Birth-Death Structural Learning for 3D Gaussian Splatting*

Quan Tran Hong 26quan.th@vinuni.edu.vn
Long Nguyen-Chi lc.stilldomath@gmail.com
Binh T. Nguyen binh.nt2@vinuni.edu.vn
Center for AI Research, VinUniversity
Hanoi, Vietnam

Abstract

3D Gaussian Splatting (3DGS) has become a standard representation for real-time novel-view synthesis. Yet, its quality–efficiency trade-off still relies heavily on adaptive density control. Existing cloning, splitting, and pruning rules use proxy signals—such as position gradients, opacity, image-space error. Although effective in practice, these signals do not directly answer the structural question behind density control: where should new Gaussians be allocated to reduce the multi-view reconstruction loss, and which existing primitives can be pruned with minimal penalty. To address this, we propose first-variation birth–death control for 3DGS, a principled approach that replaces heuristic structural decisions with a variational score derived from the reconstruction loss. At each structural update, we freeze the current Gaussian population, attach auxiliary mass coordinates to each primitive, and differentiate the multiview loss with respect to these coordinates. We demonstrate that this score is a coordinate of the first variation of Splat Regression Model, giving a rigorous local descent surrogate for birth-death moves: high-score Gaussians are selected as birth sites, low-score low-opacity Gaussians are death. Across standard 3DGS benchmarks, integrating this score into 3DGS-MCMC yields state-of-the-art reconstruction quality and outperforms existing methods, delivering the most significant gains under tight Gaussian budgets. Ultimately, our work provides compelling evidence that gradient-based variational metrics offer a principled alternative to proxy heuristics in 3DGS.

1 Introduction

High-quality photorealistic scene modeling from images is a fundamental task in computer vision and graphics, with applications spanning immersive rendering, robotics, digital humans, and geometric reconstruction [4, 38]. 3D Gaussian Splatting (3DGS) [15] has emerged as a fast explicit radiance-field representation that models scenes as anisotropic Gaussian primitives and renders them by differentiable rasterization. Its speed and explicitness have led to rapid extensions in dynamic rendering, anti-aliasing, generative 3D content, and SLAM [9, 11, 14, 24, 27, 28, 32, 34, 37].

Despite this progress, 3DGS training still relies on heuristic *adaptive density control* (ADC). Training a 3DGS model interleaves two distinct optimization mechanisms. Continuous optimization adjusts the parameters (position, covariance, color) of existing Gaussians to fit the image data. Structural control, meanwhile, changes the population itself by creating or destroying primitives. In standard 3DGS, these population updates rely on indirect proxies: high accumulated position gradients trigger cloning or splitting, while low opacity triggers pruning [15]. These signals correlate with regions needing refinement, but they do not directly measure how a discrete birth or death event affects the photometric objective. As a result, primitive placement is sensitive to hand-tuned thresholds.

*The code is available at: https://github.com/quanth238/birth-death-3dgs.

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

The model frequently misses high-frequency details, creates floaters or needle-like artifacts, and wastes redundant Gaussians to regions that are already well covered [10, 25, 36]. The cost is most visible under tight primitive budgets, where redundant placement directly competes with missing detail [2, 17, 19].

While recent methods improve these signals through pixel-aware gradients, error-guided insertion, and budgeted densification [2, 22, 36], they remain mathematically decoupled from the empirical risk. The Splat Regression Model (SRM) framework [6] provides a rigorous lens to understand this limitation: it models splat fitting as optimization over mixing measures structured by Wasserstein-Fisher-Rao (WFR) geometry. Under this view, standard 3DGS successfully performs continuous *transport* (moving Gaussian parameters via gradient descent), but completely ignores the WFR *mass* component when deciding where to spawn or destroy primitives. Because disconnected heuristics govern these mass perturbations, a fundamental allocation problem remains unresolved: *How can we identify which discrete primitives to kill, and where to spawn new ones guided by a first-order descent surrogate for the multiview reconstruction loss?*

Contributions. In this work, we propose reframing 3DGS structural updates from a collection of proxy heuristics into a formal component of empirical risk minimization. Motivated by the WFR geometry, we formulate structural control as a finite-particle birth–death process. At each control step, we freeze the continuous scene parameters and attach an auxiliary mass coordinate to each Gaussian. Differentiating the multiview reconstruction risk with respect to these coordinates yields a per-primitive first-variation score. This score is the negative directional derivative of the objective, evaluated at the frozen parameter state. It mathematically predicts the first-order descent achieved by applying an infinitesimal mass perturbation to that specific primitive. We then use score gaps to guide the birth-death process: capacity is reclaimed by killing low-score, low-opacity Gaussians and reborn at high-score sites. This provides a criterion for birth–death decisions. Because finite structural updates lack strict continuous descent guarantees, we execute them via the state transitions of 3DGS-MCMC [17]. This work bridges the theoretical view of splat fitting as mixing-measure optimization with practical 3DGS training. Our contributions are:

- We replace heuristic proxy signals with a first-variation score for 3DGS structural control, directly tying primitive allocation to the multiview empirical risk.
- We theoretically justify this framework by discretizing the SRM first variation, and by proving that restricting death to low-opacity Gaussians bounds the immediate rendering disturbance.
- We demonstrate that variational allocation outperforms proxy heuristics when representation capacity is constrained. Across standard benchmark datasets, our method achieves state-of-the-art reconstruction quality against matched 3DGS-MCMC and recent budget-aware baselines.

1.1 Related Work

Adaptive density control and densification. The original 3DGS framework couples continuous photometric optimization with adaptive density control: accumulated position-gradient statistics trigger cloning or splitting, while low opacity triggers pruning [15]. A massive body of recent work improves this controller. *Revising Densification* and Pixel-GS replace or reweight the growth criterion using pixel-level error or pixel-aware gradients [25, 36], while AbsGS and Improved ADC refine gradient aggregation, threshold schedules, and pruning rules [10, 33]. Other methods manage the primitive budget explicitly through resource-aware schedules, progressive propagation, or high-error insertion rules [2, 5, 22]. While these methods engineer better proxy signals or insertion heuristics, our work diverges fundamentally: we abandon proxies in favor of the structural derivative. By explicitly differentiating the multiview risk with respect to Gaussian mass, we formalize capacity allocation as a direct objective rather than a heuristic byproduct.

Relocation, MCMC, and compact representations. Several methods improve 3DGS efficiency by pruning redundant Gaussians or compressing attributes post-training [8, 12, 20, 23], while others avoid incremental growth entirely by relying on dense initializations [19]. The closest backend to our method is 3DGS-MCMC [17], which rewrites densification and pruning as relocation/addition transitions. Our approach adopts these discrete transitions from 3DGS-MCMC, but completely replaces its structural selection statistic. Rather than relying on opacity-sampling to blindly select

relocation targets, we use our first-variation score to predict where released capacity is most likely to reduce the reconstruction risk under a local first-order model.

Variational splat models and first variation. The Splat Regression Model (SRM) formalizes splat models as optimization over mixing measures, deriving Wasserstein–Fisher–Rao gradient flows and explicit first-variation characterizations of mass rescaling [6]; related birth–death discretizations also appear in sampling theory [21]. However, while SRM establishes the continuous, measure-level WFR geometry, it does not prescribe a finite, budget-constrained training algorithm for discrete rasterizers. Our contribution bridges this gap: we derive a concrete, renderer-level mass-coordinate score and a finite score-gap birth–death controller, making this continuous variational geometry actionable for practical 3DGS training.

2 Background

We fix notation for the 3DGS renderer and the 3DGS-MCMC relocation backend. The SRM first-variation viewpoint is recalled at the end of the section.

3D Gaussian Splatting. At iteration t, a scene is represented by a finite population $\mathcal{G}_t = \{g_i\}_{i=1}^{M_t}$, where

$$g_i = (\mu_i, s_i, R_i, \alpha_i, c_i), \qquad \mu_i \in \mathbb{R}^3, \quad s_i \in \mathbb{R}^3_{>0}, \quad R_i \in SO(3), \quad \alpha_i \in (0,1).$$

Here s_i and R_i determine the anisotropic covariance $\Sigma_i = R_i \,\mathrm{Diag}(s_i^2) R_i^\top$, and c_i denotes the view-dependent color parameters. For a fixed training view n and pixel x, let $G_i(x)$ be the projected Gaussian footprint, $a_i(x) := \alpha_i G_i(x)$, and let $j \prec_n i$ denote front-to-back ordering along the ray. Suppressing the view index on projected footprints and transmittances, the rendered color is

$$\hat{I}_n(x) = \sum_{i=1}^{M_t} T_i(x)\, a_i(x)\, c_i(\mathbf{d}_n), \qquad T_i(x) := \prod_{j \prec_n i} \big(1 - a_j(x)\big). \tag{1}$$

We write $\hat{I}_n = \Phi(\mathcal{G}_t, n)$ for the rasterized image. At training iteration t, a view n_t is sampled and the photometric loss is $\widehat{\mathcal{L}}_{\mathrm{img}}^{(t)} = \ell_1(\Phi(\mathcal{G}_t, n_t), I_{n_t})$. The 3DGS-MCMC backend additionally uses opacity and scale regularizers to encourage unused Gaussians to disappear; we retain these terms in training but do not expand them here, since our score pass is defined from the image objective in Section 3.

3DGS-MCMC: Markov view and relocation/addition operations. 3DGS-MCMC [17] interprets the Gaussian population as a sample state of a Markov chain. For a regularized training objective $\mathcal{L}$, the formal target density is

$$p_\beta(\mathcal{G}) = Z_\beta^{-1} \exp\big(-\beta \mathcal{L}(\mathcal{G})\big),$$

where Z_β is the normalizing constant and $\beta > 0$ is an inverse temperature. Under this view, the usual gradient update is interpreted as the zero-noise limit of an SGLD step. For a parameter block θ,

$$\theta \leftarrow \theta - \eta\, \widehat{\nabla}_\theta \mathcal{L}(\mathcal{G}) + \sigma_\theta(\mathcal{G})\, \xi, \qquad \xi \sim \mathcal{N}(0, I),$$

where $\widehat{\nabla}$ is a stochastic gradient and $\sigma_\theta(\mathcal{G})$ is the exploration scale. In the original 3DGS-MCMC implementation, this noise is applied to Gaussian centers, with scale depending on the current Gaussian state.

The structural mechanism of 3DGS-MCMC realizes densification and pruning through deterministic state transitions. Let $\mathcal{D}_t := \{i : \alpha_i \le \alpha_{\mathrm{death}}\}$ and $\mathcal{A}_t := \{i : \alpha_i > \alpha_{\mathrm{death}}\}$ denote the candidate dead and live pools, respectively. To reallocate capacity, r_j recycled slots from $\mathcal{D}_t$ (or fresh slots during addition) are assigned to a target Gaussian j, yielding $N_j := r_j + 1$ co-located copies parameterized as:

$$\mu'_{j,\ell} = \mu_j, \qquad \alpha'_{j,\ell} = 1 - \big(1 - \alpha_j\big)^{1/N_j}, \qquad \Sigma'_{j,\ell} = \gamma(N_j, \alpha_j)\, \Sigma_j, \tag{2}$$

for $\ell = 1, \dots, N_j$, with color and orientation inherited from the target. The closed-form scalar $\gamma(N_j, \alpha_j)$ rescales the covariance such that the aggregate opacity footprint of the co-located cluster matches that of the original primitive, minimizing instantaneous rendering perturbations.

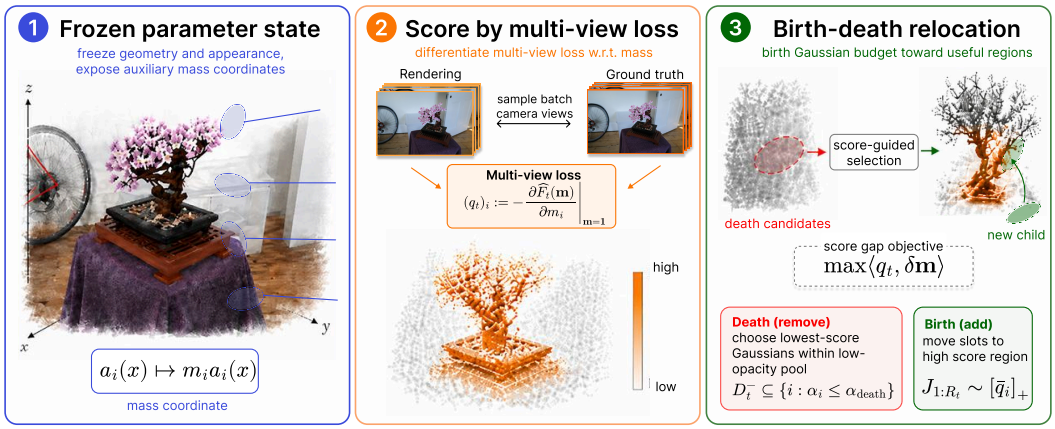


Figure 1: Overview of the score-guided birth–death controller. At each structural update, we freeze the current Gaussian parameters and expose one auxiliary mass coordinate per primitive, implemented as $a_i(x) \mapsto m_i a_i(x)$. A batched multiview backward pass computes the first-variation score $q_i = -\partial \widehat{F}_t(\mathbf{m})/\partial m_i|_{\mathbf{m}=\mathbf{1}}$. Higher scores explicitly predict a larger first-order loss reduction from assigning mass to that primitive. Guided by these scores, the controller kills low-opacity, low-score slots and births new capacity at high-score parent sites using the 3DGS-MCMC transitions.

In our framework, we decouple structural decision-making from physical execution: we adopt these closed-form transitions strictly as a *conservative realization backend*, while entirely replacing the heuristic opacity proposals with our objective-derived first-variation score.

SRM viewpoint: splat measures and first variation. Splat Regression Models (SRM) [6] formalize "splat" representations as optimization over mixing measures. Let ρ be a centered isotropic "mother" density on $\mathbb{R}^3$. For $(A, b) \in \mathbb{R}^{3\times3} \times \mathbb{R}^3$, define the transformed splat $\rho_{A,b} := (A(\cdot) + b)_{\#}\rho$. A (vector-valued) splat measure μ over weights and splat parameters induces the mixture model

$$f_\mu(x) := \int v\, \rho_{A,b}(x)\, \mu(dv, dA, db), \qquad x \in \mathbb{R}^3.$$

SRM endows the space of such measures with a Wasserstein–Fisher–Rao (WFR) geometry in which gradient flow decomposes into a *transport* component (moving A, b) and a *mass* component (rescaling v). The first-order object governing the mass component is the *first variation* of the data-fit functional. Inspired by the SRM first-variation view of splat measures, our score pass instantiates a finite-dimensional coordinate of this first-variation field for the specific mass direction that scales opacities (see Section 3.1).

3 Method

3.1 First-Variation Score from Splat Measures

Let $F_{\mathcal{D}}(\mathcal{G}) := N^{-1}\sum_{n=1}^{N} \ell_{\text{img}}(\Phi(\mathcal{G}, n), I_n)$ denote the full multiview empirical risk. The Splat Regression Model (SRM) shows that optimizing a splat-based representation involves two distinct operations: moving parameters (transport) and changing primitive weights (mass). Standard 3DGS handles transport natively via continuous gradient descent. To handle mass, we evaluate how infinitesimal changes to a Gaussian's weight affect the empirical risk.

At a controller step t, we freeze the current population $\mathcal{G}_t = \{g_i\}_{i=1}^{M_t}$ and attach an auxiliary mass coordinates $\mathbf{m}_i$ for each Gaussian, initialized to the all-ones vector $\mathbf{1}$. We define a mass-perturbed population $\mathcal{G}_t^{\mathbf{m}}$ by scaling the opacity contribution of each Gaussian i by m_i, i.e., $a_i(x) \mapsto m_i a_i(x)$. We estimate the minibatch multiview risk and its negative mass derivative from a batch of views $\mathcal{B}_t$:

$$\widehat{F}_t(\mathbf{m}) := \frac{1}{B}\sum_{n\in\mathcal{B}_t} \ell_{\text{img}}(\Phi(\mathcal{G}_t^{\mathbf{m}}, n), I_n)\,, \qquad (q_t)_i := -\left.\frac{\partial \widehat{F}_t(\mathbf{m})}{\partial m_i}\right|_{\mathbf{m}=\mathbf{1}}. \tag{3}$$

All scene parameters are detached during this computation, so a single backward pass returns the score vector q_t. The sign of $(q_t)_i$ gives the local mass interpretation: increasing m_i decreases the empirical risk by $\varepsilon(q_t)_i + o(\varepsilon)$.

Fix a view (I_n, π_n). Under mass coordinates $\mathbf{m}$, the renderer induces a map $f_{n,t}(\cdot; \mathbf{m}) := \Phi(\mathcal{G}_t^{\mathbf{m}}, \pi_n) \in L^2(\Omega; \mathbb{R}^3)$. Define the per-view functional $F_n(f) := \ell_{\text{img}}(f, I_n)$. Following SRM, the relevant first-order object governing mass creation is the first variation.

Definition 1 (First variation on images [6, Def. 2])**.** *Let $F : L^2(\Omega; \mathbb{R}^3) \to \mathbb{R}$ be Fréchet differentiable at f. Its first variation $\delta F[f] \in L^2(\Omega; \mathbb{R}^3)$ is the Riesz representative of the differential: for all $\xi \in L^2(\Omega; \mathbb{R}^3)$,*

$$\frac{d}{d\varepsilon} F(f + \varepsilon\xi)\Big|_{\varepsilon=0} = \int_\Omega \langle \xi(x),\, \delta F[f](x) \rangle \, dx.$$

Remark 1 (On differentiability and discretization)**.** *Definition 1 assumes Fréchet differentiability. The practical image loss contains nonsmooth components such as ℓ_1 and differentiable image-structure terms such as DSSIM. The implemented renderer is evaluated on a pixel grid, in which case the integrals below are finite sums over pixels.*

Applying the chain rule to $F_n \circ f_{n,t}$ yields a coordinate expression for the score.

Proposition 1 (Score as a first-variation coordinate)**.** *Assume each F_n admits a first variation and $f_{n,t}(\cdot; \mathbf{m})$ is differentiable in $\mathbf{m}$ at $\mathbf{m} = \mathbf{1}$. Then the score in* (3) *satisfies*

$$(q_t)_i = -\frac{1}{B} \sum_{n \in \mathcal{B}_t} \int_\Omega \left\langle \delta F_n\big[f_{n,t}(\cdot; \mathbf{1})\big](x),\ \frac{\partial f_{n,t}(x; \mathbf{m})}{\partial m_i}\Big|_{\mathbf{m}=\mathbf{1}} \right\rangle dx.$$

The proof is provided in Appendix A. Proposition 1 connects the SRM first-variation viewpoint to renderer-level mass coordinates. Intuitively, the integral pairs the image-space loss gradient with the image change induced by rescaling the i-th Gaussian. Ranking Gaussians by $(q_t)_i$ therefore orders the predicted loss decrease for equal infinitesimal mass increments at the frozen state. Unlike accumulated position gradients or plain opacity [15, 17], the score directly evaluates this mass direction, providing an objective-derived criterion for allocating capacity.

In the specific context of 3DGS alpha-compositing [39], applying the chain rule to the renderer gives a concrete pixel-level expression for this score:

$$(q_t)_i = -\frac{1}{B} \sum_{n \in \mathcal{B}_t} \sum_x \left\langle r_n(x),\ T_i(x) a_i(x) \big(c_i(\mathbf{d}_n) - \hat{I}_i^{\text{back}}(x)\big) \right\rangle, \tag{4}$$

where $\hat{I}_i^{\text{back}}(x)$ is the accumulated background color and $r_n(x)$ is the image-space loss gradient. Equation (4) gives the geometric intuition: a Gaussian receives a high score when its footprint is visible, meaning $T_i(x)a_i(x)$ is large; when its color differs from the accumulated background behind it; and when the resulting color change aligns with the negative loss gradient $-r_n(x)$.

3.2 Birth–Death Allocation via Score Gaps

The score q_t provides a continuous mass derivative, but structural control requires a discrete allocation decision. We model reclaiming capacity from death coordinates D and assigning it to birth coordinates $J_{1:R_t}$ through the following balanced mass perturbation:

$$\delta\mathbf{m}(J, D) = \kappa \left(\sum_{r=1}^{R_t} e_{J_r} - \sum_{i \in D} e_i \right), \qquad |D| = R_t.$$

Here e_i is the i-th coordinate vector and $\kappa > 0$ sets the scale of the mass perturbation. The following bound identifies its first-order loss term under a smooth local model.

Proposition 2 (Local descent upper bound under smoothness)**.** *If $F(\mathbf{m}) := \widehat{F}_t(\mathbf{m})$ is L_{sm}-smooth on a neighbourhood containing the segment $\{\mathbf{1} + \tau\delta\mathbf{m} : \tau \in [0, 1]\}$, then*

$$F(\mathbf{1} + \delta\mathbf{m}) \le F(\mathbf{1}) - \langle q_t, \delta\mathbf{m} \rangle + \frac{L_{\text{sm}}}{2} \|\delta\mathbf{m}\|_2^2.$$

Proposition 2 (proof in Appendix A) identifies the score gap as the first-order predicted loss decrease:

$$\langle q_t, \delta \mathbf{m} \rangle = \kappa \left(\sum_{r=1}^{R_t} (q_t)_{J_r} - \sum_{i \in D} (q_t)_i \right). \tag{5}$$

Proposition 3 (Bounded influence of low-opacity Gaussians)**.** *Fix a view n and let $\hat{I}_n$ be the rendered image from* (1)*. Let $\hat{I}_n^{(-i)}$ be the image obtained by setting $\alpha_i = 0$ while keeping all other Gaussians unchanged. Assume rendered colors and the terminal background color are bounded in $[0,1]^3$, and define the per-image mean absolute error*

$$\|J\|_{1,\mathrm{mean}} := \frac{1}{|\Omega_{\mathrm{pix}}|} \sum_{x \in \Omega_{\mathrm{pix}}} \frac{1}{3} \|J(x)\|_1.$$

Then

$$\|\hat{I}_n - \hat{I}_n^{(-i)}\|_{1,\mathrm{mean}} \le \alpha_i.$$

Consequently, if $\ell_{\mathrm{img}}(\cdot, I_n)$ is L_ℓ-Lipschitz with respect to $\|\cdot\|_{1,\mathrm{mean}}$ in a neighbourhood of $\hat{I}_n$, then

$$\left| \ell_{\mathrm{img}}(\hat{I}_n, I_n) - \ell_{\mathrm{img}}(\hat{I}_n^{(-i)}, I_n) \right| \le L_\ell \, \alpha_i.$$

Proposition 3 (proof in Appendix A) bounds the disturbance from an individual low-opacity removal. This motivates the candidate pool $\mathcal{P}_t := \{i : \alpha_i \le \alpha_{\mathrm{death}}\}$. Within this pool, smaller scores indicate lower retention value under the local mass model.

The final variational controller merges this continuous ideal with discrete constraints. We stabilize multi-view stochasticity using an exponentially moving averaged (EMA) score $\bar{q}_t$. The death set $\mathcal{D}_t^-$ selects the R_t Gaussians with the lowest scores $(q_t)_i$ *within* the pool $\mathcal{P}_t$. The birth multiset $J_{1:R_t}$ is sampled from live Gaussians proportional to their positive scores $[(\bar{q}_t)_i]_+$.

3.3 Discrete Realization by the 3DGS-MCMC Kernels

Our variational policy dictates *which* primitives are retired and *where* new capacity is sampled to approximate first-order loss descent. Once the death set $\mathcal{D}_t^-$ is selected from the lowest-scoring candidates in $\mathcal{P}_t$ and the birth multiset $J_{1:R_t}$ is sampled proportional to positive scores $[(\bar{q}_t)_i]_+$, the discrete state transition is realized by the footprint-preserving relocation kernel:

$$\mathcal{G}_{t+} = \mathcal{K}_{\mathrm{reloc}}\big(\mathcal{G}_t;\, \mathcal{D}_t^-,\, J_{1:R_t}\big). \tag{6}$$

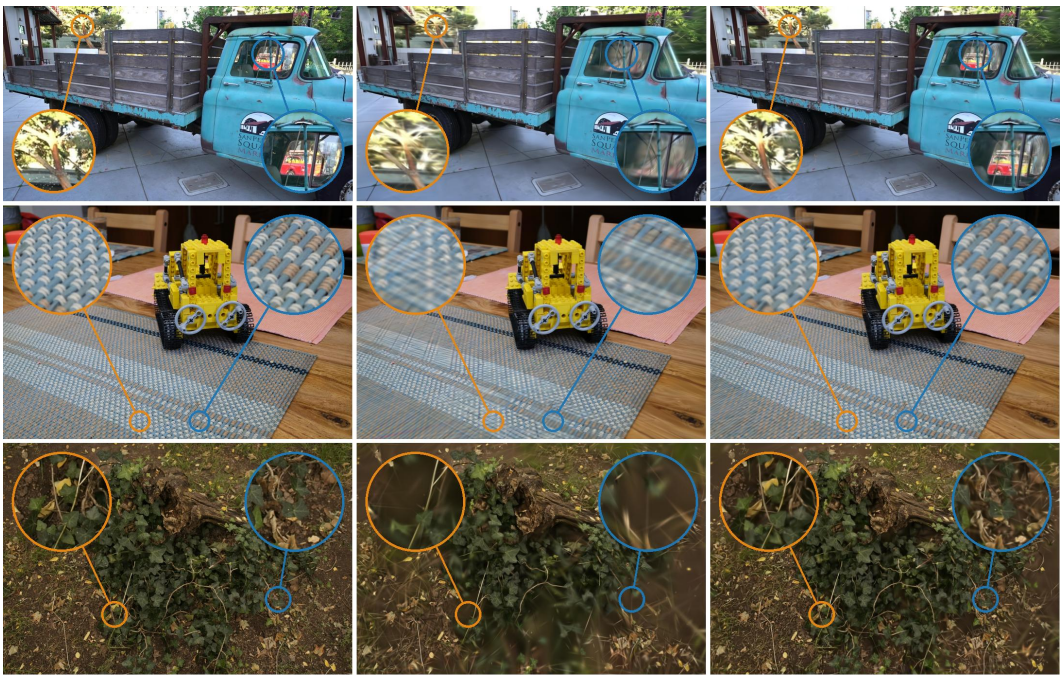
Each recycled slot in $\mathcal{D}_t^-$ is transferred to a sampled birth site in $J_{1:R_t}$, where the closed-form opacity and covariance corrections (2) preserve the aggregate optical footprint to minimize instantaneous rendering disturbance. When the population is below the prescribed cap, the addition kernel directly activates new slots at sites sampled from $[(\bar{q}_t)_i]_+$ without requiring paired deaths.

Algorithm 1 **First-variation controller.**

Input: Δ, B, α_{death}
for $t = 1, 2, \ldots$ **do**
  Take a gradient step on continuous parameters of $\mathcal{G}_t$
  if $t \bmod \Delta = 0$ **then**
    Freeze $\mathcal{G}_t$ and sample multi-view batch $\mathcal{B}_t$.
    Compute q_t ▷ Eq. (3)
    Form $\mathcal{P}_t = \{i : \alpha_i \le \alpha_{\mathrm{death}}\}$.
    Select $\mathcal{D}_t^-$ as the lowest-score elements in $\mathcal{P}_t$ ▷ Eq. (5)
    Sample birth sites $J_{1:R_t}$ from live Gaussians using $[(q_t)_i]_+$ ▷ Eq. (5)
    Apply $\mathcal{K}_{\mathrm{reloc}}$ ▷ Eq. (6)
  end if
end for

4 Experiments

Our experiments test a single claim: the first variation of the multiview reconstruction risk is a better birth–death signal than opacity or gradient proxies. We fix the renderer, optimizer, and MCMC transitions to isolate this statistic. This allows us to answer three direct questions: Does the variational controller improve reconstruction quality at the same primitive budgets? Is the gain explicitly caused by the WFR score? And does the allocation behavior align with local descent predictions?



(a) Ground Truth **(b) 3DGS-MCMC** **(c) Ours**

Figure 2: Qualitative comparison on Tanks & Temples (Truck, top row) and Mip-NeRF360 (middle and bottom rows) at a 200k-Gaussian cap with random initialization. All methods use the same view, scale, and crop coordinates in each row. Additional examples are in Appendix K.

4.1 Experimental setup

Datasets and metrics. We evaluate on Mip-NeRF360 [3], Tanks & Temples [18], and DeepBlending [13]. Our main fixed-budget Mip-NeRF360 averages use seven scenes: Bicycle, Garden, Stump, Room, Counter, Kitchen, and Bonsai; Flowers and Treehill are excluded. Tanks & Temples uses Truck and Train, and DeepBlending uses Dr. Johnson and Playroom. We report PSNR, SSIM [30], and LPIPS [35] on held-out test views. Per-scene results are in Appendix H; compact-model comparisons use all nine Mip-NeRF360 scenes.

Budgeted evaluation. We evaluate under matched Gaussian caps $K \in \{100\text{k}, 500\text{k}, 1\text{M}\}$. The fixed-budget setting isolates the structural decision. Every birth consumes finite capacity. Poor allocation cannot be hidden by unbounded growth.

Baselines and protocol. We compare against 3DGS [15] with SfM initialization, Improved ADC [10], 3DGS-MCMC [17] with SfM initialization, EDGS [19], and ConeGS [2]. Rows marked † use published fixed-budget benchmark numbers from ConeGS under the same dataset split and Gaussian cap. Claims about the effect of the proposed controller are based on matched 3DGS-MCMC reruns: the renderer, optimizer, initialization, cap, and transition maps remain identical. Only the structural selection statistic changes.

Implementation. Our implementation builds on 3DGS-MCMC. We keep the renderer, training loss, optimizer, relocation/addition kernels, and event schedule unchanged. All methods are trained for $30{,}000$ iterations. Hyperparameters, hardware, and timing details are in Appendix B.

4.2 Fixed-budget reconstruction quality

Table 1 establishes fixed-budget reconstruction quality. At the tightest 100k cap, our controller outperforms 3DGS-MCMC baseline, EDGS, and ConeGS across all three datasets. This severe scarcity highlights our advantage: reallocating via WFR score derivatives yields the largest impact.

Table 1: Fixed-budget comparison under matched Gaussian caps with SfM initialization for Ours. Mip-NeRF360 averages use the seven scenes listed in Section 4.1. Higher PSNR and SSIM are better; lower LPIPS is better. We highlight the best, second best and third best results within each dataset-budget block.

Method	Mip-NeRF360			Tanks & Temples			DeepBlending		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Number of Gaussians limited to 100k									
3DGS [15] (SfM)†	23.61	0.693	0.413	22.38	0.774	0.333	24.65	0.827	0.412
Improved ADC [10]	26.34	0.781	0.322	22.08	0.784	0.280	28.25	0.885	0.315
3DGS-MCMC [17] (SfM)†	27.06	0.800	0.303	22.50	0.780	0.332	28.94	0.876	0.333
EDGS [19]†	27.09	0.798	0.296	22.32	0.777	0.324	28.43	0.872	0.337
ConeGS [2]†	27.74	0.809	0.285	23.12	0.791	0.310	29.44	0.880	0.328
Ours	**27.94**	**0.818**	**0.276**	**23.60**	**0.831**	**0.225**	**29.45**	**0.892**	**0.303**
Number of Gaussians limited to 500k									
3DGS [15] (SfM)†	28.22	0.821	0.260	23.54	0.816	0.265	29.25	0.882	0.302
Improved ADC [10]	28.46	0.843	0.232	23.66	0.842	0.192	28.95	0.896	**0.260**
3DGS-MCMC [17] (SfM)†	28.82	0.861	0.214	23.68	0.825	0.259	29.44	0.887	0.308
EDGS [19]†	28.82	0.865	0.193	23.47	0.831	0.229	29.33	0.889	0.286
ConeGS [2]†	29.08	**0.870**	**0.190**	23.69	0.829	0.229	29.86	0.891	0.285
Ours	**29.11**	0.869	0.203	**24.05**	**0.857**	**0.178**	**30.04**	**0.906**	0.262
Number of Gaussians limited to 1M									
3DGS [15] (SfM)†	28.71	0.846	0.222	23.60	0.826	0.245	28.93	0.881	0.292
Improved ADC [10]	28.94	0.858	0.205	24.06	0.855	0.168	29.18	0.902	**0.248**
3DGS-MCMC [17] (SfM)†	29.23	0.875	0.190	23.93	0.836	0.233	29.67	0.887	0.288
EDGS [19]†	29.18	0.877	**0.168**	23.66	0.842	0.200	29.43	0.889	0.271
ConeGS [2]†	29.37	0.880	**0.168**	23.70	0.835	0.207	29.69	0.889	0.273
Ours	**29.49**	**0.883**	0.178	**24.17**	**0.865**	**0.157**	**30.14**	**0.911**	**0.248**

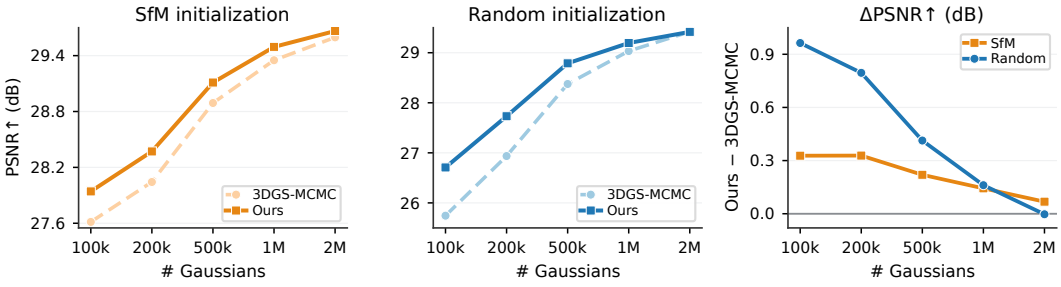


Figure 3: Matched PSNR–budget comparison on the seven-scene Mip-NeRF360 subset. The renderer, optimizer, relocation kernel, and training schedule are fixed; only the birth–death statistic changes from opacity to first variation.

At 500k and 1M, our method has the highest PSNR across all three datasets in Table 1. SSIM and LPIPS are metric-dependent: on Mip-NeRF360 at 500k, ConeGS has slightly higher SSIM (0.870 vs. 0.869), and competing methods achieve lower LPIPS at both larger caps. These results support the benefit of score-guided allocation under tight budgets without implying uniform superiority. Figure 3 shows the matched PSNR–budget frontier. Full metrics, random-initialization results, and runtimes are in Appendix I.

4.3 Does the first-variation score matter?

To isolate the effect of the WFR score on these gains, we perform a controlled drop-in ablation on Mip-NeRF360 at a 100k cap. We freeze the renderer, optimizer, training schedule, and MCMC relocation kernel; only the structural selection statistic changes. Table 2 shows that replacing opacity sampling with position gradients yields a +0.04 dB gain, while the post-activation opacity gradient yields +0.11 dB. The latter already measures local loss sensitivity, but to an additive opacity change. Our mass score $q_i = -\alpha_i \, \partial \widehat{F}_t / \partial \alpha_i$ instead measures relative mass rescaling and achieves +0.33 dB, an

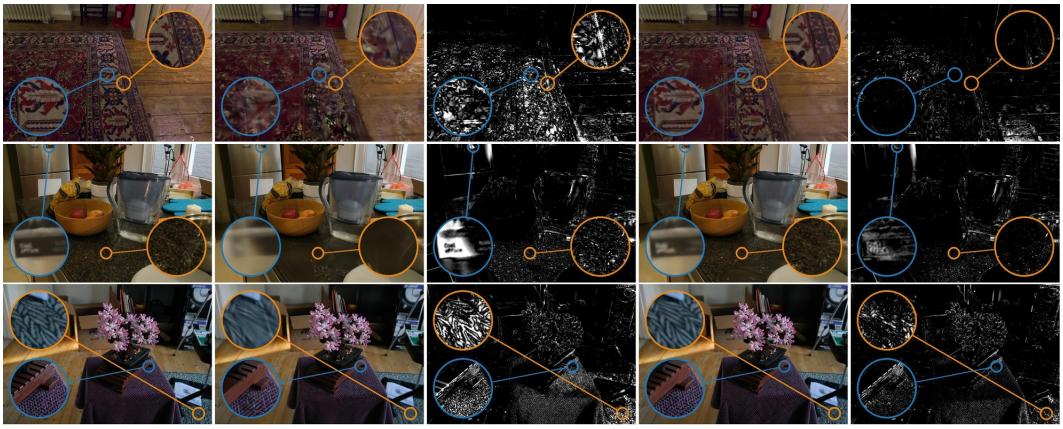
Table 2: Controller-statistic ablation on Mip-NeRF360 (100k cap). Within-sweep gains over the matched opacity-controller rerun.

Variant	Selection statistic	ΔPSNR (dB)↑
(a) MCMC baseline	α_i	–
(b) Position gradient	$\|\nabla_{\mu_i} \widehat{F}_t\|$	+0.04
(c) Opacity gradient (post-act.)	$-\partial \widehat{F}_t / \partial \alpha_i$	+0.11
(d) Birth only	$(q_i)_+$	+0.28
(e) Death only	q_i	+0.05
(f) Ours (Full)	q_i	**+0.33**

additional +0.22 dB over the plain gradient under the matched protocol. Additional score diagnostics, including copy-gain, recycle-hit, and removal-agreement probes, are provided in Appendix J.

4.4 Qualitative behavior

Figure 2 gives representative crops, while Figure 4 adds absolute error maps across scenes and budgets. The matched comparisons show improved reconstruction of fine textures and boundaries with score-guided allocation.



Ground Truth **3DGS-MCMC** **L_1 Error** **Ours** **L_1 Error**

Figure 4: Qualitative comparison with absolute L_1 error maps. Rows, top to bottom: Dr. Johnson (1M), Counter (200k), and Bonsai (100k); all use random initialization. Brighter error-map pixels indicate larger error. Each row uses the same Gaussian cap, test view, and crop across methods.

4.5 Compaction without Post-Training Pruning

Post-training compaction methods reduce a larger trained model [8, 12, 31], whereas our method trains under a fixed cap. Table 3 compares our 500k models with external compact-model results from GHAP [29]. Mip-NeRF360 averages all nine scenes in this comparison, including Flowers and Treehill. Gaussian counts and training protocols differ across rows.

Our controller produces compact models directly during training. The matched-cap comparisons in Table 1 and the matched-controller ablation in Table 2 provide the controlled evidence for its quality gains; Table 3 provides additional context across training pipelines.

4.6 Discussion and Limitations

Transfer beyond MCMC. The first-variation score measures local mass utility at existing parent sites, while the backend determines how new capacity is realized. Replacing Vanilla 3DGS's position-gradient criterion with q_i, while retaining native cloning, splitting, and pruning, provides empirical evidence of transfer beyond MCMC. Protocols and results are provided in Appendix D.

Method	Tanks & Temples				Mip-NeRF360				DeepBlending			
	#G	PSNR↑	SSIM↑	LPIPS↓	#G	PSNR↑	SSIM↑	LPIPS↓	#G	PSNR↑	SSIM↑	LPIPS↓
Vanilla 3DGS [15]	1577	23.79	0.853	0.169	2627	27.55	0.813	0.221	2475	29.82	0.907	0.238
LightGaussian-20% [8]	315	22.49	0.779	0.271	526	26.35	0.765	0.288	495	28.01	0.873	0.315
PUP-3DGS-20% [12]	315	22.60	0.809	0.228	525	26.67	0.790	0.257	495	29.72	0.905	**0.248**
Trimming the Fat-20% [1]	315	22.50	0.819	0.232	524	26.49	0.781	0.280	494	29.08	0.900	0.272
MesonGS-20% [31]	314	20.70	0.822	0.207	527	25.01	0.776	0.262	496	28.70	0.897	0.262
3DGS+GHAP-20% [29]	314	23.62	0.835	0.212	527	26.97	0.788	0.275	496	29.86	**0.907**	0.252
3DGS-MCMC [17]	315	22.14	0.779	0.282	263	25.96	0.763	0.309	496	28.98	0.885	0.298
Ours	**500**	**24.05**	**0.857**	**0.178**	**500**	**27.61**	**0.815**	**0.238**	**500**	**30.04**	0.906	0.262

Table 3: Compact-model comparison with external results from GHAP [29]. Mip-NeRF360 averages all nine scenes here, including Flowers and Treehill. Ours uses a 500k cap; external counts and training protocols differ. Counts are in thousands. Colors denote 1st, 2nd, and 3rd among compact rows; Vanilla is excluded.

Limitations and future work. While the first-variation score provides a principled surrogate for mass allocation, several natural frontiers remain. First, as a scalar mass derivative evaluated at existing sites, q_i dictates *where* to allocate capacity rather than parameterizing the optimal geometric displacement of split children; extending this to joint geometry-mass proposal scoring is an exciting direction. Second, because the score tracks photometric residuals, severe cross-view inconsistencies (e.g., auto-exposure in unconstrained captures) can occasionally misguide allocation at high capacities (Appendices G.2 and E). Coupling variational density control with exposure- or uncertainty-aware models presents a promising solution. Finally, adaptive score schedules that decay update frequency as geometry stabilizes could further amortize the auxiliary backward-pass overhead (Appendix G.3).

5 Conclusion

We reframed 3DGS structural control from a collection of heuristics into a formal mass-allocation problem. By evaluating the first variation of the multiview reconstruction risk, we replaced heuristic cloning and pruning with a principled birth–death controller. Our experiments expose the fundamental limitation of standard 3DGS: proxy signals succeed when excess capacity hides their inefficiencies. When the primitive budget is constrained, querying the objective's mass derivative is essential to preserve complex geometry and thin structures. This demonstrates that 3DGS structural updates belong in the same rigorous empirical risk minimization framework as continuous parameters. Moving forward, future compact and dynamic radiance fields can bypass heuristic bottlenecks by letting the reconstruction risk govern structural growth.

Acknowledgments and Disclosure of Funding

Acknowledgments. The authors thank the anonymous reviewers and the Area Chair for their helpful feedback. This work is supported by grant VUNI.2526.CAIR.04.

Competing interests. The authors declare no competing interests.

References

[1] M. S. Ali, M. Qamar, S.-H. Bae, and E. Tartaglione. Trimming the fat: Efficient compression of 3d gaussian splats through pruning. In *British Machine Vision Conference*, 2024. URL `https://arxiv.org/abs/2406.18214`.

[2] B. Baranowski, S. Esposito, P. Gschoßmann, A. Chen, and A. Geiger. Conegs: Error-guided densification using pixel cones for improved reconstruction with fewer primitives. *arXiv preprint arXiv:2511.06810*, 2025.

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[4] G. Chen and W. Wang. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024. Accepted by ACM Computing Surveys.

[5] K. Cheng, X. Long, K. Yang, Y. Yao, W. Yin, Y. Ma, W. Wang, and X. Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 8123–8140. PMLR, 2024.

[6] M. Daniels and P. Rigollet. Splat regression models. In *International Conference on Learning Representations (ICLR)*, 2026. Poster.

[7] I. Deutsch, N. Moënne-Loccoz, G. State, and Z. Gojcic. Ppisp: Physically-plausible compensation and control of photometric variations in radiance field reconstruction, 2026.

[8] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, and Z. Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[9] L. Franke, L. Fink, and M. Stamminger. Vr-splatting: Foveated radiance field rendering via 3d gaussian splatting and neural points. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 8(1), 2025. doi: 10.1145/3728302.

[10] G. Grubert, F. Barthel, A. Hilsmann, and P. Eisert. Improving adaptive density control for 3d gaussian splatting. In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications – Volume 2: VISAPP*, pages 610–621. SCITEPRESS, 2025. doi: 10.5220/0013308500003912.

[11] A. Guédon and V. Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[12] A. Hanson, A. Tu, V. Singla, M. Jayawardhana, M. Zwicker, and T. Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2406.10219.

[13] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics*, 37(6):257:1–257:15, 2018. doi: 10.1145/3272127.3275084.

[14] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[15] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. SIGGRAPH 2023.

[16] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering: Official implementation. `https://github.com/graphdeco-inria/gaussian-splatting`, 2023.

[17] S. Kheradmand, D. Rebain, G. Sharma, W. Sun, Y.-C. Tseng, H. Isack, A. Kar, A. Tagliasacchi, and K. M. Yi. 3d gaussian splatting as markov chain monte carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2404.09591.

[18] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[19] D. Kotovenko, O. Grebenkova, and B. Ommer. Edgs: Eliminating densification for efficient convergence of 3dgs. *arXiv preprint arXiv:2504.13204*, 2025.

[20] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21719–21728, 2024.

[21] Y. Lu, D. Slepčev, and L. Wang. Birth–death dynamics for sampling: global convergence, approximations and their asymptotics. *Nonlinearity*, 36(11):5731–5772, 2023. doi: 10.1088/1361-6544/acf988.

[22] S. S. Mallick, R. Goel, B. Kerbl, F. V. Carrasco, M. Steinberger, and F. De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. doi: 10.1145/3680528.3687694.

[23] S. Niedermayr, J. Stumpfegger, and R. Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10349–10358, 2024.

[24] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[25] S. Rota Bulò, L. Porzi, and P. Kontschieder. Revising densification in gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024. arXiv:2404.06109.

[26] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20626–20636, 2023.

[27] Z. Shao, Z. Wang, Z. Li, D. Wang, X. Lin, Y. Zhang, M. Fan, and Z. Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[28] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations (ICLR)*, 2024.

[29] T. Wang, M. Li, G. Zeng, C. Meng, and Q. Zhang. Gaussian herding across pens: An optimal transport perspective on global gaussian reduction for 3dgs, 2025. URL `https://arxiv.org/abs/2506.09534`.

[30] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[31] S. Xie, W. Zhang, C. Tang, Y. Bai, R. Lu, S. Ge, and Z. Wang. Mesongs: Post-training compression of 3d gaussians via efficient attribute transformation. In *European Conference on Computer Vision*. Springer, 2024. doi: 10.1007/978-3-031-73414-4_25. URL `https://arxiv.org/abs/2409.09756`.

[32] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

[33] Z. Ye, W. Li, S. Liu, P. Qiao, and Y. Dou. Absgs: Recovering fine details for 3d gaussian splatting. In *ACM International Conference on Multimedia (ACM MM)*, 2024.

[34] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[35] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[36] Z. Zhang, W. Hu, Y. Lao, T. He, and H. Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

[37] L. Zhu, Z. Wang, J. Cui, Z. Jin, G. Lin, and L. Yu. Endogs: Deformable endoscopic tissues reconstruction with gaussian splatting. In *MICCAI Workshop on Embodied AI and Robotics for HealTHcare (EARTH)*, 2024.

[38] S. Zhu, G. Wang, X. Kong, D. Kong, and H. Wang. 3d gaussian splatting in robotics: A survey. *arXiv preprint arXiv:2410.12262*, 2024.

[39] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.

A Deferred Proofs from Section 3

A.1 Proof of Proposition 1

Proof. Fix $i \in \{1, \dots, M_t\}$. For each $n \in \mathcal{B}_t$, define

$$\varphi_n(\mathbf{m}) := F_n\big(f_{n,t}(\cdot; \mathbf{m})\big) = \ell_{\text{img}}\big(\Phi(\mathcal{G}_t^{\mathbf{m}}, \pi_n), I_n\big).$$

By assumption, $F_n : L^2(\Omega; \mathbb{R}^3) \to \mathbb{R}$ is Fréchet differentiable at $f_{n,t}(\cdot; \mathbf{1})$, with differential $DF_n[f_{n,t}(\cdot; \mathbf{1})]$. Moreover, $f_{n,t}(\cdot; \mathbf{m})$ is differentiable in $\mathbf{m}$ at $\mathbf{m} = \mathbf{1}$, so its i-th partial derivative

$$\xi_{i,n} := \left.\frac{\partial f_{n,t}(\cdot; \mathbf{m})}{\partial m_i}\right|_{\mathbf{m}=\mathbf{1}} \in L^2(\Omega; \mathbb{R}^3)$$

is well-defined. The chain rule in Banach spaces gives

$$\left.\frac{\partial \varphi_n(\mathbf{m})}{\partial m_i}\right|_{\mathbf{m}=\mathbf{1}} = DF_n\big[f_{n,t}(\cdot; \mathbf{1})\big](\xi_{i,n}).$$

By Definition 1, the Riesz representative of DF_n is $\delta F_n[f_{n,t}(\cdot; \mathbf{1})]$, hence

$$DF_n\big[f_{n,t}(\cdot; \mathbf{1})\big](\xi_{i,n}) = \int_\Omega \Big\langle \xi_{i,n}(x),\, \delta F_n\big[f_{n,t}(\cdot; \mathbf{1})\big](x) \Big\rangle\, dx.$$

Differentiating the finite sum in (3) term by term gives

$$\left.\frac{\partial \widehat{F}_t(\mathbf{m})}{\partial m_i}\right|_{\mathbf{m}=\mathbf{1}} = \frac{1}{B} \sum_{n \in \mathcal{B}_t} \int_\Omega \Big\langle \delta F_n\big[f_{n,t}(\cdot; \mathbf{1})\big](x),\ \left.\frac{\partial f_{n,t}(x; \mathbf{m})}{\partial m_i}\right|_{\mathbf{m}=\mathbf{1}} \Big\rangle\, dx.$$

Multiplying by -1 yields the claimed expression for $(q_t)_i$. □

A.2 Proof of Proposition 2

Proof. Let $x := \mathbf{1}$ and $y := \mathbf{1} + \delta\mathbf{m}$. Since F is differentiable on a neighbourhood containing the segment $\{x + \tau(y - x) : \tau \in [0, 1]\}$, the fundamental theorem of calculus yields

$$F(y) - F(x) = \int_0^1 \Big\langle \nabla F\big(x + \tau(y - x)\big),\, y - x \Big\rangle\, d\tau.$$

Add and subtract $\nabla F(x)$ inside the integrand:

$$F(y) - F(x) = \langle \nabla F(x),\, y - x \rangle + \int_0^1 \Big\langle \nabla F\big(x + \tau(y - x)\big) - \nabla F(x),\, y - x \Big\rangle\, d\tau.$$

By Cauchy–Schwarz and L_{sm}-smoothness,

$$\Big| \Big\langle \nabla F\big(x + \tau(y - x)\big) - \nabla F(x),\, y - x \Big\rangle \Big| \le L_{\text{sm}}\, \tau\, \|y - x\|_2^2.$$

Integrating τ from 0 to 1 gives

$$F(y) \le F(x) + \langle \nabla F(x),\, y - x \rangle + \frac{L_{\text{sm}}}{2} \|y - x\|_2^2.$$

Finally, $\nabla F(\mathbf{1}) = -q_t$ by definition of q_t, and $y - x = \delta\mathbf{m}$, which yields the stated bound. □

A.3 Proof of Proposition 3

Proof. Fix a view n and a pixel $x \in \Omega_{\text{pix}}$. For readability we suppress the dependence on (n, x) and write

$$a_k := a_k(x) = \alpha_k G_k(x) \in [0, 1], \qquad T_i := \prod_{k \prec_n i} (1 - a_k) \in [0, 1], \qquad w_i := T_i a_i.$$

Define tail colors recursively by choosing a terminal color $U_{M_t+1}(x) \in [0, 1]^3$ and setting

$$U_k := a_k\, c_k(\mathbf{d}_n) + (1 - a_k)\, U_{k+1}, \qquad k = M_t, \dots, 1.$$

Then $\hat{I}_n(x) = U_1$, and $U_k \in [0,1]^3$ for all k by backward induction.

Let $\hat{I}_n^{(-i)}$ denote the image obtained by setting $a_i = 0$ at every pixel. At the fixed pixel x, the corresponding tail recursion satisfies $U_k^{(-i)} = U_k$ for $k \geq i+1$ and

$$U_i^{(-i)} = U_{i+1}.$$

Because the transmittance before i depends only on Gaussians in front of i, it is unchanged by setting $a_i = 0$. Decomposing the pixel value into the front contribution and the tail starting at i gives

$$\hat{I}_n(x) = \sum_{k \prec_n i} w_k \, c_k(\mathbf{d}_n) + T_i U_i, \qquad \hat{I}_n^{(-i)}(x) = \sum_{k \prec_n i} w_k \, c_k(\mathbf{d}_n) + T_i U_i^{(-i)}.$$

Subtracting and using $U_i - U_i^{(-i)} = U_i - U_{i+1} = a_i(c_i(\mathbf{d}_n) - U_{i+1})$ yields

$$\hat{I}_n(x) - \hat{I}_n^{(-i)}(x) = T_i a_i \big(c_i(\mathbf{d}_n) - U_{i+1}\big) = w_i \big(c_i(\mathbf{d}_n) - U_{i+1}\big).$$

Since $G_i(x) \leq 1$ and $T_i \leq 1$, we have $w_i = T_i a_i \leq a_i \leq \alpha_i$. Since both $c_i(\mathbf{d}_n)$ and U_{i+1} lie in $[0,1]^3$,

$$\frac{1}{3} \big\| \hat{I}_n(x) - \hat{I}_n^{(-i)}(x) \big\|_1 \leq w_i \leq \alpha_i.$$

Averaging over $x \in \Omega_{\text{pix}}$ gives $\|\hat{I}_n - \hat{I}_n^{(-i)}\|_{1,\text{mean}} \leq \alpha_i$.

Finally, if $\ell_{\text{img}}(\cdot, I_n)$ is L_ℓ-Lipschitz with respect to $\|\cdot\|_{1,\text{mean}}$, then

$$\big|\ell_{\text{img}}(\hat{I}_n, I_n) - \ell_{\text{img}}(\hat{I}_n^{(-i)}, I_n)\big| \leq L_\ell \, \|\hat{I}_n - \hat{I}_n^{(-i)}\|_{1,\text{mean}} \leq L_\ell \, \alpha_i,$$

as claimed. □

B Implementation and Reproducibility

This section gives the default hyperparameters used in all experiments. The main paper analyzes the variational birth–death decision; the finite relocation, optimizer, regularizers, and event schedule are inherited from the 3DGS-MCMC backend and are kept fixed in the matched comparisons.

Table 4: Experimental configuration and default hyperparameters.

Item	**Value**
Training iterations	30,000
Gaussian caps	100k, 200k, 500k, 1M, 2M
Initialization / seed	SfM or random initialization, seed 0
Controller interval Δ	40 iterations
Score batch size B	16 views
EMA coefficient ρ	0.9
Clipping quantile τ	0.99 absolute quantile
Dead-opacity threshold α_{death}	0.015
Birth-site weights	$(\bar{q}_t)_i$
Death policy	Lowest $(\bar{q}_t)_i$ within the low-opacity pool
Jitter scale η	1.0
Photometric loss	$(1-\lambda_{\text{dssim}})\ell_1 + \lambda_{\text{dssim}}(1-\text{SSIM})$
λ_{dssim}	0.2
Opacity regularizer weight λ_α	0.01
Scale regularizer weight λ_s	0.01
Optimizer	3DGS Adam parameter groups with unchanged learning rates
GPU	NVIDIA RTX 4090
CUDA / PyTorch	CUDA 12.8 / PyTorch 2.11.0
Evaluation metrics	PSNR, SSIM, LPIPS

C Matched 3DGS-MCMC Drop-in Comparison

The main text uses the matched 3DGS-MCMC comparison to isolate the effect of the structural statistic. Table 5 keeps the renderer, loss, optimizer, relocation kernel, schedule, initialization type, and Gaussian cap fixed; only the controller statistic changes from opacity to the first-variation score. The gains are largest in the low-budget regime, where the birth–death allocation decision is most constrained. At larger caps, the effect becomes dataset- and metric-dependent, consistent with the main-paper interpretation that additional capacity can partially mask allocation errors.

Table 5: Controlled drop-in comparison against our matched 3DGS-MCMC rerun. Positive deltas indicate improvement: ΔPSNR = Ours − MCMC, ΔSSIM = Ours − MCMC, and ΔLPIPS = MCMC − Ours.

Dataset	Cap	3DGS-MCMC rerun			Ours			Gain		
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	ΔPSNR	ΔSSIM	ΔLPIPS
Mip-NeRF360	100k	27.61	0.814	0.281	27.94	0.818	0.276	+0.33	+0.004	+0.005
	200k	28.04	0.835	0.254	28.37	0.841	0.249	+0.33	+0.005	+0.006
	500k	28.89	0.864	0.209	29.11	0.869	0.203	+0.22	+0.005	+0.006
	1M	29.35	0.878	0.185	29.49	0.883	0.178	+0.14	+0.005	+0.006
	2M	29.60	0.885	0.168	29.66	0.890	0.163	+0.07	+0.004	+0.005
Tanks & Temples	100k	23.21	0.826	0.232	23.60	0.831	0.225	+0.39	+0.005	+0.006
	200k	23.45	0.833	0.221	23.68	0.836	0.216	+0.23	+0.004	+0.005
	500k	24.06	0.854	0.184	24.05	0.857	0.178	-0.01	+0.003	+0.006
	1M	24.43	0.867	0.160	24.17	0.865	0.157	-0.26	-0.001	+0.003
	2M	24.66	0.875	0.141	24.21	0.870	0.143	-0.45	-0.005	-0.002
DeepBlending	100k	29.14	0.891	0.305	29.45	0.892	0.303	+0.32	+0.001	+0.002
	200k	29.71	0.899	0.283	29.72	0.899	0.284	+0.01	-0.000	-0.001
	500k	29.68	0.905	0.262	30.04	0.906	0.262	+0.35	+0.001	-0.001
	1M	29.94	0.909	0.249	30.14	0.911	0.248	+0.20	+0.002	+0.000
	2M	30.05	0.910	0.239	30.19	0.912	0.240	+0.14	+0.002	-0.001

D Transfer beyond 3DGS-MCMC

Local utility and spatial realization. The first-variation score measures the utility of rescaling a Gaussian's contribution at its current support, independently of the MCMC transition. The backend realizes the selected capacity through co-located relocation or Vanilla cloning and spatial splitting [15]. This separates parent selection from spatial realization: the score does not determine a split child's displacement.

Vanilla protocol. We replace Vanilla's position-gradient criterion with q_i, retaining native splitting geometry, pruning thresholds, and the 30k training schedule within each comparison. The uncapped experiment uses $q_i > 10^{-7}$ on three Mip-NeRF360 scenes. Separately, we evaluate parent selection with Vanilla clone/split geometry at a 500k cap on five scenes, and native clone/split/prune at a 100k cap on Bicycle.

Table 6: Uncapped Vanilla 3DGS transfer. Each quality entry reports PSNR↑ / SSIM↑ / LPIPS↓; final Gaussian counts are in millions.

	Reconstruction quality		Final #G (M)	
Scene	Vanilla	Vanilla + q_i	Vanilla	+ q_i
Bicycle	25.13 / 0.747 / 0.242	25.29 / 0.774 / 0.202	4.769	3.271
Bonsai	32.46 / 0.947 / 0.180	32.67 / 0.952 / 0.151	2.167	1.064
Flowers	21.36 / 0.589 / 0.359	21.61 / 0.627 / 0.278	3.206	2.850

Results. Table 6 shows improved quality with fewer final Gaussians; it is not a matched-population comparison. Table 7 shows gains across all metrics on the five 500k scenes. At 100k on Bicycle, SSIM and LPIPS improve while PSNR decreases slightly. This supports transfer, not uniform superiority.

Table 7: Capped Vanilla transfer. Entries report PSNR↑ / SSIM↑ / LPIPS↓; the mean uses the five 500k scenes only. Final Bicycle/100k populations are 99,812 (position gradient) and 99,910 (q_i).

Scene	Position gradient	First-variation score q_i
500k parent-selection diagnostic		
Bonsai	29.81 / 0.908 / 0.277	30.20 / 0.912 / 0.266
Counter	27.67 / 0.874 / 0.267	27.95 / 0.877 / 0.261
Garden	24.51 / 0.689 / 0.361	24.58 / 0.691 / 0.357
Kitchen	28.05 / 0.878 / 0.218	28.94 / 0.883 / 0.210
Room	30.17 / 0.894 / 0.277	30.67 / 0.897 / 0.273
Mean	28.04 / 0.849 / 0.280	28.47 / 0.852 / 0.274
100k native clone/split/prune		
Bicycle	23.12 / 0.580 / 0.445	23.03 / 0.605 / 0.414

Scope and future work. A scalar mass derivative does not evaluate the full geometric change of a finite split or the child's eventual reconstruction benefit. The Bicycle trade-off is consistent with this distinction, but does not establish displacement as its cause. Future work could score finite proposals jointly with their geometry, rather than parent mass alone.

E High-Capacity Overfitting and Structural-Update Cutoff

To test the effect of late structural updates, we stop birth–death relocations at iteration 15k while continuing continuous parameter optimization until 30k. Both variants reach their respective Gaussian caps before the cutoff, and other settings are held fixed within this diagnostic. Table 8 reports the Train results.

Table 8: Structural-update cutoff on Tanks & Temples Train. Continuous optimization runs for 30k iterations in both variants. These are within-diagnostic comparisons, not replacements for the default benchmark results.

Cap	Structural updates	PSNR↑	SSIM↑	LPIPS↓
500k	Full updates	21.88	0.820	0.216
500k	Stop at 15k	22.36	0.823	0.217
1M	Full updates	22.06	0.835	0.190
1M	Stop at 15k	22.24	0.831	0.196

Stopping structural updates improves PSNR in these runs, while some SSIM and LPIPS values worsen. This supports a contribution from late structural updates, but does not establish a complete remedy or isolate photometric variation as the sole cause. We retain the default schedule for the main benchmark and treat the cutoff as a diagnostic variant. An adaptive schedule based on cross-view score reliability is a direction for future work.

F Extended Comparison to Compact 3DGS

As introduced in Section 4.5, our WFR-derived controller intrinsically prevents the primitive bloat typical of standard 3DGS. Rather than training a massive, computationally expensive model only to prune it post-hoc, our method allocates finite capacity efficiently from scratch. This section provides an extended comparison against recent compaction baselines in the extreme low-primitive regime ($\sim$10% retention) across all three datasets.

Protocol distinction. Table 9 contextualizes our results against representative compact 3DGS baselines reported by GHAP [29]. The experimental protocols differ fundamentally: baselines like LightGaussian [8], PUP-3DGS [12], and MesonGS [31] prune and compress a fully trained backbone. In contrast, our method trains directly from scratch under a strict Gaussian cap. We include this

comparison to demonstrate a key point: principled, first-variation allocation during training achieves highly competitive compactness without the wasted overhead of an over-parameterized initial phase.

Table 9: Extreme low-primitive comparison (∼10% retention) on Mip-NeRF360, Tanks & Temples, and DeepBlending. External compact-baseline values are from GHAP [29]. Vanilla 3DGS is shown only as a full-model reference and is not included in the rank highlighting. Rank colors are computed only among the low-primitive rows: best, second best and third best. Higher PSNR/SSIM and lower LPIPS are better. All Gaussian counts are in thousands.

Method	Tanks & Temples				Mip-NeRF360				DeepBlending			
	#G	PSNR↑	SSIM↑	LPIPS↓	#G	PSNR↑	SSIM↑	LPIPS↓	#G	PSNR↑	SSIM↑	LPIPS↓
Vanilla 3DGS [15]	1577	23.79	0.853	0.169	2627	27.55	0.813	0.221	2475	29.82	0.907	0.238
LightGaussian [8]	158	22.11	0.756	0.306	263	25.67	0.735	0.331	248	28.01	0.869	0.327
PUP-3DGS [12]	158	21.52	0.767	0.280	262	25.33	0.753	0.309	248	29.15	0.895	0.274
Trimming the Fat [1]	156	21.54	0.776	0.293	263	25.26	0.731	0.343	247	28.06	0.887	0.302
MesonGS [31]	157	20.71	0.811	**0.208**	263	24.92	0.773	**0.264**	248	28.69	0.896	**0.264**
3DGS+GHAP [29]	157	23.31	0.818	0.242	263	26.40	0.764	0.314	248	29.65	**0.905**	**0.264**
3DGS-MCMC [17]	157	22.14	0.779	0.282	263	25.96	0.763	0.309	248	28.98	0.885	0.298
Ours	**200**	**23.68**	**0.836**	0.216	**200**	**26.93**	**0.783**	0.287	**200**	**29.72**	0.899	0.284

Extreme low-primitive comparison. On the Mip-NeRF360 benchmark, our method uses significantly fewer primitives than the baselines (200k vs. ∼263k) yet obtains the highest PSNR and SSIM among the low-primitive models. This trend holds across Tanks & Temples and DeepBlending, where our model delivers the best PSNR and competitive SSIM. While post-training attribute transformation methods (e.g., MesonGS) achieve lower LPIPS in specific settings, our controller establishes a highly competitive distortion-oriented reconstruction frontier straight out of training.

Higher-budget comparison. The equivalent comparison for the 500k-budget regime (∼20% retention) is provided in Section 4.5 of the main text. Together, these extreme and higher-budget comparisons reinforce the central thesis of the paper: when the representation is constrained to a compact regime, the WFR mass derivative provides a vastly superior structural statistic for capacity allocation than standard heuristics.

G Discussion

G.1 Mass score and opacity-gradient parameterization

The proposed mass-coordinate score is closely related to standard opacity gradients, but it provides a more appropriate coordinate space for structural allocation.

In the frozen score pass, the auxiliary coordinate m_i modulates the renderer through the opacity footprint: $a_i(x;\mathbf{m}) = m_i\alpha_i G_i(x)$. Suppressing the view index and batch average for readability, the chain rule yields a direct equivalence to the post-activation opacity derivative:

$$q_i = -\frac{\partial \widehat{F}_t}{\partial m_i}\bigg|_{\mathbf{m}=\mathbf{1}} = -\sum_x \frac{\partial \widehat{F}_t}{\partial a_i(x)}\alpha_i G_i(x) = -\alpha_i \frac{\partial \widehat{F}_t}{\partial \alpha_i}.$$

This scaling by α_i explains the practical stability of the mass-coordinate score. The raw opacity derivative $\partial \widehat{F}_t/\partial\alpha_i$ evaluates a scalar perturbation without accounting for the primitive's current presence in the scene. In practice, raw gradients often spike on nearly invisible "floaters" attempting to fit high-frequency noise. By measuring the relative rescaling of the primitive's mass, the factor α_i acts as an objective-level confidence weight. A nearly transparent primitive with a noisy gradient is correctly suppressed, forcing the controller to prioritize structurally established Gaussians for birth sites. For death decisions, the logic inverts cleanly: we first filter for safe slots via the low-opacity pool, then use the score to cull the slots whose removal will least impact the reconstruction loss.

Crucially, the mass score also bypasses a hidden artifact in standard continuous optimization. 3DGS typically optimizes opacity via a logit ζ_i, where $\alpha_i = \sigma(\zeta_i)$. The gradient is therefore:

$$\frac{\partial \widehat{F}_t}{\partial \zeta_i} = \alpha_i(1-\alpha_i)\frac{\partial \widehat{F}_t}{\partial \alpha_i}.$$

Because of the $(1 - \alpha_i)$ factor, logit gradients saturate and vanish for highly opaque, solid Gaussians ($\alpha_i \approx 1$). If a density controller relied on the optimizer's internal logit gradient to identify structural candidates, it would inherently penalize the most established, reliable primitives in the scene.

Because our mass score q_i is derived directly from the linear mass coordinate m_i, it measures the true first-order descent direction, isolated from the saturation artifacts of the activation function. We therefore use q_i exclusively for discrete birth–death selection, leaving continuous opacity optimization to the inherited 3DGS-MCMC backend.

G.2 High-budget behavior on Tanks & Temples Train

At larger capacities, our method occasionally trails the opacity-driven 3DGS-MCMC baseline on Tanks & Temples. Table 10 isolates the main source of this variance. The high-budget regression is concentrated on *Train*: at 2M Gaussians, *Train* has 13 severe views where our method trails by more than 1 dB, while *Truck* has none. Mip-NeRF360 remains stable overall at the same capacity.

Table 10: Diagnostic evidence for the high-budget Tanks & Temples Train behavior. ΔPSNR denotes Ours minus the matched 3DGS-MCMC rerun. The exposure-compensation row reruns both 3DGS-MCMC and Ours under the same exposure-compensation protocol; for Train at 2M, MCMC/Ours PSNR changes from $22.95/22.10$ to $24.95/24.65$. This is a paired diagnostic, not a replacement for the default main-table protocol. Severe views denote $\Delta\text{PSNR} < -1$ dB.

Setting	Protocol	Mean ΔPSNR	Severe views	View-luma std.	Mean bright-pixel frac.
Train, 2M	Default	-0.85	$13/38$	0.088	0.094
Train, 2M	Exp. comp.	-0.29	$6/38$	–	–
Truck, 2M	Default	-0.05	$0/32$	0.061	0.048
Mip-NeRF360 avg., 2M	Default	$+0.06$	$3/246$	0.062	0.031

This pattern is consistent with the capture conditions. Mip-NeRF360 was captured with fixed exposure and controlled lighting to reduce photometric variation [3]. Tanks & Temples is a realistic outside-the-lab benchmark, and *Train* was captured under sunny illumination with automatic ISO to handle varying exposure [18]. Our image statistics confirm this trend: *Train* exhibits higher view-level brightness variation and a larger bright-pixel fraction than *Truck* or the Mip-NeRF360 average.

This exposes a useful boundary of residual-driven density control. The first-variation score directly follows the empirical reconstruction loss. This is highly desirable when residuals correspond to missing geometry, underrepresented texture, or thin structures. However, at high capacity on *Train*, many easy structural errors have already been resolved. The remaining residuals are heavily weighted toward exposure changes, saturated sky regions, shadows, and view-dependent train surfaces. In that regime, a large image residual is not always reliable birth-site evidence. Robust radiance-field literature makes a related observation: lighting variation, shadows, and other non-persistent effects produce view-dependent floaters when treated as ordinary reconstruction residuals [26]. Recent image-formation work similarly shows that photometric variation can be absorbed as geometric artifacts when no compensation model is present [7].

The exposure-compensation diagnostic in Figure 5 narrows the performance gap, supporting the interpretation that photometric inconsistency contributes to the high-budget regression. Once the residual is made photometrically consistent, our method gains $+2.55$ dB—significantly outpacing MCMC's $+2.00$ dB recovery. (We retain this as a diagnostic because the official 3DGS exposure setting alters the standard train/test split, breaking comparability with standard benchmarks [16]).

G.3 Compute trade-off and scalability

Replacing heuristic proxies with a WFR derivative requires an auxiliary backward pass. In our default protocol, this frozen score pass adds a $1.21\times$ to $1.27\times$ overhead to the total training time. We view this as a highly favorable trade-off. By paying a fixed computational cost during training, the controller secures higher reconstruction quality under strict primitive caps. Because the final model retains the same Gaussian cap and introduces no additional scoring operation at inference, the method preserves the inference-time architecture of the baseline.

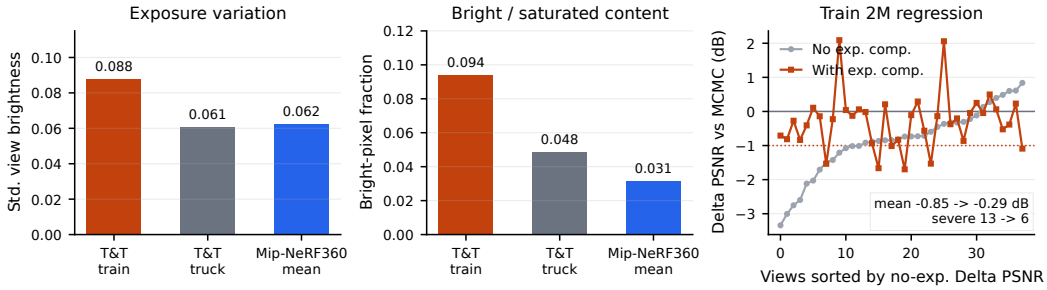


Figure 5: Photometric diagnostic for the high-budget *Train* case. Left and middle: *Train* has higher view-level brightness variation and a larger bright-pixel fraction than *Truck* and the Mip-NeRF360 average. Right: paired exposure-compensation reruns for both 3DGS-MCMC and Ours reduce the mean Train/SfM/2M gap from -0.85 dB to -0.29 dB and reduce severe views from 13 to 6. This supports the interpretation that the high-budget regression is tied to residual reliability under photometric variation, not to a general breakdown of first-variation allocation.

H Full Fixed-Budget Comparisons

This section expands the fixed-budget comparison from the main text. The wide table reports all dataset-level averages, including the 2M cap and random initialization rows. The per-scene tables show where the aggregate gains come from and make clear which scenes and metrics are responsible for non-uniform behavior at larger budgets.

Table 11: Full fixed-budget comparison across datasets, capacities, and methods. Each cell reports PSNR↑ / SSIM↑ / LPIPS↓. Rows marked † are published ConeGS benchmark values; Improved ADC and Ours are evaluated at the same Gaussian caps.

Dataset	Method	100k (P / S / L)	500k (P / S / L)	1M (P / S / L)	2M (P / S / L)
Mip-NeRF360	3DGS [15] (SfM)†	23.61 / 0.693 / 0.413	28.22 / 0.821 / 0.260	28.71 / 0.846 / 0.222	29.01 / 0.861 / 0.200
	Improved ADC [10]	26.34 / 0.781 / 0.322	28.46 / 0.843 / 0.232	28.94 / 0.858 / 0.205	29.21 / 0.871 / 0.184
	3DGS-MCMC [17] (SfM)†	27.06 / 0.800 / 0.303	28.82 / 0.861 / 0.214	29.23 / 0.875 / 0.190	29.52 / 0.883 / 0.172
	EDGS [19]†	27.09 / 0.798 / 0.296	28.82 / 0.865 / 0.193	29.18 / 0.877 / 0.168	29.38 / 0.883 / 0.155
	ConeGS [2]†	27.74 / 0.809 / 0.285	29.08 / 0.870 / 0.190	29.37 / 0.880 / 0.168	29.43 / 0.881 / 0.158
	Ours (SfM)	27.94 / 0.818 / 0.276	29.11 / 0.869 / 0.203	29.49 / 0.883 / 0.178	29.66 / 0.890 / 0.163
	Ours (Random)	26.71 / 0.769 / 0.337	28.79 / 0.856 / 0.222	29.19 / 0.874 / 0.190	29.42 / 0.885 / 0.170
Tanks & Temples	3DGS [15] (SfM)†	22.38 / 0.774 / 0.333	23.54 / 0.816 / 0.265	23.60 / 0.826 / 0.245	23.84 / 0.831 / 0.235
	Improved ADC [10]	22.08 / 0.784 / 0.280	23.66 / 0.842 / 0.192	24.06 / 0.855 / 0.168	24.11 / 0.862 / 0.153
	3DGS-MCMC [17] (SfM)†	22.50 / 0.780 / 0.332	23.68 / 0.825 / 0.259	23.93 / 0.836 / 0.233	24.11 / 0.846 / 0.211
	EDGS [19]†	22.32 / 0.777 / 0.324	23.47 / 0.831 / 0.229	23.66 / 0.842 / 0.200	23.47 / 0.846 / 0.183
	ConeGS [2]†	23.12 / 0.791 / 0.310	23.69 / 0.829 / 0.229	23.70 / 0.835 / 0.207	23.69 / 0.837 / 0.192
	Ours (SfM)	23.60 / 0.831 / 0.225	24.05 / 0.857 / 0.178	24.17 / 0.865 / 0.157	24.21 / 0.870 / 0.143
	Ours (Random)	22.19 / 0.778 / 0.287	23.58 / 0.840 / 0.197	23.14 / 0.828 / 0.200	23.96 / 0.857 / 0.160
DeepBlending	3DGS [15] (SfM)†	24.65 / 0.827 / 0.412	29.25 / 0.882 / 0.302	28.93 / 0.881 / 0.292	29.04 / 0.883 / 0.281
	Improved ADC [10]	28.25 / 0.885 / 0.315	28.95 / 0.896 / 0.260	29.18 / 0.902 / 0.248	29.60 / 0.905 / 0.239
	3DGS-MCMC [17] (SfM)†	28.94 / 0.876 / 0.333	29.44 / 0.887 / 0.308	29.67 / 0.887 / 0.288	29.53 / 0.886 / 0.279
	EDGS [19]†	28.43 / 0.872 / 0.337	29.33 / 0.889 / 0.286	29.43 / 0.889 / 0.271	29.38 / 0.887 / 0.258
	ConeGS [2]†	29.44 / 0.880 / 0.328	29.86 / 0.891 / 0.285	29.69 / 0.889 / 0.273	29.76 / 0.886 / 0.261
	Ours (SfM)	29.45 / 0.892 / 0.303	30.04 / 0.906 / 0.262	30.14 / 0.911 / 0.248	30.19 / 0.912 / 0.240
	Ours (Random)	28.32 / 0.875 / 0.325	29.74 / 0.899 / 0.274	29.96 / 0.909 / 0.254	29.92 / 0.909 / 0.246

The 100k per-scene results in Table 12 support the low-budget claim from the main text: when capacity is tight, the first-variation controller improves most scenes and gives especially strong LPIPS gains on Tanks & Temples and DeepBlending. Tables 13 and 14 show the larger-cap regime. The method remains strong on PSNR and SSIM, while LPIPS is more metric-dependent on Mip-NeRF360, matching the calibrated interpretation in the main text.

Table 12: Per-scene fixed-budget comparison at 100k Gaussians. Each cell reports PSNR / SSIM / LPIPS. Baseline values for ConeGS, 3DGS (SfM), 3DGS-MCMC (SfM), and EDGS are published ConeGS per-scene results; Ours is evaluated at the same cap.

Scene	ConeGS†	3DGS (SfM)†	3DGS-MCMC (SfM)†	EDGS†	**Ours**
			Mip-NeRF360		
Bicycle	24.05 / 0.652 / 0.379	20.99 / 0.470 / 0.543	23.34 / 0.630 / 0.400	23.68 / 0.635 / 0.393	**23.57 / 0.636 / 0.393**
Garden	24.69 / 0.709 / 0.336	22.87 / 0.589 / 0.445	24.51 / 0.710 / 0.350	24.47 / 0.706 / 0.341	**25.12 / 0.747 / 0.306**
Stump	25.71 / 0.707 / 0.351	22.79 / 0.547 / 0.498	25.16 / 0.680 / 0.370	25.08 / 0.673 / 0.377	**25.41 / 0.695 / 0.363**
Room	31.09 / 0.903 / 0.254	27.02 / 0.859 / 0.328	30.37 / 0.900 / 0.270	30.08 / 0.896 / 0.265	**30.94 / 0.903 / 0.261**
Counter	28.22 / 0.879 / 0.251	25.64 / 0.845 / 0.307	27.83 / 0.880 / 0.260	27.77 / 0.877 / 0.251	**28.44 / 0.895 / 0.232**
Kitchen	29.73 / 0.896 / 0.183	20.53 / 0.671 / 0.440	28.37 / 0.890 / 0.210	28.84 / 0.891 / 0.186	**30.63 / 0.917 / 0.152**
Bonsai	30.67 / 0.920 / 0.241	25.41 / 0.866 / 0.331	29.84 / 0.910 / 0.260	29.73 / 0.906 / 0.256	**31.48 / 0.934 / 0.225**
			Tanks & Temples		
Truck	24.54 / 0.822 / 0.291	23.59 / 0.803 / 0.318	23.86 / 0.810 / 0.316	23.33 / 0.799 / 0.314	**25.12 / 0.861 / 0.200**
Train	21.70 / 0.759 / 0.328	21.17 / 0.744 / 0.349	21.13 / 0.749 / 0.347	21.31 / 0.756 / 0.334	**22.08 / 0.801 / 0.251**
			DeepBlending		
Dr Johnson	28.78 / 0.875 / 0.338	23.58 / 0.808 / 0.440	28.16 / 0.869 / 0.343	27.66 / 0.865 / 0.347	**29.22 / 0.887 / 0.298**
Playroom	30.09 / 0.885 / 0.318	25.72 / 0.845 / 0.384	29.72 / 0.883 / 0.323	29.19 / 0.879 / 0.327	**29.68 / 0.897 / 0.308**

Table 13: Per-scene fixed-budget comparison at 500k Gaussians. Each cell reports PSNR / SSIM / LPIPS. Baseline values for ConeGS, 3DGS (SfM), 3DGS-MCMC (SfM), and EDGS are published ConeGS per-scene results; Ours is evaluated at the same cap.

Scene	ConeGS†	3DGS (SfM)†	3DGS-MCMC (SfM)†	EDGS†	**Ours**
			Mip-NeRF360		
Bicycle	25.25 / 0.758 / 0.242	24.06 / 0.651 / 0.370	24.90 / 0.740 / 0.282	24.95 / 0.751 / 0.251	**25.14 / 0.753 / 0.265**
Garden	26.87 / 0.837 / 0.157	25.16 / 0.743 / 0.296	26.36 / 0.823 / 0.189	26.49 / 0.833 / 0.161	**26.79 / 0.837 / 0.170**
Stump	27.03 / 0.793 / 0.215	25.27 / 0.688 / 0.348	26.81 / 0.779 / 0.250	26.50 / 0.768 / 0.237	**27.21 / 0.795 / 0.229**
Room	32.02 / 0.924 / 0.205	31.47 / 0.913 / 0.234	31.72 / 0.921 / 0.221	31.36 / 0.923 / 0.203	**31.84 / 0.922 / 0.219**
Counter	28.87 / 0.906 / 0.194	28.86 / 0.901 / 0.213	28.95 / 0.907 / 0.205	29.05 / 0.912 / 0.184	**29.06 / 0.911 / 0.200**
Kitchen	31.38 / 0.929 / 0.124	30.76 / 0.918 / 0.143	31.04 / 0.921 / 0.141	31.35 / 0.928 / 0.122	**31.57 / 0.926 / 0.136**
Bonsai	32.16 / 0.944 / 0.191	31.95 / 0.936 / 0.218	31.97 / 0.939 / 0.210	32.02 / 0.942 / 0.191	**32.17 / 0.942 / 0.205**
			Tanks & Temples		
Truck	25.41 / 0.858 / 0.204	24.80 / 0.842 / 0.253	25.24 / 0.851 / 0.242	24.66 / 0.850 / 0.212	**26.03 / 0.888 / 0.145**
Train	21.97 / 0.799 / 0.253	22.28 / 0.790 / 0.277	22.12 / 0.798 / 0.275	22.27 / 0.812 / 0.246	**22.07 / 0.826 / 0.211**
			DeepBlending		
Dr Johnson	29.20 / 0.890 / 0.291	28.85 / 0.881 / 0.307	28.71 / 0.884 / 0.313	28.60 / 0.888 / 0.292	**29.70 / 0.902 / 0.255**
Playroom	30.51 / 0.892 / 0.279	29.66 / 0.883 / 0.298	30.16 / 0.889 / 0.303	30.06 / 0.890 / 0.281	**30.38 / 0.911 / 0.270**

Table 14: Per-scene fixed-budget comparison at 1M Gaussians. Each cell reports PSNR / SSIM / LPIPS. Baseline values for ConeGS, 3DGS (SfM), 3DGS-MCMC (SfM), and EDGS are published ConeGS per-scene results; Ours is evaluated at the same cap.

Scene	ConeGS†	3DGS (SfM)†	3DGS-MCMC (SfM)†	EDGS†	**Ours**
			Mip-NeRF360		
Bicycle	25.45 / 0.778 / 0.198	24.40 / 0.688 / 0.325	25.34 / 0.768 / 0.238	25.26 / 0.778 / 0.199	**25.51 / 0.780 / 0.220**
Garden	27.42 / 0.861 / 0.115	26.70 / 0.827 / 0.172	26.83 / 0.847 / 0.147	27.05 / 0.857 / 0.118	**27.45 / 0.862 / 0.126**
Stump	27.23 / 0.802 / 0.184	25.64 / 0.719 / 0.301	27.21 / 0.798 / 0.213	26.71 / 0.783 / 0.203	**27.56 / 0.814 / 0.193**
Room	32.30 / 0.928 / 0.196	31.62 / 0.918 / 0.221	32.09 / 0.926 / 0.208	31.71 / 0.928 / 0.191	**32.10 / 0.927 / 0.205**
Counter	29.21 / 0.911 / 0.181	29.11 / 0.907 / 0.201	29.24 / 0.913 / 0.191	29.27 / 0.917 / 0.171	**29.34 / 0.917 / 0.187**
Kitchen	31.66 / 0.932 / 0.117	31.18 / 0.924 / 0.131	31.44 / 0.927 / 0.130	31.83 / 0.933 / 0.114	**32.00 / 0.932 / 0.124**
Bonsai	32.33 / 0.945 / 0.183	32.34 / 0.941 / 0.205	32.46 / 0.944 / 0.199	32.44 / 0.947 / 0.181	**32.49 / 0.947 / 0.194**
			Tanks & Temples		
Truck	25.34 / 0.864 / 0.181	25.06 / 0.851 / 0.234	25.57 / 0.862 / 0.217	25.02 / 0.862 / 0.181	**26.21 / 0.897 / 0.120**
Train	22.05 / 0.805 / 0.232	22.13 / 0.801 / 0.255	22.29 / 0.811 / 0.249	22.29 / 0.823 / 0.220	**22.12 / 0.834 / 0.194**
			DeepBlending		
Dr Johnson	28.98 / 0.886 / 0.282	28.85 / 0.884 / 0.293	29.06 / 0.884 / 0.291	28.75 / 0.890 / 0.277	**29.73 / 0.907 / 0.242**
Playroom	30.39 / 0.891 / 0.264	29.02 / 0.877 / 0.290	30.28 / 0.890 / 0.284	30.11 / 0.889 / 0.265	**30.55 / 0.915 / 0.255**

I Budget Curves and Runtime

This section supports the main-paper claim that the effect of the controller is largest when the Gaussian budget is tight and reports the training overhead introduced by the score pass. Figure 6 extends the main PSNR–budget frontier to PSNR, SSIM, and LPIPS under both SfM and random initialization.

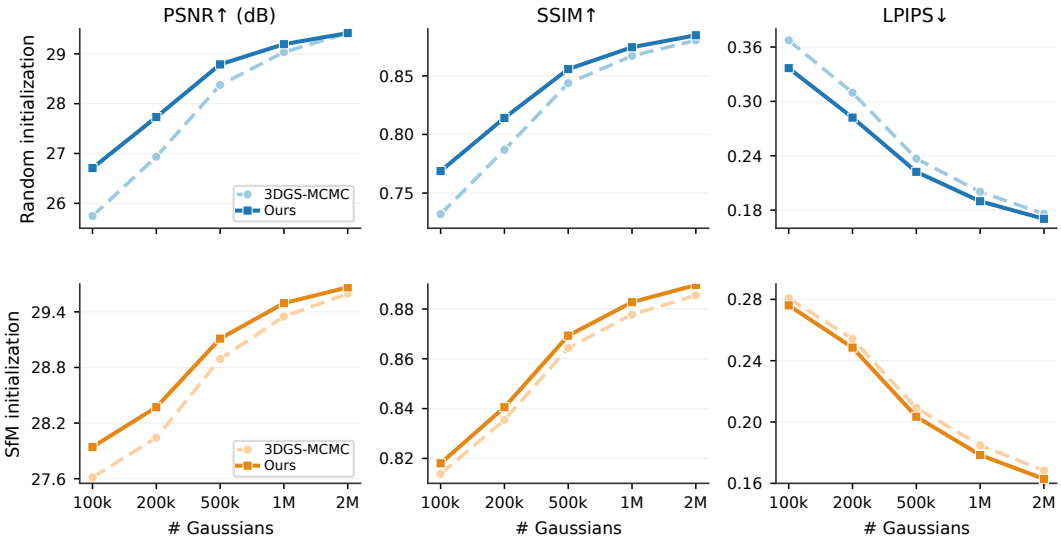


Figure 6: Quality as a function of Gaussian budget on the Mip-NeRF360 comparison set under random and SfM initialization. Each point is a seed-0 dataset average at a fixed Gaussian cap; error bands are not shown because matched multi-seed runs are not available for this sweep. Panels use tight y-axis ranges within each initialization and metric to make the matched Ours-vs-3DGS-MCMC gap readable. Across PSNR, SSIM, and LPIPS, the proposed controller improves over the matched 3DGS-MCMC backend, with the largest gains at lower budgets.

Figure 7 and Table 15 show the cost of the additional score pass. Training time increases by roughly 1.21–1.27× in this protocol, while inference cost remains tied to the same fixed Gaussian budget.

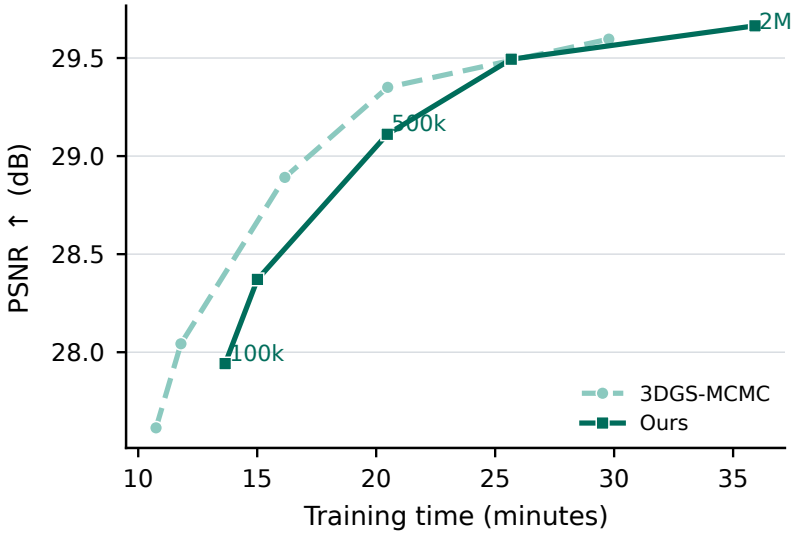


Figure 7: Quality–time tradeoff on the Mip-NeRF360 SfM setting. Ours improves PSNR at matched Gaussian caps while adding score-pass overhead during training; inference cost remains tied to the same fixed Gaussian budget.

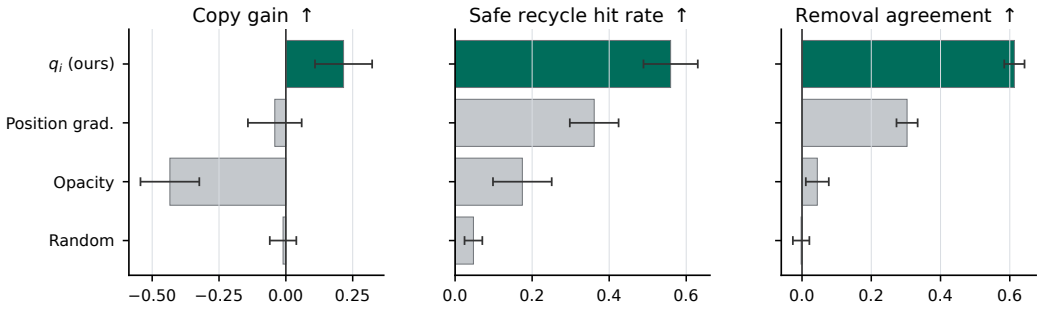
Table 15: Quality–time and computational efficiency values on the Mip-NeRF360 SfM setting. MCMC time is estimated by subtracting measured score-pass time from the matched Ours wall-clock logs. Peak memory and FPS are omitted because complete matched measurements were not logged for this timing protocol.

Cap	Method	PSNR↑	SSIM↑	LPIPS↓	Train time	Overhead	Score-pass	Base train
100k	3DGS-MCMC	27.61	0.814	0.281	10.75m	1.00×	–	–
	Ours	**27.94**	**0.818**	**0.276**	13.65m	1.27×	2.90m	10.75m
200k	3DGS-MCMC	28.04	0.835	0.254	11.79m	1.00×	–	–
	Ours	**28.37**	**0.841**	**0.249**	15.01m	1.27×	3.22m	11.79m
500k	3DGS-MCMC	28.89	0.864	0.209	16.16m	1.00×	–	–
	Ours	**29.11**	**0.869**	**0.203**	20.47m	1.27×	4.31m	16.16m
1M	3DGS-MCMC	29.35	0.878	0.185	20.48m	1.00×	–	–
	Ours	**29.49**	**0.883**	**0.178**	25.68m	1.25×	5.19m	20.48m
2M	3DGS-MCMC	29.60	0.885	0.168	29.78m	1.00×	–	–
	Ours	**29.66**	**0.890**	**0.163**	35.92m	1.21×	6.14m	29.78m

J Score Validation and Controller Sensitivity

The main ablation proves that the WFR statistic improves final reconstruction quality. This section isolates the score itself to verify that it acts as a true local descent surrogate. To test the score's predictive power, we design three empirical probes: **Copy gain** measures the immediate held-out loss reduction achieved by duplicating a high-score Gaussian, **Recycle hit rate** tracks how often removing a low-score Gaussian safely leaves the loss unaffected or improved, **Removal agreement** computes the rank correlation between our score and the actual finite damage caused by deleting a primitive.

Figure 8 shows the results. High-score Gaussians are demonstrably better birth sites. Low-score Gaussians are significantly safer to recycle. Crucially, the WFR score strongly agrees with actual finite removal damage, whereas opacity and position gradients show weak or near-random correlation.



Method	Copy gain↑	Recycle hit rate↑	Removal agree.↑
Ours score (q_i)	$\mathbf{0.216 \pm 0.107}$	$\mathbf{0.560 \pm 0.070}$	$\mathbf{0.613 \pm 0.029}$
Position grad.	-0.041 ± 0.101	0.361 ± 0.063	0.303 ± 0.031
Opacity	-0.434 ± 0.110	0.175 ± 0.076	0.044 ± 0.033
Random	-0.010 ± 0.050	0.048 ± 0.023	-0.003 ± 0.024

Figure 8: Utility-score validation for the fixed-budget controller. Top: choosing high-score Gaussians gives larger held-out loss reduction after a small test copy; choosing low-score Gaussians better identifies slots that are safe to recycle; and the score agrees with finite removal damage. Bottom: mean ± standard error over disjoint-view probes. Copy and recycle use 18 samples; removal agreement uses 16 finite Spearman correlations.

Table 16 separates same-view and disjoint-view validation. Same-view probes are naturally easier because the derivative and probe loss share the same views. However, performance on disjoint and held-out test views remains positive, confirming that the WFR score identifies true geometric utility rather than merely memorizing the local score batch.

Table 16: Effect of probe-view choice in utility-score validation. The main paper uses disjoint probe views to avoid measuring the derivative on the same views used to compute the score. Values report mean ± standard error over 18 validation samples.

Probe setting	**Spearman↑**	**Top-10% gain↑**	**Death P@10↑**
Same views as score batch	0.954 ± 0.007	1.442 ± 0.081	0.929 ± 0.026
Disjoint probe views	0.613 ± 0.029	0.216 ± 0.107	0.560 ± 0.070
Held-out test views	0.632 ± 0.026	0.600 ± 0.108	0.615 ± 0.055

K Additional Qualitative Results

The qualitative results illustrate the visual consequences of the matched controller comparison. They are not used as the primary evidence for the first-variation claim; the quantitative tables and controller ablations serve that role. The figures below show that the same fixed-cap controller improves fine structures, boundaries, and repeated textures across datasets and initialization regimes.

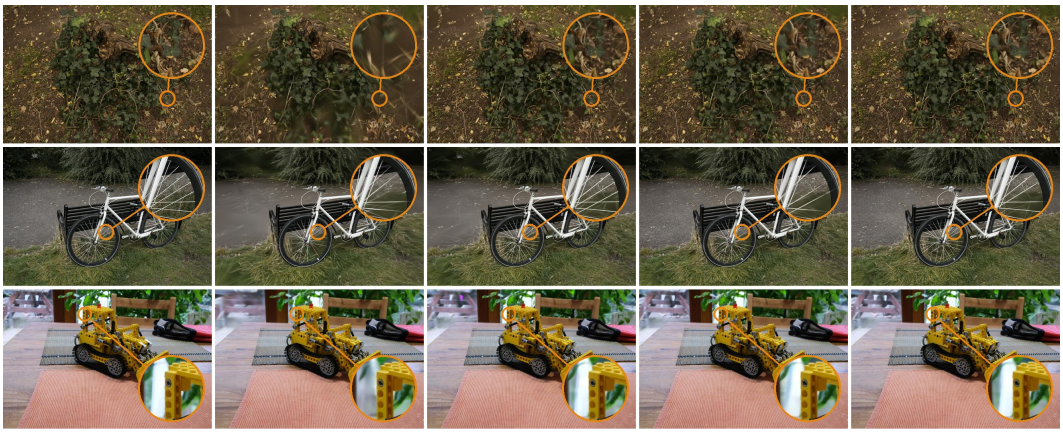
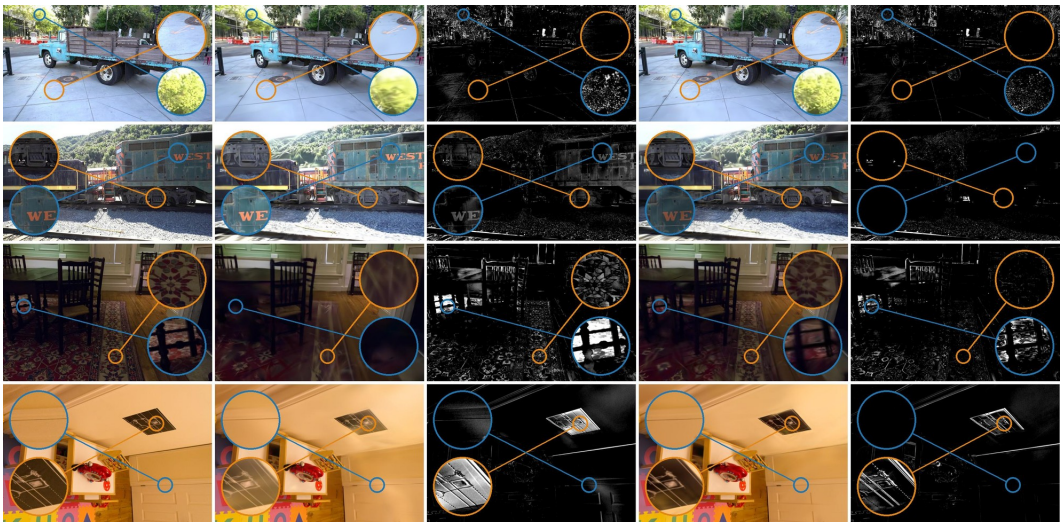


Figure 9: Qualitative comparison under random and SfM initialization. For each scene, we compare the matched 3DGS-MCMC controller and our score-guided controller under both initialization regimes. This comparison verifies that the visual gains are not tied to a particular initialization choice. All columns use the same Gaussian cap, test view, and crop coordinates within each row.



Ground Truth **3DGS-MCMC** L_1 **Error** **Ours** L_1 **Error**

Figure 10: Qualitative comparison across scenes and budgets. Each row shows one test-view crop with a fixed scene, Gaussian cap, and random initialization. From top to bottom, the rows correspond to *Dr. Johnson* (1M, random), *Garden* (500k, random), *Counter* (200k, random), *Bonsai* (100k, random), and *Bicycle* (100k, random). From left to right, we show the ground-truth image, the matched 3DGS-MCMC rendering, the per-pixel absolute L_1 error map $|I_{\mathrm{GT}} - I_{\mathrm{MCMC}}|$, our rendering, and the per-pixel absolute L_1 error map $|I_{\mathrm{GT}} - I_{\mathrm{ours}}|$. Brighter pixels in the error maps indicate larger reconstruction error. All methods use the same test view, scale, and crop coordinates within each row.



Ground Truth **3DGS-MCMC** L_1 **Error** **Ours** L_1 **Error**

Figure 11: All-scene qualitative comparison on Tanks & Temples and DeepBlending at a 200k-Gaussian cap with random initialization. Each row uses the same test view and crop coordinates for the ground truth, 3DGS-MCMC, and our method. These examples show that the score-guided controller also improves visual fidelity outside Mip-NeRF360, especially on thin structures, object boundaries, and repeated textures.



Ground Truth **3DGS-MCMC** **L_1 Error** **Ours** **L_1 Error**

Figure 12: All-scene qualitative comparison on Mip-NeRF360 at a 200k-Gaussian cap with random initialization. Each row corresponds to one Mip-NeRF360 scene, and all methods use identical test views and crop coordinates. This figure complements the representative examples in Figure 2 by showing that the visual improvement is not restricted to a small number of selected scenes.

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 Section 1 state the proposed first-variation birth–death controller, its fixed-budget 3DGS scope, and the main empirical claim. The method is formalized in Section 3, especially Eqs. (3)–(6), and the claims are supported by Table 1, Table 2, Figure 3, and Appendices C, H, and J.

   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 states that the score is a local mass-coordinate descent surrogate rather than a finite-step global guarantee in Section 3.2 and Section 3.3. Additional limitations and diagnostics are discussed in Appendix G.1, Appendix G.2, and Appendix G.3.

   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: Proposition 1, Proposition 2, and Proposition 3 state their assumptions in Section 3. Complete proofs are provided in Appendix A.

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: Section 4.1 specifies datasets, metrics, budgets, baselines, and the matched-comparison protocol. Appendix B gives the code-faithful algorithm, hyperparameters, optimizer/training details, hardware/software environment, and the anonymized supplementary code reproduces the proposed method.

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: An anonymized supplementary code package is submitted with training and evaluation scripts, configuration files, dependency instructions, and commands for reproducing the proposed method. The standard benchmark datasets are not redistributed; the supplementary README gives public data-access and preprocessing instructions, and rows marked † use cited published benchmark numbers as described in Section 4.1.

   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: Section 4.1 specifies the benchmark datasets, held-out metrics, Gaussian caps, baseline protocol, and training length. Appendix B, Table 4, and Algorithm 1 provide the controller interval, score batch size, EMA/clipping settings, opacity threshold, regularization weights, optimizer setting, GPU, CUDA/PyTorch versions, and evaluation metrics.

   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 main fixed-budget reconstruction sweeps report dataset-level and per-scene averages rather than multi-seed error bars, because full matched multi-seed 3DGS sweeps are computationally expensive. The paper nevertheless provides controlled matched reruns, per-scene tables in Appendix H, and mean $\pm$ standard-error score-validation probes in Figure 8 and Table 16.

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: [No]

   Justification: Appendix B reports the GPU and software environment, and Appendix I reports wall-clock training times and score-pass overhead in Figure 7 and Table 15. Complete peak-memory measurements and total project-level compute are not available for all experiments, and Table 15 explicitly discloses omitted peak-memory/FPS logs.

   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 research conforms to the NeurIPS Code of Ethics: it uses public benchmark datasets, does not collect new human-subject data, does not involve crowdsourcing, and submits anonymized supplementary code for review. The work is a reconstruction-method paper and does not release private data or high-risk pretrained models.

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: Section 1 notes positive applications such as immersive rendering, robotics, digital humans, and geometric reconstruction. Potential negative uses include more accurate reconstruction of privacy-sensitive scenes or surveillance settings; this work mitigates direct risk by using only public benchmarks and releasing research code rather than private scene data or sensitive pretrained assets.

    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 a high-risk pretrained model, image generator, language model, scraped dataset, or private dataset. The released asset is anonymized research code for training and evaluating a 3D reconstruction method on public 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 paper cites the original datasets, metrics, baselines, and codebases used in Section 4.1 and the references. The supplementary README lists the corresponding data/code sources, versions or commits when available, licenses or terms of use, and states that benchmark datasets are not redistributed.

    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: [Yes]

    Justification: The new asset is the anonymized supplementary source code. The code package includes documentation, environment setup, training/evaluation commands, configuration files, and license/terms information.

    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, human-subject experiments, or newly collected human-subject data. All experiments use existing public benchmark 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 research with human subjects, so IRB approval or equivalent review is not applicable.

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: LLMs are not an important, original, or non-standard component of the core method, experiments, code, or analysis. Any use of LLMs only for writing, editing, or formatting does not affect the methodology, scientific rigor, or originality of the research.

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.