

Gradient-Driven Natural Selection for Compact 3D Gaussian Splatting

Xiaobin Deng Qiuli Yu Changyu Diao* Min Li Duanqing Xu*
Zhejiang University

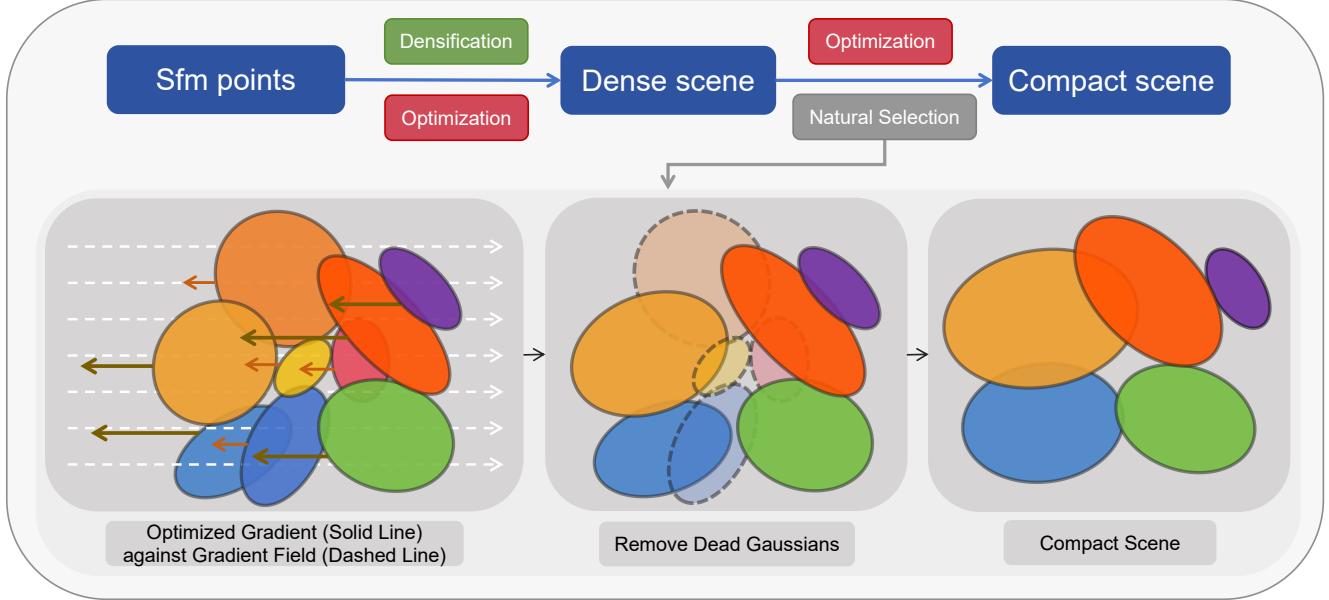


Figure 1. Our method’s pipeline consists of two parts. The upper section shows the full workflow: starting from sparse SfM point clouds, we densify and optimize to obtain a high-quality dense scene, which is then refined via a natural selection mechanism to produce a compact, high-fidelity representation. The lower section details this natural selection framework: a globally consistent regularization gradient field (dashed lines) is applied to all Gaussian opacities, guiding optimization gradients (solid lines) to identify and prune Gaussians whose opacity falls below a survival threshold. Those Gaussians with smaller optimized gradients will gradually decay in opacity until reaching the death threshold, after which they are permanently removed. The result is a high-quality, compact scene.

Abstract

3DGS employs a large number of Gaussian primitives to fit scenes, resulting in substantial storage and computational overhead. Existing pruning methods rely on manually designed criteria or introduce additional learnable parameters, yielding suboptimal results. To address this, we propose a natural selection inspired pruning framework that models survival pressure as a regularization gradient field applied to opacity, allowing the optimization gradients—driven by the goal of maximizing rendering quality—to autonomously determine which Gaussians to retain or prune. This process is fully learnable and requires no human intervention. We further introduce an opacity decay technique with a finite opacity prior, which accelerates the selection process without compromising pruning effectiveness. Compared to 3DGS, our method achieves over

0.6 dB PSNR gain under 15% budgets, establishing state-of-the-art performance for compact 3DGS. Project page <https://xiaobin2001.github.io/GNS-web>.

1. Introduction

Novel view synthesis is a fundamental yet challenging task in computer vision, with the core objective of reconstructing scene images from novel viewpoints given a limited set of input views. This technology demonstrates significant value in critical domains such as virtual reality (VR), digital twins, and autonomous driving. Neural Radiance Fields (NeRF) [12], based on implicit neural representations, achieve a breakthrough in high-fidelity view synthesis from sparse input views by parameterizing the light transport function using Multilayer Perceptron (MLP). However, this method struggles to achieve real-time rendering. In contrast, 3D Gaussian Splatting (3DGS) [7], as a recon-

*Corresponding authors.



Figure 2. Qualitative comparison between our method and 3DGS on the garden scene

struction technique utilizing explicit representation, simultaneously ensures both rendering quality and real-time performance. Initialized from sparse point clouds obtained through SfM (Structure from Motion) [14], 3DGS models the scene as a collection of 3D Gaussian primitives with optimizable parameters. Each primitive is characterized by its center position μ , covariance matrix Σ , opacity α , and color attributes encoded via spherical harmonics.

3DGS typically employs millions of Gaussians to fit a scene for high-quality rendering. However, the large number of Gaussian ellipsoids significantly increases rendering and storage costs, hindering the adoption of downstream applications. Existing compact 3DGS variants reduce the number of Gaussians through manually designed pruning criteria or by introducing additional learnable masks. Inspired by natural selection, we propose a learnable pruning technique that requires neither manually designed pruning criteria nor additional parameters, thereby enhancing the performance of compact 3DGS.

Figure 1 illustrates the overall pipeline of our proposed method. Inspired by natural selection, where environmental pressures screen for the fittest genes, we model a sparsity metric as a uniform survival pressure applied to all Gaussians, gradually reducing their opacity. Concurrently, optimization gradients aimed at maximizing rendering quality counteract the negative gradients induced by this pressure. This process favors Gaussians that contribute more significantly to rendering quality, as those receiving stronger optimization gradients survive longer under high survival pressure. To confine the natural selection process within a limited number of training iterations, we design a minimal-prior opacity decay technique to simulate environmental pressure. Under the same training iterations, our method uses only 15% of the budget and one-third of the training time compared to the original 3DGS, while achieving a 0.6 dB improvement in PSNR. Figure 2 compares our method with the original 3DGS.

In summary, our contributions are as follows:

- We propose a learnable pruning technique that requires neither manually designed pruning criteria nor additional parameters, yielding highly competitive performance.
- We design a opacity decay technique with finite that accelerates the pruning process without compromising the resulting quality.
- Our method achieves state-of-the-art performance in compact 3DGS and is highly portable.

2. Related Works

Quality Optimization of 3DGS: 3DGS already achieves a balance between rendering quality and real-time performance, yet there remains room for optimization. AbsGS [16] addresses the inherent reconstruction blur in 3DGS by introducing an absolute gradient-based densification evaluation criterion. TamingGS [11] incorporates multiple evaluation metrics for the densification process and optimizes the rendering kernel of 3DGS, significantly improving training speed. Improved-GS [3] reconstructs the densification process of 3DGS from three perspectives, significantly enhancing rendering quality. ScaffoldGS [10] combines implicit and explicit representations by using neural networks to implicitly store the parameters of Gaussian primitives. Mip-Splatting [17] introduces a 3D smoothing filter and a 2D mipmap filter to mitigate aliasing artifacts that may occur during magnification in 3DGS. GaussianPro [2] utilizes optimized depth and normal maps to initialize new Gaussians via reprojection. It further enhances geometric reconstruction through planar regularization. 2DGS [6] employs 2D Gaussian primitives to represent scenes, improving the application of Gaussian Splatting in geometric reconstruction. Spec-Gaussian [15] adopts an anisotropic spherical Gaussian appearance field for Gaussian color modeling, significantly enhancing rendering quality in complex scenes with specular reflections and anisotropic surfaces.

Compact 3DGS: The approaches to achieving compact 3DGS primarily fall into two categories. One builds upon the neural representation introduced by ScaffoldGS, utilizing octrees [13] or hash grids [1] for compression. This approach relies on specific neural representations and cannot be generalized to standard 3DGS. The other approach involves pruning techniques to eliminate redundant Gaussians, thereby reducing the total number of Gaussians in the scene to achieve compactness.

Compact3DGS [8] aims to reduce both rendering and storage overhead in 3DGS. It proposes a learnable masking strategy to decrease the number of Gaussians, while further compressing storage through neural color representations and vector quantization. LightGaussian [4] employs a global rendering weight combined with Gaussian volume to assist in pruning. LP-3DGS introduces the Gumbel-Sigmoid activation function into 3DGS to replace the Sigmoid function for masking or pruning scor-

ing. Gumbel-Sigmoid pushes values closer to 0 or 1, providing a good approximation for binary masks. Compact3DGS uses the straight-through estimator to address the non-differentiability issue in the binarization process, whereas MaskGS [9] directly employs probabilistic masks and derives gradients for mask parameters, achieving better mask pruning results. MaskGS also utilizes Gumbel-Sigmoid for activating mask parameters. Mini-Splatting [5] introduces depth reinitialization to address the uneven spatial distribution of Gaussians and uses maximum rendering contribution area and global rendering weight as criteria for pruning. GaussianSPA [18] employs the Alternating Direction Method of Multipliers (ADMM) to gradually attenuate the opacity of Gaussians scheduled for pruning to zero during optimization. This replaces explicit pruning with a smooth optimization process, contributing to improved rendering quality. Unlike previous methods, we avoid manual pruning rules and extra parameters.

Based on the different technical approaches adopted, we select the following works as comparative benchmarks: Compact3DGS, as a representative of binary masking; Mini-Splatting, as a representative of scene reorganization for sparsification; MaskGS, as a representative of probabilistic masking; and GaussianSPA, as a representative of progressive sparsification. LP-3DGS [19] only introduces a new activation function, which is also used in MaskGS, and is therefore not included in the comparison. The rendering importance pruning employed by LightGaussian will be compared in the ablation experiments.

3. Methods

3.1. Preliminaries

In 3DGS framework, a scene is represented by a collection of anisotropic 3D Gaussian primitives:

$$G(x) = \exp\left(-\frac{1}{2}(x)^T \Sigma^{-1}(x)\right), \quad (1)$$

where x denotes the offset from the Gaussian’s mean position, and Σ is its 3D covariance matrix. To guarantee that Σ remains positive semi-definite, 3DGS expresses it through a decomposition involving a rotation matrix R and a scaling matrix S :

$$\Sigma = RSS^T R^T. \quad (2)$$

Here, the scaling matrix S is parameterized by a 3D vector s , and the rotation matrix R is derived from a unit quaternion q . When rendering an image from a given camera viewpoint, the final color of a pixel p is computed by alpha-compositing N Gaussians $\{G_i \mid i = 1, \dots, N\}$ that project onto p , ordered from front to back, according to:

$$C = \sum_{i=1}^N c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (3)$$

where α_i is the opacity contribution of G_i at pixel p (obtained by evaluating the projected Gaussian and scaling by its intrinsic opacity), and c_i is the color of G_i , encoded using spherical harmonics (SH) coefficients.

3.2. Compact 3DGS Objective

Background and Challenges: Under a constrained Gaussian budget, the core objective of compact 3DGS remains to achieve high-fidelity rendering. Traditional 3DGS densification strategies identify blurry regions and insert additional Gaussians, which are inherently unconstrained local operations. However, when the total number of Gaussians (budget) is limited, global balance of rendering quality becomes essential. Simply restricting densification growth is therefore insufficient to meet the requirement of compact 3DGS (see Sec. 4.3).

Current approaches typically adopt a two-stage strategy: first, generate a high-quality scene representation through unconstrained densification; second, perform pruning to balance quality and budget.

Bi-Objective Decomposition: In 3DGS, scene optimization relies on back-propagation, where each Gaussian only receives gradient signals from pixels it contributes to. Consequently, the retained subset of Gaussians after pruning, denoted as G_C , fundamentally determines the achievable rendering upper bound. Once a Gaussian representing unique local details is mistakenly pruned, its contribution is almost unrecoverable.

Based on this, we decompose the overall objective of compact 3DGS into two sub-goals:

1. Selecting a superior retained subset G_C to maximize the potential quality upper bound;
2. Ensuring stable convergence of the pruning-optimization process within a limited number of iterations.

The following sections address these two goals respectively. Sec. 3.3 introduces a natural selection-based adaptive pruning framework for achieving Goal 1, while Sec. 3.4 presents a regularization gradient field with finite priors to accelerate convergence (Goal 2).

3.3. Adaptive Pruning via Natural Selection

To fulfill Goal 1—selecting a superior subset G_C —we draw inspiration from natural selection in biological evolution (excluding genetic inheritance and mutation) and propose an adaptive pruning framework. This framework simulates the competition between environmental pressure and individual fitness, progressively eliminating low-contribution Gaussians and retaining high-importance ones to maximize rendering quality under a strict budget.

Problem Formulation: Let the complete Gaussian set be $G = \{g_i\}_{i=1}^N$, where each Gaussian g_i is parameterized as $\theta_i = (\alpha_i, \mu_i, \Sigma_i, c_i)$ (opacity, mean, covariance, and color). The pruning objective is formulated as:

$$\min_{G_C \subset G} \mathcal{L}_{\text{render}}(G_C) \quad \text{s.t.} \quad |G_C| \leq B, \quad (4)$$

where $\mathcal{L}_{\text{render}}$ denotes the rendering loss and B is the Gaussian budget.

Natural Selection Mechanism: We model the biological analogy through the following definitions:

- **Vitality:** the opacity $\alpha_i \in [0, 1]$ represents the existence strength of each Gaussian. This is because opacity directly determines the rendering contribution of each primitive.
- **Environmental Pressure:** a globally consistent regularization gradient field $\mathcal{L}_{\text{reg}}(\alpha)$ applied to all Gaussians (see Sec. 3.4 for details).
- **Fitness:** the rendering gradient $\nabla_{\alpha_i} \mathcal{L}_{\text{render}}$ received by Gaussian g_i , representing its dynamic importance.

The overall loss is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{render}}(\Theta) + \mathcal{L}_{\text{reg}}(\alpha), \quad (5)$$

and the net gradient on the opacity of each Gaussian is:

$$\nabla_{\alpha_i}^{\text{net}} = \nabla_{\alpha_i} \mathcal{L}_{\text{render}} + \nabla_{\alpha} \mathcal{L}_{\text{reg}}. \quad (6)$$

Here, the first term is Gaussian-specific, while the second term is a global constant. In practice, the regularization gradient is applied every $N = 50$ rendering iterations. This allows fitness signals to accumulate from multiple training views, producing more stable survival decisions before applying environmental pressure.

If the rendering gradient $\nabla_{\alpha_i} \mathcal{L}_{\text{render}}$ is opposite in direction to the regularization gradient $\nabla_{\alpha} \mathcal{L}_{\text{reg}}$, increasing opacity improves rendering quality, and the Gaussian tends to survive. Conversely, when both gradients are aligned, the Gaussian contributes negatively to the rendering and is rapidly suppressed. Since the regularization gradient magnitude is globally constant, the relative differences between rendering gradients dominate the competition—only high-fitness Gaussians maintain strong vitality and survive.

At the end of densification, when the scene has largely converged, the natural selection process is activated.

Selection Process and Survival Criterion:

During optimization, a Gaussian is permanently removed if its opacity falls below a survival threshold: $\alpha_i < \tau$. We empirically set $\tau = 0.001$, smaller than the 0.005 threshold used in 3DGS, as Gaussians below this level contribute negligibly to rendering.

Strong environmental pressure gradually removes weak Gaussians until $|G_C| \leq B$. The resulting subset G_C thus consists solely of high-fitness, high-contribution primitives.

Importantly, this framework requires no pre-defined importance score; instead, global competition among gradients naturally yields an optimal subset.

Key Innovations: Prior works regularize opacity in two common ways, both fundamentally different from ours:

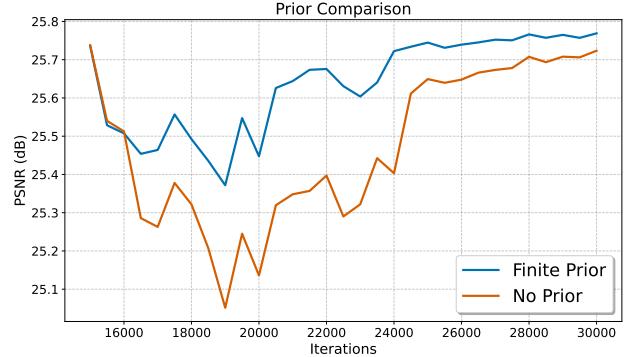


Figure 3. The figure illustrating the optimization speed improvement of finite prior over no prior.

1. **Weak Regularization:** introduces low-intensity penalties to reduce average opacity and encourage color aggregation from multiple primitives.
2. **Auxiliary Pruning Regularization:** applies regularization as a post-processing tool for numerical stability, but it does not participate in pruning decisions.

In contrast, our method applies a strong, globally consistent regularization gradient that explicitly competes with the rendering gradient. This introduces a novel gradient-competition-driven pruning mechanism, enabling:

1. **Automatic Selection:** weak Gaussians are eliminated via gradient competition;
2. **Smooth Optimization:** opacity decays continuously, avoiding instability from discrete pruning.

As shown in Section 4.3, this mechanism consistently outperforms hand-designed pruning heuristics and learnable-mask approaches across multiple datasets.

3.4. Gradient Field with Finite Priors

Introduction of Finite Priors: In the natural selection mechanism, the gradient of the optimization loss with respect to opacity parameters is used to counteract the environmental pressure, while other parameters continue to be optimized to minimize the rendering loss. To achieve faster optimization convergence (Goal 2 in Section 3.2), it is essential to design an appropriate environmental pressure that ensures the natural selection process does not hinder the optimization of opacity parameters for surviving Gaussians.

In the natural-selection framework, environmental pressure should be applied uniformly to all Gaussian opacities to guarantee fairness of the selection process. We define the opacity decay ratio:

$$R_o = \frac{\alpha_{t-1} - \alpha_t}{\alpha_{t-1}} \quad (7)$$

to quantify the degree of influence that the environmental pressure exerts on an individual. Ideally, applying the same

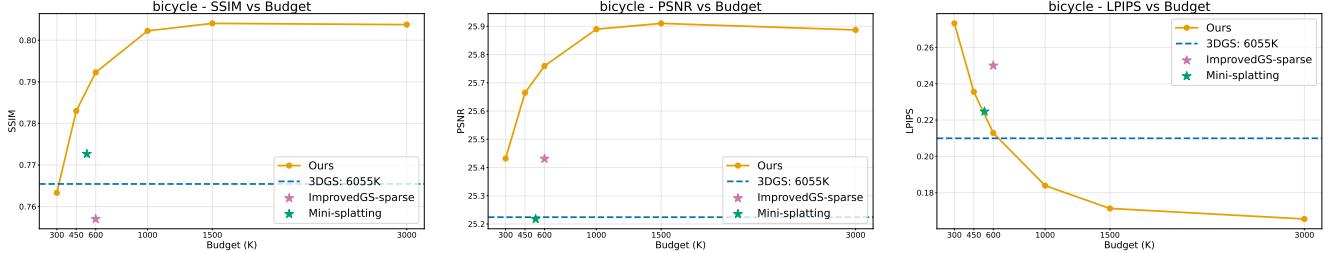


Figure 4. Performance under varying Gaussian budgets. Results for the remaining scenes can be found in appendix.

decay ratio to every Gaussian accurately models the environmental pressure. Because the regularization gradient received by every Gaussian has the same magnitude and direction, it can be regarded as a gradient field. However, if, by the end of the natural-selection stage, the opacities of surviving Gaussians are generally suppressed too strongly, this will substantially degrade subsequent optimization convergence speed and numerical stability.

To satisfy Goal 2, we introduce a finite prior designed to accelerate the overall process without breaking the fairness of natural selection. There exists a correlation between a Gaussian’s current opacity and its fitness: high-opacity Gaussians tend to contribute more to rendering and are more likely to be important primitives; moreover, in the late stage of natural selection opacity already reflects fitness. We exploit this weak correlation as a limited prior to speed up both selection and the optimization recovery (see Figure 3). Concretely, we introduce the finite prior by applying the gradient field to the pre-activation opacity parameter v instead of α directly,

Opacity α is activated by v through the sigmoid function:

$$\alpha = S(v) = \frac{1}{1 + e^{-v}}. \quad (8)$$

Let the global regularization gradient be ∇v and the learning rate be lr . Ignoring rendering gradients, the update rule is:

$$v_{t+1} = v_t + \nabla v \cdot lr, \quad (9)$$

yielding the decay ratio:

$$R_o^{t+1} = \frac{S(v_t) - S(v_t + \nabla v \cdot lr)}{S(v_t)}. \quad (10)$$

When $\nabla v \cdot lr \rightarrow 0$,

$$R_o^{t+1} \approx (1 - \alpha) \cdot |\nabla v \cdot lr|. \quad (11)$$

Thus, the decay ratio R_o is linearly proportional to $1 - \alpha$:

- As $\alpha \rightarrow 1$, $R_o^{t+1} \rightarrow 0$, protecting consistently high-fitness Gaussians;
- As $\alpha \rightarrow 0$, $R_o^{t+1} \rightarrow |\nabla v \cdot lr|$, meaning the decay ratio approaches its maximum value. For example, compared

to Gaussians with average opacity $\alpha = 0.5$ (whose decay ratio is $0.5 \cdot |\nabla v \cdot lr|$), low-opacity Gaussians ($\alpha \rightarrow 0$) exhibit up to twice the decay magnitude. This accelerates the elimination of less fit individuals.

This limited prior speeds up convergence while preserving the fairness and adaptivity of natural selection.

Implementation Details: To prevent Gaussians with extremely high opacity from evading selection during the initial phase, we apply prior-free regularization under a low learning rate in the early stages of natural selection. After introducing the prior, a higher learning rate is subsequently used to accelerate the screening process. The regularization gradient ∇v is derived from:

$$\mathcal{L}_{\text{reg}} = (\mathbb{E}[v] - T)^2, \quad (12)$$

where $\mathbb{E}[v]$ denotes the mean value of all v parameters and T is the regularization target. Its gradient is:

$$\nabla v = 2(\mathbb{E}[v] - T), \quad (13)$$

When $T \gg \mathbb{E}[v]$, ∇v is primarily dominated by T , remaining stable throughout the process. In practice, we set $T = -20$. Since the natural selection process terminates when the number of surviving Gaussians meets the budget requirement, T here primarily serves to provide a gradient direction and a stable magnitude. Because $\nabla v \propto T$ (when $|T| \gg |\mathbb{E}[v]|$), increasing $|T|$ is equivalent to linearly amplifying the strength of the gradient field, an effect similar to increasing the learning rate lr .

4. Experiments

4.1. Datasets and Metrics

Following 3DGS and other compact 3DGS baselines, we select 13 scenes from the Mip-NeRF 360, Deep Blending, and Tanks and Temples datasets for evaluation. In each experiment, every 8th image is used as the test set, and the remaining images form the training set.

For quantitative evaluation, we adopt three widely used metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). All metrics are computed using the official evaluation framework provided by 3DGS.

Dataset Method—Metric	Mip-NeRF360					Deep Blending					Tanks&Temples				
	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow	Num \downarrow	Time \downarrow	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow	Num \downarrow	Time \downarrow	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow	Num \downarrow	Time \downarrow
3DGS	0.816	27.50	0.216	3320453	35.9	0.904	29.56	0.244	2819180	32.5	0.849	23.72	0.177	1835092	19.9
Compact-3DGS	0.807	27.33	0.227	1516172	31.3	0.904	29.61	0.249	1251516	28.9	0.847	23.67	0.180	933650	18.0
Mini-splatting	0.822	27.36	0.217	493466	26.8	0.910	30.07	0.241	554179	23.2	0.847	23.47	0.180	300481	17.3
MaskGS	0.815	27.43	0.218	1582926	31.9	0.907	29.76	0.245	910162	27.6	0.847	23.73	0.180	740683	17.4
GaussianSPA	0.817	27.31	0.229	421427	34.3	0.913	30.00	0.242	443740	28.8	0.850	23.40	0.171	424801	23.3
ImprovedGS	0.814	27.66	0.233	466667	8.1	0.911	30.23	0.244	450000	6.8	0.856	24.39	0.179	450000	5.7
Ours	0.833	28.13	0.207	466667	12.5	0.914	30.15	0.233	450000	9.4	0.871	24.63	0.154	450000	8.5

Table 1. Quantitative results on the Mip-NeRF 360, Deep Blending, and Tanks and Temples datasets. Cells are highlighted as follows: best , and second best . Per-scene detailed data can be found in the appendix.

4.2. Implementation Details

Our method is independent of specific CUDA kernels and introduces no additional trainable parameters, making it easily transferable to advanced 3DGS variants. We adopt Improved-GS as the base model, since it provides faster and higher-quality densification. To reduce the influence of initial opacity before natural selection, we increase the opacity learning rate to 4 times the original value, ensuring that survival is primarily determined by dynamic gradient competition. The opacity learning rate will be restored to its original value after natural selection is completed.

The natural selection stage begins after densification (at 15K iterations) and continues until the number of remaining Gaussians meets the target budget. Empirically, this process converges within 5K–8K iterations, achieving the best rendering quality. Based on this observation, we tune the regularization learning rate per scene, while all other hyperparameters remain shared across scenes. **A detailed discussion of hyperparameter choices is provided in appendix.**

All experiments are conducted on a single RTX A5000 GPU, with a total of 30K training iterations for all methods. Competing methods are reimplemented using the official configurations provided by their authors. During the reproduction of GaussianSPA, several non-trivial issues were observed and are discussed in the appendix. As noted in Section 3.2, densification alone cannot meet the compact 3DGS requirement. Thus, we also include a sparse variant of Improved-GS—obtained by lowering the peak densification budget—as an additional baseline.

4.3. Quantitative Comparison

The quantitative results are summarized in Table 1. Despite using a minimal Gaussian budget, our method achieves the best overall performance across PSNR, SSIM, and LPIPS. Compared with 3DGS, our approach attains a $> 0.6dB$ gain in PSNR while using only 15% of the Gaussian budget. Relative to Improved-GS, our method shows comparable training time but significantly better rendering quality.

Mini-Splatting reaches a level similar to 3DGS with low budget usage, whereas GaussianSPA, due to its ADMM-

Dataset Method—Metric	Outdoors			Indoors		
	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow
ImprovedGS (sparse)	0.728	24.85	0.260	0.922	31.18	0.198
SPA Pruning	0.733	24.95	0.261	0.928	31.70	0.184
MaskGS Pruning	0.749	25.11	0.242	0.927	31.67	0.184
Opacity Pruning	0.722	24.75	0.275	0.924	31.52	0.190
Render Pruning	0.726	24.67	0.261	0.923	31.39	0.191
Edge Pruning	0.728	24.81	0.263	0.924	31.50	0.193
Natural Selection (Ours)	0.753	25.20	0.234	0.930	31.78	0.178

Table 2. Ablation Study of unified base model.

Dataset Method—Metric	Outdoors			Indoors		
	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow	SSIM \uparrow	PSNR \uparrow	LPIPS \downarrow
No Prior	0.751	25.18	0.242	0.928	31.70	0.183
Strong Prior	0.747	25.10	0.245	0.929	31.75	0.181
Finite Prior (Ours)	0.753	25.20	0.234	0.930	31.78	0.178

Table 3. Ablation Study of prior.

based optimization, requires substantially longer iterations and performs pruning only at the end, resulting in longer training time. Compared to CompactGS, MaskGS achieves comparable or superior rendering quality with a similar or smaller budget. Overall, our method achieves SOTA performance in compact 3DGS.

Figure 4 illustrates performance curves under different budget settings. Notably, in some scenes, our approach surpasses the original 3DGS using only 5% of its budget.

4.4. Qualitative Comparison

Representative qualitative results are shown in Figure 5.

Garden: Our method achieves higher rendering quality than 3DGS while using only 10% of the Gaussian budget. Although Mini-Splatting and GaussianSPA consume less budget, their reconstruction quality is far inferior. Compared to ImprovedGS, our approach restores finer scene details and avoids large blurry regions.

Room: Among all methods, our approach produces fewer artifacts and is the only one that successfully restores the details of the globe in the upper left corner. **Train:** Our method delivers superior reconstruction quality while maintaining the lowest budget among all methods. Except for ImprovedGS-sparse, all other baselines exhibit visible artifacts and fail to reconstruct the distant mountain contours.

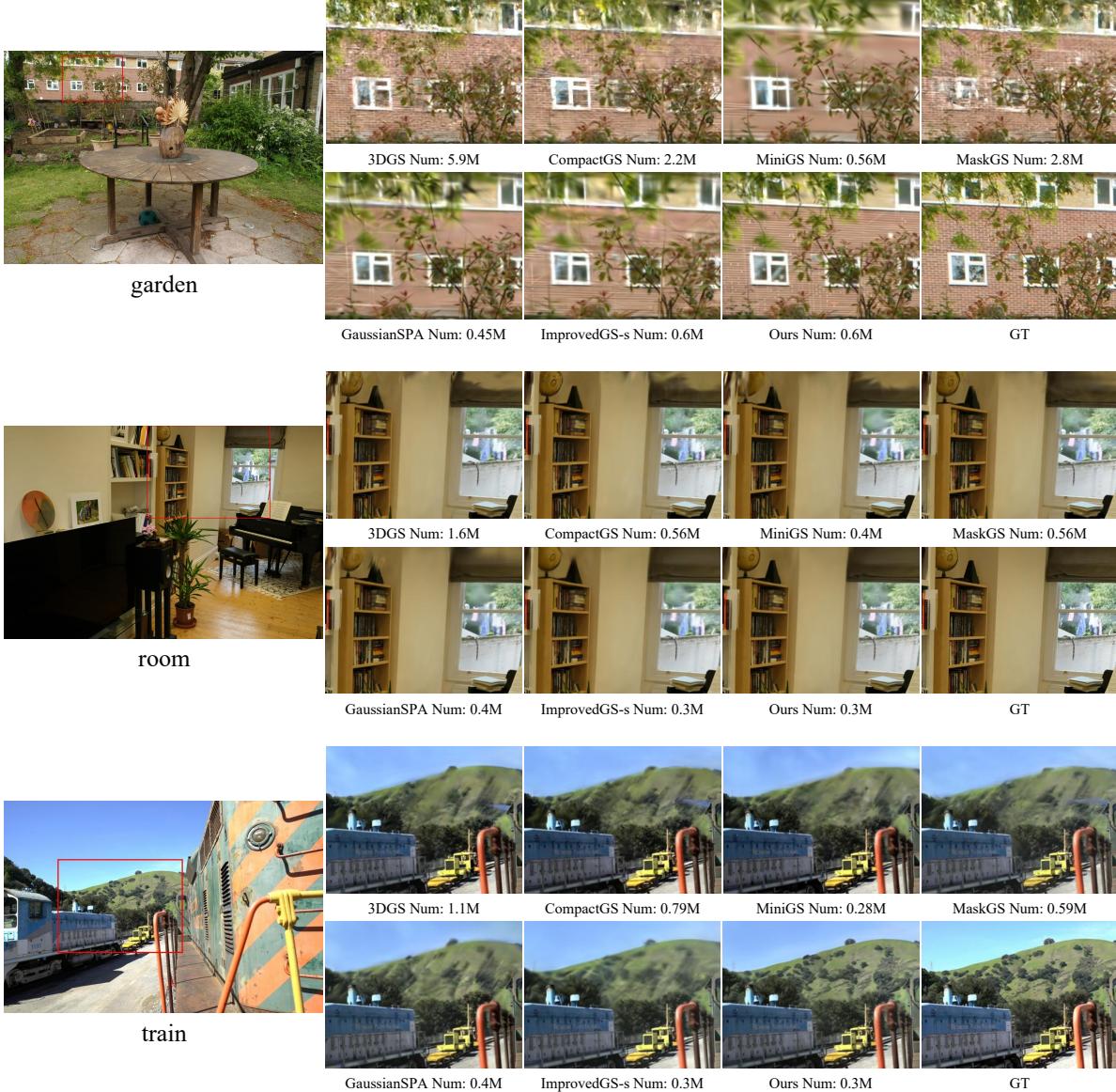


Figure 5. Qualitative comparison results among scenes garden, room, train. Num is final Gaussian count after training. The results for the remaining scenes can be found in appendix.

accurately. Our approach produces more faithful geometry and textures.

4.5. Ablation Studies

Unified Base Model: To eliminate the influence of inherent model differences (e.g., Improved-GS, 3DGS, MiniGS-D) and ensure a fair comparison of pruning techniques, we apply all pruning methods on a unified Improved-GS (15K iteration) base model. We choose MaskGS as the representative of mask-based pruning and SPA for smooth pruning. Additionally, we test three non-learnable pruning weights: (1) opacity-based weight (used in LightGaussian and Mini-

Splatting), (2) rendering-weight, and (3) edge-weight from Improved-GS densification. These weights are used as relative retention probabilities to prevent structural voids.

As shown in Table 2, our method consistently outperforms all others. Among the baselines, MaskGS performs better in outdoor scenes, while SPA is slightly superior in indoor scenes. This indicates that in complex scenes, selecting the right Gaussians is critical, whereas in simpler scenes, smoothness of pruning plays a larger role. Our approach effectively balances both aspects, achieving the best results. We also observe that single-weight heuristics differ little among themselves but remain substantially worse than

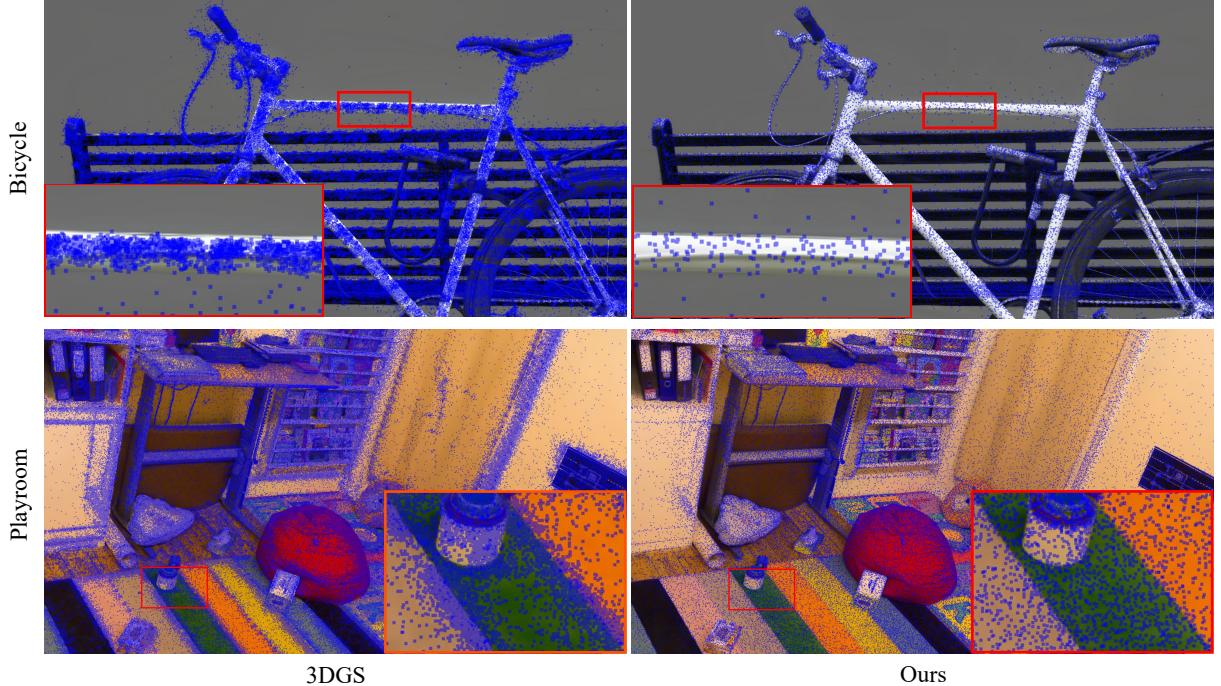


Figure 6. Comparison of the point cloud distributions in scenes trained with 3DGS and our method. For the bicycle scene, the background has been removed to more clearly highlight the differences between the two approaches.

learnable methods.

Finite Prior Ablation: To validate the effectiveness of the finite prior design, we conduct two ablation variants:

1. No Prior: By manually compensating gradients, the opacity attenuation magnitude of all Gaussians is maintained strictly consistent, thereby eliminating prior influence. Since direct parameter modification invalidates the optimizer, manual gradient compensation serves as an approximation.
2. Strong Prior: Using opacity as the sampling probability for exemption from natural selection in the current round.

As shown in Table 3, the No Prior variant fails to achieve efficient convergence (Goal 2) and performs poorly in indoor scenes. The Strong Prior variant disrupts the fairness and adaptability of natural selection (Goal 1), leading to significant performance drops in outdoor scenes. Our finite prior strikes the optimal balance between fairness and efficiency. More ablation study results can be found in appendix.

4.6. Spatial Distribution of Point Cloud

Figure 6 compares the spatial distribution of Gaussians between 3DGS and our method after training. Apart from a significant reduction in the total number of Gaussians, our method yields more uniform distributions. In 3DGS, Gaussian points often form dense clusters, as observed in the *bicycle* and *playroom* scenes.

Bicycle scene, where many Gaussians are concentrated in low-frequency areas such as the bicycle frame, resulting in redundancy. Mini-Splatting mitigates this problem via depth reinitialization, whereas our method eliminates redundancy without any external intervention. During training, the positions of Gaussians in edge regions undergo repeated fine-tuning, causing these areas to receive high gradients and form excessively dense distributions (e.g., along table edges and carpet boundaries in the *Playroom* scene). Our method avoids such over-clustering effectively.

5. Conclusion

We presented a biologically inspired simplification framework for 3D Gaussian Splatting that leverages natural selection principles to autonomously prune redundant Gaussians. By applying uniform survival pressure through opacity regularization and using optimization gradients as fitness indicators, our method eliminates the need for manual criteria or additional parameters. Accelerated by finite-prior opacity decay, it achieves SOTA rendering quality with only 15% of the original Gaussian budget and one-third of the training time, while improving PSNR by over 0.6 dB. This efficient, portable solution significantly lowers computational barriers, enabling broader adoption of 3DGS in resource-constrained applications. Future work will explore integrating our natural-selection framework into dynamic or sparse-view reconstruction.

References

- [1] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, pages 422–438. Springer, 2024. [2](#)
- [2] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning*, 2024. [2](#)
- [3] Xiaobin Deng, Changyu Diao, Min Li, Ruohan Yu, and Duanqing Xu. Improving densification in 3d gaussian splatting for high-fidelity rendering. *arXiv preprint arXiv:2508.12313*, 2025. [2](#)
- [4] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, Zhangyang Wang, et al. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37: 140138–140158, 2024. [2](#)
- [5] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pages 165–181. Springer, 2024. [3](#)
- [6] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. [2](#)
- [7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. [1](#)
- [8] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21719–21728, 2024. [2](#)
- [9] Yifei Liu, Zhihang Zhong, Yifan Zhan, Sheng Xu, and Xiao Sun. Maskgaussian: Adaptive 3d gaussian representation from probabilistic masks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 681–690, 2025. [3](#)
- [10] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. [2](#)
- [11] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. [2](#)
- [12] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. [1](#)
- [13] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. [2](#)
- [14] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. [2](#)
- [15] Ziyi Yang, Xinyu Gao, Yang-Tian Sun, Yihua Huang, Xiaoyang Lyu, Wen Zhou, Shaohui Jiao, Xiaojuan Qi, and Xiaogang Jin. Spec-gaussian: Anisotropic view-dependent appearance for 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:61192–61216, 2024. [2](#)
- [16] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1053–1061, 2024. [2](#)
- [17] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19447–19456, 2024. [2](#)
- [18] Yangming Zhang, Wenqi Jia, Wei Niu, and Miao Yin. Gausianspa: An “optimizing-sparsifying” simplification framework for compact and high-quality 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26673–26682, 2025. [3](#)
- [19] Zhaoliang Zhang, Tianchen Song, Yongjae Lee, Li Yang, Cheng Peng, Rama Chellappa, and Deliang Fan. Lp-3dgs: Learning to prune 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:122434–122457, 2024. [3](#)

A. Appendix

A.1. Quantitative Comparison per Scene

Scene—Methods	3DGS	CompactGS	MiniGS	MaskGS	SPA	ImprovedGS-s	Ours
bicycle	0.765	0.745	0.773	0.765	0.761	0.757	0.792
flowers	0.606	0.592	0.625	0.604	0.609	0.610	0.640
garden	0.867	0.855	0.848	0.866	0.840	0.838	0.867
stump	0.773	0.756	0.805	0.774	0.796	0.790	0.813
treehill	0.632	0.628	0.654	0.634	0.656	0.648	0.664
bonsai	0.942	0.941	0.939	0.941	0.940	0.939	0.947
counter	0.908	0.907	0.905	0.907	0.906	0.908	0.916
kitchen	0.928	0.926	0.926	0.926	0.923	0.919	0.931
room	0.919	0.918	0.921	0.919	0.922	0.921	0.929
playroom	0.907	0.908	0.914	0.910	0.916	0.913	0.916
drjohnson	0.901	0.901	0.907	0.904	0.910	0.909	0.911
train	0.815	0.813	0.812	0.812	0.813	0.819	0.843
truck	0.882	0.880	0.883	0.881	0.887	0.892	0.899

Table 4. The SSIM scores for all works in each scene.

Scene—Methods	3DGS	CompactGS	MiniGS	MaskGS	SPA	ImprovedGS-s	Ours
bicycle	25.22	24.92	25.22	25.21	25.07	25.43	25.76
flowers	21.62	21.39	21.53	21.55	21.56	21.59	21.86
garden	27.41	27.13	26.91	27.39	26.80	27.17	27.79
stump	26.64	26.31	27.24	26.66	27.05	27.05	27.38
treehill	22.44	22.48	22.73	22.52	23.03	23.00	23.16
bonsai	32.26	32.14	31.49	32.08	31.40	32.22	32.77
counter	29.03	28.98	28.65	28.96	28.56	29.31	29.68
kitchen	31.48	31.15	31.26	31.14	30.99	31.06	32.25
room	31.43	31.49	31.25	31.40	31.36	32.14	32.52
playroom	29.96	30.06	30.57	30.15	30.49	30.66	30.63
drjohnson	29.16	29.17	29.57	29.37	29.51	29.79	29.66
train	22.00	22.01	21.61	22.09	21.34	22.43	22.74
truck	25.44	25.33	25.33	25.36	25.47	26.35	26.52

Table 5. The PSNR scores for all works in each scene.

Scene—Methods	3DGS	CompactGS	MiniGS	MaskGS	SPA	ImprovedGS-s	Ours
bicycle	0.210	0.235	0.225	0.212	0.251	0.250	0.213
flowers	0.335	0.351	0.327	0.337	0.346	0.337	0.307
garden	0.106	0.124	0.150	0.108	0.168	0.164	0.127
stump	0.214	0.237	0.199	0.216	0.225	0.217	0.197
treehill	0.327	0.335	0.313	0.328	0.331	0.333	0.306
bonsai	0.203	0.205	0.200	0.206	0.199	0.215	0.193
counter	0.200	0.204	0.198	0.203	0.198	0.206	0.189
kitchen	0.126	0.128	0.129	0.129	0.134	0.150	0.126
room	0.218	0.223	0.212	0.222	0.208	0.221	0.202
playroom	0.243	0.248	0.238	0.247	0.239	0.248	0.235
drjohnson	0.244	0.249	0.243	0.243	0.245	0.240	0.230
train	0.207	0.210	0.222	0.212	0.216	0.219	0.190
truck	0.146	0.151	0.139	0.149	0.126	0.139	0.119

Table 6. The LPIPS scores for all works in each scene.

Scene—Methods	ImprovedGS-s	SPA	MaskGS	Opacity Pruning	Render Pruning	Edge Pruning	Natural Selection(Ours)
bicycle	0.757	0.759	0.781	0.751	0.755	0.757	0.792
flowers	0.610	0.618	0.636	0.606	0.610	0.613	0.637
garden	0.838	0.844	0.861	0.842	0.845	0.845	0.867
stump	0.790	0.799	0.808	0.783	0.783	0.788	0.813
treehill	0.648	0.647	0.658	0.628	0.637	0.636	0.657
bonsai	0.939	0.945	0.944	0.941	0.939	0.941	0.947
counter	0.908	0.913	0.912	0.908	0.907	0.908	0.916
kitchen	0.919	0.926	0.928	0.925	0.925	0.924	0.930
room	0.921	0.926	0.924	0.923	0.921	0.922	0.929

Table 7. The SSIM scores for all ablation studies in each scene.

Scene—Methods	ImprovedGS-s	SPA	MaskGS	Opacity Pruning	Render Pruning	Edge Pruning	Natural Selection(Ours)
bicycle	25.43	25.38	25.63	25.25	25.25	25.39	25.80
flowers	21.59	21.74	21.85	21.46	21.41	21.55	21.85
garden	27.17	27.33	27.68	27.29	27.23	27.39	27.80
stump	27.05	27.19	27.26	26.90	26.74	26.95	27.42
treehill	23.00	23.12	23.15	22.87	22.71	22.78	23.16
bonsai	32.22	32.74	32.63	32.40	32.19	32.46	32.74
counter	29.31	29.61	29.59	29.42	29.47	29.55	29.69
kitchen	31.06	32.08	32.08	31.96	31.81	31.74	32.22
room	32.14	32.36	32.40	32.30	32.10	32.25	32.45

Table 8. The PSNR scores for all ablation studies in each scene.

Scene—Methods	ImprovedGS-s	SPA	MaskGS	Opacity Pruning	Render Pruning	Edge Pruning	Natural Selection(Ours)
bicycle	0.250	0.260	0.236	0.265	0.253	0.256	0.214
flowers	0.337	0.330	0.311	0.339	0.332	0.330	0.311
garden	0.164	0.161	0.138	0.171	0.155	0.161	0.128
stump	0.217	0.217	0.206	0.236	0.229	0.225	0.197
treehill	0.333	0.338	0.320	0.364	0.336	0.344	0.318
bonsai	0.215	0.196	0.198	0.205	0.207	0.206	0.194
counter	0.206	0.194	0.196	0.204	0.203	0.205	0.189
kitchen	0.150	0.135	0.130	0.137	0.135	0.141	0.126
room	0.221	0.208	0.214	0.216	0.217	0.218	0.203

Table 9. The LPIPS scores for all ablation studies in each scene.

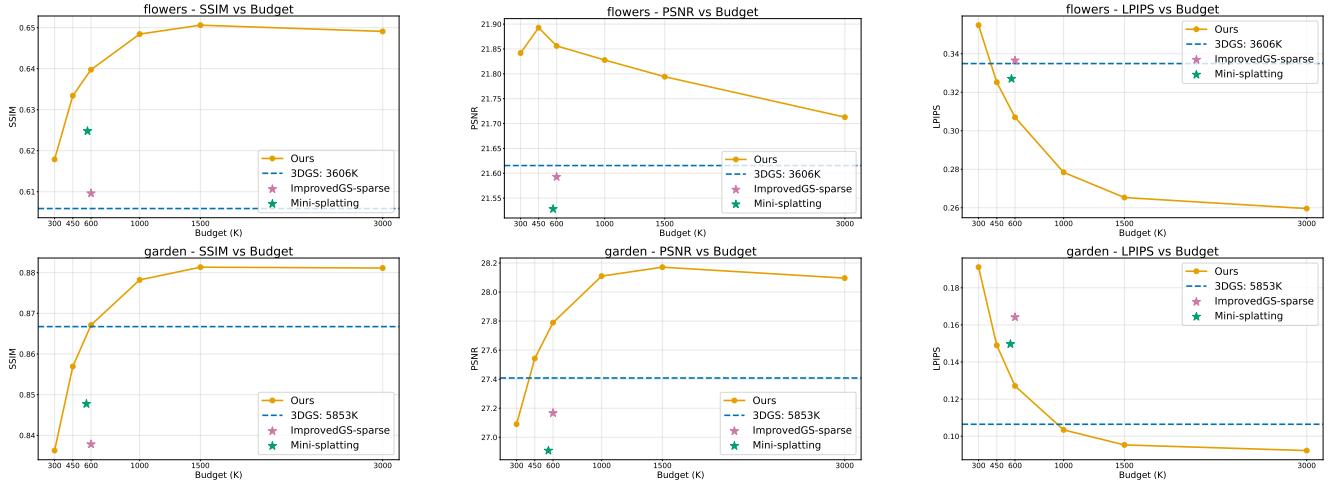


Figure 7. Performance under varying Gaussian budgets of 2 scenes.

A.2. Performance Under Varying Gaussian Budgets

Figure 7 and 8 present the Performance Under Varying Gaussian Budgets across 8 scenes from the Mip-NeRF 360 dataset. Due to the inherent high volatility of the T & T and DB datasets, similar comparisons are not conducted for these.

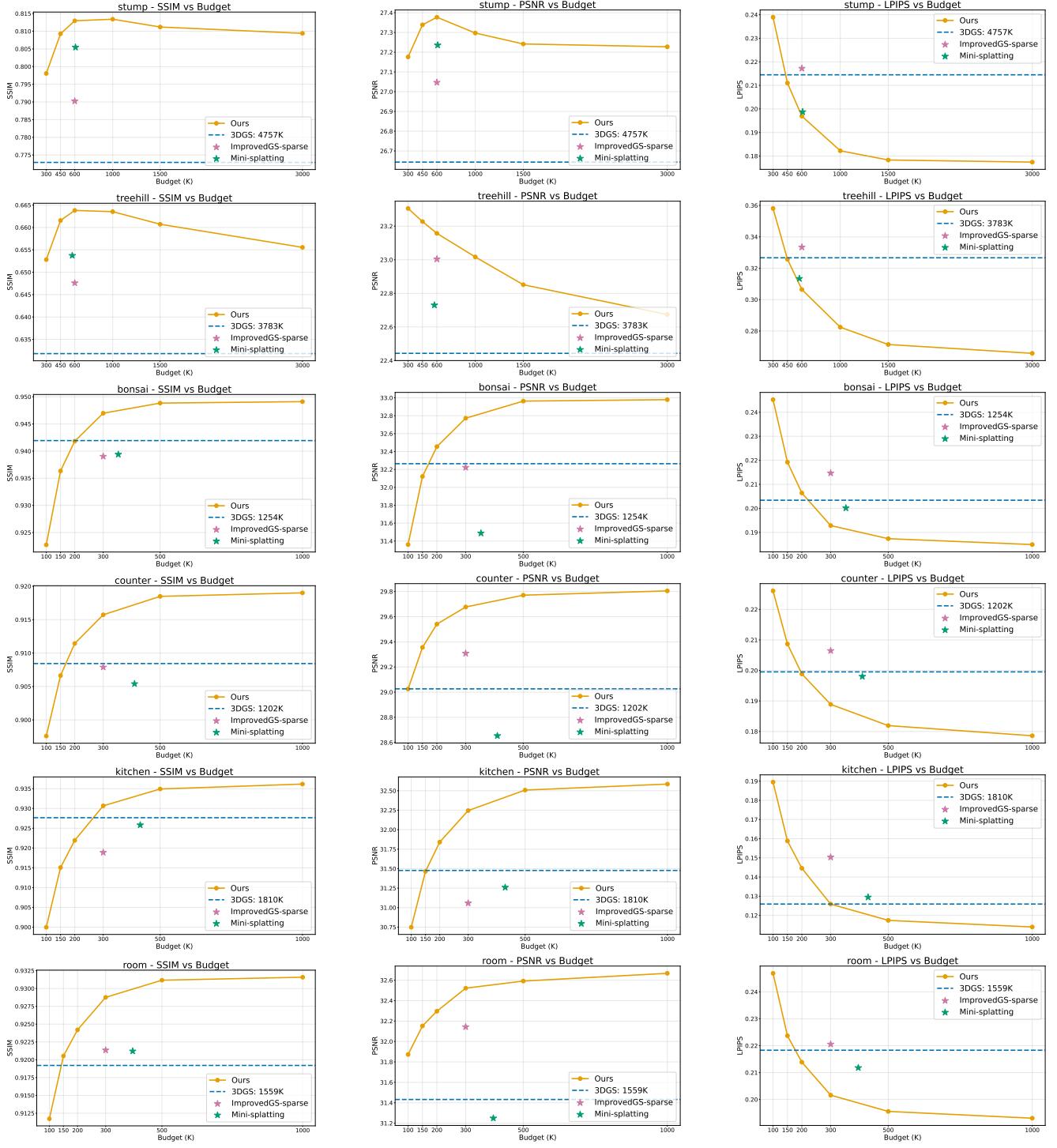


Figure 8. Performance under varying Gaussian budgets of 6 scenes.

A.3. More Qualitative Comparisons

Figure 9 and 10 present qualitative comparisons of the remaining 10 scenes. Our method achieves optimal rendering results across all scenes.

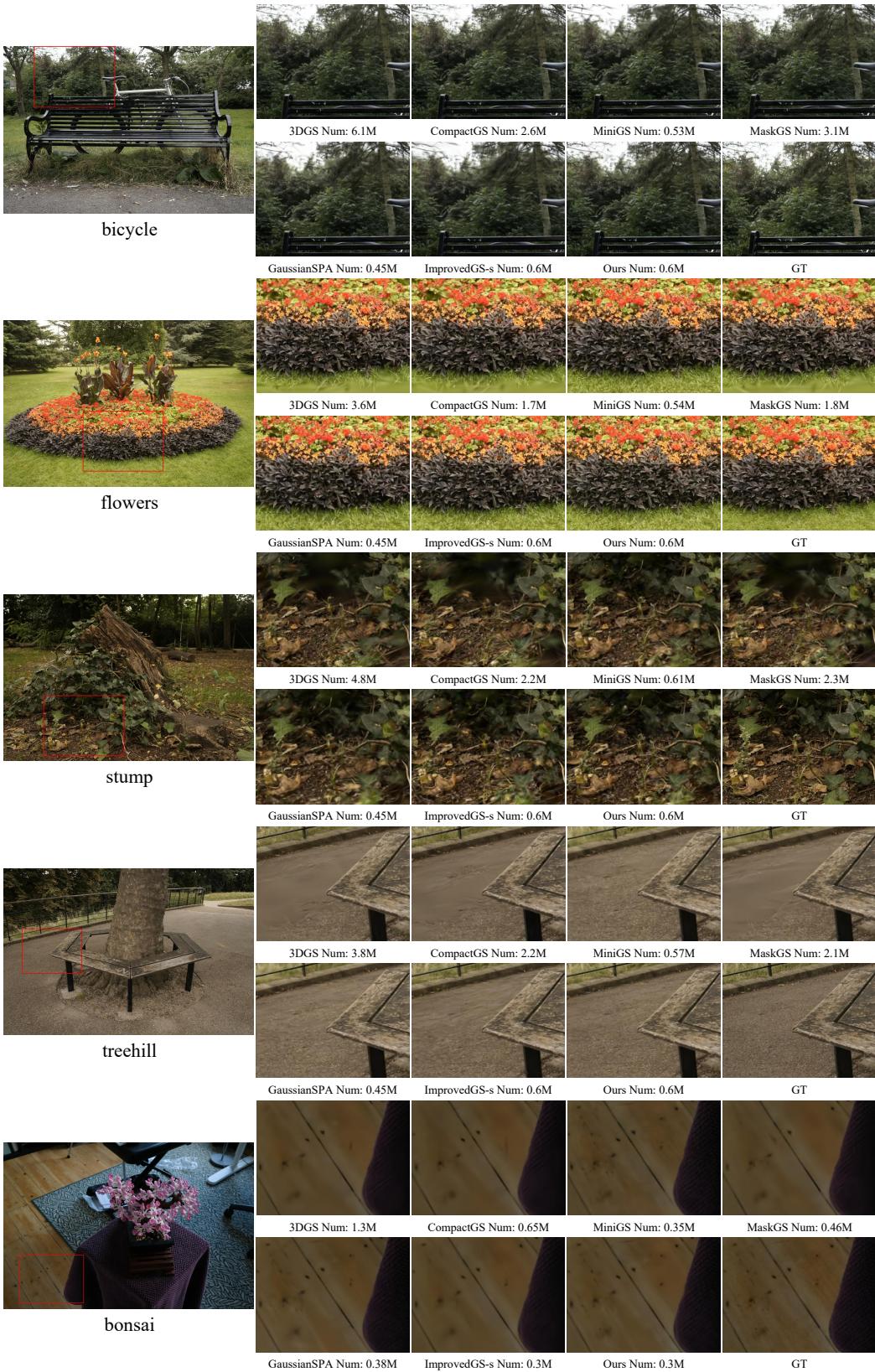


Figure 9. Qualitative comparison results among scenes bicycle, flowers, stump, treehill, bonsai.

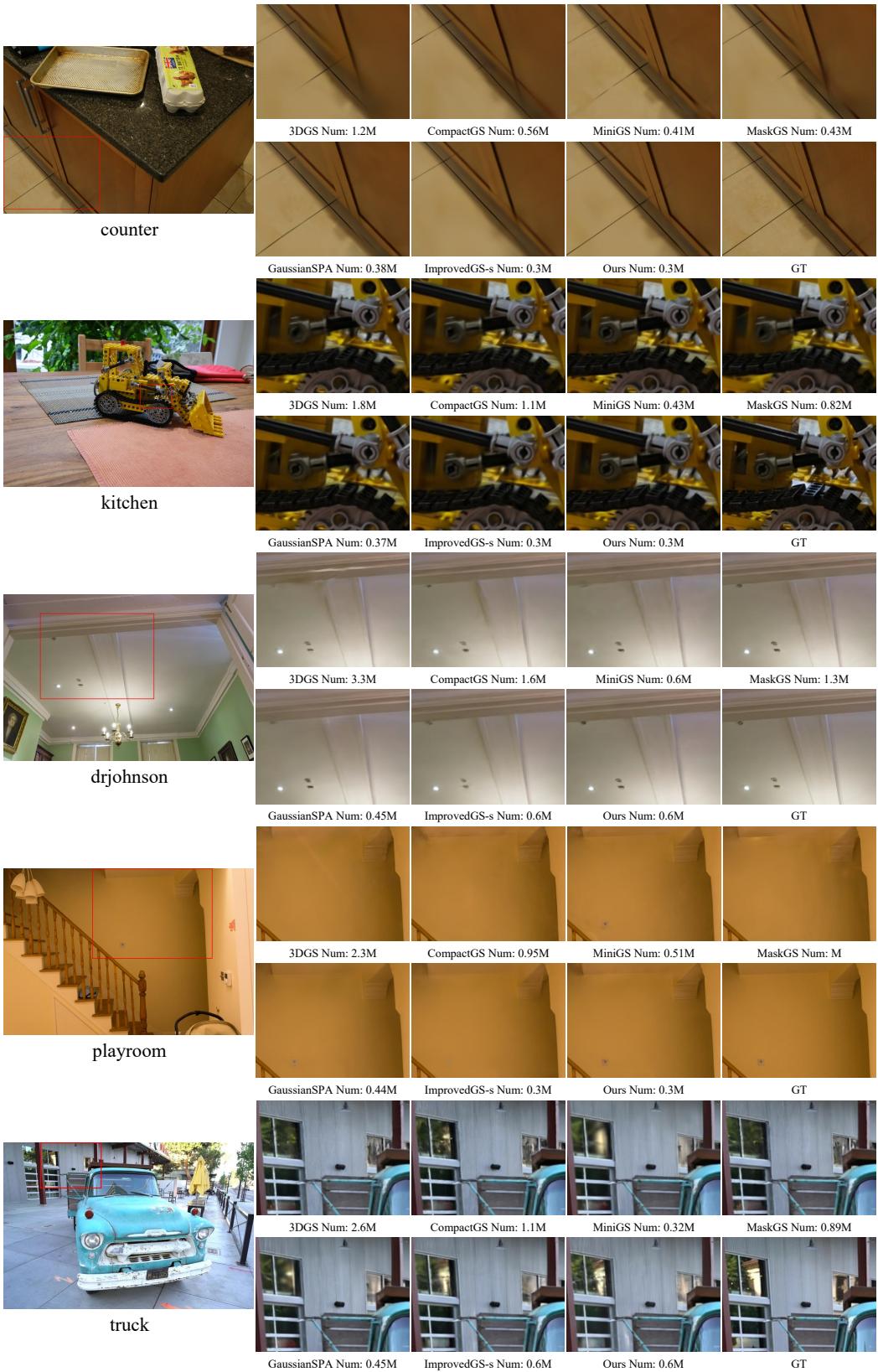


Figure 10. Qualitative comparison results among scenes counter, kitchen, drjohnson, playroom, truck.

A.4. Derivation of the Finite Prior

This section derives the linear relation between the decay ratio and the current opacity under the finite prior.

Opacity is activated from the pre-activation variable v by the sigmoid function:

$$\alpha = S(v) = \frac{1}{1 + e^{-v}}. \quad (14)$$

Ignoring rendering gradients, the update under environmental pressure is:

$$v_{t+1} = v_t + \nabla v \cdot lr, \quad (15)$$

which yields the decay ratio:

$$R_o^{t+1} = \frac{S(v_t) - S(v_t + \nabla v \cdot lr)}{S(v_t)}. \quad (16)$$

Let $\Delta v = \nabla v \cdot lr$ and assume it is small, i.e., $|\Delta v| \ll 1$. A first-order Taylor expansion of $S(v_t + \Delta v)$ at v_t gives:

$$S(v_t + \Delta v) \approx S(v_t) + S'(v_t)\Delta v. \quad (17)$$

Since the derivative of the sigmoid is

$$S'(v) = S(v)[1 - S(v)] = \alpha(1 - \alpha), \quad (18)$$

we obtain

$$S(v_t + \Delta v) \approx \alpha_t + \alpha_t(1 - \alpha_t)\Delta v. \quad (19)$$

Substituting into the decay ratio:

$$\begin{aligned} R_o^{t+1} &= \frac{\alpha_t - S(v_t + \Delta v)}{\alpha_t} \\ &\approx -(1 - \alpha_t)\Delta v. \end{aligned} \quad (20)$$

Environmental pressure enforces $\Delta v < 0$, so we take the magnitude:

$$R_o^{t+1} \approx (1 - \alpha_t)|\Delta v| = (1 - \alpha)|\nabla v \cdot lr|. \quad (21)$$

When $|\Delta v| \rightarrow 0$, the approximation error is $\mathcal{O}(|\Delta v|^2)$, and the first-order expression becomes accurate.

Thus, the decay ratio is linearly proportional to $(1 - \alpha)$:

- As $\alpha \rightarrow 1$, the decay ratio vanishes, preserving consistently high-fitness Gaussians;
- As $\alpha \rightarrow 0$, the decay ratio approaches its maximum $|\nabla v \cdot lr|$, accelerating the removal of low-fitness individuals.
This finite prior therefore accelerates convergence while maintaining the fairness and adaptivity of natural selection.

A.5. Pruning based on 3DGS

Our method employs Improved-GS as the densification strategy to achieve the best compact 3DGS rendering quality, but this does not imply that our approach relies on specific prior work. Table 10 also presents the results of our method on 3DGS, with the budget uniformly set to 1/4 of the peak densification budget in 3DGS. Even within the 3DGS framework, our method achieves highly competitive pruning performance.

Dataset Method—Metric	$SSIM^\uparrow$	$PSNR^\uparrow$	$LPIPS^\downarrow$	Num^\downarrow
3DGS	0.816	27.50	0.216	3320453
Compact-3DGS	0.807	27.33	0.227	1516172
MaskGS	0.815	27.43	0.218	1582926
Ours(3DGS)	0.815	27.45	0.222	830000

Table 10. Quantitative results based on 3DGS densification method.

A.6. Opacity Learning Rate Scaling

Necessity. To complete the natural selection process within a limited number of training iterations, a sufficiently strong regularization gradient is required. If the original opacity learning rate is maintained, the selection outcome is heavily influenced by the *initial* opacity distribution, since the final opacity can be decomposed as:

$$\alpha_{\text{final}} = \alpha_{\text{init}} + \text{cumulative recovery} - \text{cumulative decay}.$$

Increasing the opacity learning rate during natural selection proportionally amplifies both the recovery and decay terms, thereby increasing the relative influence of dynamic optimization gradients and reducing the dependence on initial opacity.

Factors	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$
1×	0.786	25.71	0.227
2×	0.790	25.72	0.217
3×	0.792	25.80	0.214
4×	0.793	25.80	0.214
5×	0.792	25.76	0.213

Table 11. The impact of different opacity learning rate scaling factors on the rendering quality of the bicycle scene.

On the *bicycle* scene, we evaluate different scaling factors (Table 11). A scale of 4× already approaches the upper quality bound. However, excessively large learning rates also increase per-step decay, which may violate the assumption that gradients are approximately zero under the finite prior. Thus, we adopt a scaling factor of 4× as a balanced choice. After natural selection concludes, we allow the scene an additional 1,000 iterations for opacity recovery, after which the opacity learning rate will revert to its original value.

A.7. Learning Rate of the Regularization Gradient Field

lr	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$
0.002	0.773	25.51	0.239
0.0025	0.792	25.78	0.213
0.003	0.793	25.78	0.213
0.004	0.791	25.78	0.217
0.006	0.787	25.72	0.226
0.008	0.784	25.71	0.232

Table 12. The impact of regularization gradient field learning rate on the rendering quality of the bicycle scene.

Effect of the learning rate. The learning rate of the regularization gradient field directly controls the rate at which natural selection proceeds. After selection completes, a short period of fine-tuning is still required. Empirically, allowing natural selection to run for approximately **5K–8K** iterations yields the highest pruning quality.

We set the latest possible termination point at **23K** iterations. If the budget is not met by that point, a one-shot pruning based on opacity is applied. In the *bicycle* scene, six learning rates are tested (Table 12, Fig. 11). The second-smallest learning rate—used in our main experiments—enables natural selection to finish just before the latest allowed iteration.

Results show:

- Natural selection lasting 5K–8K iterations consistently achieves the best rendering quality.
- Insufficient selection time degrades pruning quality, though moderately.
- If the process does not finish on time and must fall back to one-shot pruning, rendering quality drops substantially due to the harshness and instability of one-shot removal.

Learning rate selection. Across scenes, we observe that **higher redundancy** corresponds to **lower required learning rates**. Redundancy depends on scene complexity (resolution, detail density) and the extent to which high Gaussian counts actually improve rendering quality. Indoor scenes usually exhibit lower complexity and redundancy, while higher-resolution scenes contain richer details and thus lower redundancy.

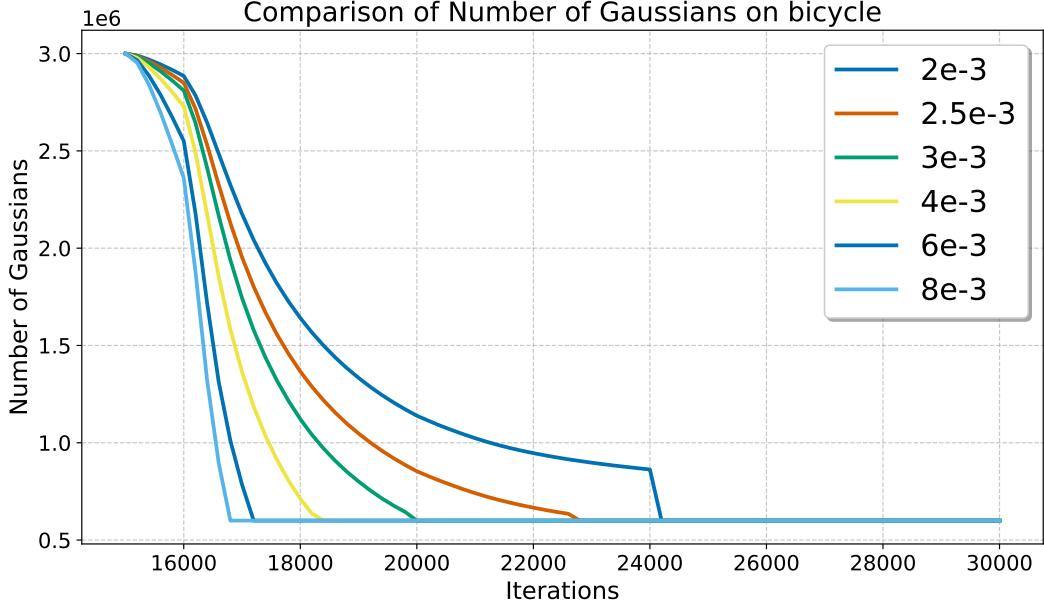


Figure 11. In the bicycle scene, pruning curves corresponding to different regularization learning rates.

A.8. Automated Learning Rate

By tracking the opacity change curve of Gaussians ranked at the final budget under the optimal learning rate, we observe that the overall trend is linear. Consequently, we can dynamically adjust the learning rate by comparing the opacity of the current target Gaussian with the preset curve. This version was completed after the paper submission and has been included in the code provided in Supplementary Material. Additionally, this version sets the densification budget to three times the final budget, meaning that for each scene, only the final budget parameter needs to be specified. This parameter can be easily automated by adjusting the gradient threshold for densification. However, manual control of the final budget remains a desirable feature, so we have chosen to retain it.

The automated learning rate version achieves rendering quality nearly identical to that of manual parameter tuning. However, since the main text version still employs manual parameter tuning, the results provided in the appendix are also based on manually tuned parameters.

A.9. FPS Evaluation

Methods	ImprovedGS-s	SPA	MaskGS	Opacity Pruning	Render Pruning	Edge Pruning	Natural Selection(Ours)
FPS	153	174	177	197	150	153	193

Table 13. FPS under the same budget and with Improved-GS as the shared pre-stage.

Table 13 reports FPS under the same budget and with Improved-GS as the shared pre-stage. Our method achieves slightly higher FPS than Improved-GS.

FPS is primarily determined by the average length of the per-pixel rendering queue. MiniGS significantly improves FPS by employing depth reinitialization and a “max-contribution-only” retention rule, both of which shorten rendering queues. However:

- depth reinitialization incurs considerable extra optimization time, and
- retaining only the maximum-contribution Gaussian contradicts the objective of preserving the highest rendering quality.

Since our focus is on improving **quality** in compact 3DGs rather than maximizing FPS, direct comparison with MiniGS is not entirely appropriate. Compared with other pruning methods, our approach yields no noticeable FPS disadvantage while achieving superior quality.

Dataset Method—Metric	Mip-NeRF360			
	$SSIM^\uparrow$	$PSNR^\uparrow$	$LPIPS^\downarrow$	Num^\downarrow
MiniGS	0.822	27.36	0.217	493466
GaussianSPA-30K	0.817	27.31	0.229	421427
GaussianSPA-40K	0.821	27.58	0.224	421086

Table 14. Quantitative results based on 3DGS densification method.

A.10. Reproduction Issue of GaussianSPA

The GaussianSPA paper states that pruning is based solely on opacity. However, the official implementation instead ranks Gaussians using:

$$\text{current opacity} + \text{accumulated decay offset}.$$

In each pruning step, Gaussians within the top budget are exempted, whereas the others undergo decay. The accumulated offset corresponds to past decay, and due to the implemented ranking rule, Gaussians that have undergone multiple decays (i.e., originally low-opacity Gaussians) gradually move forward in the ranking, making them increasingly likely to be retained.

Thus, the implementation effectively becomes:

$$\text{opacity pruning} + \text{partial random pruning},$$

introducing unintended stochasticity. When we correct this logic, performance noticeably decreases—likely because the stochastic component implicitly mimics the gradient-competition behavior of our natural selection, thereby improving performance “by accident.”

SPA’s true contribution lies in its ADMM-based smooth attenuation mechanism, which is independent of any specific pruning weight. Although a MiniGS-based pruning weight is provided, it requires computing per-view maximum contribution counts, making training several times slower. For fairness, we report the original SPA results in the main paper and explain the discrepancy in the appendix.

Additionally, SPA uses **40K** iterations in its paper while all comparison methods use **30K**. According to the authors, SPA continues to improve beyond 30K whereas other methods overfit. For fair comparison, we report SPA at 30K in the main paper, and its 40K results are included in Table 14.

Method—Metric	$SSIM^\uparrow$	$PSNR^\uparrow$	$LPIPS^\downarrow$	Num^\downarrow
base	0.792	25.78	0.213	
$T2.5 \times \text{and} lr0.4 \times$	0.792	27.78	0.212	

Table 15. Different combinations of T and learning rate in quality.

A.11. Effect of the Finite Prior Parameter T

Table 15 presents results obtained using different combinations of T and learning rate. Increasing T is theoretically equivalent to increasing the learning rate because, under the finite-prior formulation:

$$\nabla v = 2(\mathbb{E}[v] - T).$$

When $T \gg \mathbb{E}[v]$, the gradient magnitude is dominated by T , making $|\nabla v|$ proportional to $|T|$.

When T is scaled to 2.5 times its original value, the effective gradient scaling factor becomes:

$$\frac{2.5T - \mathbb{E}[v]}{T - \mathbb{E}[v]} = 2.5 + \frac{1.5\mathbb{E}[v]}{T - \mathbb{E}[v]}.$$

Since $\mathbb{E}[v] < 0$ in practice (as described in the main text), the second term is positive, meaning the effective growth factor is greater than 2.5.

Thus, if T increases by $2.5 \times$ while the learning rate is reduced by $2.5 \times$, the actual natural-selection speed becomes slightly faster—consistent with the observed results in Fig. 12.

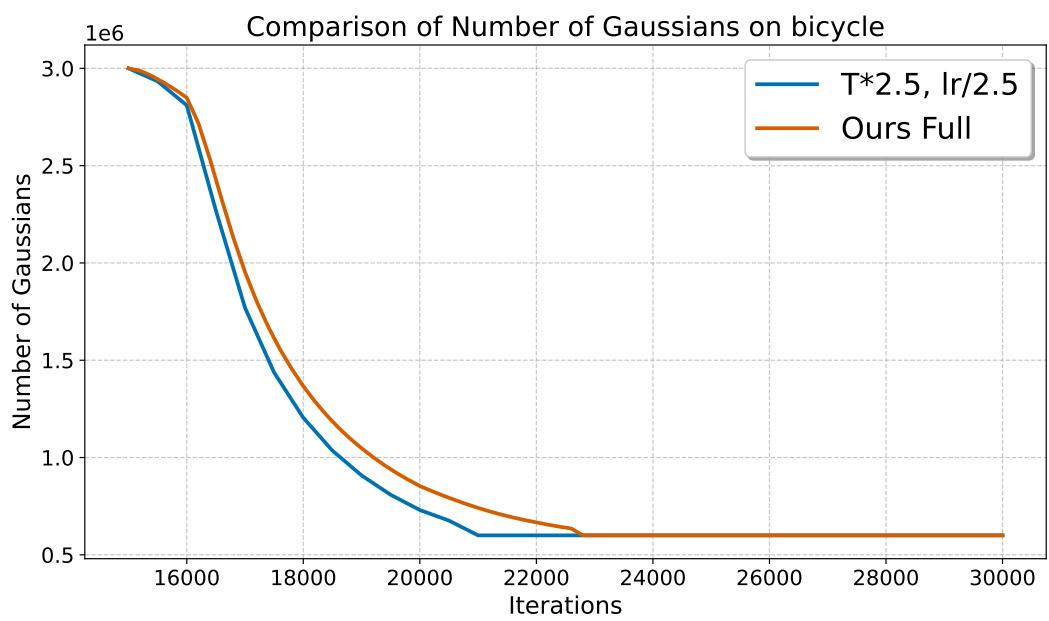


Figure 12. Different combinations of T and learning rate in pruning speed.