VDGS: IMPROVING FIDELITY IN FAST 3D GAUSSIAN SPLATTING TRAINING VIA VIRTUAL DENSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fast 3D Gaussian Splatting training must improve fidelity with limited densification. Existing candidate criteria often aggregate gradients or errors at the parent level. Visible-view averages do not directly capture cross-view frequency, while parent aggregation obscures demand across the sides of a prescribed split. We propose VDGS, based on Virtual Densification (VD), to rank candidate splits before creating children. Parent gradient statistics first nominate candidates; the prescribed split geometry then partitions their pixelwise backward signals into two virtual child domains. VDGS combines two-sided activity and net-gradient direction differences, accumulates this evidence over a selection window, and ranks splits for submission. It reuses the regular backward pass without temporary children or trial training. In a common low-budget framework, full VDGS improves mean PSNR over direct AbsGS growth by approximately 0.45 dB on nine Mip-NeRF 360 scenes, while also improving SSIM and LPIPS.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) grows a sparse initialization into a dense representation during training. Additional Gaussians capture finer details but increase cost, so time-sensitive reconstruction limits densification. Every low-value expansion then consumes a scarce slot, making growth allocation central to fast training.

Prior work selects densification candidates using parent gradients, reconstruction errors, edge responses, and cross-view statistics. These signals locate unresolved optimization pressure, but a large gradient or error does not establish whether a particular expansion can use its new parameters to resolve that demand. Parent-level aggregation removes how pixelwise responses are distributed between potential child domains. Averaging only over visible views describes activity when observed, without directly measuring demand across the full set of training views. Because growth slots serve the whole reconstruction, strong demand in a few views and recurring demand across many views deserve different ranking opportunities. We ask whether cross-view support and responses on both sides of a prescribed split can guide this allocation before splitting.

Under strong parent-level pressure, two potential child domains may have different or conflicting demands. A single parent struggles to satisfy both, whereas separate children can respond independently. We therefore propose VDGS, centered on Virtual Densification (VD). Parent statistics first nominate high-demand Gaussians; VD then evaluates each prescribed split by projecting its hypothetical child centers, partitioning the parent's covered pixels into virtual domains, and aggregating gradient evidence on both sides. A score combines two-sided activity with net-gradient direction differences. Accumulating scores over the selection window lets both response strength and cross-view frequency influence slot allocation. Figure 1 contrasts split potential under equal parent demand and illustrates virtual evaluation. VD ranks candidate splits using the normal backward pass, without creating children or running trial training.

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.

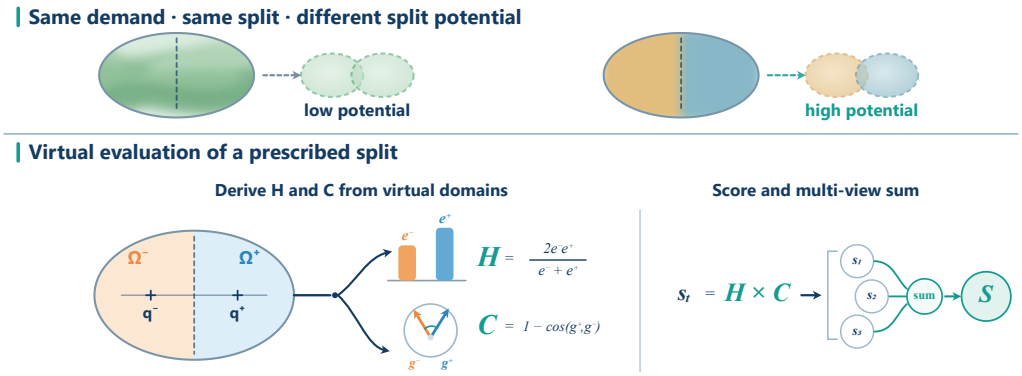

Figure 1: **VDGS insight and virtual evaluation.** Top: Equal parent demand and the same prescribed split can yield different split potential. Bottom: Projected hypothetical child centers define two virtual domains; their activity H and net-direction difference C give $s_t = HC$, which is aggregated across views for ranking. Colors indicate response differences, dashed outlines prospective children, and scores the real parent's backward contributions.

On Mip-NeRF 360, full VDGS takes 1.75 minutes on average and achieves higher PSNR and SSIM than accelerated 3DGS (3DGS-Accel). It also improves quality over FastGS at a similar training time. Under a common candidate pipeline, LAS operator, budget schedule, and capacity ceiling, the two-sided score improves SSIM and LPIPS over parent scores with matched visibility normalization; window accumulation improves quality further.

Our main contributions are:

- A ranking principle based on both sides of a prescribed split and their accumulated cross-view evidence.
- A lightweight VD probe that scores two-sided activity and direction from split geometry without temporary children.
- A low-budget system with favorable quality and training time, supported by controlled tests of two-sided scoring and window accumulation.

2 Related Work

Gaussian representation and rendering. 3DGS (Kerbl et al., 2023) combines an explicit Gaussian representation, differentiable rasterization, and adaptive density control to achieve high-quality real-time novel-view synthesis. Mip-Splatting (Yu et al., 2024) suppresses aliasing with 3D smoothing and 2D filtering; Scaffold-GS (Lu et al., 2024) organizes view-dependent Gaussians through neural anchors; and Speedy-Splat (Hanson et al., 2025) improves rendering efficiency through tighter rasterization bounds and sparse primitives.

Densification and pruning. For a given resolution and training setup, both the number of Gaussians and their growth trajectory strongly affect training cost. Pixel-GS (Zhang et al., 2024), AbsGS (Ye et al., 2024), and ReAct-GS (Cheng et al., 2025) improve densification statistics through pixel-coverage weighting, absolute pixelwise gradients, and rendering-contribution weighting, respectively. FDS-GS (Zeng et al., 2025) relates density and scale reparameterization to local frequency; Perceptual-GS (Zhou & Ni, 2025) and ImprovedGS (Deng et al., 2026) guide growth using perceptual sensitivity and image edges, respectively. MCMC-3DGS (Kheradmand et al., 2024) adjusts primitive configurations through sampling and relocation. For pruning, LightGaussian (Fan et al., 2024) selects primitives by contribution score, while Compact-3DGS (Lee et al., 2024), MaskGaussian (Liu et al., 2025), and GNS (Deng et al., 2025) obtain compact representations through learnable masking, existence probabilities, and sparse competition in native opacity, respectively. VDGS focuses on growth and asks how to allocate limited low-budget slots to splits that can better use the added degrees of freedom.

Fast training. Taming-3DGS (Mallick et al., 2024) combines budgeted growth with efficient backpropagation, while Faster-GS (Hahlbohm et al., 2026) optimizes training execution and memory access. DashGaussian (Chen et al., 2025) jointly schedules image resolution and Gaussian growth, and Turbo-GS (Dhiman et al., 2026) combines sparse-pixel rendering with convergence-aware budget scheduling. Mini-Splatting (Fang & Wang, 2024) and Mini-Splatting2 (Fang & Wang, 2026) progress from spatial reorganization and simplification to rapid early growth and visibility-based culling. FastGS (Ren et al., 2026) coordinates densification and pruning using high-error pixels across views, while SAD-GS (Lyu et al., 2026) selects subdivision axes and child counts according to the match between projected scale and image frequency. These methods reduce training cost through execution and structural changes; VDGS targets quality within a fast-training setup by improving the use of a limited densification budget.

Split utility and budget allocation. SteepGS (Wang et al., 2025) evaluates local descent potential using a second-order split matrix and creates children along the selected direction. AdpSplit (Lee et al., 2026) adapts children to error regions. LeGS (Ning et al., 2026) executes structural actions, evaluates their descendant primitives, and accumulates rewards to train an online decision policy. GDAGS (Zhou et al., 2026) and DC4GS (Jeong et al., 2025) use gradient-direction coherence within the parent domain to assist densification; DC4GS also compares directional conflict within subdomains to determine split positions. VDGS fixes the LAS geometry under evaluation, measures activity support and differences in net response between its prescribed sides from the real parent's backward signals, and allocates growth slots according to the coverage of this evidence across sampled views.

3 METHOD

VDGS separates parent-demand nomination from split evaluation. Absolute position gradients nominate candidates; prescribed split geometry defines two virtual receiving domains, whose backward evidence is accumulated for budgeted split decisions.

3.1 REPRESENTATION, OPTIMIZATION DEMAND, AND SPLIT OPERATOR

Gaussian representation and densification. 3DGS represents a scene using Gaussians with independent centers $\boldsymbol{\mu}_i$, covariances $\Sigma_i = R_i \operatorname{diag}(\boldsymbol{\sigma}_i^2) R_i^\top$, opacities o_i, and spherical-harmonic colors. Here R_i is the rotation, $\boldsymbol{\sigma}_i$ contains principal-axis scales, and diag forms a diagonal matrix from their squares. Projected Gaussians are composited front to back; pixelwise opacity and transmittance determine each color contribution, and the remaining transmittance composites the background.

For pixel $\mathbf{p}$, let $\mathcal{N}(\mathbf{p})$ be the contributing Gaussians and $j \prec i$ mean that j precedes i in depth. $\alpha_i(\mathbf{p})$ is the opacity response determined by o_i and the projected 2D Gaussian, $T_i(\mathbf{p})$ the incoming transmittance, and $\mathbf{c}_i$ the view-dependent color. With background transmittance T_{bg} and color $\mathbf{c}_{\mathrm{bg}}$, the rendered color is

$$\widehat{\mathbf{I}}(\mathbf{p}) = \sum_{i \in \mathcal{N}(\mathbf{p})} T_i(\mathbf{p}) \alpha_i(\mathbf{p}) \mathbf{c}_i + T_{\mathrm{bg}}(\mathbf{p}) \mathbf{c}_{\mathrm{bg}}, \qquad T_i(\mathbf{p}) = \prod_{j \prec i} [1 - \alpha_j(\mathbf{p})]. \tag{1}$$

New Gaussians add independent local capacity; under a constrained budget, densification must place it where representation is insufficient.

Locating demand with absolute gradients. Position gradients locate adjustment demand, but opposing pixels can cancel at the parent level. Following AbsGS (Ye et al., 2024), we take componentwise absolute pixel gradients before summation and measure the resulting vector norm as activity.

Let $\mathbf{u}_{i,t,p} \in \mathbb{R}^2$ be pixel p's reconstruction-loss gradient contribution to Gaussian i's screen-position channel at iteration t, expressed in normalized image coordinates. $\mathcal{P}_{i,t}$ contains pixels contributing to the parent's backward sum; $|\cdot|$ is componentwise absolute value and $\|\cdot\|_2$ the Euclidean norm. Single-view activity is

$$d_{i,t} = \left\| \sum_{p \in \mathcal{P}_{i,t}} |\mathbf{u}_{i,t,p}| \right\|_2 . \tag{2}$$

Between adjacent densification events, a window's visible-view mean activity determines eligibility. If eligible candidates exceed the cap, accumulated activity weights sampling without replacement; otherwise all remain (Appendix A.2). VD then examines their activity on both sides of a prescribed split.

Choosing an operator with explicit split geometry. The original 3DGS clone overlaps its parent, while random splitting needs a fixed draw to evaluate a particular split in advance. We use ImprovedGS Long-Axis Split (LAS) (Deng et al., 2026), which places child centers symmetrically along the longest parent axis and shrinks their scales, giving each parent a predictable split geometry.

Let $k_i = \arg\max_k \sigma_{i,k}$ index the longest scale axis, $\mathbf{e}_{k_i}$ be its coordinate basis vector, and δ be a dimensionless split-distance parameter with $0 < \delta < 1$. The two child centers $\boldsymbol{\mu}_i^+$ and $\boldsymbol{\mu}_i^-$ lie on opposite sides of this axis:

$$\boldsymbol{\mu}_i^{\pm} = \boldsymbol{\mu}_i \pm 3\delta\sigma_{i,k_i} R_i \mathbf{e}_{k_i}. \tag{3}$$

The child scale on the split axis is multiplied by $1-\delta$ and the other scales by $\sqrt{1-\delta^2}$. VD evaluates this prescribed split only. The axis index is fixed at nomination, while evaluation and submission use the then-current trained parent parameters. Other child rules appear in Appendix A.5.

3.2 Virtual Densification: split-aligned utility evaluation

VD evaluates whether both sides of a prescribed split have sufficient activity and different net responses. It partitions parent pixel evidence into two virtual domains and scores their activity and directional difference.

Constructing virtual receiving domains. Projected hypothetical child centers partition the parent's pixelwise contributions by proximity, exposing demand associated with each prospective child location.

At sampled iteration t, Eq. (3) projects to $\mathbf{q}_{i,t}^+$ and $\mathbf{q}_{i,t}^-$. Their midpoint is $\mathbf{m}_{i,t}$, the half-displacement from negative to positive is $\boldsymbol{\Delta}_{i,t}$, and their perpendicular bisector partitions pixels into $\Omega_{i,t}^+$ and $\Omega_{i,t}^-$, assigning ties to the positive side:

$$\begin{aligned} &\mathbf{m}_{i,t} = \tfrac{1}{2}(\mathbf{q}_{i,t}^+ + \mathbf{q}_{i,t}^-), \qquad \boldsymbol{\Delta}_{i,t} = \tfrac{1}{2}(\mathbf{q}_{i,t}^+ - \mathbf{q}_{i,t}^-), \\ &\Omega_{i,t}^+ = \{\mathbf{p} \in \mathcal{P}_{i,t} : (\mathbf{p} - \mathbf{m}_{i,t})^\top \boldsymbol{\Delta}_{i,t} \geq 0\}, \qquad \Omega_{i,t}^- = \mathcal{P}_{i,t} \setminus \Omega_{i,t}^+. \end{aligned} \tag{4}$$

The view-dependent boundary follows updated parent parameters and the axis index fixed at nomination; the partition applies to the real parent's backward contributions.

Retaining activity and net direction. Each domain retains a signed gradient sum for net response and a componentwise ABS sum for activity that could cancel. Let $\mathbf{g}_{i,t}^{\pm}$ and $\mathbf{a}_{i,t}^{\pm}$ be these vectors, and $e_{i,t}^{\pm}$ the norm of the ABS sum:

$$\mathbf{g}_{i,t}^{\pm} = \sum_{p \in \Omega_{i,t}^{\pm}} \mathbf{u}_{i,t,p}, \qquad \mathbf{a}_{i,t}^{\pm} = \sum_{p \in \Omega_{i,t}^{\pm}} |\mathbf{u}_{i,t,p}|, \qquad e_{i,t}^{\pm} = \|\mathbf{a}_{i,t}^{\pm}\|_2. \tag{5}$$

Direction differences supported by activity. We favor splits with activity on both sides and distinct net directions. One-sided activity offers little support for the other child, while angles alone could reward negligible gradients. An activity term H measures two-sided support, a direction term C compares net gradients, and $s = HC$ combines them.

H is a harmonic mean constrained by the weaker side; C is a cosine difference that grows with the angle. Suppressing i, t and numerical stabilization (Appendix A.4), for nonzero inputs we have

$$H = \frac{2e^+e^-}{e^+ + e^-}, \qquad C = 1 - \frac{(\mathbf{g}^+)^\top \mathbf{g}^-}{\|\mathbf{g}^+\|_2 \|\mathbf{g}^-\|_2}, \qquad s = HC. \tag{6}$$

For fixed $e^+ + e^-$, H peaks when both sides are balanced and vanishes with either side's activity. C equals 0, 1, and 2 for aligned, orthogonal, and opposite net responses; near-zero denominators are handled in Appendix A.4. Together, these terms distinguish one-sided demand, aligned two-sided responses, and active sides with differing responses, even at similar parent activity. This is a proxy for how the prescribed split could accommodate those demands independently.

Reusing the regular backward pass. Statistics are collected before parent-level gradient reduction using the regular forward and backward passes. Given parent signed and ABS totals, VDGS accumulates one side, recovers the other by subtraction, and computes HC (Appendix A.3).

3.3 From sampled evidence to budgeted splits

A single view reveals only its own optimization demand. Nomination averages parent ABS activity over visible samples to find candidates with strong demand when observed. Final ranking instead retains the accumulated split evidence across the entire evaluation window, so frequency and evidence strength jointly determine the order. Let $\mathcal{W}$ be the sampled iterations in the evaluation window. Candidate i receives the window score

$$S_i = \sum_{t \in \mathcal{W}} s_{i,t}. \tag{7}$$

Invisible samples, samples without activity on both sides, and samples with undefined net-gradient directions contribute zero. Window accumulation lets both frequency and the two-sided response when visible affect ranking. A candidate never observed has score zero and fails the validity condition. At least one valid sample is required to enter the HC ranking. Because 2D gradients from different views live in different projected coordinate systems, directions are compared within each view; only the resulting scalar scores are aggregated across views.

Nomination, evaluation, and submission proceed in order: at a densification event, parent-level demand from the preceding window nominates candidates; these candidates receive virtual evaluation in the following window and compete for splits at the next event. The number of candidates is capped at a fixed multiple of the expected growth slots. An external growth schedule supplies K slots at the current event. VD allocates them using S_i and the demand-protection and fill-in rules below. Each binary split adds one Gaussian net, while the underlying LAS operator and growth schedule remain unchanged.

Ranking by accumulated evidence can lower the priority of candidates with strong demand in a small number of visible views. We therefore add demand protection and deficit filling alongside HC ranking. Using mean ABS activity saved at nomination, the protection rule first includes candidates exceeding a high-demand threshold, then fills remaining slots by HC score and demand. All routes share the same K slots. The thresholds and selection order are specified in Appendix A.5.

Finally, selected candidates undergo LAS using their current parent parameters and the fixed axis. Children follow the base operator's initialization rules, while unselected parents continue training.

4 Experiments

4.1 Experimental setup

Datasets and metrics. We report PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and training time on nine Mip-NeRF 360 (Barron et al., 2022) scenes, Train and Truck from Tanks and Temples (Knapitsch et al., 2017), and Playroom and Dr Johnson from Deep Blending (Hedman

et al., 2018). We use the 3DGS (Kerbl et al., 2023) $4\times/2\times$ downsampled outdoor/indoor Mip-NeRF 360 images and its provided versions of the other datasets. Every eighth image by sorted filename is held out for testing. Methods share test views and image sizes; quality uses the original 3DGS evaluator, and dataset means weight scenes equally.

Complete methods and criterion comparisons. The main table compares complete systems: 3DGS-Accel, Taming-3DGS (Mallick et al., 2024), DashGaussian (Chen et al., 2025), Mini-Splatting2 (Fang & Wang, 2026), FastGS (Ren et al., 2026), SAD-GS (Lyu et al., 2026), and VDGS. These systems use their authors' original code. Ablated criteria are ported from the respective papers and code, with output differences below 10^{-7} on identical inputs. Two low-budget Mip-NeRF 360 ablations test VDGS decisions: one compares full densification pipelines against AbsGS (Ye et al., 2024); the other fixes candidates to compare scores, visible-view means, window accumulation, and auxiliary budget rules against Parent ABS. All ablations share scene-wise budgets and growth curves.

Training configuration. VDGS trains from SfM for 20K iterations on a black background using $0.8L_1 + 0.2(1 - \text{SSIM})$. The main system prunes 50% with GNS (Deng et al., 2025) after densification; both low-budget ablations disable GNS. All experiments use an RTX 4090 GPU with the data device set to GPU.

4.2 Overall results and densification decisions

Table 1: Complete systems on 9/2/2 equally weighted scenes. Time is in minutes; bold marks the best quality per dataset and blue marks VDGS.

Method	PSNR↑	SSIM↑	LPIPS↓	Time↓
Mip-NeRF 360				
3DGS-Accel	27.59	0.8168	0.2163	10.46
Taming-3DGS	27.37	0.7969	0.2583	3.99
DashGaussian	27.71	0.8175	0.2216	3.71
Mini-Splatting2	27.40	0.8213	0.2162	1.81
FastGS	27.53	0.7985	0.2620	1.73
SAD-GS	27.25	0.8214	**0.1988**	1.02
VDGS	**27.95**	**0.8274**	0.2192	1.75
Tanks and Temples				
3DGS-Accel	23.85	0.8538	0.1697	6.74
Taming-3DGS	24.06	0.8410	0.2050	2.36
DashGaussian	24.06	0.8511	0.1816	2.87
Mini-Splatting2	23.22	0.8422	0.1855	1.27
FastGS	24.20	0.8429	0.2098	1.38
SAD-GS	23.61	0.8577	**0.1469**	1.59
VDGS	**24.39**	**0.8629**	0.1698	1.25
Deep Blending				
3DGS-Accel	29.86	0.9078	0.2423	8.63
Taming-3DGS	29.69	0.9030	0.2762	2.36
DashGaussian	30.18	0.9078	0.2518	2.34
Mini-Splatting2	30.24	0.9132	0.2435	1.41
FastGS	30.17	0.9066	0.2716	1.28
SAD-GS	29.96	0.9109	**0.2414**	0.72
VDGS	**30.32**	**0.9142**	0.2447	1.24

Reconstruction quality under fast training. Table 1 shows that VDGS reduces training time by approximately 81%–86% relative to 3DGS-Accel across the three datasets, while achieving higher PSNR and SSIM and similar LPIPS. Compared with FastGS at a similar training time, VDGS im-

proves PSNR, SSIM, and LPIPS on all three datasets. Thus, virtual split evaluation improves reconstruction quality within a fast-training time range. SAD-GS achieves the lowest LPIPS on all three datasets and the shortest training time on Mip-NeRF 360 and Deep Blending, but has lower PSNR and SSIM than VDGS. Overall, VDGS offers a favorable balance between quality and time in fast training.

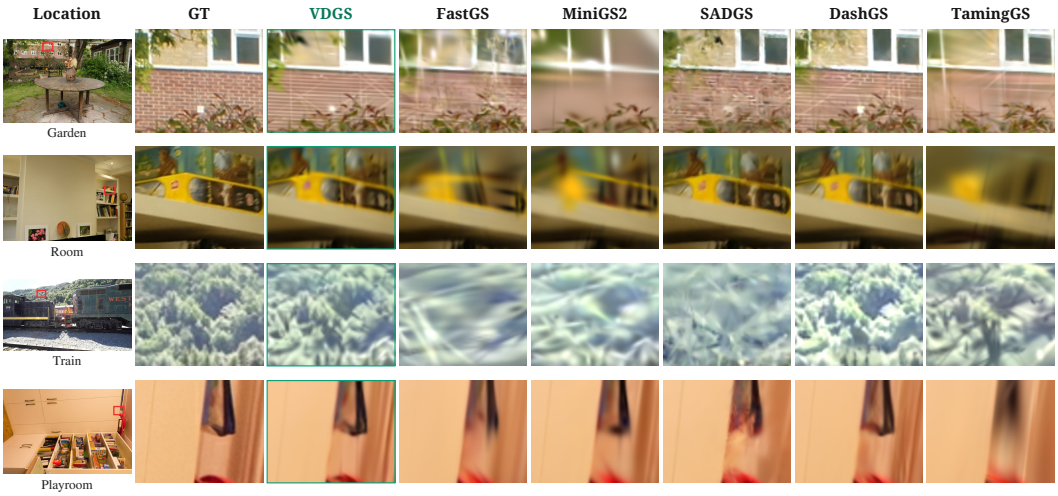

Figure 2: **Complete-system qualitative comparison.** Test-set examples from Garden, Room, Train, and Playroom use the same view and crop within each row, enlarged with bicubic interpolation. Red boxes locate crops; green borders highlight VDGS. MiniGS2, SADGS, DashGS, and TamingGS denote Mini-Splatting2, SAD-GS, DashGaussian, and Taming-3DGS.

Local details. Figure 2 shows differences in details beyond the aggregate metrics. On the brick wall in Garden, objects on the shelf in Room, vegetation in Train, and the edge of the hanging object in Playroom, VDGS and DashGaussian retain more complete local structure. Textures and edges are relatively blurred in FastGS, Mini-Splatting2, and Taming-3DGS, while SAD-GS shows more grainy artifacts. Together with Table 1, these visible details show that VDGS's fast-training quality is reflected in the rendered images.

Table 2: Low-budget densification decisions on nine Mip-NeRF 360 scenes. Time is in minutes; SteepGS uses its own split geometry and all others use LAS.

Method	PSNR↑	SSIM↑	LPIPS↓	Time↓
SteepGS	26.16	0.7369	0.3297	2.86
3DGS	26.81	0.7561	0.3127	1.22
TamingGS criterion	27.33	0.7857	0.2772	1.58
AbsGS	27.22	0.7822	0.2830	1.29
EAS	27.52	0.8057	0.2455	1.52
FastGS gate	27.39	0.7899	0.2733	1.40
VDGS	**27.67**	**0.8135**	**0.2398**	1.40

Densification decision comparison. The first ablation compares the 3DGS gradient, the AbsGS absolute gradient, the TamingGS criterion, the FastGS gate, the edge-aware sampling criterion (EAS) of ImprovedGS (Deng et al., 2026), and the complete VD decision pipeline. The multi-view high-error pixel count of FastGS, the multi-view edge-weighted contributions of EAS, and the view and count terms in the combined importance of TamingGS each incorporate view coverage into their criteria. SteepGS (Wang et al., 2025) selects candidates by the minimum eigenvalue of a

split matrix and splits along its associated eigenvector. This ablation retains its split geometry; the other configurations use LAS.

As Table 2 shows, VDGS has better PSNR, SSIM, and LPIPS than the other compared methods under the same scene-wise budgets and growth curves. Relative to AbsGS, PSNR rises from 27.22 to 27.67 dB, SSIM from 0.7822 to 0.8135, and LPIPS falls from 0.2830 to 0.2398. All three metrics improve in each of the nine scenes (Appendix Table 6). The full VDGS pipeline therefore allocates the limited growth slots more effectively. SteepGS has lower mean quality on all three metrics than 3DGS in this comparison and takes 2.86 minutes, versus 1.22 minutes for 3DGS.

4.3 Two-sided scoring versus parent statistics

To test the benefit of two-sided scoring over parent statistics, we compare Parent ABS, the GCR score used in the official GDAGS implementation (Zhou et al., 2026), the DCC criterion of DC4GS (Jeong et al., 2025), and mean HC within a common candidate pipeline. Let $A_{i,t}$ be the parent's single-view ABS activity, $G_{i,t} = \| \sum_p \mathbf{u}_{i,t,p} \|_2$ its signed-gradient norm, $\mathcal{W}_i^{\mathrm{vis}}$ the visible samples in the window, and $V_i = |\mathcal{W}_i^{\mathrm{vis}}|$; an overbar denotes the mean over visible samples. GCR measures parent-level gradient cancellation using the ratio of signed to ABS activity and assigns the score

$$Q_i^{\mathrm{GCR}} = \bar{G}_i \big[0.8 + 25(1 - r_i)^{15}\big], \qquad r_i = \frac{\bar{G}_i + 10^{-8}}{\bar{A}_i + 10^{-8}}. \tag{8}$$

DCC first computes $r_{i,t}$, the length of the mean direction of unit pixel gradients in each view, then combines directional conflict and parent activity as

$$Q_i^{\mathrm{DCC}} = \frac{1}{V_i} \sum_{t \in \mathcal{W}_i^{\mathrm{vis}}} (1 - r_{i,t}) A_{i,t}. \tag{9}$$

Both criteria pool gradient evidence at the parent level. The same parent totals can arise when opposing responses occupy opposite LAS sides or are mixed within each side; their net responses differ after the prescribed split partitions the pixels. HC retains this geometric distinction through two-sided activity H and net-direction difference C. All four scores use visible-sample means.

Table 3: Parent statistics versus two-sided scoring on nine Mip-NeRF 360 scenes. Scores use visible-sample means; time is in minutes.

Method	PSNR↑	SSIM↑	LPIPS↓	Time↓
Parent ABS	27.40	0.7933	0.2679	1.34
GDAGS GCR	27.40	0.7940	0.2656	1.36
DC4GS DCC	**27.43**	0.7943	0.2662	1.37
Mean HC	27.42	**0.7972**	**0.2621**	1.33

In Table 3, the four scores give similar PSNR, while mean HC obtains the highest SSIM (0.7972) and lowest LPIPS (0.2621). GCR and DCC increase mean SSIM over Parent ABS from 0.7933 to 0.7940 and 0.7943, respectively; mean HC reaches 0.7972. Its LPIPS of 0.2621 is also below GCR's 0.2656 and DCC's 0.2662. Relative to Parent ABS, mean HC improves SSIM and LPIPS in all nine scenes. Its mean LPIPS reduction is 0.0087 across the five outdoor scenes and 0.0022 across the four indoor scenes (Appendix Table 7). Two-sided HC provides more effective candidate ranking in the detail-rich outdoor scenes. Its training time is close to Parent ABS: 1.33 versus 1.34 minutes.

4.4 Window accumulation and demand protection

We compare visible-sample averaging and window accumulation within the same two-window candidate pipeline. Under accumulation, we test parent ABS, H, C, HC, and the full configuration with demand protection and deficit filling.

Table 4: Window accumulation and demand protection on nine Mip-NeRF 360 scenes. Accumulated rows use the same evaluation window; Full VDGS adds demand protection and deficit filling. Time is in minutes.

Method	PSNR↑	SSIM↑	LPIPS↓	Time↓
Mean HC	27.42	0.7972	0.2621	1.33
Accumulated Parent ABS	27.62	0.8128	0.2416	1.44
Accumulated H	27.60	0.8127	0.2412	1.40
Accumulated C	26.71	0.7904	0.2602	1.44
Accumulated HC	27.59	**0.8137**	**0.2389**	1.41
Full VDGS	**27.67**	0.8135	0.2398	1.40

Table 4 shows that, within our two-window pipeline, nominating candidates by visible-mean parent ABS and ranking them by accumulated Parent ABS already reaches 27.62 dB PSNR, 0.8128 SSIM, and 0.2416 LPIPS. This configuration surpasses all prior methods evaluated in Table 2 across the three mean quality metrics, including direct AbsGS and EAS. The nomination and accumulated parent-demand ranking therefore form an effective low-budget growth strategy.

With the same candidates and accumulation, HC reaches 27.59 dB PSNR, 0.8137 SSIM, and 0.2389 LPIPS: PSNR remains close to accumulated Parent ABS, while SSIM improves by 0.0009 and LPIPS by 0.0027. LPIPS improves in eight of nine scenes, and Bicycle gains 0.0059 SSIM and 0.0100 LPIPS; the mean LPIPS gain is 0.0038 outdoors versus 0.0012 indoors (Appendix Table 7). The average incremental benefit is modest, with a clearer effect in this detailed outdoor scene. Mean HC to accumulated HC also shows the value of cross-view frequency; training time changes from 1.33 to 1.41 minutes.

Under the same accumulation rule, H alone gives nearly the same PSNR as HC, while adding C raises SSIM from 0.8127 to 0.8137 and lowers LPIPS from 0.2412 to 0.2389. C alone performs worse than either H or HC, so activity supplies the main ranking signal and direction refines structural and perceptual quality.

Kitchen PSNR falls to 31.27 dB with accumulated HC, from 31.70 dB with mean HC. Demand protection and deficit filling raise it to 31.84 dB. Full VDGS averages 27.67 dB PSNR, 0.8135 SSIM, and 0.2398 LPIPS in 1.40 minutes, compared with 1.44 minutes for accumulated Parent ABS. Per-scene results appear in Appendix Table 7.

5 Conclusion

VDGS ranks prescribed splits under a limited growth budget by separating parent-demand nomination from virtual two-sided evaluation and accumulating response strength across views. The probe uses activity and net-gradient directions from regular backpropagation before creating children. Complete-system comparisons show high PSNR and SSIM with a compact representation. In low-budget ablations, our two-window pipeline with accumulated parent ABS already improves substantially over direct AbsGS; two-sided scoring further improves SSIM and LPIPS, while demand protection and deficit filling recover PSNR. VD thereby provides a split-geometry-aware route to higher fidelity within fast 3DGS training.

AI Use Statement

Generative AI tools assisted with code writing, figure and table preparation, manuscript revision and English translation, and provided feedback on the method, ablation design, and interpretation of results. The authors reviewed all AI-assisted work, checked the algorithm, experimental data, and manuscript claims, and take responsibility for the final content.

Ethics Statement

This work studies 3D reconstruction on established benchmark datasets. We are not aware of ethical issues specific to the proposed method or its evaluation.

Reproducibility Statement

Appendix A serves as a reproduction guide, providing the complete VDGS procedure, candidate and budget rules, numerical handling, and general hyperparameters. Section 4.1 specifies the datasets, training and evaluation setup, while Appendix B reports per-scene results for the main comparison and ablations.

References

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://jonbarron.info/mipnerf360/.

Youyu Chen, Junjun Jiang, Kui Jiang, Xiao Tang, Zhihao Li, Xianming Liu, and Yinyu Nie. Dash-Gaussian: Optimizing 3D Gaussian Splatting in 200 Seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11146–11155, 2025.

Yuxin Cheng, Binxiao Huang, Wenyong Zhou, Taiqiang Wu, Zhengwu Liu, Graziano Chesi, and Ngai Wong. Re-Activating Frozen Primitives for 3D Gaussian Splatting. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 7578–7586, 2025. doi: 10.1145/3746027.3754958.

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. URL https://arxiv.org/abs/2511.16980v1.

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, 2026.

Ankit Dhiman, Tao Lu, R Srinath, Emre Arslan, Angela Xing, Yuanbo Xiangli, Venkatesh Babu Radhakrishnan, and Srinath Sridhar. Turbo-GS: Accelerating 3D Gaussian Fitting for High-Resolution Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15454–15464, 2026.

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 *Advances in Neural Information Processing Systems*, volume 37, 2024.

Guangchi Fang and Bing Wang. Mini-Splatting: Representing Scenes with a Constrained Number of Gaussians. In *European Conference on Computer Vision*, 2024.

Guangchi Fang and Bing Wang. Efficient Scene Modeling via Structure-Aware and Region-Prioritized 3D Gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48 (4):4623–4641, 2026. doi: 10.1109/TPAMI.2025.3646473.

Florian Hahlbohm, Linus Franke, Martin Eisemann, and Marcus Magnor. Faster-GS: Analyzing and Improving Gaussian Splatting Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18946–18957, June 2026.

Alex Hanson, Allen Tu, Geng Lin, Vasu Singla, Matthias Zwicker, and Tom Goldstein. Speedy-Splat: Fast 3D Gaussian Splatting with Sparse Pixels and Sparse Primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21537–21546, June 2025.

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), 2018. URL https://www-sop.inria.fr/reves/Basilic/2018/HPPFDB18/.

Moonsoo Jeong, Dongbeen Kim, Minseong Kim, and Sungkil Lee. DC4GS: Directional Consistency-Driven Adaptive Density Control for 3D Gaussian Splatting. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D Gaussian Splatting as Markov Chain Monte Carlo. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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. URL https://www.tanksandtemples.org/.

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 (CVPR)*, pp. 21719–21728, June 2024.

Yongjae Lee, Jingxing Li, Abhay Kumar Yadav, Rama Chellappa, and Deliang Fan. AdpSplit: Error-Driven Adaptive Splitting for Faster Geometry Discovery in 3D Gaussian Splatting. *arXiv preprint arXiv:2605.06876*, 2026. URL https://arxiv.org/abs/2605.06876v1.

Yifei Liu, Zhihang Zhong, Yifan Zhan, Sheng Xu, and Xiao Sun. MaskGaussian: Adaptive 3D Gaussian Representation from Probabilistic Masks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 681–690, June 2025.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20654–20664, June 2024.

Linjie Lyu, Ayush Tewari, Jianchun Chen, Thomas Leimkühler, and Christian Theobalt. Faster 3D Gaussian Splatting Convergence via Structure-Aware Densification. In *SIGGRAPH Conference Papers*, 2026. doi: 10.1145/3799902.3811212.

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Francisco Vicente Carrasco, Markus Steinberger, and Fernando de la Torre. Taming 3DGS: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. doi: 10.1145/3680528.3687694.

Zhenhua Ning, Xin Li, Jun Yu, Guangming Lu, Yaowei Wang, and Wenjie Pei. Beyond Heuristics: Learnable Density Control for 3D Gaussian Splatting. *arXiv preprint arXiv:2605.00408*, 2026. URL https://arxiv.org/abs/2605.00408v2.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. FastGS: Training 3D Gaussian Splatting in 100 Seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26094–26103, June 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26663–26672, 2025.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004. URL https://ece.uwaterloo.ca/~z70wang/publications/ssim.html.

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*, pp. 1053–1061, 2024. doi: 10.1145/3664647.3681361.

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 (CVPR)*, pp. 19447–19456, June 2024.

Zhaojie Zeng, Yuesong Wang, Lili Ju, and Tao Guan. Frequency-Aware Density Control via Reparameterization for High-Quality Rendering of 3D Gaussian Splatting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(9):9833–9841, 2025. doi: 10.1609/aaai.v39i9.33066.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

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*, 2024.

Hongbi Zhou and Zhangkai Ni. Perceptual-GS: Scene-adaptive Perceptual Densification for Gaussian Splatting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79089–79114, 2025. URL https://proceedings.mlr.press/v267/zhou25s.html.

Zheng Zhou, Yu-Jie Xiong, Jia-Chen Zhang, Chun-Ming Xia, Xihe Qiu, and Hongjian Zhan. Gradient-Direction-Aware Density Control for 3D Gaussian Splatting. In *International Conference on Learning Representations*, 2026.

A COMPLETE VDGS PROCEDURE

This section specifies candidate nomination, virtual evaluation, and budgeted submission.

A.1 GROWTH EVENTS AND SLOTS

VDGS trains for 20K iterations. In the main-table configuration, growth ends at 9K and GNS then prunes 50% of Gaussians by 12K. In the ablations, growth continues to 12K, followed by training at a fixed capacity. Structural events occur every 100 iterations, beginning at iteration 600. The last events for the main table and ablations are at iterations 8900 and 11900, respectively. The growth budget follows the square-root progress curve of ImprovedGS (Deng et al., 2026), which provides the allowed total number of Gaussians B_t at each structural event. Subtracting the current count gives the available net growth slots K. One LAS replaces one parent with two children and thus adds one Gaussian net, consuming one slot per selected candidate. No growth is submitted when no slots remain.

A.2 CANDIDATE NOMINATION

The accumulated parent ABS norm over a window is divided by the visible count; existing parents that meet the densification threshold are eligible. Non-finite accumulated values are reset to zero. The candidate cap is 16 times the expected growth deficit at the next structural event, rounded up. If fewer parents qualify, all are kept; otherwise candidates are sampled without replacement, weighted by accumulated ABS activity. Eligibility uses mean activity, while sampling weights use accumulated activity.

At each structural event, VDGS first submits candidates evaluated during the previous window, then nominates the next group from existing parents retained after the structural update. Newly created children do not enter this nomination. The first event only nominates candidates; no candidates are nominated for a final window that cannot be evaluated. During the following 100 iterations, candidates are probed every two iterations, giving 50 samples in a typical complete window. Nomination fixes the axis index from the largest of the three actual scales. Parent centers, scales, and rotations keep optimizing; probing and final submission use that index with the then-current parent parameters.

A.3 FOUR COMPONENTS FOR ONE SIDE

The probe uses the screen-position channel of the regular backward pass. Its horizontal and vertical components include $W/2$ and $H/2$ scaling, respectively, where W, H are the image width and height. For each candidate, VDGS additionally accumulates four components on the positive side: $(g_x^+, g_y^+, a_x^+, a_y^+)$. The negative-side signed sum is recovered by subtracting the positive sum from the parent total. The negative-side ABS sum is recovered componentwise as $\max(\mathbf{a}^{\text{parent}} - \mathbf{a}^+, 0)$. Directions are compared within each view, while single-view HC scores are accumulated across views. Invalid or invisible samples contribute zero.

A.4 DEGENERATE CASES AND NUMERICAL STABILIZATION

Homogeneous projection of virtual child centers divides by clip-$w + 10^{-7}$. Normalized horizontal coordinate n is converted to pixels using $((n+1)W - 1)/2$, and analogously for the vertical coordinate. Quaternion normalization clamps the norm denominator below at 10^{-12}. If the two centers coincide exactly, Eq. (4) assigns all pixels to the positive side and the negative side has zero evidence.

Let $u^\pm = \|\mathbf{g}^\pm\|_2$, retain $e^\pm$ as defined in the main text, and set $\epsilon = 10^{-12}$. The implemented numerical form is

$$H = \frac{2e^+e^-}{\max(e^+ + e^-, \epsilon)}, \qquad C = 1 - \text{clip}\left(\frac{(\mathbf{g}^+)^\top \mathbf{g}^-}{\max(u^+u^-, \epsilon)}, -1, 1\right). \tag{10}$$

H requires $e^+, e^- > \epsilon$, C requires $u^+, u^- > \epsilon$, and HC requires both conditions. Only finite scores are accumulated, along with the number of valid samples. A candidate needs at least one valid sample to enter the ordinary Top-K ranking; a valid score of zero remains eligible.

A.5 Demand protection and split submission

Let k be the smaller of the current slot count K and candidate count. To estimate the protection threshold, we form a reference set of up to k highest-scoring candidates with valid HC. Let D_i be the mean visible-sample ABS demand saved at nomination. We take natural logarithms of the finite positive demands in the reference set and use linearly interpolated quantiles Q_{25}, Q_{50}, Q_{75} to define

$$\widehat{\sigma} = \frac{Q_{75} - Q_{25}}{1.3489795003921634}, \qquad \tau = \exp(Q_{50} + 2\widehat{\sigma}). \tag{11}$$

If there are fewer than four reference values, we set $\tau = +\infty$. From outside the reference set, we first take up to k candidates with finite positive demand $D_i > \tau$, in descending order of D_i; this includes high-demand candidates with invalid HC. We then fill unselected slots in descending HC order among valid candidates. Any remaining deficit is filled using other finite positive demands. The final selection may remain below budget. Full VDGS uses both demand protection and deficit filling within the original slots, with a threshold coefficient of 2. Selected candidates are split according to the ImprovedGS LAS rules. Each iteration executes rendering and backward propagation, parameter update, reporting and saving, and then structural update, in that order.

A.6 Inherited training parameters

VDGS uses the ImprovedGS position learning-rate schedule, which is better suited to LAS splitting, and retains the official 3DGS settings for the other learning rates. All ablation configurations use the same learning-rate settings. The ABS densification threshold for nomination is 3×10^{-4}, inherited from ImprovedGS. LAS split parameters and child initialization also follow ImprovedGS.

B Per-Scene Results

Table 5 reports quality, training time, and final Gaussian count for the complete systems on all 13 scenes. Tables 6 and 7 expand the two Mip-NeRF 360 ablations to their nine per-scene results. All three tables use the metric conventions of the main text; final counts are rounded to thousands of Gaussians.

Table 5: Per-scene results of complete systems on 13 scenes. Training time is in minutes; final Gaussian counts are in thousands.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	Count (K)
Mip-NeRF 360						
Bicycle	3DGS-Accel	25.27	0.76749	0.20758	16.03	4904
	Taming-3DGS	24.85	0.71868	0.29120	3.90	813
	DashGS	25.39	0.77081	0.21277	5.10	4253
	Mini-Splatting2	25.26	0.77125	0.22145	1.45	672
	FastGS	24.83	0.72265	0.29667	1.53	482
	SAD-GS	25.21	0.77719	0.19228	1.05	5285
	VDGS	25.66	0.78280	0.22145	1.67	600
Flowers	3DGS-Accel	21.57	0.60499	0.33643	9.89	2901
	Taming-3DGS	21.10	0.55954	0.40429	3.27	575
	DashGS	21.94	0.61699	0.32819	3.70	2491
	Mini-Splatting2	21.44	0.61298	0.33110	1.42	615
	FastGS	21.35	0.57517	0.38937	1.62	445
	SAD-GS	21.74	0.62629	0.28680	1.09	5140
	VDGS	21.85	0.62257	0.34244	1.56	450

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	Count (K)
Garden	3DGS-Accel	27.50	0.86826	0.10638	15.45	4155
	Taming-3DGS	27.49	0.85979	0.12447	7.22	1927
	DashGS	27.62	0.86381	0.12110	4.53	2828
	Mini-Splatting2	26.87	0.85015	0.14178	1.77	727
	FastGS	27.32	0.84779	0.15743	2.07	660
	SAD-GS	27.19	0.86588	0.10449	1.16	4883
	VDGS	27.51	0.85944	0.13654	1.91	600
Stump	3DGS-Accel	26.67	0.77241	0.21572	13.36	4278
	Taming-3DGS	25.89	0.73140	0.29331	2.76	480
	DashGS	27.08	0.78400	0.21559	3.14	3466
	Mini-Splatting2	27.10	0.79756	0.20527	1.34	671
	FastGS	26.52	0.75239	0.28160	1.35	350
	SAD-GS	26.39	0.77415	0.20407	0.91	4775
	VDGS	27.31	0.80702	0.20316	1.65	600
Treehill	3DGS-Accel	22.73	0.63599	0.32517	10.66	3254
	Taming-3DGS	22.90	0.62638	0.38094	3.78	785
	DashGS	23.04	0.63977	0.32031	4.23	3241
	Mini-Splatting2	22.77	0.65220	0.32086	1.32	583
	FastGS	22.86	0.60806	0.41691	1.43	363
	SAD-GS	22.59	0.64184	0.28445	1.00	5060
	VDGS	23.26	0.65973	0.32703	1.48	450
Room	3DGS-Accel	31.65	0.92158	0.22013	6.68	1299
	Taming-3DGS	31.46	0.91045	0.25069	3.06	225
	DashGS	31.80	0.91646	0.23347	2.80	1033
	Mini-Splatting2	31.43	0.92424	0.20961	2.00	524
	FastGS	31.76	0.91395	0.24207	1.58	213
	SAD-GS	31.19	0.92361	0.20592	0.91	1765
	VDGS	32.15	0.92577	0.21424	1.50	300
Counter	3DGS-Accel	29.12	0.90994	0.20140	6.87	1081
	Taming-3DGS	29.03	0.90297	0.21947	3.70	311
	DashGS	29.00	0.90429	0.21138	2.87	791
	Mini-Splatting2	28.63	0.90898	0.19451	2.29	549
	FastGS	29.09	0.90151	0.22124	1.81	212
	SAD-GS	28.72	0.91115	0.18912	1.04	1691
	VDGS	29.28	0.91142	0.20119	1.76	300
Kitchen	3DGS-Accel	31.50	0.92834	0.12654	9.33	1599
	Taming-3DGS	31.38	0.92434	0.13927	4.57	482
	DashGS	31.44	0.92129	0.14208	4.37	1256
	Mini-Splatting2	31.38	0.93012	0.12578	2.56	604
	FastGS	31.79	0.92520	0.13675	2.41	392
	SAD-GS	30.78	0.92945	0.12588	1.09	1576
	VDGS	32.16	0.93278	0.12327	2.09	450
Bonsai	3DGS-Accel	32.26	0.94193	0.20724	5.88	1075
	Taming-3DGS	32.22	0.93812	0.22087	3.69	413
	DashGS	32.05	0.94022	0.20971	2.63	842
	Mini-Splatting2	31.70	0.94422	0.19514	2.17	597
	FastGS	32.20	0.93932	0.21572	1.81	279
	SAD-GS	31.46	0.94304	0.19657	0.94	1686
	VDGS	32.37	0.94485	0.20357	2.09	300

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	Count (K)
Tanks and Temples						
Truck	3DGS-Accel	25.49	0.88533	0.14302	7.88	2054
	Taming-3DGS	25.35	0.87042	0.18167	2.12	272
	DashGS	25.84	0.88527	0.15248	2.66	1401
	Mini-Splatting2	25.03	0.87656	0.14922	1.22	348
	FastGS	25.80	0.87744	0.17778	1.33	253
	SAD-GS	25.44	0.89115	0.11257	1.71	3014
	VDGS	26.20	0.89305	0.13563	1.38	450
Train	3DGS-Accel	22.20	0.82228	0.19640	5.59	1089
	Taming-3DGS	22.77	0.81148	0.22837	2.60	365
	DashGS	22.28	0.81686	0.21068	3.08	1000
	Mini-Splatting2	21.41	0.80783	0.22173	1.31	362
	FastGS	22.60	0.80836	0.24187	1.42	230
	SAD-GS	21.77	0.82426	0.18119	1.47	2240
	VDGS	22.59	0.83269	0.20404	1.12	300
Deep Blending						
Dr Johnson	3DGS-Accel	29.53	0.90623	0.23955	10.37	3103
	Taming-3DGS	29.33	0.90031	0.27187	2.71	404
	DashGS	29.69	0.90602	0.25049	2.67	2543
	Mini-Splatting2	29.78	0.91133	0.24049	1.57	811
	FastGS	29.57	0.90146	0.27528	1.35	252
	SAD-GS	29.37	0.90775	0.24435	0.79	2119
	VDGS	29.86	0.91227	0.24351	1.35	450
Playroom	3DGS-Accel	30.20	0.90930	0.24508	6.88	1846
	Taming-3DGS	30.04	0.90574	0.28050	2.00	185
	DashGS	30.67	0.90966	0.25315	2.01	1360
	Mini-Splatting2	30.69	0.91501	0.24658	1.25	487
	FastGS	30.76	0.91168	0.26793	1.20	181
	SAD-GS	30.55	0.91396	0.23839	0.65	1567
	VDGS	30.79	0.91604	0.24590	1.12	300

Table 6: Per-scene densification-decision results on nine Mip-NeRF 360 scenes. Training time is in minutes; final Gaussian counts are in thousands. SteepGS retains its own split geometry.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	Count (K)
Mip-NeRF 360						
Bicycle	SteepGS	23.61	0.60573	0.40526	2.46	599
	3DGS	24.14	0.63726	0.38698	1.14	600
	TamingGS criterion	24.62	0.68544	0.33667	1.47	600
	AbsGS	24.58	0.69366	0.33130	1.16	600
	EAS	25.17	0.74121	0.26901	1.43	600
	FastGS gate	24.77	0.70665	0.31971	1.26	600
	VDGS	25.36	0.75699	0.25536	1.33	600

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	Count (K)
Flowers	SteepGS	19.68	0.44334	0.51381	2.46	450
	3DGS	19.99	0.46993	0.49017	1.01	450
	TamingGS criterion	20.94	0.54080	0.42097	1.38	450
	AbsGS	20.75	0.52636	0.43813	1.10	450
	EAS	21.34	0.58161	0.36832	1.36	450
	FastGS gate	20.88	0.53692	0.42712	1.18	450
	VDGS	21.57	0.59666	0.36688	1.25	450
Garden	SteepGS	25.64	0.76111	0.26782	2.60	599
	3DGS	26.13	0.77943	0.25402	1.22	600
	TamingGS criterion	26.79	0.81860	0.19963	1.58	600
	AbsGS	26.68	0.81494	0.20946	1.32	600
	EAS	26.86	0.82886	0.17444	1.60	600
	FastGS gate	26.89	0.82547	0.19304	1.45	600
	VDGS	27.02	0.83810	0.16513	1.50	600
Stump	SteepGS	24.05	0.62179	0.40351	2.34	599
	3DGS	24.98	0.67082	0.37182	1.05	600
	TamingGS criterion	26.15	0.73563	0.29388	1.36	600
	AbsGS	25.95	0.72410	0.31709	1.12	600
	EAS	26.75	0.77847	0.23018	1.39	600
	FastGS gate	26.33	0.74750	0.28857	1.22	600
	VDGS	27.01	0.78974	0.22617	1.29	600
Treehill	SteepGS	22.59	0.56682	0.46803	2.45	450
	3DGS	22.74	0.58231	0.45282	1.04	450
	TamingGS criterion	23.03	0.61140	0.41317	1.35	450
	AbsGS	22.94	0.60281	0.42470	1.10	450
	EAS	22.99	0.63187	0.36570	1.31	450
	FastGS gate	22.94	0.60707	0.41884	1.18	450
	VDGS	23.10	0.64563	0.35299	1.21	450
Room	SteepGS	30.66	0.90325	0.26207	3.09	299
	3DGS	31.76	0.91180	0.24936	1.22	300
	TamingGS criterion	31.93	0.91542	0.24259	1.66	300
	AbsGS	31.86	0.91562	0.23987	1.29	300
	EAS	31.93	0.91978	0.22896	1.48	300
	FastGS gate	32.03	0.91794	0.23534	1.42	300
	VDGS	32.10	0.92104	0.22626	1.32	300
Counter	SteepGS	28.17	0.88993	0.24225	3.29	300
	3DGS	28.69	0.89869	0.22862	1.37	300
	TamingGS criterion	28.96	0.90295	0.22066	1.76	300
	AbsGS	28.93	0.90167	0.22176	1.44	300
	EAS	29.13	0.90621	0.21231	1.68	300
	FastGS gate	29.09	0.90413	0.21721	1.60	300
	VDGS	29.09	0.90557	0.21373	1.50	300
Kitchen	SteepGS	30.57	0.91670	0.15260	3.77	449
	3DGS	31.30	0.92174	0.14457	1.60	450
	TamingGS criterion	31.71	0.92473	0.13913	1.97	450
	AbsGS	31.66	0.92550	0.13639	1.68	450
	EAS	31.64	0.92699	0.13541	1.92	450
	FastGS gate	31.83	0.92751	0.13288	1.84	450
	VDGS	31.84	0.92837	0.13102	1.76	450

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	Count (K)
Bonsai	SteepGS	30.48	0.92322	0.25215	3.32	300
	3DGS	31.52	0.93284	0.23588	1.34	300
	TamingGS criterion	31.87	0.93608	0.22767	1.66	300
	AbsGS	31.67	0.93517	0.22858	1.36	300
	EAS	31.86	0.93616	0.22551	1.51	300
	FastGS gate	31.78	0.93552	0.22708	1.47	300
	VDGS	31.94	0.93891	0.22058	1.42	300

Table 7: Per-scene scoring ablation under a common candidate pipeline on nine Mip-NeRF 360 scenes. Training time is in minutes; final Gaussian counts are in thousands. Full VDGS adds demand protection and deficit filling.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	Count (K)
Mip-NeRF 360						
Bicycle	Parent ABS	24.81	0.71460	0.30878	1.23	600
	GDAGS GCR	24.80	0.71701	0.30408	1.25	600
	DC4GS DCC	24.82	0.71655	0.30582	1.25	600
	Mean HC	24.78	0.71978	0.30206	1.21	600
	Accumulated Parent ABS	25.34	0.75159	0.26473	1.38	600
	Accumulated HC	25.34	0.75748	0.25477	1.33	600
	VDGS	25.36	0.75699	0.25536	1.33	600
Flowers	Parent ABS	20.96	0.54378	0.41890	1.14	450
	GDAGS GCR	20.98	0.54607	0.41580	1.16	450
	DC4GS DCC	20.96	0.54584	0.41678	1.17	450
	Mean HC	21.10	0.56120	0.40169	1.14	450
	Accumulated Parent ABS	21.59	0.59666	0.36799	1.29	450
	Accumulated HC	21.56	0.59613	0.36655	1.25	450
	VDGS	21.57	0.59666	0.36688	1.25	450
Garden	Parent ABS	26.93	0.82880	0.18720	1.40	600
	GDAGS GCR	26.90	0.82845	0.18512	1.43	600
	DC4GS DCC	26.96	0.83068	0.18403	1.43	600
	Mean HC	26.95	0.83278	0.17957	1.38	600
	Accumulated Parent ABS	26.95	0.83552	0.16884	1.54	600
	Accumulated HC	26.98	0.83812	0.16398	1.51	600
	VDGS	27.02	0.83810	0.16513	1.50	600
Stump	Parent ABS	26.30	0.74788	0.28468	1.20	600
	GDAGS GCR	26.26	0.74708	0.28495	1.21	600
	DC4GS DCC	26.33	0.74892	0.28367	1.22	600
	Mean HC	26.36	0.75269	0.27684	1.19	600
	Accumulated Parent ABS	27.04	0.79047	0.22549	1.31	600
	Accumulated HC	26.98	0.79055	0.22423	1.29	600
	VDGS	27.01	0.78974	0.22617	1.29	600

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	Count (K)
Treehill	Parent ABS	23.05	0.61676	0.40456	1.15	450
	GDAGS GCR	22.99	0.61788	0.40192	1.17	450
	DC4GS DCC	23.09	0.61763	0.40221	1.18	450
	Mean HC	23.04	0.61854	0.40071	1.14	450
	Accumulated Parent ABS	23.17	0.64745	0.35407	1.24	450
	Accumulated HC	23.13	0.64604	0.35247	1.21	450
	VDGS	23.10	0.64563	0.35299	1.21	450
Room	Parent ABS	31.97	0.91903	0.23240	1.33	300
	GDAGS GCR	32.01	0.91983	0.22894	1.34	300
	DC4GS DCC	32.01	0.91920	0.23146	1.36	300
	Mean HC	31.99	0.91972	0.22882	1.30	300
	Accumulated Parent ABS	32.02	0.92084	0.22645	1.37	300
	Accumulated HC	32.01	0.92187	0.22331	1.32	300
	VDGS	32.10	0.92104	0.22626	1.32	300
Counter	Parent ABS	29.07	0.90436	0.21641	1.49	300
	GDAGS GCR	29.10	0.90462	0.21491	1.52	300
	DC4GS DCC	29.10	0.90474	0.21538	1.53	300
	Mean HC	29.04	0.90517	0.21373	1.47	300
	Accumulated Parent ABS	29.09	0.90696	0.21126	1.58	300
	Accumulated HC	29.07	0.90736	0.20970	1.55	300
	VDGS	29.09	0.90557	0.21373	1.50	300
Kitchen	Parent ABS	31.76	0.92831	0.13099	1.77	450
	GDAGS GCR	31.80	0.92843	0.12985	1.79	450
	DC4GS DCC	31.78	0.92842	0.13049	1.79	450
	Mean HC	31.70	0.92849	0.13017	1.76	450
	Accumulated Parent ABS	31.43	0.92695	0.13431	1.81	450
	Accumulated HC	31.27	0.92654	0.13512	1.79	450
	VDGS	31.84	0.92837	0.13102	1.76	450
Bonsai	Parent ABS	31.79	0.93603	0.22683	1.39	300
	GDAGS GCR	31.79	0.93629	0.22501	1.39	300
	DC4GS DCC	31.79	0.93635	0.22566	1.40	300
	Mean HC	31.80	0.93649	0.22519	1.37	300
	Accumulated Parent ABS	31.98	0.93855	0.22101	1.43	300
	Accumulated HC	31.97	0.93890	0.22022	1.41	300
	VDGS	31.94	0.93891	0.22058	1.42	300