L1RESPRUNE: GENERAL PRUNING FOR EXPLICIT SCENE REPRESENTATION PRIMITIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we present L1 Loss Residual Selective Pruning, termed L1ResPrune, a general pruning method for explicit scene representation primitives. L1ResPrune is applicable to both static and dynamic scenes and is compatible with representations based on Gaussians as well as other types of primitives. Building upon the standard alpha-blending formulation, we derive an importance score for each primitive based on the change in the L1 loss induced by its removal. This formulation provides a theoretically grounded criterion for pruning and does not rely on the specific characteristics of any particular representation primitive. Extensive experiments demonstrate that, compared with existing 3D Gaussian Splatting based pruning methods, L1ResPrune consistently preserves higher rendering quality while achieving superior SSIM and LPIPS performance.

1 INTRODUCTION

Unlike implicit Neural Radiance Fields (Mildenhall et al., 2020), 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has become a mainstream approach for novel view rendering. However, explicit representations often contain substantial redundant primitives, leading to unnecessary memory and computational costs. To obtain compact representations, many pruning methods have been proposed. Nevertheless, existing approaches suffer from two major limitations. First, their pruning criteria are mainly heuristic, relying on empirical observations of alpha blending rather than analyzing the actual impact of primitive removal on rendering quality. Second, most methods are tailored to static Gaussian-based representations and lack general applicability to dynamic scenes or other primitive-based representations.

In this paper, we propose a theoretical and general pruning framework for explicit scene representations. Based on the standard alpha-blending formulation, we derive a primitive importance score by measuring the change in L1 residual caused by primitive removal. This theoretically grounded criterion enables more accurate pruning decisions while avoiding heuristic assumptions. Furthermore, since our formulation depends only on the general rendering process rather than primitive-specific properties, it can be applied to diverse explicit representations, including Gaussian and non-Gaussian primitives.

2 RELATED WORK

Primitive redundancy is a fundamental challenge in explicit scene representations, and pruning has become an important strategy for obtaining compact representations. Among existing approaches, 3D Gaussian Splatting (3DGS) has attracted significant attention due to its explicit and fast rendering. The redundancy of Gaussians mainly originates from the densification process and subsequent optimization. Existing pruning methods can be broadly classified into three categories.

2.1 ADAPTIVE DENSITY CONTROL

The first category focuses on improving Adaptive Density Control (ADC) during training to prevent excessive Gaussian generation. Methods such as Taming3DGS (Mallick et al., 2024) and SteepestGS (Wang et al., 2025a) modify densification criteria or introduce explicit Gaussian budgets to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

constrain primitive growth. However, these methods face an inherent trade-off: insufficient primitives may limit optimization capability, while additional primitives introduced for better convergence can increase redundancy.

2.2 Mask-based Pruning

Mask-based approaches, such as Compact3DGS (Lee et al., 2024) and MaskGS (Liu et al., 2025), introduce learnable masks and remove primitives with low mask values. Since mask optimization is usually performed during training, early pruning may affect convergence and limit the achievable pruning ratio.

2.3 Post-training Pruning

The third category removes redundant primitives after optimization. Existing methods primarily rely on importance scores, which constitute the predominant approach to primitive pruning.

Importance-score-based methods define explicit criteria to evaluate primitive contribution and prune according to their importance scores. Trimming The Fat (Ali et al., 2024) removes Gaussians with low opacity and gradient values. LightGS (Fan et al., 2025), EAGLES (Girish et al., 2024), MesonGS (Xie et al., 2024), and RadSplat (Niemeyer et al., 2025) construct importance scores based on opacity, scale, and transmittance. Mini-Splatting (Fang & Wang, 2024) further considers ray-wise contribution and normalized importance for efficient pruning. PUP3DGS (Hanson et al., 2025b) introduces a theoretically motivated criterion by estimating second-order gradients of the L2 loss with respect to Gaussian parameters, and subsequent works (Hanson et al., 2025a; Wang et al., 2026) extend this framework with additional gradient or frequency information. Other methods instead explore different criteria, including multi-view consistency (Ren et al., 2025), heuristic scale-opacity rules (Du et al., 2026), global optimization toward sparse representations (Wang et al., 2025b), and evolution-inspired pruning strategies (Deng et al., 2025).

2.4 Other Approaches

Beyond direct pruning, several methods explore alternative strategies. Distilled-3DGS (Xiang et al., 2025) applies knowledge distillation to reduce Gaussian numbers but requires additional teacher models and training cost. LP-3DGS (Zhang et al., 2024a) uses Gumbel-Sigmoid to automatically determine scene-dependent pruning ratios, while LocoGS (Shin et al., 2025) exploits local coherence among Gaussians for compact representation.

Despite extensive progress, existing pruning methods largely rely on empirically motivated importance criteria. Although some approaches introduce theoretically motivated formulations, their criteria are often coupled with Gaussian-specific properties or particular optimization objectives. Furthermore, most methods are developed for static Gaussian scenes and cannot be directly extended to dynamic scenes or other primitive-based representations. In contrast, our method derives primitive importance from the general alpha-blending formulation and the L1 residual caused by primitive removal, providing a principled and representation-agnostic pruning criterion.

3 Background

3.1 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) Kerbl et al. (2023) is a novel-view synthesis method that represents a scene explicitly using a set of 3D Gaussian primitives. Each Gaussian is parameterized by a center position $x^{3D} \in \mathbb{R}^3$, a rotation quaternion $q \in \mathbb{R}^4$, a scale vector $s \in \mathbb{R}+^3$, an opacity $o \in \mathbb{R}+$, and a set of spherical harmonic coefficients for view-dependent color representation. Gaussian primitive $G^{3D}(x)$ is formulated as:

$$G^{3D}(x) = e^{-\frac{1}{2}(x-x^{3D})^T(\Sigma^{3D})^{-1}(x-x^{3D})}, \tag{1}$$

and its covariance matrix is parameterized as:

$$\Sigma^{3D} = RSS^TR^T, \tag{2}$$

where Σ^{3D} denotes the 3D covariance matrix, R is the rotation matrix corresponding to quaternion q, and S is the diagonal scale matrix constructed from s.

During rendering, given a camera viewpoint θ, each visible 3D Gaussian is projected onto the image plane and represented as a 2D Gaussian $G_\theta^{2D}(x)$:

$$G_\theta^{2D}(x) = e^{-\frac{1}{2}(x-x_\theta^{2D})^T(\Sigma_\theta^{2D})^{-1}(x-x_\theta^{2D})}, \tag{3}$$

where

$$\Sigma_\theta^{2D} = JW_\theta\Sigma^{3D}W_\theta^T J^T. \tag{4}$$

Here, Σ_θ^{2D} denotes the covariance matrix of the projected 2D Gaussian, J is the Jacobian of the perspective projection, x_θ^{2D} is the projected Gaussian center on the image plane, and W_θ is the transformation matrix from the world coordinate system to the camera coordinate system under viewpoint θ.

The N_θ visible 2D Gaussians are sorted according to their depth and composited using alpha blending. The rendered color C_p^θ at pixel p is given by:

$$C_p^\theta = \sum_{i=1}^{N_\theta} c_{i,p}^\theta \alpha_{i,p}^\theta \prod_{j=1}^{i-1}(1-\alpha_{j,p}^\theta), \tag{5}$$

where $c_{i,p}^\theta$ denotes the color contributed by the i-th Gaussian at pixel p under viewpoint θ, and $\alpha_{i,p}^\theta$ denotes its opacity contribution at that pixel. In 3DGS, the opacity contribution is defined as

$$\alpha_{i,p}^\theta = o_i G_{\theta,i}^{2D}(p), \tag{6}$$

while $c_{i,p}^\theta$ is computed from the spherical harmonic coefficients of the i-th Gaussian according to the viewing direction.

3.2 Non-Gaussian Explicit Representations

To address the limitations of 3D Gaussian primitives in explicit scene representation, particularly in modeling geometric structures, several works have explored alternative primitives, including 2D Gaussians (Huang et al., 2024), triangles (Held et al., 2025a), and convex hulls (Held et al., 2025b). Despite their differences in geometric representation, these primitives can also be rendered through alpha blending, where the opacity field $\alpha_{i,p}^\theta$ is defined according to the corresponding primitive. For example, the opacity field can be defined over a 2D Gaussian, a triangle, or a convex hull. This shared alpha-blending formulation provides a general basis for analyzing the contribution of different explicit representation primitives.

4 Method

In this section, we present L1 Loss Residual Selective Pruning (L1ResPrune). The key idea is to evaluate the importance of each primitive from the L1 loss residual caused by removing its contribution. Specifically, we first analyze the primitive contribution at a single pixel under a single viewpoint. Based on this analysis, we derive a per-pixel single-view importance score. Finally, these scores are aggregated across pixels and viewpoints to obtain the global importance score for primitive pruning. For low-ratio pruning, we further introduce efficient simplification schemes.

4.1 Mathematical Model

Given a camera viewpoint θ, we consider the alpha contribution $\alpha^\theta = (\alpha_{i,p}^\theta)_{i=1,\dots,N,p\in P}$ of all primitives. N is the number of primitives and P denotes the set of image pixels. We introduce a binary mask $M^\theta = (m_{i,p}^\theta)$ to indicate whether the contribution of primitive i at pixel p is retained. Specifically, $m_{i,p}^\theta = 0$ means that the contribution is removed, whereas $m_{i,p}^\theta = 1$ means that it is retained.

Let $L_{1,p}^\theta(\alpha^\theta)$ denote the pixel-wise L_1 loss under viewpoint θ, and $L_{1,p}^\theta(M^\theta \odot \alpha^\theta)$ denotes the pixel-wise L_1 loss after removing a subset of Gaussian primitives. Our objective is to maximize the number of removed primitives while keeping the induced loss variation negligible:

$$\left|L_{1,p}^\theta(M^\theta \odot \alpha^\theta) - L_{1,p}^\theta(\alpha^\theta)\right| \approx 0. \tag{7}$$

Here, $\odot$ denotes element-wise multiplication. Eq.7 indicates that primitives can be safely removed when their removal introduces only a negligible pixel-wise residual.

4.2 Analysis

The key problem is to characterize the residual $L^{\theta}_{1,p}(M^{\theta} \odot \alpha^{\theta}) - L^{\theta}_{1,p}(\alpha^{\theta})$. It can be viewed as evaluating the function $L^{\theta}_{1,p}(x)$ at a perturbed point $x = M^{\theta} \odot \alpha^{\theta}$ around α^{θ}. Therefore, the residual can be analyzed using Taylor expansion. The standard L_1 loss is non-differentiable when the rendered color equals the ground-truth color. To enable the following analysis, we introduce an infinitesimal perturbation $n^{\theta}_{c,p}$ to the ground-truth image that satifies $C^{\theta}_{c,p} \neq qC^{gt,\theta}_{c,p} + n^{\theta}_{c,p}$, where $c \in \{r, g, b\}$. The modified pixel-wise loss is defined as:

$$L^{\theta}_{1,p}(\alpha^{\theta}) = \frac{1}{3} \sum_{c \in \{r,g,b\}} \left| C^{\theta}_{c,p} - (C^{gt,\theta}_{c,p} + n^{\theta}_{c,p}) \right|. \tag{8}$$

Then the residual can be bounded by the Taylor expansion terms:

$$\left| L^{\theta}_{1,p}(M^{\theta} \odot \alpha^{\theta}) - L^{\theta}_{1,p}(\alpha^{\theta}) \right| \leq \sum_{k=1}^{\infty} |T_k| \tag{9}$$

The k-th order Taylor term is:

$$|T_k| = \frac{1}{k!} \sum_{I_k} \left\{ \left| \frac{\partial^k L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_1,p} ... \partial \alpha^{\theta}_{i_k,p}} \right| \prod_{i \in I_k} (1 - m^{\theta}_{i,p}) \alpha^{\theta}_{i,p} \right\}, \tag{10}$$

where $I_k = (i_1, ..., i_k)$ represents the ordered set of primitives involved in the k-th order interaction.

To satisfy Eq. 7, the Taylor residual terms should be minimized. We first analyze the first-order term:

$$|T_1| = \sum_i (1 - m^{\theta}_{i,p}) \alpha^{\theta}_{i,p} \left| \frac{\partial L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i,p}} \right|. \tag{11}$$

Therefore, $\alpha^{\theta}_{i,p} \left| \frac{\partial L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i,p}} \right|$ measures the partial influence of primitive i on pixel p under viewpoint θ. When the value is small, $m^{\theta}_{i,p}$ can be set to zero to eliminate its contribution. Otherwise, it is retained. However, the first-order term ignores the interaction among multiple primitives, i.e. higher-order terms. Therefore, we further analyze higher-order terms. For alpha blending, the higher-order derivatives can be decomposed into primitive-wise independent components. The general form of the higher-order terms is:

$$|T_k| = \frac{1}{k!} \sum_{I_k} \{ (1 - m^{\theta}_{i_1,p}) \alpha^{\theta}_{i_1,p} \left| \frac{\partial L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i_1,p}} \right| \prod_{\substack{i \in I_k \\ i \neq i_1}} (1 - m^{\theta}_{i,p}) \frac{\alpha^{\theta}_{i,p}}{1 - \alpha^{\theta}_{i,p}} \}, \quad k \geq 2. \tag{12}$$

The detailed derivation is provided in the appendix. We denote the deepest primitive in the primitive set I_k as i_1. Thus, Eq. 12 shows that the primitive with the largest depth contribution is equivalent to the first-order influence term, while other primitives contribute through the transmittance compensation term $\frac{\alpha^{\theta}_{i,p}}{1 - \alpha^{\theta}_{i,p}}$. Therefore, to jointly consider first-order and higher-order residuals, we modify the primitive influence term as $\frac{\alpha^{\theta}_{i,p}}{1 - \alpha^{\theta}_{i,p}} \left| \frac{\partial L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i,p}} \right|$.

4.3 Per-Pixel Single-View Importance Score

Based on the above analysis, we define the importance of primitive i at pixel p under viewpoint θ as:

$$score_n(i, \theta, p) = \left(\frac{\alpha^{\theta}_{i,p}}{1 - \alpha^{\theta}_{i,p}} \left| \frac{\partial L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i,p}} \right| \right)^n. \tag{13}$$

The exponent n controls the sensitivity to large responses. A larger n emphasizes pixels where the primitive has a stronger influence, while reducing the contribution from pixels with consistently low importance.

4.4 GLOBAL IMPORTANCE SCORE

The above score only measures the importance of a primitive from a single pixel and viewpoint. To obtain the final primitive importance score, we aggregate the scores over pixels and viewpoints. We consider three aggregation strategies below.

Global Aggregation. Global aggregation directly accumulates all pixel-wise scores:

$$S(i) = \sum_{\theta} \sum_{p} score_n(i, \theta, p). \tag{14}$$

This strategy considers both the contribution magnitude and observation frequency of primitives. However, it may introduce bias when the training views are unevenly sampled. This aggregation strategy has been widely adopted by existing methods (Fan et al., 2025; Hanson et al., 2025b; Fang & Wang, 2024; Hanson et al., 2025a; Wang et al., 2026).

Visibility Aggregation. To reduce the influence of uneven viewpoint sampling, we average the importance only over viewpoints where the primitive is visible:

$$S(i) = \frac{\sum_{\theta \in V_i} \sum_{p} score_n(i, \theta, p)}{|V_i|}, \tag{15}$$

where V_i denotes the set of viewpoints containing primitive i. This strategy provides a more stable estimation of primitive importance, as it is independent of view density.

Maximum Aggregation. When n becomes sufficiently large, the aggregation is dominated by the maximum single-view single-pixel response:

$$S(i) = \max_{\theta, p} score_n(i, \theta, p). \tag{16}$$

This strategy preserves primitives that have strong contributions in at least one viewpoint, which is the aggregation strategy adopted by RadSplat.(Niemeyer et al., 2025).

4.5 PRUNING STRATEGY

After obtaining the global importance score, pruning can be performed according to the score distribution. Pruning strategies are categorized into fixed-ratio and adaptive pruning. Fixed-ratio pruning removes a predefined percentage of primitives with the lowest global importance scores. It can be applied to arbitrary importance aggregation but requires manually selecting the pruning ratio for different scenes. Adaptive pruning removes primitives whose importance scores are below a predefined threshold. This enables automatic pruning ratios across different scenes when the score distribution is consistent. In general, visibility aggregation and maximum aggregation yield stable global importance scores and can be used for adaptive pruning, whereas global aggregation does not.

4.6 SIMPLIFICATION(OPTIONAL)

Computing Eq. 13 requires back-propagation to obtain the gradient term. For low-ratio pruning, we observe that the gradient term is upper-bounded by 1 and has limited influence on primitive ranking. Therefore, we remove the gradient term and obtain:

$$score_n(i, \theta, p) = \left(\frac{\alpha_{i,p}^{\theta}}{1 - \alpha_{i,p}^{\theta}} \right)^n. \tag{17}$$

We refer to Eq. 17 as Non-Gradient Simplification (NGS). Furthermore, for Gaussian Splatting, the pixel summation can be approximated by integrating the continuous Gaussian field:

$$\sum_{p} score_n(i, \theta, p) \approx \iint \left(\frac{o_i G_{\theta,i}^{2D}(x)}{1 - o_i G_{\theta,i}^{2D}(x)} \right)^n dx = 2\pi \sqrt{\lambda_1 \lambda_2} \int_{1-o_i}^{1} \frac{(1-u)^{n-1}}{u^n} du, \tag{18}$$

where λ_1 and λ_2 are eigenvalues of the projected 2D covariance matrix. We refer to Eq. 18 as Integral Approximation Simplification (IAS). With IAS, only primitive projection and covariance computation are required, without rendering or back-propagation. Other primitives can be treated similarly, as long as the corresponding quantities can be computed.

Table 1: Comparison of static 3DGS pruning methods. Low- and high-ratio pruning use per-scene adaptive ratios aligned across methods, while extreme-ratio pruning uses a fixed 90% ratio. The number of Gaussian primitives for each scene is provided in the appendix. **Bold** and underlined denote the best and second-best results, respectively.

Method	MipNeRF-360				Tanks & Temples				Deep Blending			
	PSNR↑	SSIM↑	LPIPS↓	GS(M)↓	PSNR↑	SSIM↑	LPIPS↓	GS(M)↓	PSNR↑	SSIM↑	LPIPS↓	GS(M)↓
Backbone 3DGS (Kerbl et al., 2023)	27.55	0.813	0.221	2.646	23.67	0.852	0.170	1.572	29.76	0.906	0.238	2.478
Low-ratio pruning												
LightGS (Fan et al., 2025)	27.12	0.803	0.244	1.137	23.03	0.822	0.227	0.481	27.40	0.879	0.295	0.720
Compact3DGS (Lee et al., 2024)	27.05	0.798	0.246	1.354	23.40	0.834	0.202	0.841	**29.91**	0.904	0.255	1.044
MaskGS (Liu et al., 2025)	27.41	0.811	**0.227**	1.161	23.61	0.845	0.184	0.549	29.78	0.908	0.247	0.572
Mini-Splatting (Fang & Wang, 2024)	27.34	0.810	0.229	1.137	23.47	0.841	0.194	0.481	29.81	0.907	0.247	0.720
PUP3DGS (Hanson et al., 2025b)	27.42	0.809	0.229	1.137	22.80	0.827	0.208	0.481	29.90	**0.910**	**0.242**	0.720
GHAP (Wang et al., 2025b)	27.36	0.805	0.243	1.137	**23.71**	0.843	0.199	0.481	29.83	0.908	0.249	0.720
Speedy-Splat (Hanson et al., 2025a)	27.52	**0.813**	0.227	1.137	23.70	0.846	0.188	0.481	29.59	0.906	0.249	0.720
L1ResPrune	**27.55**	**0.813**	**0.224**	1.137	23.62	**0.850**	**0.181**	0.481	29.86	0.909	**0.242**	0.720
High-ratio pruning												
Mini-Splatting (Fang & Wang, 2024)	27.11	0.802	0.248	0.541	23.37	0.835	0.207	0.310	29.68	0.905	0.253	0.500
PUP3DGS (Hanson et al., 2025b)	26.67	0.796	0.252	0.541	22.26	0.811	0.228	0.310	29.81	**0.909**	0.246	0.500
GHAP (Wang et al., 2025b)	27.05	0.789	0.274	0.541	**23.69**	0.837	0.212	0.310	29.76	0.907	0.254	0.500
Speedy-Splat (Hanson et al., 2025a)	**27.34**	0.804	0.250	0.541	23.61	0.840	0.201	0.310	29.58	0.907	0.252	0.500
L1ResPrune	27.31	**0.806**	**0.242**	0.541	23.45	**0.845**	**0.191**	0.310	**29.83**	**0.909**	**0.245**	0.500
Extreme-ratio pruning												
Mini-Splatting (Fang & Wang, 2024)	26.56	0.779	0.286	0.265	22.93	0.812	0.243	0.158	**29.69**	0.901	0.267	0.248
PUP3DGS (Hanson et al., 2025b)	25.25	0.765	0.299	0.265	21.26	0.775	0.275	0.158	29.29	0.902	0.266	0.248
GHAP (Wang et al., 2025b)	26.48	0.766	0.314	0.265	23.32	0.818	0.245	0.158	29.58	0.904	0.266	0.248
Speedy-Splat (Hanson et al., 2025a)	**26.74**	0.776	0.298	0.265	**23.33**	0.813	0.247	0.158	29.54	0.903	0.270	0.248
L1ResPrune	26.62	**0.785**	**0.281**	0.265	23.31	**0.822**	**0.234**	0.158	29.51	**0.905**	**0.258**	0.248

5 EXPERIMENT

5.1 EXPERIMENTAL SETTINGS AND DETAILS

Datasets and Metrics. We evaluate L1ResPrune on five benchmark datasets, including three static scene datasets: Mip-NeRF360 (Barron et al., 2022), Tanks & Temples (Knapitsch et al., 2017), and Deep Blending (Hedman et al., 2018), as well as two dynamic scene datasets: D-NeRF (Pumarola et al., 2020) and NeRF-DS (Yan et al., 2023). We use peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) (Wang et al., 2004), and learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018) as rendering quality metrics. Additionally, we report the number of remaining primitives to evaluate the compactness of the reconstructed representation. All methods are trained and evaluated using the same image resolution and camera poses. The input camera poses are obtained from COLMAP (Schönberger & Frahm, 2016).

Experimental Setup. We evaluate L1ResPrune from three perspectives: (1) pruning performance on static scenes, (2) pruning performance on dynamic scenes, and (3) generalization to different explicit primitive-based representations. Additionally, we evaluate the effectiveness of L1ResPrune when applied to 3DGS improvement methods. These additional experiments and the analysis of different importance score variants, including Full, NGS, and IAS, are provided in the appendix.

Implementation Details of Comparative Methods. For fair comparison, all pruning methods are implemented using the same backbone, training iterations, pruning schedules, and target pruning ratios. For methods, i.e, Compact3DGS and MaskGS whose pruning ratios cannot be exactly controlled, we report the closest achievable pruning ratio. To ensure that the comparison focuses only on pruning effectiveness, we exclude additional components such as quantization or quality-enhancement modules from all compared methods.

Implementation Details of L1ResPrune. For static and dynamic scene experiments, we adopt the Full version of our importance score with $n = 2$, corresponding to Eq. 13. The global importance score is computed using global aggregation. This setting is selected because most compared pruning methods accumulate importance scores directly over all training views. For experiments involving different explicit primitive representations, we adopt the NGS version with $n = 2$, corresponding to Eq. 17, and compute the global importance score using visibility aggregation.

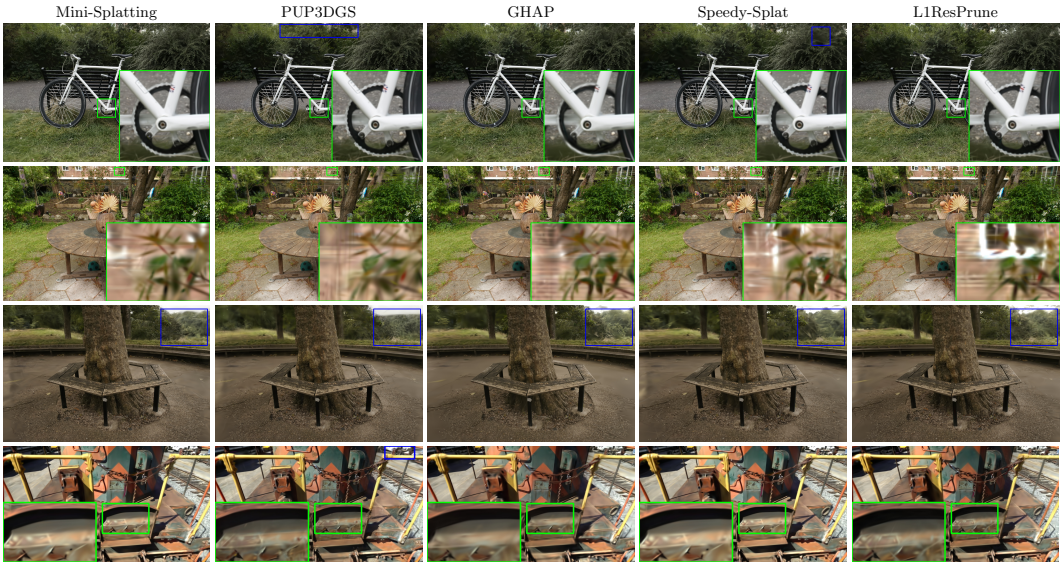


Figure 1: Dynamic visualization results.

5.2 Comparison of Static Scenes

We first evaluate L1ResPrune on static scenes represented by 3D Gaussian Splatting, comparing it with representative pruning methods, including LightGS (Fan et al., 2025), Compact3DGS (Lee et al., 2024), MaskGS (Liu et al., 2025), Mini-Splatting (Fang & Wang, 2024), PUP3DGS (Hanson et al., 2025b), GHAP (Wang et al., 2025b), and Speedy-Splat (Hanson et al., 2025a).

We consider three compression levels: low-ratio, high-ratio, and extreme-ratio pruning, covering both moderate and aggressive primitive reduction. For fair comparison, all methods are evaluated under a unified primitive budget. The target number of remaining Gaussians except for extreme-ratio pruning, is determined per scene by L1ResPrune using visibility aggregation and the IAS importance score with $n = 2$, and the same budget is applied to all methods. Compact3DGS and MaskGS optimize pruning masks during training and do not support arbitrary primitive budgets. Thus, they are evaluated only under low-ratio pruning. For all other methods, pruning is performed once at 15K iterations, followed by fine-tuning to 30K iterations.

We additionally evaluate MobileGS (Du et al., 2026), FastGS (Ren et al., 2025), and PruneWisely (Wang et al., 2026). While these methods achieve reasonable performance, their gains over the baseline are relatively modest. We therefore include their results in the appendix.

Quantitative and Quality Results. Tab. 1 summarizes the quantitative comparisons under different pruning ratios. L1ResPrune achieves competitive or superior performance compared with existing pruning methods, especially in terms of SSIM and LPIPS metrics. As the pruning ratio increases, the performance gap becomes more evident, demonstrating that our method can better preserve visually important primitives under aggressive compression.

The visual comparisons under extreme-ratio pruning are shown in Fig. 1. Compared with other approaches, L1ResPrune preserves more fine structures and produces fewer noticeable artifacts, further verifying the effectiveness of our importance score.

Analysis. Mask-based methods, including Compact3DGS and MaskGS, learn masks to identify redundant Gaussians, but operate primarily on opacity and do not explicitly model pixel-wise alpha contributions. Mini-Splatting considers both average and maximum Gaussian contributions, enabling aggressive pruning by preserving view-specific primitives, but its criteria are manually designed. PUP3DGS estimates pruning sensitivity using the Fisher information matrix, mainly based on position and scale gradients, considering only second-order terms with additional simplifications, while GHAP performs Global Reduction by optimizing a compact Gaussian set, which can alleviate structural disruption but may alter well-optimized geometry under high compression. In contrast,

L1ResPrune directly derives primitive importance from the rendering L1 residual, explicitly modeling the effect of primitive removal. Compared with Speedy-Splat, L1ResPrune achieves better SSIM and LPIPS under high- and extreme-ratio pruning. Notably, Speedy-Splat's importance formulation $\sum_\theta \sum_p (\nabla_{G^{2D,\theta}_{i,p}} I^\theta)^2 = \sum_\theta \sum_p (o_i \frac{\partial L_1^\theta}{\partial \alpha_i})^2$ shares a similar form with our global aggregation in Eq. 14 for $n = 2$, suggesting that it implicitly captures part of the L1-residual-based importance. However, it omits the complete alpha contribution term derived in our analysis, which helps explain the superior robustness of L1ResPrune under extreme pruning. An additional benefit of our method is that it provides a theoretical interpretation of the performance gains observed with Speedy-Splat, which were not explicitly characterized in the original work. Overall, L1ResPrune provides a simple and theoretically grounded pruning criterion that better preserves rendering quality.

5.3 Comparison of Dynamic Scenes

To evaluate the generalization of L1ResPrune beyond static scenes, we conduct experiments on dynamic scene datasets. We compare with SpeeDe3DGS (Tu et al., 2026), a dynamic 3DGS pruning method that extends Speedy-Splat with temporal information, and Mini-Splatting (Fang & Wang, 2024) adapted to dynamic scenes. All methods use Deformable3DGS (Yang et al., 2023) as the backbone. Pruning is performed at 15K iterations, followed by fine-tuning to 40K iterations on D-NeRF and 30K on NeRF-DS. For SpeeDe3DGS, we only use its pruning components, Temporal Sensitivity Pruning (TSP) and Temporal Sensitivity Sampling (TSS), excluding components that modify optimization to ensure a fair comparison of pruning strategies.

L1ResPrune uses the same configuration as in static scenes: the full from importance score with $n = 2$, fix-ratio pruning, and global aggregation. This directly evaluates whether its importance formulation generalizes to dynamic representations without scene-specific modifications.

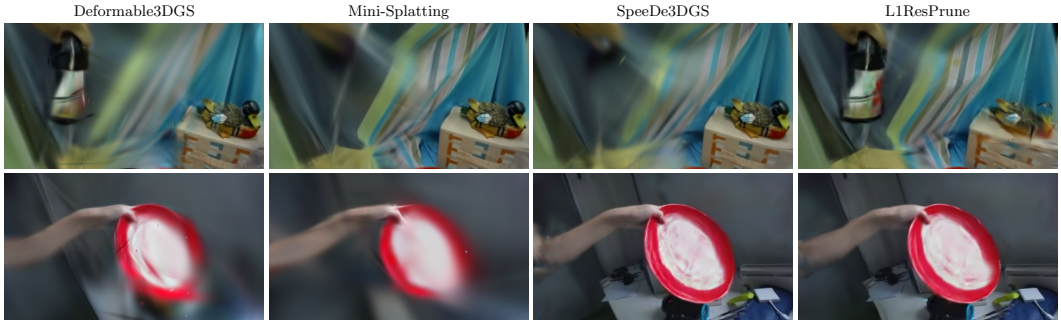


Figure 2: Dynamic visualization results.

Quantitative and Quality Results. Tab. 2 and Fig. 2 show the quantitative and qualitative results, respectively. Under the same primitive budget, L1ResPrune consistently achieves strong performance. On D-NeRF dataset, it obtains comparable PSNR to SpeeDe3DGS while achieving better SSIM and LPIPS. On the more challenging NeRF-DS, it outperforms all compared methods, improving over SpeeDe3DGS from 23.34 to 23.74 PSNR, 0.836 to 0.850 SSIM, and 0.207 to 0.178 LPIPS, demonstrating better preservation of structural and perceptual details.

Unlike SpeeDe3DGS, which introduces dynamic-specific temporal sensitivity modeling, L1ResPrune derives primitive importance directly from the rendering residual of the general alpha-blending formulation. It therefore requires no temporal assumptions or scene-specific modifications and can be directly applied to both static and dynamic explicit representations.

6 Applications to Non-Gaussian Primitive Representations

Unlike pruning methods designed specifically for Gaussian primitives, L1ResPrune is derived from the general alpha-blending formulation and does not rely on primitive-specific properties. It can therefore be applied to different explicit scene representations. We evaluate four representative types: AAA-Gaussian (Steiner et al., 2025), 2DGS (Huang et al., 2024), 3DCS (Held et al., 2025b),

Table 2: Comparison of dynamic 3DGS pruning methods under the same per-scene primitive budget. **Bold** and underlined denote the best and second-best results, respectively.

Method	D-NeRF				NeRF-DS			
	PSNR↑	SSIM↑	LPIPS↓	GS(K)↓	PSNR↑	SSIM↑	LPIPS↓	GS(K)↓
Backbone Deformable3DGS (Yang et al., 2023)	38.88	0.990	0.012	145	23.18	0.824	0.232	232
Mini-Splatting (Fang & Wang, 2024)	38.63	0.988	0.017	29	22.99	0.824	0.233	46
SpeeDe3DGS (Tu et al., 2026)	38.73	0.988	**0.016**	29	23.34	0.836	0.207	46
L1ResPrune	**38.74**	**0.989**	**0.016**	29	**23.74**	**0.850**	**0.178**	46

Table 3: Pruning results on non-Gaussian primitive representations.

Method	MipNeRF-360			
	PSNR↑	SSIM↑	LPIPS↓	Num(M)↓
AAA-Gaussians (Steiner et al., 2025)	27.78	0.831	0.190	3.167
AAA-Gaussians+L1ResPrune	27.76	0.827	0.212	0.856
2DGS (Huang et al., 2024)	26.81	0.794	0.260	1.852
2DGS+L1ResPrune	26.74	0.791	0.265	1.137
3DCS(w/ p) (Held et al., 2025b)	27.13	0.783	0.240	1.032
3DCS+L1ResPrune	27.06	0.780	0.245	1.032
Triangle Splatting(w/o P) (Held et al., 2025a)	26.81	0.782	0.238	4.000
+Official Prune	27.03	0.804	0.200	3.668
Triangle Splatting(w/o P)+L1ResPrune	27.03	0.802	0.206	2.334

and Triangle Splatting (Held et al., 2025a), which use Gaussian, 2D disk, convex hull, and triangular primitives, respectively.

All methods prune at 15K iterations and fine-tune to 30K iterations. Since importance score distributions vary across representations, we use per-scene adaptive pruning with fixed thresholds of 0.02, 0.2, and 0.2 for AAA-Gaussian, 2DGS, and Triangle Splatting, respectively, while fixed-ratio pruning is used for 3DCS to match its official pruning strategy.

Tab. 3 shows the quantitative results. L1ResPrune consistently reduces the number of primitives while maintaining comparable rendering quality across different representations. For Triangle Splatting, it achieves stronger compression than the official pruning method with similar rendering quality. For 3DCS, L1ResPrune performs comparably but slightly underperforms the official method, mainly because the latter progressively prunes and optimizes the remaining primitives, whereas L1ResPrune prunes in a single stage. Multi-stage pruning with intermediate optimization could further reduce this gap. Overall, these results demonstrate that L1ResPrune generalizes to non-Gaussian primitives without modifying the pruning criterion.

7 LIMITATION

A limitation of our method lies in the adaptive pruning strategy. Although the pruning threshold can be generalized across different scenes represented by the same type of primitive under a fixed rendering resolution, it may require re-calibration when the primitive representation or rendering resolution changes. This is because different primitive types and resolutions can lead to different importance score distributions. Developing a universal threshold selection strategy independent of primitive type and rendering resolution is an interesting direction for future work.

8 CONCLUSION

In this paper, we propose L1ResPrune, a general pruning framework for explicit scene representations based on the L1 residual caused by primitive removal. By deriving primitive importance directly from the alpha-blending formulation, L1ResPrune provides a theoretically grounded criterion beyond heuristic pruning strategies and is applicable to various explicit primitives. Extensive experiments on both static and dynamic scenes demonstrate that L1ResPrune achieves superior quality preservation under different pruning ratios compared with existing pruning methods.

AI USE STATEMENT

In this work, we used generative AI tools for improve readability and coding. We have not used generative AI tools for method development, experimental design or settings, theoretical derivations or proofs. We have reviewed all AI-assisted work, including the language as well as the inspection and testing of AI-assisted code. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate the reproducibility of our work. The source code and configuration files are included in the supplementary material.

REFERENCES

Muhammad Salman Ali, Maryam Qamar, Sung-Ho Bae, and Enzo Tartaglione. Trimming the fat: Efficient compression of 3d gaussian splats through pruning. In *British Machine Vision Conference*. BMVA, 2024.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5460–5469, June 2022.

Xiaobin Deng, Qiuli Yu, Changyu Diao, Min Li, and Duanqing Xu. Gradient-driven natural selection for compact 3d gaussian splatting. *arXiv preprint arXiv:2511.16980*, 2025.

Xiaobin Deng, Changyu Diao, Min Li, Ruohan Yu, and Duanqing Xu. Improving densification in 3d gaussian splatting for high-fidelity rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 223–232, June 2026.

Xiaobiao Du, Yida Wang, Kun Zhan, and Xin Yu. Mobile-gs: Real-time gaussian splatting for mobile devices, 2026. URL https://arxiv.org/abs/2603.11531.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: unbounded 3d gaussian compression with 15x reduction and 200+ fps. In *International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pp. 165–181, 2024.

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

Alex Hanson, Allen Tu, Geng Lin, Vasu Singla, Matthias Zwicker, and Tom Goldstein. Speedysplat: Fast 3d gaussian splatting with sparse pixels and sparse primitives. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 21537–21546, June 2025a. URL https://speedysplat.github.io/.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5949–5958, June 2025b.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics*, 37(6):257:1–257:15, 2018.

Jan Held, Renaud Vandeghen, Adrien Deliege, Abdullah Hamdi, Anthony Cioppa, Silvio Giancola, Andrea Vedaldi, Bernard Ghanem, Andrea Tagliasacchi, and Marc Van Droogenbroeck. Triangle splatting for real-time radiance field rendering. *arXiv*, 2025a.

Jan Held, Renaud Vandeghen, Abdullah Hamdi, Adrien Deliege, Anthony Cioppa, Silvio Giancola, Andrea Vedaldi, Bernard Ghanem, and Marc Van Droogenbroeck. 3d convex splatting: Radiance field rendering with 3d smooth convexes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

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

Yifei Liu, Zhihang Zhong, Yifan Zhan, Sheng Xu, and Xiao Sun. Maskgaussian: Adaptive 3d gaussian representation from probabilistic masks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 681–690, June 2025.

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*, SA '24. Association for Computing Machinery, 2024.

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. In *European Conference on Computer Vision*, 2020.

Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. In *International Conference on 3D Vision*, pp. 134–144, 2025.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. Fastgs: Training 3d gaussian splatting in 100 seconds. *arXiv preprint arXiv:2511.04283*, 2025.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, June 2016.

Seungjoo Shin, Jaesik Park, and Sunghyun Cho. Locality-aware gaussian compression for fast and high-quality rendering. In *International Conference on Learning Representations*, 2025.

Michael Steiner, Thomas Köhler, Lukas Radl, Felix Windisch, Dieter Schmalstieg, and Markus Steinberger. Aaa-gaussians: Anti-aliased and artifact-free 3d gaussian rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27650–27659, October 2025.

Allen Tu, Haiyang Ying, Alex Hanson, Yonghan Lee, Tom Goldstein, and Matthias Zwicker. Speede3dgs: Speedy deformable 3d gaussian splatting with temporal pruning and motion grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26083–26093, June 2026. URL `https://speede3dgs.github.io/`.

Haoran Wang, Guoxi Huang, Fan Zhang, David Bull, and Nantheera Anantrasirichai. Prune wisely, reconstruct sharply: Compact 3d gaussian splatting via adaptive pruning and difference-of-gaussian primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11716–11725, 2026.

Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, and Rakesh Ranjan. Steepest descent density control for compact 3d gaussian splatting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 26663–26672, June 2025a.

Tao Wang, Mengyu Li, Geduo Zeng, Cheng Meng, and Qiong Zhang. Gaussian herding across pens: An optimal transport perspective on global gaussian reduction for 3dgs. *arXiv preprint arXiv:2506.09534*, 2025b.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Lintao Xiang, Xinkai Chen, Jianhuang Lai, and Guangcong Wang. Distilled-3dgs: Distilled 3d gaussian splatting. *arXiv*, 2025.

Shuzhao Xie, Weixiang Zhang, Chen Tang, Yunpeng Bai, Rongwei Lu, Shijia Ge, and Zhi Wang. Mesongs: Post-training compression of 3d gaussians via efficient attribute transformation. In *European Conference on Computer Vision*, pp. 434–452, 2024.

Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8285–8295, 2023.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, June 2018.

Zhaoliang Zhang, Tianchen Song, Yongjae Lee, Li Yang, Cheng Peng, Rama Chellappa, and Deliang Fan. Lp-3dgs: Learning to prune 3d gaussian splatting. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 122434–122457. Curran Associates, Inc., 2024a.

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

Hongbi Zhou and Zhangkai Ni. Perceptual-gs: Scene-adaptive perceptual densification for gaussian splatting. In *International Conference on Machine Learning*, volume 267, pp. 79089–79114. PMLR, July 2025.

A Appendix

A.1 Ablation

We conduct three ablation studies. The first investigates the effect of incorporating the $\frac{1}{1-\alpha_{i,p}^{\theta}}$ term into the importance score. The second compares the performance of the Full form with its two approximations, NGS and IAS. The third examines the impact of different values of n on pruning performance.

Impact of the Alpha Term. The $\frac{1}{1-\alpha_{i,p}^{\theta}}$ term is introduced by analyzing the structure of the second- and higher-order terms in the residual expansion. To evaluate its contribution, we conduct high-ratio pruning experiments on outdoor scenes from the MipNeRF-360 dataset. We focus on outdoor scenes, high-ratio pruning and extreme-ratio pruning because the more challenging geometry and aggressive compression make differences between importance formulations more apparent, whereas indoor scenes tend to have relatively well-structured geometry and low-ratio pruning results in relatively small quality differences.

Table 4: Ablation results with and without the alpha term. w/o term denotes removing the $\frac{1}{1-\alpha_{i,p}^{\theta}}$ term. The importance score uses the Full form, and Max Aggregation ($n = +\infty$) is adopted.

Method	MipNeRF-360			
	PSNR↑	SSIM↑	LPIPS↓	Num(M)↓
L1ResPrune (high-ratio)	24.49	0.710	0.296	0.756
w/o term	24.35	0.701	0.309	0.756
L1ResPrune (extreme-ratio)	24.24	0.684	0.341	0.378
w/o term	24.15	0.675	0.352	0.378

Tab. 4 compares the pruning results with and without the $\frac{1}{1-\alpha_{i,p}^{\theta}}$ term. The results show that removing this term consistently degrades rendering quality under the same number of primitives. Although the term introduces only an additional $\frac{1}{1-\alpha_{i,p}^{\theta}}$ factor into the importance score, it leads to noticeable differences in pruning performance. This improvement is consistent with our theoretical analysis, where $\frac{1}{1-\alpha_{i,p}^{\theta}}$ characterizes the amplification of the contribution from the second- and higher-order terms in the residual expansion. Therefore, incorporating this term enables the importance score to better capture the complete effect of primitive removal, particularly under aggressive pruning.

Approximate Form. We investigate the performance differences among the Full, NGS, and IAS importance score forms. The Full form retains the complete importance score without approximation. The NGS form removes the gradient term from the Full form, while the IAS form further approximates the summation in NGS using an analytical integral. Tab. 5 compares the three forms under different pruning ratios, with visibility aggregation and fixed-ratio pruning.

At low pruning ratios, the performance gap between Full and NGS is small and can be largely ignored. This is because, when a large number of Gaussian primitives are retained, the contribution of the gradient term to the relative ranking of primitives is limited. As the pruning ratio increases, however, the gap becomes more noticeable. In our experiments, the difference between Full and NGS is larger than that caused by removing the $\frac{1}{1-\alpha_{i,p}^{\theta}}$ term. This can be attributed to the fact that the gradient term is involved throughout the first-order, whereas the $\frac{1}{1-\alpha_{i,p}^{\theta}}$ factor arises from the second- and higher-order terms. NGS eliminates the need to compute and store gradients through backpropagation, substantially reducing computational and memory overhead, particularly for scenes with many primitives or a large number of training high resolution images. Therefore, NGS provides a favorable trade-off between pruning accuracy and computational efficiency, especially for low-ratio pruning, which is the more common setting in practical applications.

IAS further approximates the summation in NGS with an analytical integral, thereby avoiding the per-pixel accumulation required by NGS. As shown in Tab. 5, the performance gap between NGS and IAS is negligible across the evaluated pruning ratios. Although the integral approximation changes the absolute magnitude of the importance scores, it preserves their relative ordering among primitives, which is the key factor for pruning. In our experiments, the IAS scores are slightly larger than the corresponding NGS scores, but their relative ranking remains highly consistent. Therefore, when the summation in NGS admits an analytical integral approximation, IAS can be adopted to further reduce computational overhead. Unlike NGS, which requires forward propagation, IAS only requires the projected primitive parameters, enabling importance estimation without either forward or backward propagation.

We also investigate the differences in computation time among the Full, NGS, and IAS importance score formulations. Tab. 6 reports the computational costs of the different importance score formulations. We select two scenes with a large number of primitives from Mip-NeRF360, namely Bicycle and Garden, and use relatively high image resolutions of (2473×1643) for Bicycle and (2594×1681) for Garden. On average, 165 views are used for importance score computation in each scene. As shown in Table A, NGS and IAS reduce the computation time by 47% and 65%, respectively, compared with Full. In large-scale scene applications, the number of training images can easily exceed 10,000, and the images are typically captured at relatively high resolutions. Since a relatively low pruning ratio (50%–60%) is generally suitable for practical applications, using NGS or IAS can substantially reduce the computational cost, making these formulations more practical for large-scale applications.

Table 5: Ablation results of different importance score forms (Full, NGS, and IAS) with $n = 2$. All methods use Visibility Aggregation and fixed-ratio pruning.

Method	MipNeRF-360			
	PSNR↑	SSIM↑	LPIPS↓	Num(M)↓
L1ResPrune (small-ratio, Full)	27.55	0.813	0.227	1.137
L1ResPrune (small-ratio, NGS)	27.52	0.812	0.229	1.137
L1ResPrune (small-ratio, IAS)	27.50	0.812	0.230	1.137
L1ResPrune (high-ratio, Full)	27.26	0.806	0.246	0.541
L1ResPrune (high-ratio, NGS)	27.06	0.793	0.263	0.541
L1ResPrune (high-ratio, IAS)	27.08	0.794	0.263	0.541

Table 6: Computation time for importance scores across different types, averaged over 165 views. Hardware: RTX 5090 D.

Score Form	Full	NGS	IAS
time(s)↓	17	9	6

Impact of Different Order. Since a single primitive can contribute to multiple pixels with different magnitudes, we introduce n to control the relative emphasis on these per-pixel importance scores. A larger n amplifies the contribution of pixels with higher importance scores. In the limiting case of $n \to +\infty$, the aggregated importance is dominated by the maximum single-pixel importance score. Tab. 7 presents the pruning results with different values of n.

When n is small, the overall performance tends to degrade. In particular, when $n \leq 1$, the importance scores of different pixels become less distinguishable; at $n = 0$, the score is determined only by the number of occupied pixels. For global aggregation, the effect of a small n is relatively limited because contributions are accumulated across multiple views, partially compensating for the reduced distinction among pixels. Moreover, the test views are sampled from the training views, which naturally gives regions observed from more views a larger accumulated importance. In contrast, visibility aggregation normalizes the contribution by the number of visible views and is therefore more sensitive to the choice of n. When n becomes excessively large, such as $n = +\infty$ corresponding to max aggregation, the performance change is also limited. We attribute this to the power formulation: $n = 2$ is already sufficient to distinguish the relative importance of Gaussian primitives, while further increasing n mainly amplifies already dominant contributions and has limited influence on their relative ranking. Based on these results, we use $n = 2$ as a suitable default choice for Gaussian primitives. For other primitive types, a different value of n may be more appropriate.

A.2 Differences Between Aggregation Strategies

Global and Visibility Aggregation. The evaluation protocol used in our experiments samples test views from the training views, which inherently favors regions observed by a larger number of views and may therefore bias the comparison toward global aggregation. For this reason, we do not directly compare the two aggregation strategies under the current evaluation protocol. Instead, we discuss their differences based on their definitions.

Global aggregation directly sums the importance scores across all visible views. Consequently, its importance magnitude depends on the number and distribution of training views, making it difficult to use a unified threshold for per-scene adaptive pruning. Moreover, in scenes with non-uniform view coverage, primitives located in sparsely observed regions may receive lower aggregated importance and thus be more likely to be pruned, potentially causing local quality degradation.

In contrast, visibility aggregation normalizes the accumulated importance by the number of views in which each primitive is visible. This reduces the dependence of the importance magnitude on view count and allows a unified threshold to be used for per-scene adaptive pruning. However, this normalization can also introduce a potential drawback: primitives that are visible in only a few views may receive unstable or overestimated importance, making their importance estimation more sensitive to limited observations.

Table 7: Pruning results with different n values and varying numbers of Gaussian primitives. The importance score is in full form and the pruning ratio is high-ratio.

Global Aggregation				
Method	PSNR↑	SSIM↑	LPIPS↓	Num(M)↓
L1ResPrune($n = 1$)	27.20	0.804	0.243	0.541
L1ResPrune($n = 2$)	27.31	0.806	0.242	0.541
L1ResPrune($n = 3$)	27.32	0.806	0.244	0.541
Visibility Aggregation				
Method	PSNR↑	SSIM↑	LPIPS↓	Num(M)↓
L1ResPrune($n = 1$)	27.10	0.791	0.263	0.541
L1ResPrune($n = 2$)	27.22	0.801	0.252	0.541
L1ResPrune($n = 3$)	27.33	0.802	0.251	0.541
Max Aggregation				
Method	PSNR↑	SSIM↑	LPIPS↓	Num(M)↓
L1ResPrune($n = +\infty$)	27.26	0.806	0.246	0.541

For real-world applications rather than benchmark evaluation, visibility aggregation is more suitable than global aggregation.

Max Aggregation. Max aggregation corresponds to the limiting case of $n \to +\infty$, where the aggregated importance is dominated by the maximum single-view importance. Unlike Global Aggregation, it is insensitive to the number of training views because only the most important view contributes substantially to the final score. At the same time, it avoids the averaging effect of visibility aggregation, which may suppress primitives whose importance is concentrated in a small number of views. Therefore, max aggregation provides an alternative aggregation strategy that emphasizes the most critical view of each primitive.

As shown in Tab. 7, Max aggregation achieves rendering quality comparable to global aggregation under our evaluation protocol, while avoiding the direct dependence of the importance magnitude on the number of views. This makes it a potentially useful choice when view coverage is highly non-uniform or when a view-count-independent importance measure is desired.

A.3 More Experimental Details

Number of Primitives. For the comparison with static 3DGS pruning methods in the main paper, we consider two pruning levels, namely low-ratio and high-ratio pruning. Since the resulting pruning ratios vary across scenes, we first obtain scene-adaptive pruning ratios using the IAS importance score with the Adaptive pruning strategy. For low-ratio pruning, we use thresholds of 0.2 for Mip-NeRF360, Tanks & Temples, and Deep Blending datasets. For high-ratio pruning, we use thresholds of 1.2, 0.5, and 0.4 for Mip-NeRF360, Tanks & Temples, and Deep Blending datasets, respectively. The resulting number of primitives is then used as the target for the corresponding comparisons with other pruning methods.

More Implementation Details of L1ResPrune. When using the IAS importance score ($n = 2$), the computation involves the determinant of the projected 2D covariance, $\sqrt{|\Sigma^{2D}|}$. Due to the Jacobian-based approximation of the 3D-to-2D projection at specific viewpoints, the estimated covariance may occasionally contain outlier values. To improve numerical stability, we clip $\sqrt{|\Sigma^{2D}|}$ to a maximum value of 10. Additionally, when using adaptive pruning, we perform pruning every 3K iterations from 15K to 27K using the same threshold. This is motivated by the possibility that redundant primitives may emerge during subsequent fine-tuning. Nevertheless, only a small number of primitives are removed in later stages, as most redundant primitives have already been eliminated at 15K iterations.

A.4 More Comparison of Static Scenes

We further compare L1ResPrune with three recent pruning methods, namely Mobile-GS Du et al. (2026), Fast-GS Ren et al. (2025), and Prune Wisely Wang et al. (2026). For Mobile-GS, we follow

Table 8: More comparison of static 3DGS pruning methods under the same per-scene primitive budget. L1ResPrune uses the full importance score with $n = 2$, fixed-ratio pruning, and global aggregation. The number of Gaussian primitives for each scene is provided in the Appendix. **Bold** and underlined denote the best and second-best results, respectively.

Method	MipNeRF-360				Tanks & Temples				Deep Blending			
	PSNR↑	SSIM↑	LPIPS↓	GS(M)↓	PSNR↑	SSIM↑	LPIPS↓	GS(M)↓	PSNR↑	SSIM↑	LPIPS↓	GS(M)↓
Backbone 3DGS (Kerbl et al., 2023)	27.55	0.813	0.221	2.646	23.67	0.852	0.170	1.572	29.76	0.906	0.238	2.478
Low-ratio pruning												
Fast-GS (Ren et al., 2025)	26.87	0.785	0.274	1.137	23.21	0.824	0.230	0.481	29.24	0.904	0.258	0.720
Mobile-GS (Du et al., 2026)	26.73	0.781	0.279	1.137	23.20	0.821	0.234	0.481	29.00	0.901	0.264	0.720
Prune Wisely (Wang et al., 2026)	**27.55**	**0.813**	0.226	1.137	**23.81**	**0.851**	0.183	0.481	29.85	**0.910**	0.243	0.720
L1ResPrune	**27.55**	**0.813**	**0.224**	1.137	23.62	0.850	**0.181**	0.481	**29.86**	0.909	**0.242**	0.720
High-ratio pruning												
Fast-GS (Ren et al., 2025)	25.58	0.728	0.345	0.541	23.20	0.812	0.251	0.310	29.07	0.902	0.266	0.500
Mobile-GS (Du et al., 2026)	25.75	0.736	0.325	0.541	22.85	0.805	0.256	0.310	28.34	0.895	0.277	0.500
Prune Wisely (Wang et al., 2026)	**27.38**	0.804	0.249	0.541	**23.68**	0.844	0.197	0.310	29.58	0.907	0.252	0.500
L1ResPrune	27.31	**0.806**	**0.242**	0.541	23.45	**0.845**	**0.191**	0.310	**29.83**	**0.909**	**0.245**	0.500
Extreme-ratio pruning												
Fast-GS (Ren et al., 2025)	23.95	0.659	0.417	0.265	22.29	0.773	0.302	0.158	28.14	0.891	0.290	0.248
Mobile-GS (Du et al., 2026)	24.83	0.703	0.372	0.265	21.90	0.766	0.301	0.158	27.04	0.882	0.308	0.248
Prune Wisely (Wang et al., 2026)	**26.71**	0.775	0.298	0.265	**23.38**	0.815	0.245	0.158	**29.66**	**0.905**	0.263	0.248
L1ResPrune	26.62	**0.785**	**0.281**	0.265	23.31	**0.822**	**0.234**	0.158	29.51	**0.905**	**0.258**	0.248

its training procedure by running an additional 5K iterations from the 15K-th iteration to compute its importance scores, and perform pruning at the 20K-th iteration. Fast-GS and Prune Wisely perform pruning at the 15K-th iteration according to their respective importance scores. For L1ResPrune, we use the same settings as in the main experiments, including the Full importance score with $n = 2$, Global Aggregation, and fixed-ratio pruning.

Tab. 8 presents the quantitative results under different pruning ratios. L1ResPrune consistently achieves competitive or superior performance across the three datasets and pruning levels, with particularly strong advantages in SSIM and LPIPS under high and extreme pruning ratios.

Fast-GS identifies Gaussians based on their reconstruction errors. While this criterion can be effective for mild pruning, its performance degrades substantially as the pruning ratio increases, indicating that reconstruction error alone is insufficient to accurately characterize primitive importance. Mobile-GS considers Gaussian opacity and scale as pruning criteria. Although these properties are intuitive indicators of redundancy, they do not explicitly characterize the influence of primitive removal on the rendered image, resulting in more severe quality degradation under aggressive pruning.

Prune Wisely extends Speedy-Splat by incorporating frequency-domain gradients into its importance estimation. In our experiments, however, its performance is generally close to that of Speedy-Splat, suggesting that the additional frequency information provides limited improvement under the evaluated settings. At low and moderate pruning ratios, Prune Wisely achieves performance comparable to L1ResPrune. At the extreme pruning ratio of 90%, Prune Wisely achieves slightly higher PSNR on all three datasets. However, L1ResPrune consistently achieves better SSIM and LPIPS, indicating better preservation of structural and perceptual quality. Since PSNR alone does not fully characterize structural and perceptual fidelity, these results demonstrate the robustness of L1ResPrune under aggressive pruning.

Overall, these comparisons further demonstrate that directly deriving primitive importance from the rendering residual provides a more reliable criterion than manually designed geometric, opacity-based, or error-based heuristics. L1ResPrune maintains strong quality preservation across a wide range of pruning ratios without introducing representation-specific modifications.

A.5 More Visual Comparisons

We present additional visual comparisons with recent competitive pruning methods. As shown in Fig. 3 and Fig. 4, we compare L1ResPrune with SpeedySplat and PruneWisely under an extreme pruning ratio of 90% in static scene and compare L1ResPrune with Mini-Splatting and SpeeDe3DGS in dynamic scene. L1ResPrune uses the same parameter settings as those in the main comparative experiments.

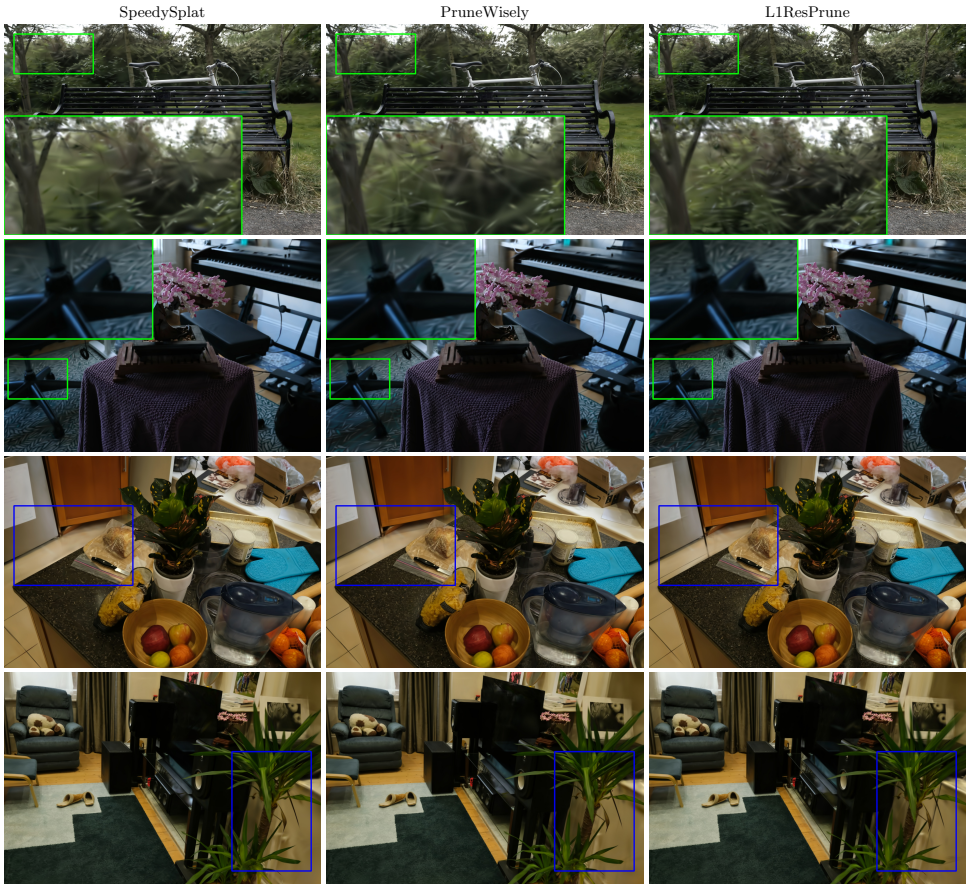


Figure 3: Additional visual comparisons on static scenes under a 90% pruning ratio.

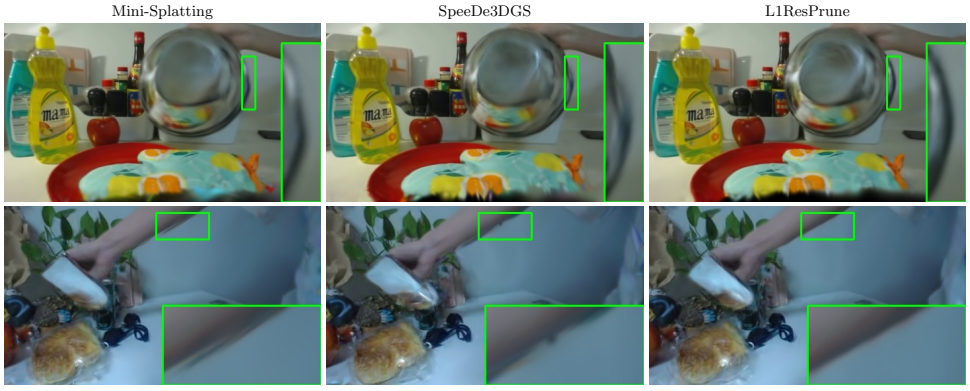


Figure 4: Additional visual comparisons on dynamic scenes under a 80% pruning ratio.

Table 9: Applications to different improved 3DGS methods. L1ResPrune($n = 2$) uses the IAS importance score, Eq. 18, with visibility aggregation and per-scene adaptive pruning (Adaptive) using a fixed threshold of 0.2.

Method	MipNeRF-360			
	PSNR↑	SSIM↑	LPIPS↓	Num(M)↓
Pixel-GSZhang et al. (2024b)	27.65	0.824	0.193	5.379
+L1ResPrune	27.64	0.824	0.198	2.073
Perceptual-GS (Zhou & Ni, 2025)	27.81	0.828	0.186	2.957
+L1ResPrune	27.79	0.829	0.194	1.432
Improved-GS (Deng et al., 2026)	28.23	0.834	0.196	1.778
+L1ResPrune	28.19	0.833	0.195	1.225
Taming3DGS (Mallick et al., 2024)	27.40	0.794	0.262	0.670
+L1ResPrune	27.32	0.792	0.267	0.420

A.6 Applications to Improved 3DGS Methods

We further evaluate L1ResPrune on several improved 3DGS methods, including Pixel-GS Zhang et al. (2024b), Perceptual-GS Zhou & Ni (2025), Improved-GS Deng et al. (2026), and Taming3DGS Mallick et al. (2024). Since no comparison of the number of Gaussians is required in this experiment, we adopt a per-scene adaptive pruning strategy, where Gaussians with global importance scores below a fixed threshold are removed. We use the IAS form of the importance score, i.e., Eq. 18, to further demonstrate the effectiveness of the simplified formulation. Pruning starts at the 15K-th iteration and is performed every 3K iterations until the 27K-th iteration, resulting in five pruning steps. Most redundant Gaussians are removed during the first pruning step, while the subsequent four steps mainly eliminate a small number of redundant Gaussians introduced during fine-tuning.

Tab. 9 summarizes the results across different improved 3DGS baselines. All methods use the same pruning threshold of 0.2. L1ResPrune achieves substantial reductions in the number of Gaussians while maintaining comparable rendering quality across all four baselines, demonstrating its compatibility with different 3DGS improvements. These results also show that the simplified IAS formulation remains effective for practical pruning. Moreover, the per-scene adaptive strategy requires only a single fixed threshold across scenes, avoiding the need for scene-specific threshold tuning and reducing the associated parameter-selection overhead.

A.7 Theoretical Proof

Proposition 1.

$$\frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_1,p} \partial \alpha^{\theta}_{i_2,p}} = -\frac{1}{1-\alpha^{\theta}_{i_1,p}} \frac{\partial L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i_2,p}}, depth(i_1,\theta) < depth(i_2,\theta). \tag{19}$$

$depth(i,\theta)$ denotes the depth of the primitive i under the camera view θ.

Proof. It is evident that:

$$\frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_1,p} \partial \alpha^{\theta}_{i_2,p}} = \frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_2,p} \partial \alpha^{\theta}_{i_1,p}}. \tag{20}$$

We compute $\frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_2,p} \partial \alpha^{\theta}_{i_1,p}}$. First, we write the rendered color $C^{\theta}_{c,p}$ in the following form:

$$C^{\theta}_{c,p} = \sum_{i \in U_{i_2,\theta}} c^{\theta}_{c,i} T^{\theta}_{i,p} \alpha^{\theta}_{i,p} + c^{\theta}_{c,i_2} T^{\theta}_{i_2,p} \alpha^{\theta}_{i_2,p} + \sum_{i \in V_{i_2,\theta}} c^{\theta}_{c,i} T^{\theta}_{i,p} \alpha^{\theta}_{i,p}. \tag{21}$$

$U_{i,\theta}$ denotes the set of primitives with depth lower than primitive i under the camera view θ, and $V_{i,\theta}$ denotes the set of primitives with depth higher than primitive i under the camera view θ. Then, we first compute the first-order gradient $\frac{\partial C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_2,p}}$. Noting that $\sum_{i \in U_{i_2,\theta}} c^{\theta}_{c,i} T^{\theta}_{i,p} \alpha^{\theta}_{i,p}$ does not contain

primitive i_2, we have:

$$\frac{\partial C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_2,p}} = c^{\theta}_{c,i_2} T^{\theta}_{i_2,p} - \sum_{i \in V_{i_2,\theta}} \frac{c^{\theta}_{c,i} T^{\theta}_{i,p}}{1-\alpha^{\theta}_{i_2,p}}. \tag{22}$$

Then, computing the second-order gradient, we have:

$$\begin{aligned}
\frac{\partial^2 C^{\theta}_{c,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_2,p} \partial \alpha^{\theta}_{i_1,p}} &= -\frac{c^{\theta}_{c,i_2} T^{\theta}_{i_2,p}}{1-\alpha^{\theta}_{i_1,p}} + \sum_{i \in V_{i_2,\theta}} \frac{c^{\theta}_{c,i} T^{\theta}_{i,p}}{(1-\alpha^{\theta}_{i_2,p})(1-\alpha^{\theta}_{i_1,p})} \\
&= -\frac{1}{1-\alpha^{\theta}_{i_1,p}} (c^{\theta}_{c,i_2} T^{\theta}_{i_2,p} - \sum_{i \in V_{i_2,\theta}} \frac{c^{\theta}_{c,i} T^{\theta}_{i,p}}{1-\alpha^{\theta}_{i_2,p}}) \\
&= -\frac{1}{1-\alpha^{\theta}_{i_1,p}} \frac{\partial C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_2,p}}.
\end{aligned} \tag{23}$$

We have derived the relationship for the second-order gradient of color $C^{\theta}_{c,p}$. Below, we compute the second-order gradient relationship of the $L^{\theta}_{1,p}(\alpha^{\theta})$ loss. According to the chain rule of differentiation, we have:

$$\frac{\partial L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_2,p}} = \sum_{c \in \{r,g,b\}} \frac{\partial L^{\theta}_{1,p}(\alpha^{\theta})}{\partial C^{\theta}_{c,p}} \frac{\partial C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_2,p}}. \tag{24}$$

Taking the derivative once more, we have:

$$\frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_2,p} \partial \alpha^{\theta}_{i_1,p}} = \sum_{c \in \{r,g,b\}} \frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{\partial (C^{\theta}_{c,p})^2} \frac{\partial C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_2,p}} \frac{\partial C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_i,p}} + \sum_{c \in \{r,g,b\}} \frac{\partial L^{\theta}_{1,p}(\alpha^{\theta})}{\partial C^{\theta}_{c,p}} \frac{\partial^2 C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_1,p} \partial \alpha^{\theta}_{i_2,p}}. \tag{25}$$

It is evident that $\frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{(\partial C^{\theta}_{c,p})^2} = 0$, thus:

$$\frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_2,p} \partial \alpha^{\theta}_{i_1,p}} = \sum_{c \in \{r,g,b\}} \frac{\partial L^{\theta}_{1,p}(\alpha^{\theta})}{\partial C^{\theta}_{c,p}} \frac{\partial^2 C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_1,p} \partial \alpha^{\theta}_{i_2,p}}. \tag{26}$$

Substituting Eq. 23 into Eq. 26, we finally obtain:

$$\begin{aligned}
\frac{\partial^2 L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_2,p} \partial \alpha^{\theta}_{i_1,p}} &= -\sum_{c \in \{r,g,b\}} \frac{\partial L^{\theta}_{1,p}(\alpha^{\theta})}{\partial C^{\theta}_{c,p}} \frac{1}{1-\alpha^{\theta}_{i_1,p}} \frac{\partial C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_2,p}} \\
&= -\frac{1}{1-\alpha^{\theta}_{i_1,p}} \sum_{c \in \{r,g,b\}} \frac{\partial L^{\theta}_{1,p}(\alpha^{\theta})}{\partial C^{\theta}_{c,p}} \frac{\partial C^{\theta}_{c,p}}{\partial \alpha^{\theta}_{i_2,p}} \\
&= -\frac{1}{1-\alpha^{\theta}_{i_1,p}} \frac{\partial L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i_2,p}}.
\end{aligned} \tag{27}$$

□

Proposition 2.

$$\frac{\partial^n L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_1,p} ... \partial \alpha^{\theta}_{i_n,p}} = \frac{\partial L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i_1,p}} \prod_{i=i_2}^{i_n} (-\frac{1}{1-\alpha^{\theta}_{i,p}}). \tag{28}$$

where i_1 is the primitive with the greatest depth under the camera view θ.

Proof. We use mathematical induction and the Eq. 27. We can easily obtain:

$$\frac{\partial^{(n+1)} L^{\theta}_{1,p}(\alpha^{\theta})}{\partial \alpha^{\theta}_{i_1,p} ... \partial \alpha^{\theta}_{i_{n+1},p}} = \frac{\partial^2 L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i_1,p} \partial \alpha^{\theta}_{i_{n+1},p}} \prod_{i=i_2}^{i_n} (-\frac{1}{1-\alpha^{\theta}_{i,p}}) = \frac{\partial L^{\theta}_{1,p}}{\partial \alpha^{\theta}_{i_1,p}} \prod_{i=i_2}^{i_{n+1}} (-\frac{1}{1-\alpha^{\theta}_{i,p}}). \tag{29}$$

□

Proposition 3.

$$\left|\frac{\partial C}{\partial \alpha_j}\right| \leq 1 \tag{30}$$

Proof. According to $C = \sum_i T_i c_i \alpha_i$ and $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$, we obtain:

$$\frac{\partial C}{\partial \alpha_j} = T_j c_j - \sum_{k \geq j+1} \frac{T_k c_k \alpha_k}{1-\alpha_j} = T_j c_j - T_j \sum_{k \geq j+1} \frac{T_k c_k \alpha_k}{(1-\alpha_j) T_j} \tag{31}$$

The mathematical meaning of $c = \sum_{k \geq j+1} \frac{T_k c_k \alpha_k}{(1-\alpha_j) T_j}$ is alpha blending starting from the $(j+1)_{th}$ primitive. Therefore, we have:

$$\left|\frac{\partial C}{\partial \alpha_j}\right| = |T_j c_j - T_j c| \leq T_j \left|c_j - c\right| \leq 1 \tag{32}$$

The upper bound of 1 is attained when $T_j c_j = 1$.

□

A.8 Integration Solution

Problem

$$\iint (\frac{o_i G_{\theta,i}^{2D}(x)}{1 - o_i G_{\theta,i}^{2D}(x)})^2 dx.$$

Solution

$$\iint (\frac{o_i G_{\theta,i}^{2D}(x)}{1 - o_i G_{\theta,i}^{2D}(x)})^2 dx = \iint (\frac{o_i e^{-\frac{1}{2}(x - x_\theta^{2D})^T (\Sigma_\theta^{2D})^{-1} (x - x_\theta^{2D})}}{1 - o_i e^{-\frac{1}{2}(x - x_\theta^{2D})^T (\Sigma_\theta^{2D})^{-1} (x - x_\theta^{2D})}})^2 dx. \tag{33}$$

Since $(\Sigma_\theta^{2D})^{-1}$ is a positive definite matrix, there exists an orthogonal matrix U such that $U^T (\Sigma_\theta^{2D})^{-1} U = \begin{pmatrix} \frac{1}{\lambda_1} & 0 \\ 0 & \frac{1}{\lambda_2} \end{pmatrix}$, where λ_1 and λ_2 are the eigenvalues of Σ_θ^{2D}. Perform a linear transformation $x = Uy + x_\theta^{2D}$, Eq. 33 can be written as:

$$\begin{aligned}
\iint (\frac{o_i G_{\theta,i}^{2D}(x)}{1 - o_i G_{\theta,i}^{2D}(x)})^n dx &= \iint (\frac{o_i e^{-\frac{1}{2}(\frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2})}}{1 - o_i e^{-\frac{1}{2}(\frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2})}})^n dy_1 dy_2 \\
&= \sqrt{\lambda_1 \lambda_2} \iint (\frac{o_i e^{-\frac{1}{2}(u^2+v^2)}}{1 - o_i e^{-\frac{1}{2}(u^2+v^2)}})^n du dv \\
&= \sqrt{\lambda_1 \lambda_2} \int_0^{2\pi} \int_0^{+\infty} r (\frac{o_i e^{-\frac{1}{2}(r^2)}}{1 - o_i e^{-\frac{1}{2}(r^2)}})^n dr d\theta \\
&= 2\pi \sqrt{\lambda_1 \lambda_2} \int_0^{+\infty} r (\frac{o_i e^{-\frac{1}{2}(r^2)}}{1 - o_i e^{-\frac{1}{2}(r^2)}})^n dr \\
&= 2\pi \sqrt{\lambda_1 \lambda_2} \int_{1-o_i}^{1} \frac{(1-u)^{n-1}}{u^n} du.
\end{aligned} \tag{34}$$

when $n = 1$:

$$\iint (\frac{o_i G_{\theta,i}^{2D}(x)}{1 - o_i G_{\theta,i}^{2D}(x)})^2 dx = 2\pi \sqrt{\lambda_1 \lambda_2} ln(\frac{1}{1 - o_i}). \tag{35}$$

when $n = 2$:

$$\iint (\frac{o_i G_{\theta,i}^{2D}(x)}{1 - o_i G_{\theta,i}^{2D}(x)})^n dx = 2\pi \sqrt{\lambda_1 \lambda_2} [ln(1 - o_i) + \frac{1}{1 - o_i} - 1]. \tag{36}$$

Table 10: PSNR per-scene results under different pruning ratios.

Methods	bicycle	bonsai	counter	garden	kitchen	room	stump	flowers	treehill	train	truck	drjohnson	playroom
3DGS (Kerbl et al., 2023)	25.18	32.25	29.20	27.40	31.43	31.78	26.69	21.41	22.64	21.97	25.37	29.45	30.08
Low-ratio pruning													
LightGS (Fan et al., 2025)	24.99	31.39	28.63	26.79	30.97	30.73	26.70	21.28	22.63	21.46	24.59	26.48	28.32
Compact3DGS (Lee et al., 2024)	24.77	31.98	28.69	26.81	30.48	30.88	26.24	21.01	22.56	21.62	25.18	29.26	30.56
MaskGS (Liu et al., 2025)	25.11	32.06	28.94	27.25	31.16	31.52	26.67	21.37	22.65	21.94	25.28	29.44	30.11
Mini-Splatting (Fang & Wang, 2024)	25.05	31.80	28.71	27.16	31.28	31.42	26.71	21.36	22.52	21.91	25.39	29.34	30.07
PUP3DGS (Hanson et al., 2025b)	25.14	32.16	28.99	27.07	31.11	31.74	26.67	21.30	22.63	21.33	24.26	29.42	30.37
GHAP (Wang et al., 2025b)	25.09	32.01	28.94	27.13	31.06	31.56	26.56	21.16	22.71	22.16	25.31	29.45	30.21
Speedy-Splat (Hanson et al., 2025a)	25.11	32.26	29.05	27.27	31.69	31.59	26.74	21.40	22.58	21.97	25.42	29.11	30.07
Fast-GS (Ren et al., 2025)	24.77	31.18	28.31	26.69	30.42	30.98	26.12	20.99	22.40	21.51	24.91	28.72	29.75
Mobile-GS (Du et al., 2026)	24.67	31.22	28.15	26.51	30.17	30.69	25.90	20.95	22.35	21.64	24.79	28.53	29.47
Prune Wisely (Wang et al., 2026)	25.18	32.44	29.09	27.30	31.59	31.85	26.70	21.43	22.58	22.21	25.40	29.46	30.23
L1ResPrune	25.16	32.37	29.10	27.35	31.54	31.77	26.68	21.40	22.58	21.86	25.38	29.55	30.17
High-ratio pruning													
Mini-Splatting (Fang & Wang, 2024)	24.81	31.41	28.52	26.64	30.95	31.21	26.62	21.23	22.64	21.72	25.02	29.20	30.15
PUP3DGS (Hanson et al., 2025b)	24.86	30.62	28.33	26.06	29.34	31.07	26.48	21.02	22.47	20.96	23.56	29.32	30.30
GHAP (Wang et al., 2025b)	24.89	31.46	28.56	26.62	30.70	31.23	26.33	20.98	22.69	22.18	25.20	29.45	30.06
p Speedy-Splat (Hanson et al., 2025a)	24.94	32.01	28.88	26.80	31.20	31.52	26.72	21.37	22.59	21.96	25.25	29.17	29.99
Fast-GS (Ren et al., 2025)	24.03	29.03	27.44	25.05	28.43	30.14	23.80	20.29	22.05	21.77	24.62	28.62	29.51
Mobile-GS (Du et al., 2026)	24.22	29.44	27.23	24.91	28.34	30.06	24.98	20.22	22.35	21.32	24.37	27.42	29.25
Prune Wisely (Wang et al., 2026)	24.99	32.07	28.93	26.76	31.17	31.60	26.75	21.38	22.75	22.06	25.29	29.35	30.15
L1ResPrune	25.10	31.98	28.88	26.67	31.19	31.44	26.75	21.32	22.50	21.72	25.23	29.41	30.25
Extreme-ratio pruning													
Mini-Splatting (Fang & Wang, 2024)	24.59	29.85	27.80	26.14	30.01	30.77	26.41	20.82	22.62	21.09	24.76	29.19	30.18
PUP3DGS (Hanson et al., 2025b)	24.50	26.23	26.12	25.40	26.87	29.98	26.11	20.31	21.71	19.92	22.60	28.91	29.66
GHAP (Wang et al., 2025b)	24.63	30.27	27.92	26.22	29.57	30.71	25.91	20.51	22.55	21.80	24.83	29.41	29.75
Speedy-Splat (Hanson et al., 2025a)	24.73	30.21	28.10	26.16	30.18	30.99	26.67	21.15	22.50	21.65	25.00	29.09	29.98
Fast-GS (Ren et al., 2025)	23.50	25.71	25.73	23.80	25.13	29.27	21.86	19.05	21.47	20.91	23.66	27.66	28.61
Mobile-GS (Du et al., 2026)	23.95	27.73	25.92	24.38	26.61	29.14	24.45	19.26	22.05	20.29	23.51	26.90	27.17
Prune Wisely (Wang et al., 2026)	24.68	30.30	28.11	26.16	29.97	31.07	26.68	21.06	22.38	21.76	24.99	29.29	30.03
L1ResPrune	24.83	29.64	27.93	26.05	29.89	30.93	26.62	21.11	22.60	21.59	25.02	29.23	29.78

when $n = 3$:

$$\iint (\frac{o_i G_{\theta,i}^{2D}(x)}{1 - o_i G_{\theta,i}^{2D}(x)})^n dx = 2\pi\sqrt{\lambda_1\lambda_2}[\frac{3}{2} - ln(1 - o_i) + \frac{-3 + 4o_i}{2(1 - o_i)^2}]. \tag{37}$$

A.9 Per-scene Results

Tab. 10, Tab. 11, Tab. 12, and Tab. 13 present the per-scene results of static scenes for PSNR, SSIM, LPIPS, and the number of primitives under low-ratio pruning, high-ratio pruning and extreme-ratio pruning respectively. Tab. 14, Tab. 15, Tab. 16, and Tab. 17 present the per-scene results of dynamic scenes for PSNR, SSIM, LPIPS, and the number of primitives under low-ratio pruning, high-ratio pruning and extreme-ratio pruning respectively.

Table 11: SSIM per-scene results under different pruning ratios.

Methods	bicycle	bonsai	counter	garden	kitchen	room	stump	flowers	treehill	train	truck	drjohnson	playroom
3DGS (Kerbl et al., 2023)	0.748	0.947	0.916	0.857	0.932	0.928	0.768	0.589	0.636	0.820	0.884	0.905	0.908
Low-ratio pruning													
LightGS (Fan et al., 2025)	0.734	0.939	0.909	0.833	0.927	0.915	0.769	0.569	0.631	0.783	0.860	0.868	0.890
Compact3DGS (Lee et al., 2024)	0.723	0.938	0.902	0.832	0.919	0.919	0.756	0.559	0.636	0.792	0.876	0.900	0.910
MaskGS (Liu et al., 2025)	0.747	0.944	0.912	0.854	0.930	0.924	0.768	0.586	0.637	0.811	0.880	0.906	0.910
Mini-Splatting (Fang & Wang, 2024)	0.743	0.943	0.909	0.851	0.930	0.924	0.770	0.584	0.636	0.817	0.881	0.904	0.910
PUP3DGS (Hanson et al., 2025b)	0.745	0.945	0.915	0.848	0.931	0.926	0.769	0.579	0.631	0.795	0.858	0.908	0.912
GHAP (Wang et al., 2025b)	0.739	0.942	0.907	0.843	0.926	0.920	0.760	0.573	0.633	0.811	0.875	0.906	0.910
Speedy-Splat (Hanson et al., 2025a)	0.747	0.946	0.914	0.854	0.932	0.926	0.773	0.590	0.638	0.812	0.880	0.903	0.908
Fast-GS (Ren et al., 2025)	0.713	0.928	0.888	0.822	0.913	0.906	0.737	0.563	0.596	0.793	0.855	0.902	0.905
Mobile-GS (Du et al., 2026)	0.706	0.927	0.884	0.812	0.908	0.903	0.727	0.557	0.601	0.788	0.853	0.898	0.903
Prune Wisely (Wang et al., 2026)	0.749	0.947	0.915	0.854	0.933	0.928	0.771	0.590	0.639	0.820	0.881	0.908	0.911
L1ResPrune	0.748	0.946	0.915	0.855	0.932	0.927	0.770	0.588	0.638	0.818	0.882	0.908	0.910
High-ratio pruning													
Mini-Splatting (Fang & Wang, 2024)	0.728	0.944	0.902	0.824	0.925	0.921	0.767	0.574	0.638	0.796	0.873	0.902	0.908
PUP3DGS (Hanson et al., 2025b)	0.727	0.933	0.905	0.806	0.920	0.919	0.761	0.558	0.633	0.782	0.839	0.906	0.912
GHAP (Wang et al., 2025b)	0.718	0.933	0.896	0.811	0.916	0.911	0.743	0.554	0.623	0.804	0.870	0.905	0.909
Speedy-Splat (Hanson et al., 2025a)	0.723	0.942	0.909	0.824	0.927	0.921	0.774	0.582	0.633	0.806	0.874	0.904	0.910
Fast-GS (Ren et al., 2025)	0.653	0.879	0.858	0.722	0.873	0.891	0.612	0.511	0.554	0.782	0.841	0.899	0.905
Mobile-GS (Du et al., 2026)	0.662	0.891	0.856	0.713	0.868	0.889	0.667	0.504	0.576	0.771	0.839	0.889	0.900
Prune Wisely (Wang et al., 2026)	0.727	0.941	0.910	0.821	0.927	0.923	0.774	0.582	0.635	0.812	0.877	0.907	0.911
L1ResPrune	0.738	0.942	0.910	0.824	0.927	0.923	0.773	0.580	0.634	0.812	0.875	0.907	0.911
Extreme-ratio pruning													
Mini-Splatting (Fang & Wang, 2024)	0.705	0.911	0.878	0.798	0.905	0.908	0.752	0.542	0.615	0.761	0.862	0.896	0.906
PUP3DGS (Hanson et al., 2025b)	0.702	0.877	0.865	0.769	0.895	0.906	0.747	0.512	0.614	0.740	0.809	0.900	0.904
GHAP (Wang et al., 2025b)	0.697	0.911	0.874	0.787	0.895	0.902	0.716	0.517	0.594	0.777	0.859	0.902	0.906
Speedy-Splat (Hanson et al., 2025a)	0.683	0.915	0.880	0.791	0.906	0.909	0.763	0.551	0.586	0.767	0.859	0.899	0.906
Fast-GS (Ren et al., 2025)	0.609	0.796	0.811	0.635	0.771	0.877	0.512	0.424	0.495	0.738	0.807	0.889	0.892
Mobile-GS (Du et al., 2026)	0.640	0.848	0.824	0.672	0.821	0.874	0.640	0.451	0.554	0.718	0.813	0.882	0.882
Prune Wisely (Wang et al., 2026)	0.688	0.910	0.883	0.787	0.903	0.911	0.762	0.545	0.588	0.769	0.860	0.904	0.906
L1ResPrune	0.714	0.910	0.889	0.793	0.910	0.913	0.764	0.555	0.613	0.776	0.867	0.904	0.905

Table 12: LPIPS per-scene results under different pruning ratios.

Methods	bicycle	bonsai	counter	garden	kitchen	room	stump	flowers	treehill	train	truck	drjohnson	playroom
3DGS (Kerbl et al., 2023)	0.241	0.180	0.189	0.122	0.116	0.195	0.243	0.359	0.345	0.197	0.143	0.236	0.241
Low-ratio pruning													
LightGS (Fan et al., 2025)	0.273	0.199	0.203	0.156	0.131	0.222	0.257	0.389	0.370	0.266	0.187	0.306	0.283
Compact3DGS (Lee et al., 2024)	0.286	0.193	0.205	0.161	0.131	0.209	0.278	0.397	0.365	0.240	0.164	0.258	0.251
MaskGS (Liu et al., 2025)	0.249	0.186	0.192	0.127	0.121	0.207	0.245	0.365	0.351	0.217	0.152	0.245	0.250
Mini-Splatting (Fang & Wang, 2024)	0.258	0.187	0.193	0.133	0.121	0.202	0.250	0.368	0.355	0.209	0.155	0.249	0.246
PUP3DGS (Hanson et al., 2025b)	0.252	0.184	0.186	0.135	0.121	0.199	0.245	0.373	0.370	0.238	0.178	0.240	0.243
GHAP (Wang et al., 2025b)	0.270	0.192	0.202	0.154	0.128	0.218	0.271	0.382	0.367	0.225	0.173	0.248	0.249
Speedy-Splat (Hanson et al., 2025a)	0.257	0.182	0.186	0.131	0.118	0.200	0.245	0.363	0.357	0.218	0.158	0.249	0.248
Fast-GS (Ren et al., 2025)	0.305	0.214	0.236	0.192	0.152	0.249	0.306	0.393	0.416	0.249	0.210	0.258	0.258
Mobile-GS (Du et al., 2026)	0.310	0.215	0.243	0.206	0.161	0.255	0.313	0.397	0.409	0.256	0.211	0.264	0.264
Prune Wisely (Wang et al., 2026)	0.254	0.182	0.185	0.131	0.117	0.198	0.245	0.363	0.356	0.210	0.155	0.241	0.245
L1ResPrune	0.250	0.181	0.185	0.129	0.118	0.197	0.244	0.363	0.350	0.209	0.152	0.240	0.244
High-ratio pruning													
Mini-Splatting (Fang & Wang, 2024)	0.289	0.199	0.207	0.180	0.132	0.213	0.262	0.385	0.362	0.246	0.168	0.256	0.249
PUP3DGS (Hanson et al., 2025b)	0.281	0.208	0.205	0.195	0.140	0.214	0.261	0.398	0.370	0.255	0.200	0.246	0.246
GHAP (Wang et al., 2025b)	0.310	0.212	0.225	0.219	0.150	0.241	0.308	0.409	0.395	0.239	0.185	0.254	0.253
Speedy-Splat (Hanson et al., 2025a)	0.304	0.194	0.198	0.186	0.130	0.213	0.259	0.382	0.386	0.231	0.171	0.253	0.251
Fast-GS (Ren et al., 2025)	0.375	0.278	0.284	0.321	0.219	0.279	0.433	0.448	0.469	0.268	0.233	0.266	0.265
Mobile-GS (Du et al., 2026)	0.359	0.267	0.286	0.328	0.226	0.280	0.279	0.449	0.448	0.279	0.232	0.284	0.269
Prune Wisely (Wang et al., 2026)	0.298	0.196	0.198	0.185	0.131	0.210	0.257	0.382	0.383	0.225	0.168	0.246	0.249
L1ResPrune	0.277	0.192	0.194	0.178	0.128	0.208	0.254	0.380	0.368	0.221	0.161	0.244	0.246
Extreme-ratio pruning													
Mini-Splatting (Fang & Wang, 2024)	0.320	0.242	0.249	0.221	0.165	0.240	0.296	0.425	0.416	0.296	0.189	0.269	0.265
PUP3DGS (Hanson et al., 2025b)	0.316	0.290	0.264	0.240	0.181	0.239	0.290	0.451	0.416	0.316	0.233	0.263	0.269
GHAP (Wang et al., 2025b)	0.341	0.251	0.266	0.259	0.190	0.261	0.353	0.455	0.450	0.283	0.207	0.264	0.267
Speedy-Splat (Hanson et al., 2025a)	0.356	0.246	0.250	0.235	0.172	0.242	0.286	0.424	0.468	0.294	0.200	0.269	0.270
Fast-GS (Ren et al., 2025)	0.415	0.361	0.348	0.394	0.342	0.303	0.528	0.529	0.531	0.326	0.278	0.288	0.291
Mobile-GS (Du et al., 2026)	0.384	0.320	0.333	0.366	0.292	0.304	0.412	0.451	0.482	0.337	0.265	0.308	0.308
Prune Wisely (Wang et al., 2026)	0.349	0.254	0.244	0.232	0.178	0.235	0.285	0.427	0.462	0.293	0.197	0.259	0.267
L1ResPrune	0.313	0.251	0.234	0.221	0.162	0.228	0.279	0.419	0.424	0.286	0.181	0.254	0.262

Table 13: Per-scene Gaussian number ratios under different pruning ratios. Other Methods: Mini-Splatting (Fang & Wang, 2024), PUP3DGS (Hanson et al., 2025b), GHAP (Wang et al., 2025b), Speedy-Splat (Hanson et al., 2025a), Fast-GS (Ren et al., 2025), Mobile-GS (Du et al., 2026), Prune Wisely (Wang et al., 2026)and L1ResPrune

Methods	bicycle	bonsai	counter	garden	kitchen	room	stump	flowers	treehill	train	truck	drjohnson	playroom
3DGS (Kerbl et al., 2023)	4.770	1.070	1.054	4.209	1.548	1.273	4.019	2.819	3.054	1.093	2.050	3.114	1.842
Low-ratio pruning													
LightGS (Fan et al., 2025)	1.800	0.540	0.480	1.620	0.690	0.480	1.730	1.320	1.570	0.450	0.511	0.780	0.660
Compact3DGS (Lee et al., 2024)	2.221	0.575	0.523	2.098	1.131	0.529	1.681	1.528	1.896	0.710	0.972	1.343	0.745
MaskGS (Liu et al., 2025)	2.221	0.350	0.295	2.160	0.532	0.242	1.704	1.429	1.522	0.451	0.648	0.821	0.323
Other Methods	1.800	0.540	0.480	1.620	0.690	0.480	1.730	1.320	1.570	0.450	0.511	0.780	0.660
High-ratio pruning													
Other Methods	0.820	0.270	0.250	0.650	0.340	0.230	0.780	0.680	0.850	0.270	0.350	0.530	0.470
Extreme-ratio pruning													
Other Methods	0.476	0.107	0.105	0.421	0.157	0.127	0.403	0.287	0.303	0.109	0.207	0.311	0.184

Table 14: Dynamic PSNR per-scene results.

Methods	bouncingballs	hellwarrior	hook	jumpingjacks	lego	mutant	standup	trex	as	basin	bell	cup	plate	press	sieve
Deformable3DGS (Yang et al., 2023)	40.98	41.40	37.17	37.48	32.57	42.53	40.98	37.94	26.12	19.60	22.65	24.60	18.61	25.51	25.20
Mini-Splatting (Fang & Wang, 2024)	40.63	40.80	36.59	37.12	32.46	41.11	42.82	37.53	26.26	19.57	22.54	24.56	17.37	25.53	25.09
SpeeDe3DGS (Tu et al., 2026)	40.71	40.79	36.71	37.09	32.14	41.60	43.14	37.64	25.85	19.50	22.45	24.61	20.33	25.29	25.35
L1ResPrune	40.99	40.86	36.68	37.39	32.08	41.64	43.49	37.54	26.28	19.48	25.48	24.69	20.17	25.38	25.08

Table 15: Dynamic SSIM per-scene results.

Methods	bouncingballs	hellwarrior	hook	jumpingjacks	lego	mutant	standup	trex	as	basin	bell	cup	plate	press	sieve
Deformable3DGS (Yang et al., 2023)	0.995	0.985	0.984	0.989	0.974	0.992	0.993	0.993	0.885	0.792	0.764	0.886	0.701	0.865	0.872
Mini-Splatting (Fang & Wang, 2024)	0.995	0.985	0.984	0.988	0.977	0.993	0.993	0.993	0.878	0.789	0.751	0.888	0.811	0.859	0.873
SpeeDe3DGS (Tu et al., 2026)	0.995	0.985	0.984	0.988	0.977	0.993	0.993	0.993	0.878	0.789	0.751	0.888	0.811	0.859	0.873
L1ResPrune	0.995	0.985	0.985	0.989	0.976	0.994	0.994	0.993	0.885	0.790	0.851	0.889	0.806	0.862	0.872

Table 16: Dynamic LPIPS per-scene results.

Methods	bouncingballs	hellwarrior	hook	jumpingjacks	lego	mutant	standup	trex	as	basin	bell	cup	plate	press	sieve
Deformable3DGS (Yang et al., 2023)	0.009	0.024	0.009	0.013	0.019	0.005	0.009	0.010	0.181	0.189	0.345	0.159	0.409	0.192	0.149
Mini-Splatting (Fang & Wang, 2024)	0.010	0.032	0.019	0.016	0.025	0.010	0.011	0.012	0.181	0.191	0.333	0.158	0.430	0.191	0.149
SpeeDe3DGS (Tu et al., 2026)	0.010	0.034	0.019	0.016	0.021	0.009	0.010	0.012	0.187	0.189	0.345	0.159	0.221	0.197	0.149
L1ResPrune	0.009	0.033	0.018	0.014	0.022	0.008	0.009	0.011	0.179	0.190	0.156	0.155	0.221	0.193	0.146

Table 17: Dynamic Gaussian numbers (K) under different pruning methods. Mini-Splatting (Fang & Wang, 2024), SpeeDe3DGS (Tu et al., 2026), and L1ResPrune use the same Gaussian budget for each scene.

Methods	bouncingballs	hellwarrior	hook	jumpingjacks	lego	mutant	standup	trex	as	basin	bell	cup	plate	press	sieve
Deformable3DGS (Yang et al., 2023)	163	38	16	89	290	171	171	220	186	255	363	204	228	186	204
Other Methods	32	7	3	18	58	34	16	45	37	51	72	41	45	37	40