3DGS	Drums	28.779±0.001	0.96174	0.02962	30k / n=3	Verifiable standard-backend baseline
Standard 3DGS	Lego	38.068±0.029	0.99073	0.00698	30k / n=3	Verifiable standard-backend baseline
Standard 3DGS	Chair	40.352±0.034	0.99608	0.00327	30k / n=3	Verifiable standard-backend baseline
Standard 3DGS	Hotdog	41.144±0.012	0.99054	0.00848	30k / n=3	Verifiable standard-backend baseline
Official 2DGS backend	Drums	25.531	0.937	0.073	3k / n=1	Drums only; different budget
Edge-fixed	Drums	21.729	0.810	0.155	300 / n=1	Diagnostic upper bound; not 30k
Edge-fixed	Lego	22.800	0.801	0.156	300 / n=1	Diagnostic upper bound; not 30k
Edge-fixed	Chair	22.959	0.827	0.106	300 / n=1	Diagnostic upper bound; not 30k

The five-seed paired matrix (Fig. 5) provides a stricter test of directional consistency. PSNR improves for 5/5 seeds in each scene; the exact two-sided Wilcoxon signed-rank test gives p=0.0625 with a rank-biserial correlation of +1.000. We therefore interpret this result as directional consistency rather than statistical significance.

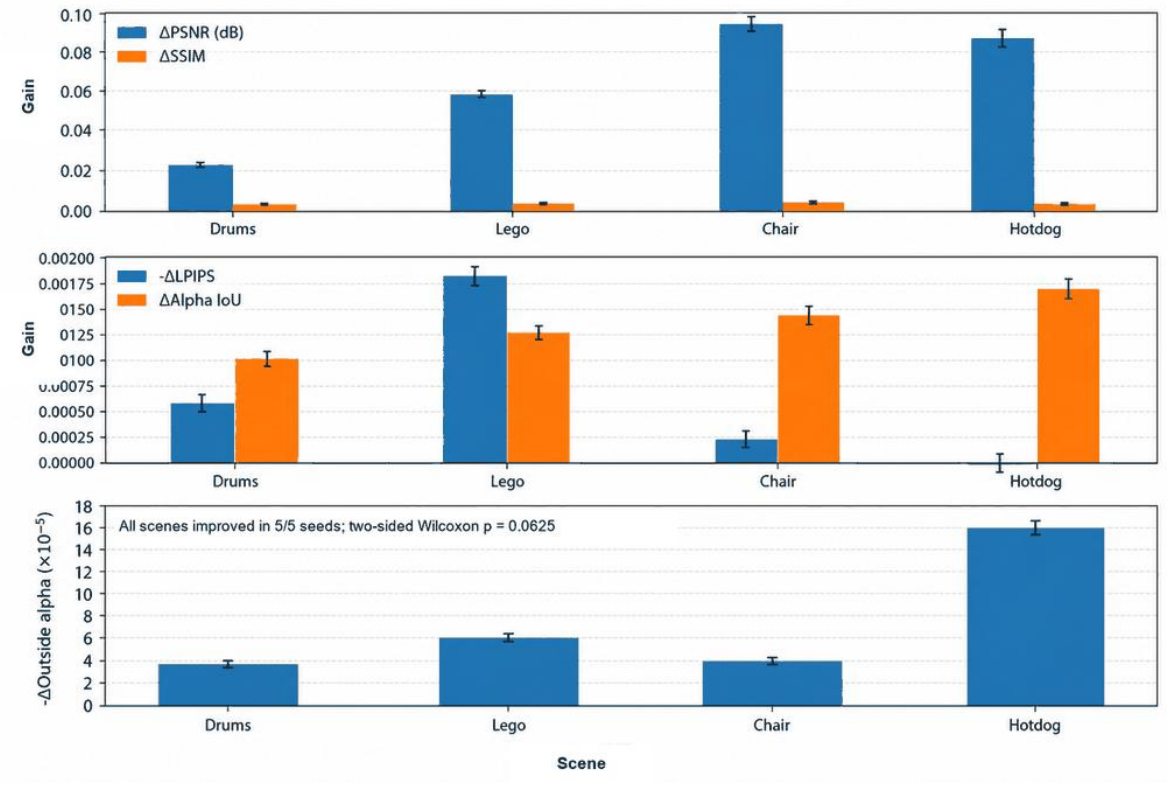


Fig. 5 Overview of five-seed paired gains over four scenes (300 iterations; error bars show the sample standard deviation over five seeds)

The SCC scan is reported in Table 11. The degradation score of Drums reaches the clipping upper bound, so its constraint coefficient remains unchanged. For Lego and Chair, the PSNR spans are no greater than 0.025 and 0.033 dB, respectively, and Alpha IoU remains essentially stable. This experiment shows that SCC does not induce pronounced performance fluctuations within the scanned range, but does not by itself establish a performance gain.

Table 11 SCC scene adaptation and sensitivity check

Scene	d_{scene}	Default constraint coefficient	$s_{\min}$-scan PSNR↑ span	Weight-scan PSNR↑ span	Alpha IoU↑
Drums	1.000	1.000	0.000 dB	0.020 dB	Stable
Lego	0.000	0.350	0.023 dB	0.033 dB	Stable
Chair	0.177	0.465	0.025 dB	0.033 dB	Stable

Ablation Study

As shown in Fig. 6, the ablation results indicate that the candidate modules do not simply contribute additively. Removing VTC increases LPIPS from 0.3711 to 0.3788, while PSNR and contour metrics change slightly in the opposite direction, suggesting that VTC mainly improves perceptual consistency rather than every metric simultaneously. Removing SG-Graph slightly increases PSNR and SSIM, and removing ACA improves all five metrics overall. Accordingly, the final simplified configuration retains VTC while disabling SG-Graph and ACA. Figure 7 is included only to illustrate the ablated SG-Graph structure.

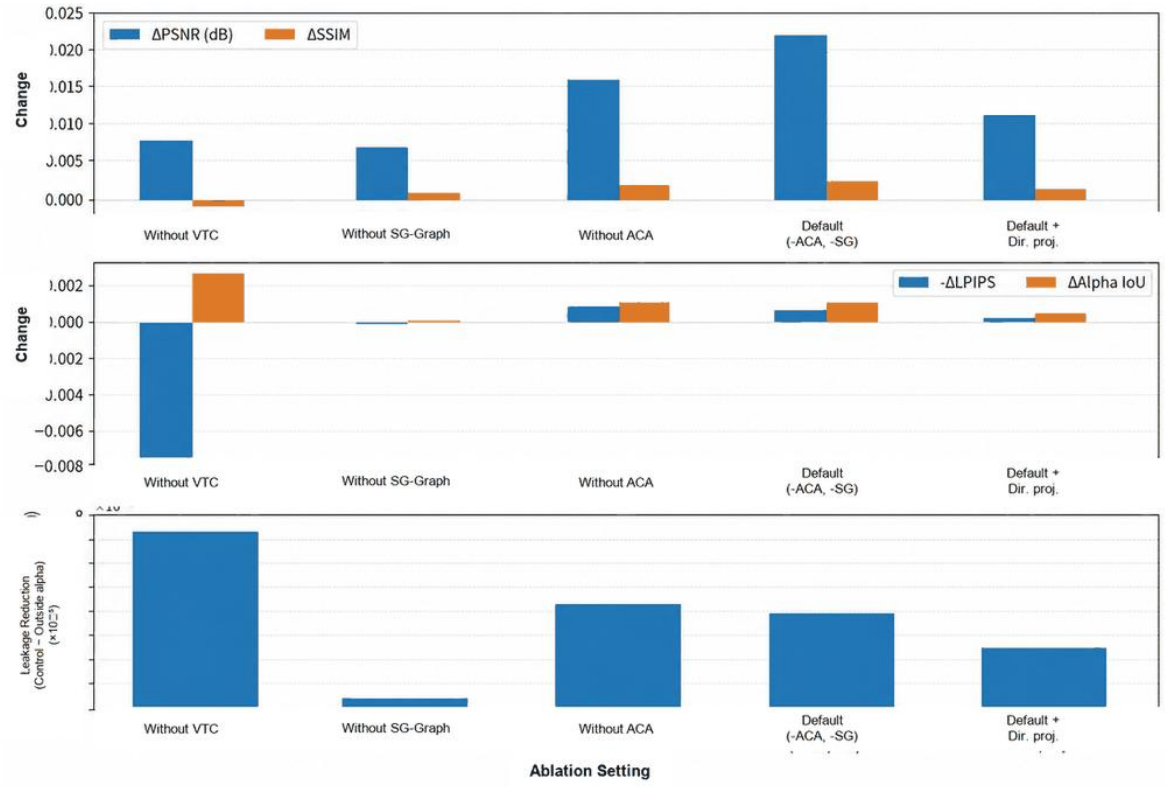


Fig. 6 Ablation changes of VTC and candidate auxiliary modules on Drums (three seeds; relative to Control)

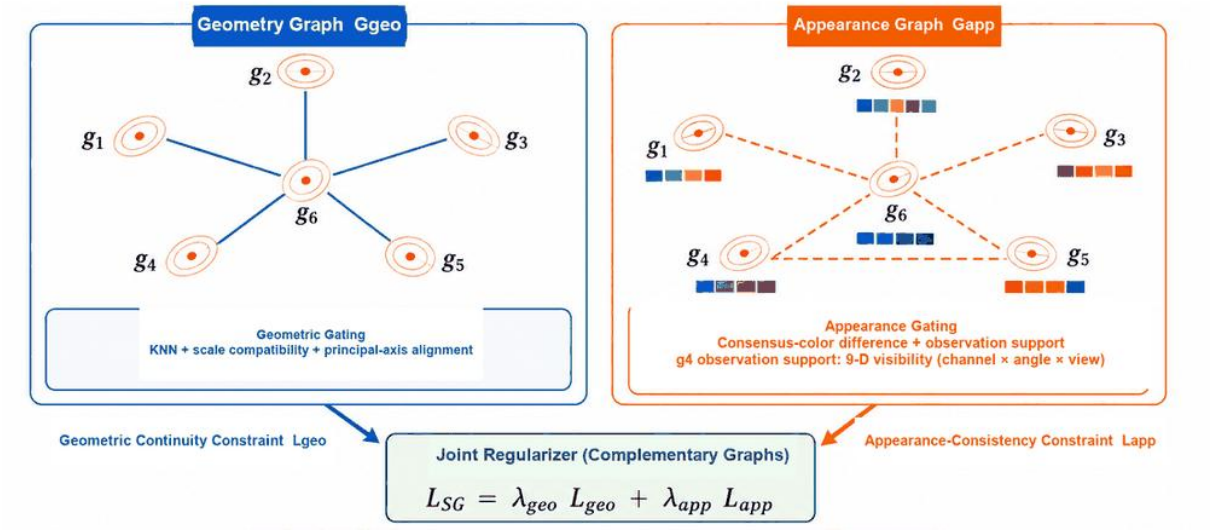


Fig. 7 SG-Graph candidate module: geometry graph, appearance graph, and joint regularization (ablation target; disabled by default)

Mechanism Analysis

This section separates mechanism evidence from performance evidence. We first test whether the directional operator changes the decomposition of center displacement, and then examine how different directional constraints and photometric-gradient routing strategies affect reconstruction quality.

With local directional projection enabled (Fig. 8), the normal-to-tangential displacement ratio decreases by 8.4% for active Gaussians and by 16.3% for high-confidence Gaussians, with the same direction of change across all three seeds. This confirms that the directional operator suppresses orthogonal displacement; whether this mechanism change translates into a performance gain must still be assessed using Fig. 9.

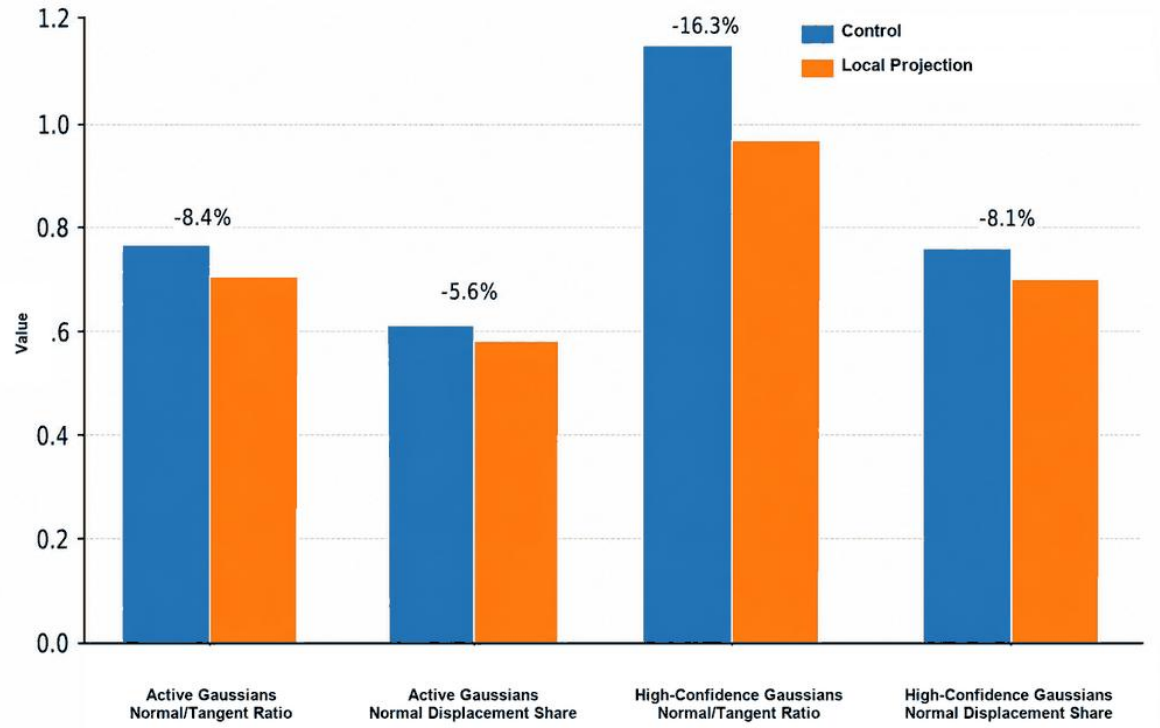


Fig. 8 Effect of local directional projection on cumulative displacement decomposition (Drums, three seeds)

Figure 9 shows that stronger suppression of orthogonal directions does not improve reconstruction quality. As the local projection coefficient is reduced from 0.80 to 0, PSNR drops from 15.118 to 15.105 dB, while SSIM, Alpha IoU, and Outside alpha deteriorate simultaneously. Equal-norm scaling, random-axis, and fixed-world-axis controls also fail to outperform Control. Directional projection is therefore retained only as a mechanism-analysis tool and remains disabled in the default configuration.

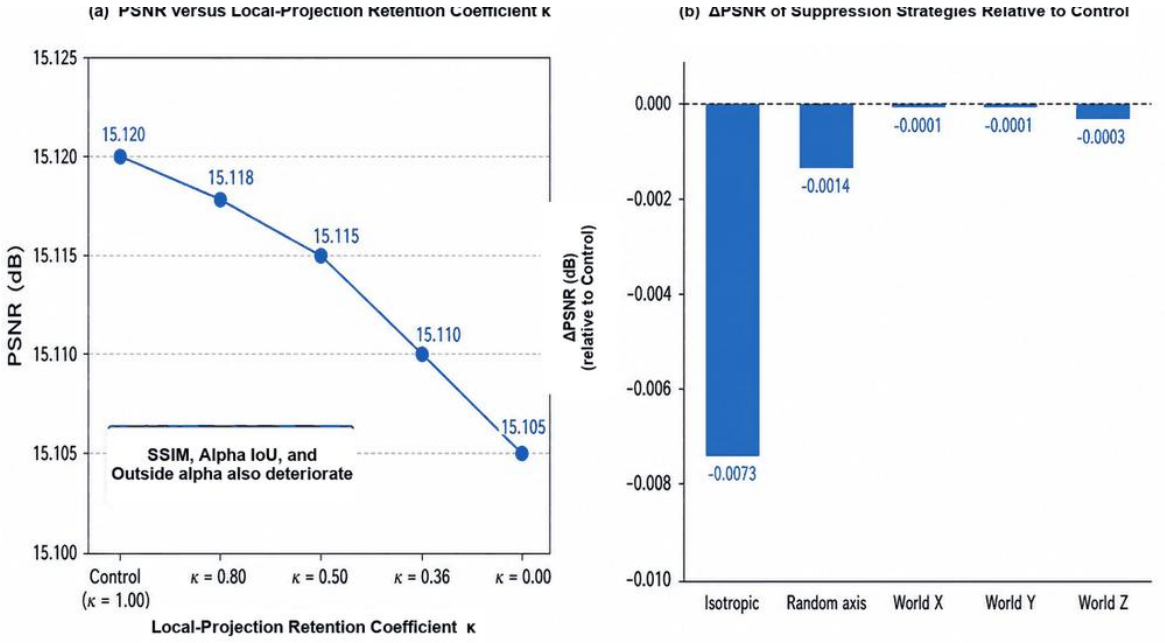


Fig. 9 Reconstruction-quality changes under different directional-suppression strategies and strengths

The parameter-level routing results (Fig. 10) further show that directly weakening the RGB gradient removes useful geometric information. Completely blocking RGB gradients to geometric parameters reduces PSNR by 0.1412 dB, and retaining 75% of the gradient still produces a 0.0183 dB decrease.

Delaying the activation of full routing does not alleviate the degradation. The final configuration therefore retains the complete photometric gradient and restricts only the VTC loss to the appearance-parameter pathway.

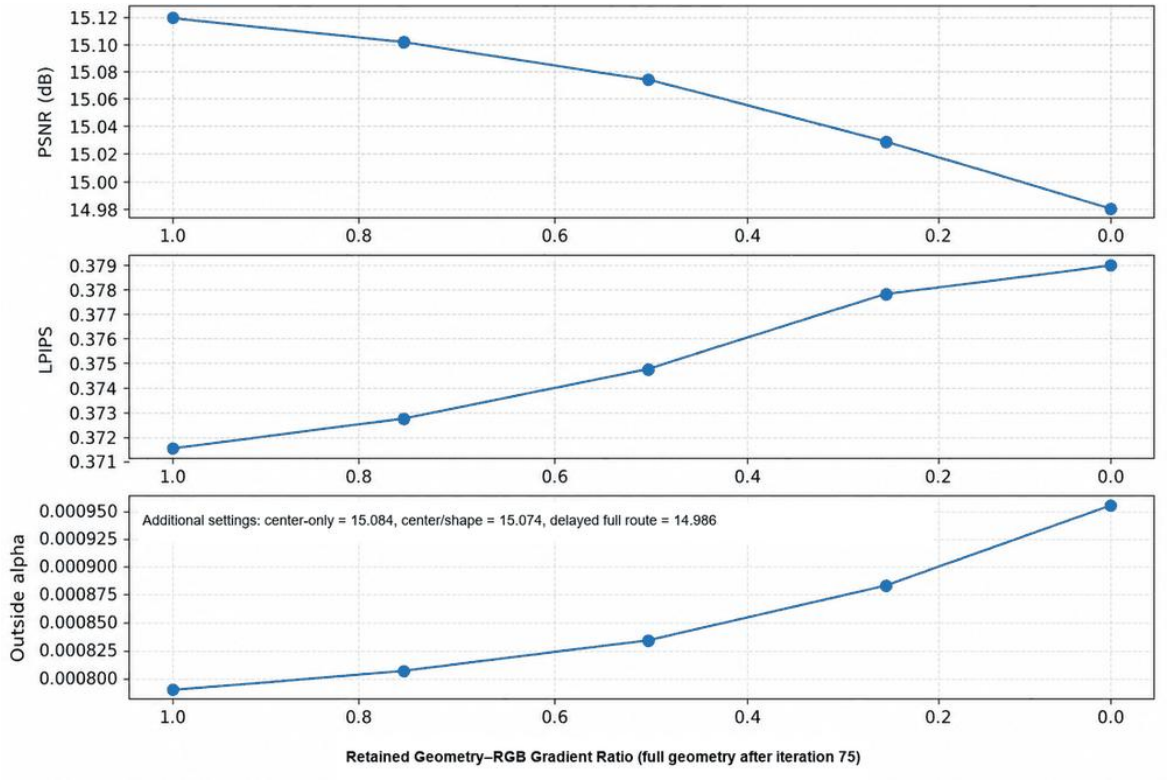


Fig. 10 Effect of the retained geometry-RGB gradient ratio on reconstruction quality (Drums, 300 iterations)

Supplementary Validation

Supplementary validation considers a geometry proxy, longer training budgets, and the standard densification backend. In the single-seed depth-proxy results (Table 13), visible-region MAE decreases on Drums, Lego, and Hotdog; Chair shows a negligible MAE increase accompanied by a lower RMSE.

The extended-budget results in Fig. 11 remain directionally consistent with the 300-iteration main experiments: at 1,200 and 3,000 iterations, PSNR improves by 0.0519 and 0.0625 dB, respectively, with the remaining metrics changing in the same favorable direction.

The standard densification backend yields a different conclusion (Table 14). In the single-seed 30k Drums run, VTC warm start decreases PSNR by 0.054 dB relative to Control, while full-time VTC decreases PSNR by 1.004 dB at the 7k checkpoint. A fixed consensus cannot be transferred directly to a continuously densifying Gaussian population; VTC is therefore disabled by default on this backend.

Table 12 Twenty-seven-run full-factorial grid for the consensus-inheritance extension (Drums, seed 17, 3,000 iterations)

No.	ρ_{inh}	T_r	κ_r	PSNR↑/dB	SSIM↑	LPIPS↓
1	0.25	500	1.5	27.375	0.9533	0.0510
2	0.25	500	3.0	27.380	0.9531	0.0515
3	0.25	500	4.5	27.137	0.9521	0.0532
4	0.25	1000	1.5	27.302	0.9527	0.0514
5	0.25	1000	3.0	27.198	0.9529	0.0525
6	0.25	1000	4.5	27.317	0.9527	0.0521
7	0.25	2000	1.5	27.372	0.9528	0.0525
8	0.25	2000	3.0	27.329	0.9527	0.0521
9	0.25	2000	4.5	27.301	0.9526	0.0519
10	0.50	500	1.5	27.329	0.9529	0.0517
11	0.50	500	3.0	27.354	0.9530	0.0520
12	0.50	500	4.5	27.412	0.9533	0.0523
13	0.50	1000	1.5	27.360	0.9532	0.0510
14	0.50	1000	3.0	27.330	0.9530	0.0515
15	0.50	1000	4.5	27.334	0.9532	0.0515
16	0.50	2000	1.5	27.325	0.9521	0.0527
17	0.50	2000	3.0	27.348	0.9527	0.0523
18	0.50	2000	4.5	27.346	0.9532	0.0515
19	0.75	500	1.5	27.350	0.9531	0.0521
20	0.75	500	3.0	27.295	0.9526	0.0518
21	0.75	500	4.5	27.328	0.9528	0.0519
22	0.75	1000	1.5	27.290	0.9526	0.0520
23	0.75	1000	3.0	27.358	0.9530	0.0519
24	0.75	1000	4.5	27.291	0.9524	0.0526
25	0.75	2000	1.5	27.375	0.9530	0.0517
26	0.75	2000	3.0	27.221	0.9530	0.0523
27	0.75	2000	4.5	27.371	0.9530	0.0526

Consensus-inheritance parameters are calibrated using a 3×3×3 full-factorial scan on Drums with seed 17 for 3,000 iterations, giving 27 training runs in total. PSNR ranges from 27.137 to 27.412 dB, SSIM from 0.9521 to 0.9533, and LPIPS from 0.0510 to 0.0532. The highest PSNR is 27.412 dB, whereas the default candidate (0.50/1000/3.0) obtains 27.330 dB, 0.9530 SSIM, and 0.0515 LPIPS. The optimal configurations differ across the three metrics, indicating that this grid is more appropriate for parameter calibration than for selecting a single universally optimal operating point.

Training logs confirm that the inheritance, refresh, and reset paths are all actually triggered. The default candidate requires 206.9 s of training time and 484.1 MiB of peak memory and ends with 110,085 Gaussians, of which 87,043 are inherited child primitives. Across three refresh events, 120,671 candidates are accepted and 92,900 are rejected, giving an acceptance rate of 56.5%; the effective support ratio at the end of training is 80.8%.

Accordingly, Table 12 is used only to verify the inheritance execution path and calibrate candidate parameters; it is not included in the main performance comparison.

Table 13 Geometry-proxy evaluation on the public Blender test depths (seed 17)

Scene	Configuration	MAE↓	RMSE↓	AbsRel↓	δ_1 ↑	Completeness	No. of Gaussians
Drums	Control	0.14392	0.26037	0.03908	0.96824	0.78230	29148
Drums	Simplified default	0.14299	0.25947	0.03885	0.96829	0.78223	29148
Lego	Control	0.15753	0.23436	0.04014	0.99060	1.00000	54726
Lego	Simplified default	0.15644	0.23313	0.03987	0.99080	1.00000	54726
Chair	Control	0.08646	0.15534	0.02182	0.99392	0.99214	35496

Scene	Configuration	MAE↓	RMSE↓	AbsRel↓	δ_1 ↑	Completeness	No. of Gaussians
Chair	Simplified default	0.08651	0.15495	0.02186	0.99402	0.99216	35496
Hotdog	Control	0.18564	0.26660	0.04698	0.98572	1.00000	42014
Hotdog	Simplified default	0.18014	0.26087	0.04554	0.98646	1.00000	42014

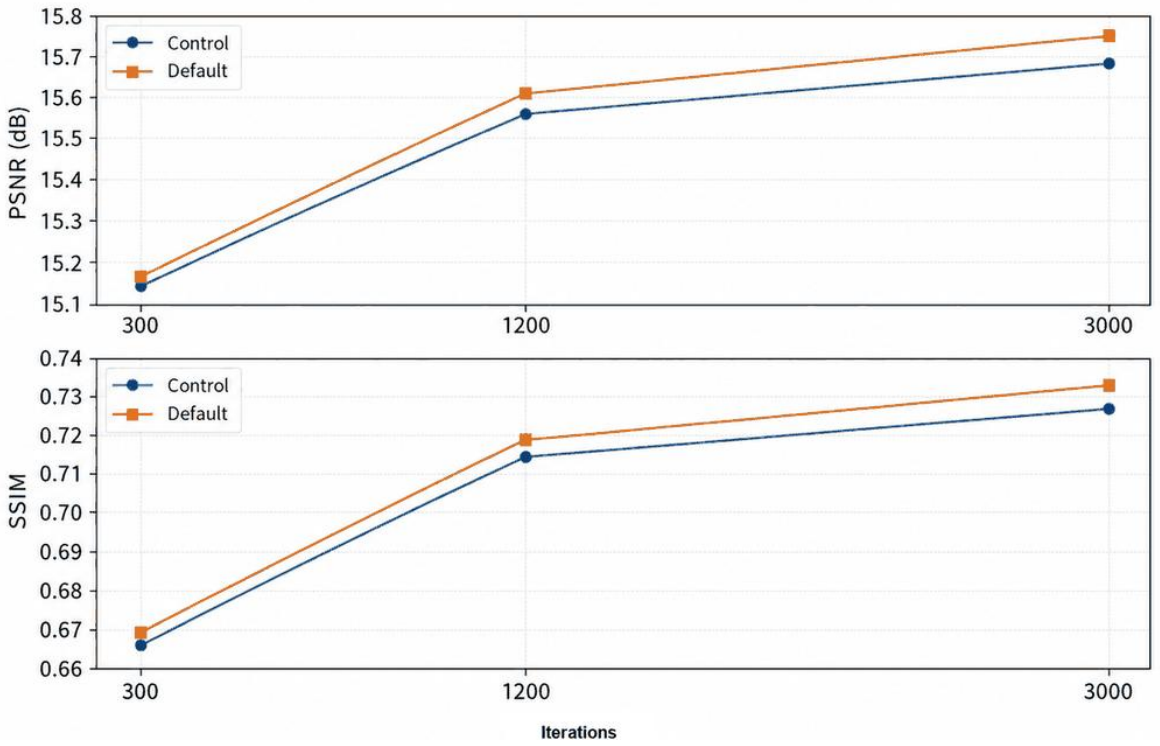


Fig. 11 PSNR and SSIM of Control and the simplified configuration under longer fixed-Gaussian training budgets

Table 14 Results on the standard 3DGS densification backend (Drums)

Iterations	Configuration	PSNR↑/dB	SSIM↑	LPIPS↓	Cumulative time / s	GPU memory / MiB	No. of Gaussians
7000	Control	28.577	0.9617	0.0386	137.3	349.0	158080
7000	Full-time VTC	27.573	0.9528	0.0539	195.3	350.4	158360
7000	VTC warm start 500	28.045	0.9569	0.0455	149.6	339.5	151159
30000	Control	29.186	0.9630	0.0294	695.5	617.1	231165
30000	VTC warm start 500	29.133	0.9625	0.0304	707.8	603.8	229024

Figure 11 reports reconstruction-quality trends under longer budgets. Because the simplified configuration combines several changes, its training-time and memory differences are not attributed to any single module.

Qualitative Analysis

Figure 12 highlights the dark drumhead, slender support rods, and foreground contour in Drums. Vanilla 3DGS shows speckles, fractures, and background leakage, whereas the compared variants exhibit visibly different artifact and contour behavior. The visualization complements the quantitative results.

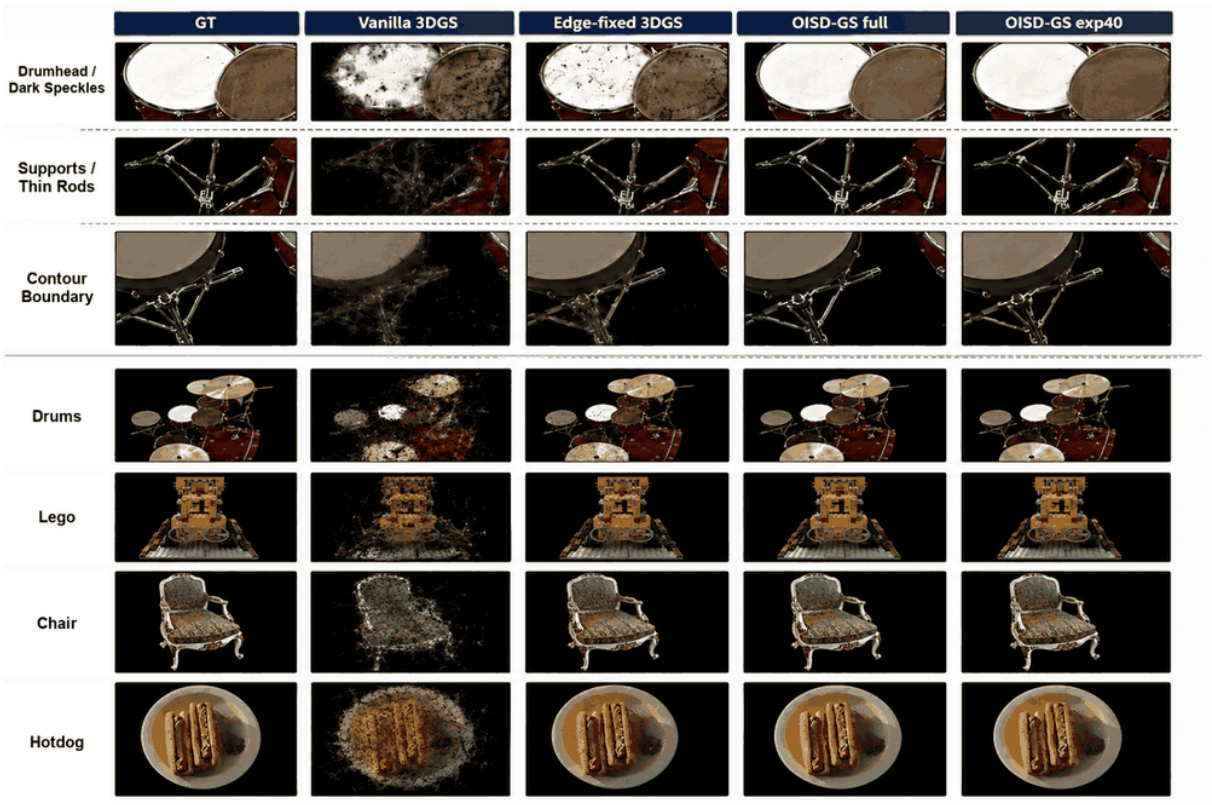


Fig. 12 Qualitative comparison on Drums (left to right: GT, Vanilla 3DGS, Edge-fixed 3DGS, OISD-GS full, and OISD-GS exp40)

Conclusion

This study analyzes how unreliable photometric supervision affects geometric optimization in texture-degraded scenes and introduces VTC, parameter-level loss routing, and local-PCA structural constraints. Under the controlled fixed-Gaussian protocol, the simplified configuration produces consistent paired improvements across four scenes and five random seeds, while the degradation and mechanism experiments support the intended observation-reliability interpretation. Ablations further show that VTC should be retained, whereas directional projection and aggressive attenuation of RGB gradients to geometry do not provide reliable quality gains and are therefore excluded from the default configuration. The current evidence also identifies a transfer boundary: fixed VTC does not improve the standard adaptive-densification backend, and the present real-data experiments validate pipeline compatibility rather than cross-dataset generalization. Overall, the method is best supported as a robust optimization strategy for texture-degraded reconstruction under controlled fixed-Gaussian settings. Future work will focus on adaptive consensus for evolving Gaussian populations and broader real-scene validation.

References

[1] ZHOU Xueyang, SHEN Xukun, HU Yong. A survey on 3D reconstruction based on neural field inverse rendering[J]. Journal of Graphics, 2026, 47(3): 449–471. (in Chinese)

[2] Mildenhall B, Srinivasan P P, Tancik M, et al. NeRF: representing scenes as neural radiance fields for view synthesis[C]//Proceedings of the European Conference on Computer Vision. Cham, Switzerland, 2020: 405-421.

[3] Barron J T, Mildenhall B, Verbin D, et al. Mip-NeRF 360: unbounded anti-aliased neural radiance fields[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2022: 5470-5479.

[4] LI Jize, CHENG Lechao, HE Jinxin, et al. Research status and prospect of neural radiance fields[J/OL]. Journal of Computer-Aided Design & Computer Graphics: 1–20 [2026-07-30]. http://kns.cnki.net/kcms/detail/11.2925.TP.20240204.1612.051.html. (in Chinese)

[5] Kerbl B, Kopanas G, Leimkühler T, et al. 3D Gaussian splatting for real-time radiance field rendering[J]. ACM Transactions on Graphics, 2023, 42(4): 1-14.

[6] LIAN Zhenhan, WANG Zhangye, NIU Jiong, CHENG Lechao. A survey on dynamic scene reconstruction based on 3D Gaussian splatting[J]. Journal of Computer-Aided Design & Computer Graphics, 2026, 38(1): 61–77. (in Chinese)

[7] Luan F, Sun S, Zhang H, et al. 3D Gaussian splatting technologies and extensions: a review[J]. Neurocomputing, 2025, 658: 131629.

[8] Li J, Zhang J, Bai X, et al. DNGaussian: optimizing sparse-view 3D Gaussian radiance fields with global-local depth normalization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2024: 20775-20785.

[9] Guédon A, Lepetit V. SuGaR: surface-aligned Gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2024: 5354-5363.

[10] Huang B, Yu Z, Chen A, et al. 2D Gaussian splatting for geometrically accurate radiance fields[C]//Proceedings of the SIGGRAPH Conference Papers. New York, USA, 2024: 1-11.

[11] Gao Z, Bian J W, Lin G, et al. SurfaceSplat: connecting surface reconstruction and Gaussian splatting[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, USA, 2025: 28525-28534.

[12] Yu Z, Chen A, Huang B, et al. Mip-Splatting: alias-free 3D Gaussian splatting[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2024: 19447-19456.

[13] Yang H, Zhang C, Wang W, et al. Improving Gaussian splatting with localized points management[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2025: 21696-21705.

[14] Schönberger J L, Frahm J M. Structure-from-motion revisited[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2016: 4104-4113.

[15] Cheng K, Long X, Yang K, et al. GaussianPro: 3D Gaussian splatting with progressive propagation[C]//Proceedings of the International Conference on Machine Learning. New York, USA, 2024: 8123-8140.

[16] Jia G, Appia V, Huang J, et al. Rendering-aware Bayesian 3D Gaussian splatting with native uncertainty and adaptive complexity control[EB/OL]. [2026-07-30]. https://arxiv.org/abs/2607.05522.

[17] Wu J, Li R, Zhu Y, et al. Sparse2DGS: geometry-prioritized Gaussian splatting for surface reconstruction from sparse views[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2025: 11307-11316.

[18] Jiang K, Sivaram V, Peng C, et al. Geometry field splatting with Gaussian surfels[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2025: 5752-5762.

[19] Li Y, Lyu C, Di Y, et al. GeoGaussian: geometry-aware Gaussian splatting for scene rendering[C]//Proceedings of the European Conference on Computer Vision. Milan, Italy, 2024: 441-457.

[20] Ververas E, Potamias R A, Song J, et al. SAGS: structure-aware 3D Gaussian splatting[C]//Proceedings of the European Conference on Computer Vision. Milan, Italy, 2024: 221-238.

[21] Zhou Q, Gong Y, Yang W, et al. MGSR: 2D/3D mutual-boosted Gaussian splatting for high-fidelity surface reconstruction under various light conditions[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, USA, 2025: 27295-27304.

[22] Zhou X, Ran X, Xiong Y, et al. SparseGS: real-time 360° sparse view synthesis using Gaussian splatting[EB/OL]. [2026-07-30]. https://arxiv.org/abs/2312.00206.

[23] Zhang J, Li J, Yu X, et al. CoR-GS: sparse-view 3D Gaussian splatting via co-regularization[C]//Proceedings of the European Conference on Computer Vision. Milan, Italy, 2024: 335-352.

[24] Wang Y, Klasson M, Turkulainen M, et al. DeSplat: decomposed Gaussian splatting for distractor-free rendering[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2025: 722-732.

[25] Rong V, Chen J, Bahmani S, et al. GStex: per-primitive texturing of 2D Gaussian splatting for decoupled appearance and geometry modeling[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Piscataway, USA, 2025: 3508-3518.

[26] Zhou H, Lähner Z. Geometry Gaussians: decoupling appearance and geometry in Gaussian splatting[EB/OL]. [2026-07-30]. https://arxiv.org/abs/2606.05124.

[27] Srinivasan P P, Deng B, Zhang X, et al. NeRV: neural reflectance and visibility fields for relighting and view synthesis[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2021: 7495-7504.

[28] Martin-Brualla R, Radwan N, Sajjadi M S M, et al. NeRF in the wild: neural radiance fields for unconstrained photo collections[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2021: 7210-7219.

[29] Sabour S, Vora S, Duckworth D, et al. RobustNeRF: ignoring distractors with robust losses[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Piscataway, USA, 2023: 866-875.

[30] Mildenhall B, Hedman P, Martin-Brualla R, et al. NeRF in the dark: high dynamic range view synthesis from noisy raw images[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2022: 16190-16199.

[31] Goli L, Reading C, Sellán S, et al. Bayes' rays: uncertainty quantification for neural radiance fields[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2024: 20061-20070.

[32] WANG Feng, et al. Lightweight reconstruction scene segmentation method based on Gaussian splatting[J/OL]. Chinese Journal of Computers, 2025. DOI: 10.11897/SP.J.1016.2025.01232. (in Chinese)

[33] Jensen R, Dahl A, Vogiatzis G, et al. Large scale multi-view stereopsis evaluation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA, 2014: 406-413.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62473278) and the 2025 “Suqian Talents” Xiongying Plan · Talent Introduction Project (SQXY202529).