COMPRESSION-AWARE EDGE DENSIFICATION FOR COMPACT 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Improving novel-view synthesis under a fixed storage budget requires deciding not only which Gaussians to retain, but also how to create them. While compression pipelines emphasize pruning, the representation built before pruning offers another opportunity to improve compact reconstruction. We introduce Compression-Aware Edge Densification (CAED), which controls early Gaussian growth for a fixed-capacity output. CAED adapts edge-guided selection and long-axis splitting from ImprovedGS, limits growth with a square-root budget, and uses the reconstruction-aware pruning and Difference-of-Gaussians refinement of PWRS to produce a compact model. With exactly the same final Gaussian count and PLY size as PWRS in every scene, CAED improves mean PSNR by 0.2384 dB and SSIM by 0.0045, and reduces LPIPS by 0.0049 across 13 scenes. LPIPS improves in 12 of the 13 scenes. Three-seed experiments on five scenes further support the overall quality gains, with an observed reduction of approximately 21% in training-command wall-clock time within the same framework. These results show that designing early densification for a compact output can improve reconstruction without increasing final model capacity.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) combines an explicit Gaussian representation with differentiable rasterization to enable high-quality, real-time novel-view synthesis. Adaptive density control repeatedly clones or splits primitives to refine poorly reconstructed regions. This growth recovers detail but also increases storage and rendering costs, so the central challenge under resource constraints is to preserve useful scene structure within a limited model capacity.

Densification methods improve the signals, geometry, or schedules used to grow the representation (Zhou et al., 2026; Wang et al., 2025; Deng et al., 2026), whereas compression methods reduce its size through compact parameterizations, sparsification, and pruning (Lee et al., 2024; Zhang et al., 2025; Hanson et al., 2025b). Prune Wisely, Reconstruct Sharply (PWRS) (Wang et al., 2026) introduces reconstruction-aware pruning, spatial–spectral scoring, and Difference-of-Gaussians (DoG) refinement after conventional densification. This raises a complementary question: *with the compression backend and final capacity held fixed, can a different early growth process produce a better compact model?*

Our starting point is simple: *models with the same final number of Gaussians need not follow the same growth trajectory.* CAED controls where new primitives are allocated, how they are initialized, and how many are created. Edge responses guide parent sampling, long-axis splitting places the children, and an intermediate budget limits growth. PWRS pruning and DoG refinement then reduce the representation to the target capacity. Densification thus becomes a way to allocate structure for a compact output, rather than merely to expand the model.

At identical per-scene Gaussian counts and PLY sizes, CAED improves overall reconstruction across 13 scenes, with mean PSNR increases of 0.2651 dB on Mip-NeRF 360 and 0.4474 dB on Tanks & Temples. Three-seed experiments and controlled ablations further characterize the design; here, "compression-aware" refers to organizing training around a fixed output capacity, with the growth budget and pruning target configured separately.

Our contributions are threefold:

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.

- We study early Gaussian generation as a design variable for fixed-capacity reconstruction under an unchanged compression backend.
- We develop a budgeted, edge-guided densification frontend that adapts ImprovedGS sampling and splitting to the PWRS compression pipeline, with separate controls for training-time growth and final capacity.
- We establish matched-capacity quality gains on 13 scenes and three-seed experiments on five scenes, and examine sampling, DoG refinement, capacity, and growth schedules through controlled ablations.

2 Related Work

3D Gaussian Splatting. 3DGS (Kerbl et al., 2023) represents radiance fields with optimizable 3D Gaussians and synthesizes views through depth-sorted differentiable splatting. Mip-Splatting (Yu et al., 2024) addresses aliasing with 3D smoothing and 2D Mip filtering, while Multi-Scale 3DGS (Yan et al., 2024) uses a scale-space representation. FlashGS (Feng et al., 2025) and DashGaussian (Chen et al., 2025) target rasterization and training efficiency, respectively. We consider novel-view synthesis of static scenes, focusing on densification under a fixed final primitive count.

Densification and structure allocation. Taming 3DGS (Mallick et al., 2024) controls growth through a budget trajectory and score-based sampling. Mini-Splatting (Fang & Wang, 2024) improves spatial distribution before simplifying the representation to a constrained number of Gaussians. Original 3DGS triggers cloning or splitting using position gradients and scale thresholds. SteepGS (Wang et al., 2025) studies optimization-theoretic density control through split conditions, split directions, and opacity normalization; GDAGS (Zhou et al., 2026) uses gradient-direction coherence to adapt splitting and cloning, addressing both over-reconstruction and over-densification; and ResGS (Lyu et al., 2025) introduces residual splitting that adds a downscaled Gaussian component while reducing the original Gaussian opacity, supported by progressive image-pyramid supervision. ImprovedGS (Deng et al., 2026) introduces edge-aware selection, long-axis splitting, and square-root growth control. CAED builds on these operators to shape the representation before PWRS pruning and recovery; we examine their role in a fixed-capacity compression pipeline, with implementation details and attribution in the appendix.

Compact representations and pruning. LightGaussian (Fan et al., 2024) combines pruning and recovery with spherical-harmonic distillation and attribute quantization. Speedy-Splat (Hanson et al., 2025a) reduces rendering work through pixel coverage and primitive sensitivity. Compact 3DGS (Lee et al., 2024) and Compressed 3DGS (Niedermayr et al., 2024) reduce storage through compact representations, quantization, or compression. GaussianSpa (Zhang et al., 2025) alternates optimization and sparsification, while PUP 3D-GS (Hanson et al., 2025b) uses uncertainty and reconstruction sensitivity for principled pruning. PWRS (Wang et al., 2026) combines adaptive reconstruction-aware pruning with 3D DoG primitives to recover detail with fewer Gaussians. GS^2 (Yang et al., 2026) further optimizes the spatial distribution of compact models, and ShorterSplatting (Liu & Han, 2026) targets shorter per-pixel or per-ray Gaussian lists. Our focus is complementary: how should training build the representation that will later be compressed to a prescribed size?

3 Method

3.1 Densification for a Fixed Output Capacity

Given training views $\{(I_v, \Pi_v)\}_{v=1}^{V}$, let $\mathcal{G} = \{g_i\}_{i=1}^{N}$ denote the scene representation. Each Gaussian has a position μ_i, rotation R_i, scale $\boldsymbol{\sigma}_i$, opacity, and spherical-harmonic appearance. We optimize reconstruction at a prescribed final capacity N_{tar}:

$$\min_{\mathcal{G}} \sum_{v=1}^{V} \mathcal{L}_{\mathrm{img}}\big(R(\mathcal{G}, \Pi_v), I_v\big), \qquad \text{s.t.} \quad |\mathcal{G}| = N_{\mathrm{tar}}. \tag{1}$$

Reconstruction-aware pruning enforces the final equality constraint. CAED additionally controls early densification, coordinating the location, geometry, and number of new primitives before compression.

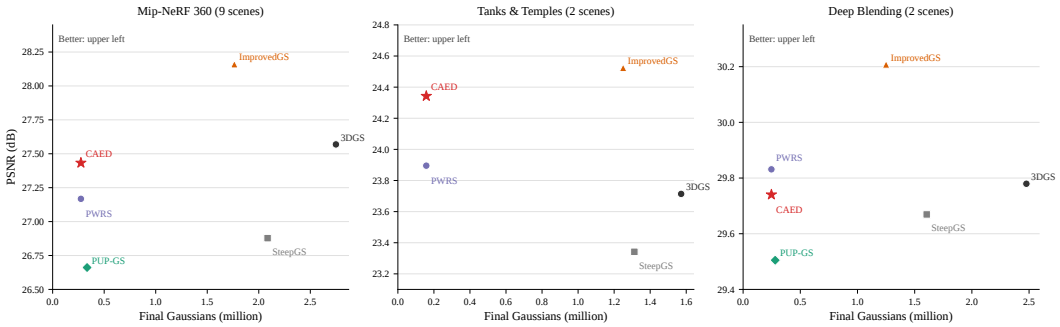


Figure 1: **Higher quality at fixed capacity.** Each point is an equally weighted mean over the scenes in a dataset; stars denote CAED. The axes show final Gaussian count (millions) and PSNR (dB), with the upper left preferred. CAED and PWRS use identical per-scene counts, so the gains on Mip-NeRF 360 and Tanks & Temples require no additional output capacity. All three datasets are shown, with separate axis ranges.

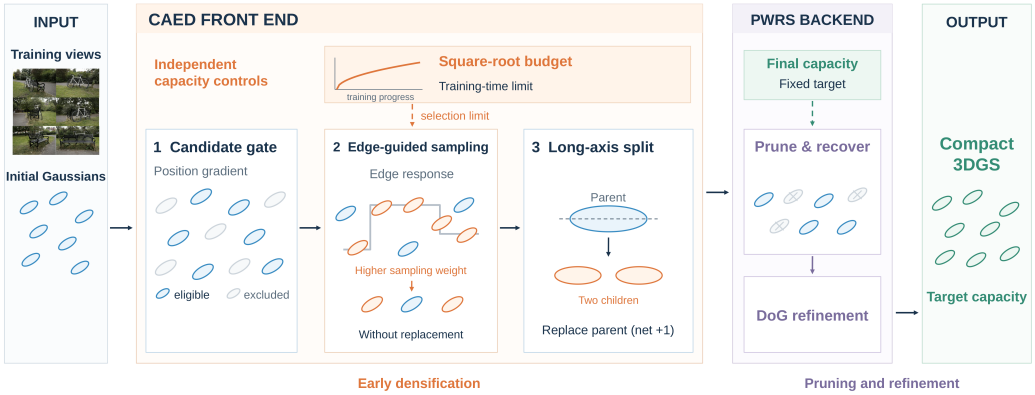


Figure 2: **CAED training pipeline.** Gradient gating identifies candidates, edge responses guide sampling without replacement, a square-root budget limits growth, and long-axis splitting creates children. Training-time growth and final capacity are configured separately. Sampling, splitting, and growth control build on ImprovedGS (Deng et al., 2026); the PWRS backend (Wang et al., 2026) performs RPS/SPS pruning and recovery before DoG refinement.

Figure 2 outlines the pipeline: a position-gradient threshold defines candidate Gaussians, an edge-weighted backward response scores them, and a square-root budget limits growth at each event. Candidates with positive weights are sampled without replacement and split along their largest-scale axis; PWRS then performs adaptive pruning and recovery during training, followed by DoG refinement. The intermediate growth budget $B(t)$ and final pruning target N_{tar} are independent configuration parameters.

Figure 1 shows the resulting quality–capacity trade-off: CAED and PWRS have identical horizontal coordinates, but CAED achieves higher PSNR on Mip-NeRF 360 and Tanks & Temples, capturing our objective of better reconstruction at the same final capacity.

3.2 Edge-Guided Parent Sampling

Image edges provide spatial cues for allocating primitives near contours and texture boundaries. CAED computes a fixed grayscale edge map E_v for each training view and samples a set of training cameras $\mathcal{V}_t$ at densification event t. ImprovedGS motivates edge-aware selection, but the expression below is the local CAED implementation rather than a claim of identity with the source formulation. For the edge-weighted RGB objective $J_v = \sum_{p,c} E_v(p) R_{vpc}$, a backward statistics interface in the

rasterizer measures each Gaussian's response:

$$q_{vi} = \sum_{p \in \mathcal{P}_{vi}} \left(\frac{\partial J_v}{\partial G_i(p)} \right)^2, \qquad e_i = \mathrm{Norm}\left[\sum_{v \in \mathcal{V}_t} q_{vi} \right], \tag{2}$$

where $G_i(p)$ is the projected Gaussian kernel value, $\mathcal{P}_{vi}$ contains the pixels actually processed, and Norm applies min–max normalization after accumulation across views. Unlike the source's absolute pixel-gradient accumulation and candidate gate, the local implementation uses the squared response of the edge-weighted kernel objective and a visible-observation average of the screen-space position-gradient norm for gating. The candidate set C_t contains Gaussians whose screen-space position-gradient norm, averaged over visible observations, reaches the threshold. Among positive-weight candidates $C_t^+ = \{i \in C_t : e_i > 0\}$, sampling without replacement uses weights

$$p_i = \frac{e_i}{\sum_{j \in C_t^+} e_j}, \qquad i \in C_t^+. \tag{3}$$

Gradient gating determines which Gaussians are eligible for expansion; edge responses allocate the available budget among them. The appendix specifies the edge kernel and handling of zero scores.

3.3 Square-Root Growth Budget

Changing parent scores alone does not constrain early model growth. Before each densification event, CAED computes the currently allowed Gaussian count:

$$B(t) = \max\left(1, \left\lfloor B_{\max} \sqrt{\mathrm{clip}\left(\frac{t - t_s}{t_e - t_s}, 0, 1 \right)} \right\rfloor \right), \tag{4}$$

where t_s and t_e delimit budget release, and $B_{\max}$ is the maximum intermediate capacity. This square-root schedule follows ImprovedGS (Deng et al., 2026): it permits relatively more growth early and releases the full budget before densification ends. If N_t is the count before the event, we select at most

$$K_t = \min\big(|C_t^+|, [B(t) - N_t]_+\big) \tag{5}$$

parent Gaussians. Each split adds one Gaussian net, so Equation 5 directly links sampling to the remaining capacity. Splitting stops when the budget is exhausted or no positive-weight candidates remain, and waits until the budget catches up when the initial count exceeds it. The training-time limit $B_{\max}$ is set independently of the final target N_{tar}.

3.4 Long-Axis Splitting

Following the long-axis construction of ImprovedGS (Deng et al., 2026), for each sampled parent g_i, CAED creates two children along its largest-scale axis:

$$\begin{aligned} k_i &= \arg\max_{k \in \{1,2,3\}} \sigma_{i,k}, \\ \mu_i^{\pm} &= \mu_i \pm 3\delta\, \sigma_{i,k_i} R_i \mathbf{e}_{k_i}. \end{aligned} \tag{6}$$

Here, $\mathbf{e}_{k_i}$ is the corresponding local unit basis vector, and R_i maps it to world coordinates. The factor 3 defines a 3σ ellipsoid semi-axis; $\delta = 0.45$ gives the displacement as a fraction of that length. Both children inherit rotation, spherical-harmonic appearance, and DoG parameters. Their long-axis scale is multiplied by $1-\delta$, the other scales by $\sqrt{1-\delta^2}$, and activated opacity by 0.6 before conversion back to parameter space. Removing the parent leaves a net increase of one Gaussian. This initialization allocates new spatial degrees of freedom along the parent's widest extent; the ablations compare it with isotropic splitting.

3.5 Compression Backend

After densification, PWRS (Wang et al., 2026) Reconstruction-aware Pruning Scheduler (RPS) and Spatio-spectral Pruning Score (SPS) perform pruning and recovery, followed by DoG refinement. RPS schedules pruning according to reconstruction recovery, while SPS prioritizes primitives for retention until the count reaches N_{tar}. Our controlled comparison fixes these rules and their configuration; actual pruning times and fractions still adapt to the count and reconstruction state produced by each frontend, so the comparison measures how early structure allocation affects the complete compact reconstruction pipeline.

Table 1: Matched-capacity comparison over 13 scenes. CAED and PWRS use the same final Gaussian target in every scene. Bold and underlined values denote the observed best and second-best results; lower training time is better.

Method	Gaussians	PLY (MB)	PSNR	SSIM	LPIPS	Train (s)†	FPS	Render Mem. (MiB)
PWRS	**252,165**	**66.57**	27.0741	0.8146	0.2682	783.2^{W}	**271.29**	**879.1**
CAED (ours)	**252,165**	**66.57**	**27.3125**	**0.8191**	**0.2633**	**655.2**^{W}	248.49	975.1

† W: training-command wall clock (initialization and saving included); I: project-internal training time (selected repeat); C: cumulative CUDA time. Time markers rank reported values only and do not imply a shared timing protocol.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate 13 scenes: nine from Mip-NeRF 360 (Barron et al., 2022), train and truck from Tanks & Temples (Knapitsch et al., 2017), and drjohnson and playroom from Deep Blending (Hedman et al., 2018). Outdoor and indoor Mip-NeRF 360 scenes use `images_4` and `images_2`, respectively; the remaining scenes use `images`. All experiments use evaluation mode, the LLFF holdout-8 split, the original resolution of the configured test images, and a black background.

Protocol and metrics. CAED uses a 40k-iteration training limit, and for each scene N_{tar} equals the final count of the corresponding PWRS model. Table 1 gives the matched-capacity comparison; Table 2 compares quality and resource use across 3DGS, SteepGS, ImprovedGS, PUP-GS, PWRS, and CAED. PUP-GS is included as an independent post-hoc uncertainty-pruning baseline; its final count is not a matched-capacity target, and its unavailable training time is shown as "–". We report PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), final Gaussian count, and PLY size in decimal MB. FPS and rendering memory are measured in a separate rendering process: FPS after warming up on three test views with synchronized CUDA rendering and no image saving, and memory as the process peak max_memory_reserved() in MiB. Training times retain their original wall-clock, internal, or cumulative CUDA definitions, so best and second-best time markers rank reported values only.

Implementation. Edge scoring uses six random training views per event and a position-gradient threshold of 2×10^{-4}. Budget release runs from $t_s = 500$ to $t_e = 14{,}500$; per-scene $B_{\max}$ and N_{tar} appear in Appendix Table 7. Densification occurs every 100 steps (actual events at 600–14,900), the pruning schedule is initialized at 15,200, and DoG is enabled at 25,200. Paired methods share data configurations, initial point clouds, and camera files; the appendix provides optimization details.

4.2 RECONSTRUCTION AT MATCHED CAPACITY

Table 1 reports the unified evaluation over 13 scenes: CAED and PWRS both average 252,165 Gaussians and 66.57 MB, with exactly matched counts and PLY sizes in every scene. CAED raises mean PSNR from 27.0741 to 27.3125 dB and SSIM from 0.8146 to 0.8191, while reducing LPIPS from 0.2682 to 0.2633. These metrics improve in 11/13, 11/13, and 12/13 scenes, respectively (Appendix Table 8). The gains therefore come from a better representation within the same final capacity, not from retaining more Gaussians.

CAED also offers a compact alternative to higher-capacity densification methods: it uses 15.7% of ImprovedGS's Gaussian count and 16.7% of its PLY size, with mean PSNR close to original 3DGS. Figure 1 shows dataset-level averages, where the identical horizontal positions of CAED and PWRS make the fixed-capacity gains explicit.

To quantify variation across scenes, we apply two-sided paired t-tests to the 13 CAED–PWRS differences. Mean changes in PSNR, SSIM, and LPIPS are $+0.2384$ dB, $+0.00445$, and -0.00490, with 95% confidence intervals of $[+0.0883, +0.3885]$, $[+0.00123, +0.00767]$, and $[-0.00748, -0.00232]$. The corresponding p-values are 0.0047, 0.0109, and 0.0014. Scenes are the paired units; these tests characterize the cross-scene trend.

Table 2: Dataset-level comparison averaged equally over scenes. 3DGS is listed first; the remaining methods are ordered by decreasing Gaussian count within each dataset. Bold and underlined values denote the observed best and second-best results among the five non-3DGS methods within each dataset; 3DGS is excluded from this ranking. PSNR, SSIM, and FPS are higher-is-better; all other columns are lower-is-better. PUP-GS training time is unavailable and is shown as "–".

Dataset	Method	Gaussians	PLY (MB)	PSNR	SSIM	LPIPS	Train (s)†	FPS	Render Mem. (MiB)
Mip-NeRF 360	3DGS	2,748,438	681.61	27.5690	0.8164	0.2144	1,765.9^{W}	113.77	2,309.8
	SteepGS	2,085,622	517.24	26.8780	0.7874	0.2552	13,160.5^{C}	160.59	1,757.6
	ImprovedGS	1,762,911	437.20	**28.1604**	**0.8363**	**0.1857**	758.8^{I}	131.92	1,774.9
	PUP-GS	334,122	82.86	26.6615	0.7865	0.2676	–	**314.23**	**433.8**
	PWRS	**274,252**	**72.40**	27.1678	0.7903	0.2799	855.9^{W}	238.06	909.8
	CAED (ours)	**274,252**	**72.40**	27.4329	0.7959	0.2740	**716.1**^{W}	214.75	1,056.2
Tanks & Temples	3DGS	1,571,870	389.83	23.7139	0.8526	0.1691	1,095.0^{W}	151.81	1,542.0
	SteepGS	1,312,220	325.43	23.3418	0.8377	0.1939	5,944.2^{C}	221.83	1,211.0
	ImprovedGS	1,250,000	310.00	**24.5242**	**0.8718**	**0.1443**	530.3^{I}	190.56	1,028.0
	PUP-GS	183,277	45.45	22.6587	0.8041	0.2431	–	**590.68**	**440.0**
	PWRS	**157,592**	**41.61**	23.8953	0.8314	0.2277	553.0^{W}	388.64	744.0
	CAED (ours)	**157,592**	**41.61**	24.3427	0.8380	0.2225	**467.5**^{W}	381.49	746.0
Deep Blending	3DGS	2,476,474	614.17	29.7795	0.9071	0.2382	1,900.5^{W}	114.47	2,078.0
	SteepGS	1,605,054	398.06	29.6693	0.9042	0.2507	11,401.9^{C}	181.38	1,333.0
	ImprovedGS	1,250,000	310.00	**30.2080**	**0.9123**	**0.2246**	582.5^{I}	188.60	1,292.0
	PUP-GS	280,585	69.59	29.5049	0.9040	0.2570	–	**542.49**	**364.0**
	PWRS	**247,350**	**65.30**	29.8313	0.9074	0.2564	686.5^{W}	303.46	876.0
	CAED (ours)	**247,350**	**65.30**	29.7404	0.9047	0.2563	**569.0**^{W}	267.33	839.0

† W: training-command wall clock; I: project-internal training time; C: cumulative CUDA time. These markers distinguish timing boundaries and do not imply a shared timing protocol.

Table 3: CAED minus PWRS under matched final capacity. Positive PSNR/SSIM and negative LPIPS indicate improvement. Training, FPS, and memory columns report resource changes.

Scope	ΔPSNR	ΔSSIM	ΔLPIPS	ΔTrain	ΔFPS	ΔMem.
13 scenes	+0.2384	+0.0045	-0.0049	-128.0	-22.80	+96.0
Mip-NeRF 360	+0.2651	+0.0056	-0.0059	-139.8	-23.31	+146.4
Tanks & Temples	+0.4474	+0.0066	-0.0051	-85.5	-7.15	+2.0
Deep Blending	-0.0908	-0.0028	-0.0001	-117.5	-36.14	-37.0

4.3 Results Across Datasets

Compact models with real-time rendering. In Table 2, CAED ties PWRS for the smallest count and PLY size on all three datasets and uses approximately 10% of original 3DGS's Gaussians (89.3% less storage). Its FPS is the third highest on each dataset, behind PUP-GS and PWRS; on Deep Blending it also uses less rendering memory than PWRS (839.0 vs 876.0 MiB). On Tanks & Temples, it achieves the second-highest PSNR while rendering at 381.49 FPS.

Relative to PWRS, CAED improves Mip-NeRF 360 by 0.2651 dB PSNR and 0.0056 SSIM, with a 0.0059 reduction in LPIPS. The corresponding changes on Tanks & Temples are $+0.4474$ dB, $+0.0066$, and -0.0051. On Deep Blending, mean LPIPS is close to PWRS, with PSNR and SSIM differences of -0.0908 and -0.0028. Our cross-dataset claim is an overall gain, with consistent improvements on Mip-NeRF 360 and Tanks & Temples.

Quality and computational cost. CAED averages 248.49 FPS and 655.2 seconds of training-command wall-clock time, compared with 271.29 FPS and 783.2 seconds for PWRS. Peak reserved memory in the separate rendering process is 975.1 versus 879.1 MiB (Table 3). We report these costs alongside reconstruction quality, since equal primitive counts do not imply equal rasterization work; training-command time includes initialization and saving, and the appendix details timing boundaries.

CAED has the lowest reported training time on each dataset under the original timing definitions. Against PWRS, which shares the training-command wall-clock definition, observed time decreases by 16.3%, 15.5%, and 17.1% on Mip-NeRF 360, Tanks & Temples, and Deep Blending; ImprovedGS internal time and SteepGS cumulative CUDA time remain separately labeled references.

Table 4: Five-scene matched-capacity frontend ablation. The table compares original ADC with budgeted frontends and then changes the scoring signal or splitting geometry. Best and second-best values are marked at the displayed precision.

Method	PSNR	SSIM	LPIPS
PW ADC	26.7904	0.7919	0.2904
Budgeted-Gradient-Isotropic	26.8036	0.7943	0.2872
Budgeted-EAS-Isotropic	26.9251	0.7961	<u>0.2861</u>
Budgeted-Gradient-LAS	**26.9801**	**0.7975**	**0.2851**
CAED	<u>26.9572</u>	<u>0.7970</u>	**0.2851**

Table 5: Per-scene frontend ablation relative to the matched PWRS ADC baseline. B-G-I is budgeted Gradient-Isotropic; B-E-I and B-G-L replace its scoring or geometry.

Scene	B-G-I ΔLPIPS	B-E-I ΔLPIPS	B-G-L ΔLPIPS	CAED ΔLPIPS	CAED ΔPSNR	CAED ΔSSIM
bicycle	-0.0041	-0.0072	-0.0092	-0.0093	+0.1651	+0.0103
drjohnson	+0.0027	+0.0023	+0.0014	+0.0012	+0.1805	-0.0003
garden	-0.0046	-0.0050	-0.0055	-0.0055	+0.3157	+0.0047
playroom	+0.0003	+0.0000	+0.0001	-0.0006	-0.2180	-0.0030
treehill	-0.0101	-0.0114	-0.0132	-0.0123	+0.3904	+0.0139

4.4 Qualitative Comparisons

Figure 3 compares playroom09, train23, and treehill05, with columns ordered as GT, 3DGS, ImprovedGS, PWRS, and CAED. Red and yellow boxes identify crops containing indoor object boundaries, locomotive railings and grilles, and vegetation and fences. PWRS and CAED have the same scene-level final count; the other methods provide visual context at their own capacities. These examples illustrate local structure; quantitative averages use the full test sets.

Separately, we measure RGB-error PSNR in fixed GT-defined edge regions across all test views of bicycle, train, and playroom. Edge-region PSNR improves by approximately 0.270 and 0.441 dB on bicycle and train (Appendix Table 13). This analysis evaluates the complete frontend on all test views; paired ablations isolate the sampling choice.

4.5 Frontend Ablations

Table 4 compares frontends at matched capacity on bicycle, garden, treehill, drjohnson, and playroom. PW ADC combines original densification with the PWRS backend. Budgeted-Gradient-Isotropic is a budgeted control using a common splitting operator. Within this frontend, Budgeted-EAS-Isotropic changes the scoring signal, Budgeted-Gradient-LAS changes the splitting geometry, and CAED uses both edge-aware selection (EAS) and long-axis splitting (LAS), separating the full frontend replacement from component changes within a shared operator.

Replacing original ADC with the budgeted frontend reduces LPIPS from 0.2904 to 0.2872. Adding EAS or LAS yields 0.2861 or 0.2851, respectively. Under isotropic splitting, EAS raises PSNR from 26.8036 to 26.9251 dB. The two LAS variants have similar perceptual quality, indicating that the effect of the scoring signal depends on the splitting geometry.

We test this relationship further with three-seed Gradient-LAS runs on the same five scenes. Relative to Gradient-LAS, CAED changes mean PSNR, SSIM, and LPIPS by $+0.00897$ dB, $+0.000152$, and -0.000200. The standard deviations of the three seed-level, five-scene mean differences are 0.07620, 0.000610, and 0.000459 (Appendix Table 14). These differences support treating EAS as a sampling design choice rather than a consistently superior scoring signal.

4.6 Robustness Across Random Seeds

Table 6 averages the five scenes for each seed, then reports the mean and sample standard deviation across seeds 0/1/2; no seed is selected by test performance. CAED achieves 26.9977 ± 0.0441 dB PSNR, 0.7974 ± 0.0006 SSIM, and 0.2849 ± 0.0001 LPIPS, improving all three overall metrics over PW ADC. LPIPS decreases in 12 of 15 scene–seed pairs. Repeats within a scene are correlated, not 15 independent scenes. Appendix Table 10 reports a second shared capacity from the same batch,

Table 6: Five-scene, three-seed results. We first average the five scenes within each seed, then report the mean and sample standard deviation over seeds 0/1/2. Seeds were not selected by test performance.

Method	PSNR	SSIM	LPIPS
PW ADC	26.8206±0.0172	0.7917±0.0004	0.2904±0.0003
CAED	26.9977±0.0441	0.7974±0.0006	0.2849±0.0001

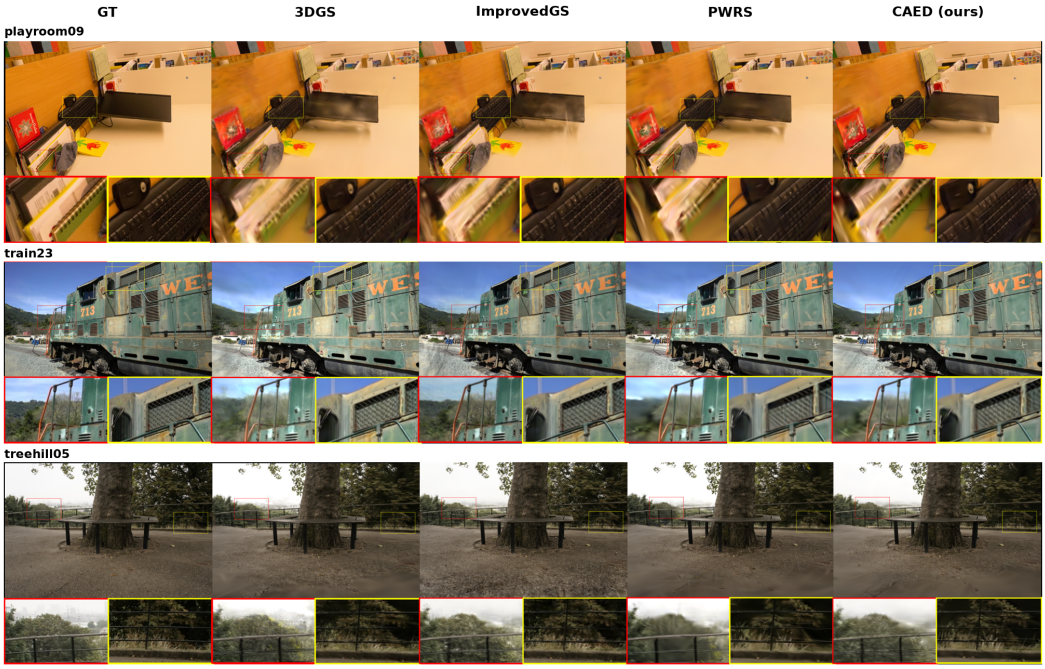


Figure 3: **Visual comparison of compact reconstructions.** Scenes from top to bottom: playroom09, train23, and treehill05. Columns from left to right: GT, 3DGS, ImprovedGS, PWRS, and CAED (ours). Each scene shows full views above the red- and yellow-box crops. View identifiers follow the supplied image labels. These examples complement the full-test-set comparison at the methods' respective capacities.

covering 60 completed runs in total; the main results never select between capacities. At the main capacity, mean training-command time falls from 846.5 to 668.3 seconds, an observed reduction of 21.1% on this subset.

4.7 Growth Trajectories

Figure 4 shows the two levels of capacity control: the frontend budget shapes densification, and adaptive pruning reaches the fixed output target. On bicycle, drjohnson, and playroom, we hold final counts fixed and compare square-root and linear schedules, as well as three square-root budget amplitudes. The default square-root schedule improves mean PSNR by 0.2536 dB and SSIM by 0.00426, and reduces LPIPS by 0.00173 relative to the linear schedule (Appendix Table 18). A smaller amplitude gives mean quality close to the default, indicating room to trade growth budget against training cost. Appendix Table 12 reports five-scene amplitude results at another final capacity.

4.8 Capacity and Backend Interactions

We also compare both frontends at three shared output capacities on bicycle, drjohnson, and playroom, with exactly matched counts within each setting (Appendix Table 17). On bicycle, all three quality metrics improve at every capacity, with LPIPS reductions of 0.00727–0.00966. Deep Blending

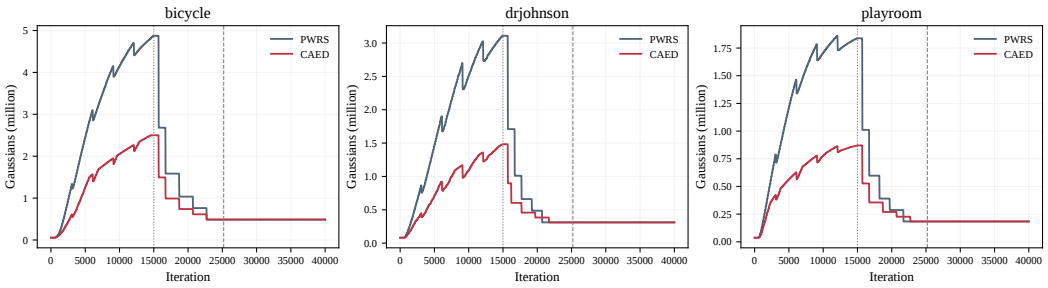


Figure 4: **From budgeted growth to fixed-capacity output.** Gaussian counts and actual pruning events (triangles) for three main-experiment scenes. Dotted lines mark the end of densification at 15k; dashed lines mark the DoG switch at 25,200. The frontends follow different growth and adaptive pruning trajectories but reach identical final counts.

responses vary: drjohnson gains PSNR with slightly higher LPIPS, while playroom lowers LPIPS at the low and reference capacities.

To isolate the backend, we disable DoG and continue ordinary Gaussian optimization in the final stage with fixed training steps and pruning rules. Both frontends achieve higher SSIM and lower LPIPS with DoG (Appendix Table 15); the frontend gain on bicycle is present without DoG, while playroom shows a positive frontend–DoG interaction in PSNR and SSIM.

5 Discussion

Capacity is more than an endpoint constraint. CAED controls early structure generation separately from output size: the former shapes the representation to be compressed, the latter specifies what remains. Early growth therefore remains a useful design variable under fixed backend rules and final capacity.

Scope. We target reconstruction quality at fixed storage capacity and report the associated rendering costs. Gains are consistent on Mip-NeRF 360 and Tanks & Temples, while Deep Blending responses are scene-dependent. Sampling ablations compare design choices, the single-seed experiments characterize three-scene responses, and per-pruning quality and rasterization-workload breakdowns remain open.

6 Conclusion

We introduced CAED for fixed-capacity reconstruction, combining budgeted early structure allocation with the PWRS backend. At identical per-scene counts and PLY sizes, it improves mean quality across 13 scenes and reduces LPIPS in 12 of them, with three-seed experiments supporting the gains. Compact reconstruction depends not only on which Gaussians survive pruning, but also on how they are created.

AI Use Statement

Generative AI assisted with experiment code, literature organization, drafting and revision, and method-diagram design. Experimental values came from saved models and logs, and reconstruction content was neither generated nor repaired. The authors reviewed all AI-assisted content and take responsibility for the final manuscript.

Reproducibility Statement

The appendix provides per-scene targets, budgets, quality and resource measurements, controlled comparisons, operator details, randomness, timing, and provenance records.

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*, pp. 5470–5479, 2022.

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.

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.

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. URL `https://arxiv.org/abs/2311.17245`.

Guangchi Fang and Bing Wang. Mini-Splatting: Representing scenes with a constrained number of gaussians. In *Computer Vision – ECCV 2024*, volume 15135, pp. 165–181, 2024. doi: 10.1007/978-3-031-72980-5_10.

Guofeng Feng, Siyan Chen, Rong Fu, Zimu Liao, Yi Wang, Tao Liu, Boni Hu, Linning Xu, Zhilin Pei, Hengjie Li, Xiuhong Li, Ninghui Sun, Xingcheng Zhang, and Bo Dai. FlashGS: Efficient 3D Gaussian Splatting for large-scale and high-resolution rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26652–26662, 2025.

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*, pp. 21537–21546, 2025a.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. PUP 3D-GS: Principled uncertainty pruning for 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5949–5958, 2025b. doi: 10.1109/CVPR52734.2025.00558.

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. doi: 10.1145/3272127.3275084.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):78:1–78:13, 2017. doi: 10.1145/3072959.3073599.

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*, pp. 21719–21728, 2024.

Jiaqi Liu and Zhizhong Han. Speeding up the learning of 3D Gaussians with much shorter gaussian lists. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1231–1240, 2026.

Yanzhe Lyu, Kai Cheng, Xin Kang, and Xuejin Chen. ResGS: Residual densification of 3D Gaussian for efficient detail recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28093–28102, 2025.

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.

Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3D Gaussian Splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10349–10358, 2024.

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 *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): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3D Gaussian Splatting for anti-aliased rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20923–20931, 2024.

Xianben Yang, Tao Wang, Yuxuan Li, Yi Jin, and Haibin Ling. GS^2: Graph-based spatial distribution optimization for compact 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33259–33268, 2026.

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*, pp. 19447–19456, 2024.

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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yangming Zhang, Wenqi Jia, Wei Niu, and Miao Yin. GaussianSpa: An "optimizing-sparsifying" simplification framework for compact and high-quality 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26673–26682, 2025.

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 Implementation and Evaluation Protocol

Edge responses. RGB values are quantized to 8 bits, converted to grayscale with weights $0.299/0.587/0.114$, and rounded. The edge map uses a 3×3 kernel with center coefficient 8 and all other coefficients -1. Zero-padded convolution is followed by clipping to $[0, 255]$ and per-image min–max normalization. The zero-initialized `scores` interface in Equation 2 returns summed squared kernel-response gradients in the backward pass without changing forward rendering. Nonfinite statistics are set to zero; constant scores normalize to zero. Position-gradient scores are used if edge scores are unavailable, whereas all-zero edge scores skip splitting. Default events stop at 14,900, so the code's late gradient-threshold fallback for $t > 14{,}900$ is not triggered.

Run provenance. Main results, component ablations, and three-seed experiments come from separate batches; seed 0 in different batches does not identify the same run. All 13 main-experiment pairs have byte-identical input point clouds and camera files, and saved data configurations agree except for model paths. The historical main runs did not save complete optimization arguments or contemporaneous source hashes; budgets in Table 7 are therefore identified as launcher settings rather than model measurements. The three-seed batch records full optimization arguments, commands, environment details, and source hashes. Current source code specifies implementation semantics but cannot retrospectively establish binary identity for unarchived runs.

Shared optimization protocol. The main PWRS/CAED comparison uses a 40k-iteration limit. Three-seed metadata records the default Adam optimizer (Kingma & Ba, 2015) and $0.8L_1 + 0.2L_{\mathrm{DSSIM}}$ loss. Initial/final position learning rates are $1.6 \times 10^{-4}/1.6 \times 10^{-6}$ with decay over 30k steps; feature, opacity, scale, and rotation rates are 0.0025, 0.025, 0.005, and 0.001. RPS checks every 500 steps, with a configured maximum wait of 1,500 and recovery factor $\beta = 0.99$. SPS retains the original spatial–spectral implementation; auxiliary residual scores and protection masks are disabled. The current code initializes the schedule at 15,200, permits the first actual pruning at 15,700, and enables DoG at 25,200. Compression therefore occurs during training, not afterward. The original paper describes an approximate 15k–25k interval; we do not assume stepwise equivalence. Strict inequalities within the configured 500–15,000 densification window give actual events at 600–14,900.

Relationship to the source methods. Both Equation (10) of ImprovedGS and our frontend use a square-root growth schedule. Candidate gating, positive-score sampling, and long-axis geometry follow its design. We configure $B_{\max}$ independently before PWRS pruning; PWRS supplies the target-capacity mechanism, RPS, SPS, and DoG. Our comparisons consistently use the local PWRS backend. Historical runs did not separately record statistics corresponding to the original per-primitive DoG degeneration threshold, so we do not report the fraction of active DoG primitives.

Published and reproduced PWRS results. On Mip-NeRF 360, the PWRS paper reports PSNR/SSIM/LPIPS of 27.16/0.789/0.285 and a size of 65.3 MB; our reproduction gives 27.1678/0.7903/0.2799 and 72.40 MB. The main comparison uses unified evaluation of locally trained models. Published numbers provide reproduction context only and are excluded from averages and rankings. We use scene-specific image resolutions, holdout-8, a black background, and the configured test resolution. Local size is actual PLY bytes divided by 10^6. Seeds, versions, serialization, and evaluation implementations can affect comparisons; the size difference is not attributed to units without evidence.

Timing and memory. Historical main results for PWRS, CAED, and original 3DGS use training-command wall-clock time, including initialization and model saving but excluding separate rendering and metrics. PUP-GS has no comparable retained training-time record and is therefore marked "–". ImprovedGS uses project-internal training time for the selected repeat, excluding reporting; the cumulative cost of its three runs is listed separately in the audit CSV. This is not the same wall-clock boundary. SteepGS uses accumulated CUDA iteration time. Main-table memory is peak reserved MiB in an independent rendering process. Three-seed training logs instead report device-memory peaks sampled with NVIDIA SMI every five seconds, including device context and potentially missing transient peaks. These measurements are not interchangeable. Stage-level timing, per-event EAS overhead, and tile-intersection counts were not recorded.

Randomness and coverage. PWRS/CAED controls run on one RTX 3090. The reference environment is Python 3.9.23, PyTorch 1.12.1+cu116, and CUDA 11.6, using seeds 0/1/2 without enforcing determinism in custom CUDA operators. Budgets retain the existing per-scene settings; their historical selection records do not establish a tuning protocol entirely independent of test information. Multi-seed experiments cover five scenes at two capacities. Three-capacity, DoG, and schedule-shape controls cover bicycle, drjohnson, and playroom with seed 0. The 13-scene main comparison uses one run per setting.

Factorial controls. Gradient-LAS adds 15 runs over five scenes and three seeds, paired with the corresponding CAED seed references. Disabling DoG adds six runs over two frontends and three

scenes, continuing ordinary Gaussian optimization after 25,200 and evaluating with the appropriate renderer. Pruning rules, final counts, and 40k training steps are fixed. Low-capacity controls add six runs, with targets rounded to 70% of the reference counts. Budget controls add nine runs: square-root schedules with amplitudes $0.7B_{\max}$ and $1.3B_{\max}$, and a linear schedule with amplitude $B_{\max}$, on three scenes. All 36 planned runs are reported. GT content, data configurations, Gaussian update code, and evaluation code were checked against the earlier reference runs. Entry-point changes are recorded in the artifacts; cross-batch runs are not treated as bitwise deterministic reproductions.

Artifacts. Figure 3 uses a supplied three-scene montage, retaining its method columns and view labels. Records include the original montage hash and lossless layout transformations. These labels have not been independently mapped to COLMAP camera IDs and specific checkpoints, so the figure does not provide or replace quantitative measurements. The regional-error table instead uses unified evaluation outputs with verified cameras and GT.

Anonymized artifacts are organized under `scripts/`, `summaries/`, and `figures/`. Manifests and CSV files record table mappings, input hashes, timing boundaries, per-view image provenance, and stage-log extraction. Renderings are not repaired or replaced with generated content; only consistent cropping, layout, and annotation are applied. Edge diagnostics use the top 20% of GT responses under the fixed kernel, excluding the outermost image border and resolving ties by pixel index. We do not compute LPIPS on images with masked regions set to black.

B PER-SCENE RESULTS AND SENSITIVITY

Table 7: Configuration for 13 scenes. Target counts come from final models; budgets come from the all-scene launcher. Historical main runs did not preserve optimization arguments in cfg_args, so launcher budgets are distinguished from measured model values.

Scene	Image directory	N_{tar}	$B_{\max}$
bicycle	images_4	487,410	2,500,000
flowers	images_4	292,389	1,500,000
garden	images_4	415,063	2,500,000
stump	images_4	442,128	2,500,000
treehill	images_4	326,286	1,500,000
room	images_2	130,527	1,000,000
counter	images_2	107,805	1,000,000
kitchen	images_2	159,648	1,200,000
bonsai	images_2	107,008	1,000,000
train	images	109,225	1,000,000
truck	images	205,960	1,500,000
drjohnson	images	310,876	1,500,000
playroom	images	183,823	1,000,000

Table 8: Per-scene quality in the main experiment. Each metric is reported as PWRS / CAED using the unified formal DoG evaluation; no cross-capacity selection is used.

Scene	PSNR(P)	PSNR(C)	SSIM(P)	SSIM(C)	LPIPS(P)	LPIPS(C)
bicycle	24.8108	25.0743	0.7176	0.7287	0.3102	0.3002
flowers	21.4141	21.6374	0.5794	0.5838	0.3949	0.3921
garden	27.0050	27.2717	0.8323	0.8369	0.1898	0.1845
stump	26.5969	26.8612	0.7695	0.7795	0.2561	0.2441
treehill	22.5655	22.9547	0.5936	0.6072	0.4387	0.4266
room	31.2914	31.7309	0.9060	0.9082	0.2590	0.2555
counter	28.6826	28.7347	0.8822	0.8842	0.2528	0.2503
kitchen	30.5824	31.1253	0.9086	0.9115	0.1705	0.1657
bonsai	31.5613	31.5058	0.9236	0.9229	0.2470	0.2468
train	22.2656	22.8086	0.7907	0.7994	0.2690	0.2616
truck	25.5250	25.8768	0.8720	0.8766	0.1863	0.1835
drjohnson	29.5308	29.7035	0.9068	0.9074	0.2514	0.2511
playroom	30.1317	29.7773	0.9080	0.9019	0.2615	0.2616

Table 9: Per-scene capacity and resources. Counts and PLY are shared by both methods; remaining entries are PWRS / CAED. Training is training-command wall-clock seconds; memory is reserved MiB in an independent rendering process.

Scene	Count	MB	Train(P)	Train(C)	FPS(P)	FPS(C)	MiB(P)	MiB(C)
bicycle	487,410	128.68	1008.0	751.0	250.6	228.9	1240.0	1302.0
flowers	292,389	77.19	773.0	640.0	264.5	230.1	682.0	1108.0
garden	415,063	109.58	1108.0	809.0	252.5	229.3	798.0	1014.0
stump	442,128	116.72	880.0	689.0	267.1	234.3	860.0	940.0
treehill	326,286	86.14	768.0	598.0	270.6	250.3	532.0	558.0
room	130,527	34.46	745.0	679.0	214.2	197.6	1238.0	1312.0
counter	107,805	28.46	779.0	747.0	197.8	179.5	694.0	724.0
kitchen	159,648	42.15	975.0	861.0	197.6	183.1	880.0	1238.0
bonsai	107,008	28.25	667.0	671.0	227.7	199.7	1264.0	1310.0
train	109,225	28.84	490.0	425.0	413.0	415.9	614.0	502.0
truck	205,960	54.38	616.0	510.0	364.2	347.1	874.0	990.0
drjohnson	310,876	82.07	778.0	600.0	268.4	272.3	928.0	912.0
playroom	183,823	48.53	595.0	538.0	338.5	262.3	824.0	766.0

Table 10: Five-scene, three-seed results at two shared capacities. We first average the five scenes within each seed, then report the mean and sample standard deviation across the three means. Absolute per-scene counts at the reference and larger capacities are in Table 11.

Capacity	Method	Mean count	PSNR	SSIM	LPIPS
Reference	PWRS ADC	344,692	26.8206±0.0172	0.7917±0.0004	0.2904±0.0003
Reference	CAED	344,692	26.9977±0.0441	0.7974±0.0006	0.2849±0.0001
Larger	PWRS ADC	448,099	26.9096±0.0147	0.8018±0.0001	0.2734±0.0002
Larger	CAED	448,099	27.0349±0.0988	0.8058±0.0005	0.2681±0.0004

Table 11: Subset capacity targets. Both methods and all three seeds share each scene target; pruning ratios are not redefined from the respective densification endpoints.

Scene	Reference	Larger
bicycle	487,410	633,633
garden	415,063	539,582
treehill	326,286	424,172
drjohnson	310,876	404,139
playroom	183,823	238,970

Table 12: Historical budget-amplitude sensitivity. The larger final capacity, EAS+LAS, and square-root schedule are fixed; only equally weighted five-scene means are shown. This is neither the main-capacity nor schedule-shape ablation.

Budget amplitude	PSNR	SSIM	LPIPS	Train (s)	Peak count
0.7	26.9598	0.8045	0.2690	651.4	1,259,471
1.0	27.1048	0.8064	0.2685	717.8	1,691,027
1.3	27.0657	0.8061	0.2690	724.0	1,995,705

Table 13: Regional-error diagnostic from existing renders. For each test view, we select the top 20% of pixels by GT edge response, compute PSNR from regional RGB MSE, and average equally over views. The mask is fixed by GT and does not depend on the method; masked LPIPS is not reported.

Scene	Method	Edge PSNR	Non-edge PSNR	Full-image PSNR
bicycle	PWRS	22.023	26.038	24.811
bicycle	CAED	22.293	26.279	25.074
train	PWRS	20.399	22.998	22.266
train	CAED	20.840	23.602	22.809
playroom	PWRS	29.217	30.440	30.132
playroom	CAED	29.089	30.026	29.777

C SAMPLING, CAPACITY, AND BACKEND CONTROLS

Table 14: Sampling control with fixed LAS and budget. Values are CAED minus Gradient-LAS, averaged over three seeds per scene; the overall standard deviation is computed from the equally weighted five-scene means for each seed. Positive PSNR/SSIM and negative LPIPS indicate improvement by CAED.

Scene	ΔPSNR	ΔSSIM	ΔLPIPS
bicycle	+0.060959	+0.001176	-0.000361
garden	+0.020189	+0.000292	-0.000263
treehill	+0.005988	-0.000477	+0.000218
drjohnson	-0.035236	-0.000118	+0.000204
playroom	-0.007048	-0.000113	-0.000800
Five-scene mean	+0.008970	+0.000152	-0.000200
Seed-mean std.	0.076195	0.000610	0.000459

Table 15: Frontend and DoG control on three scenes with seed 0. "on" is the historical anchor under the same setting; "off" adds ordinary Gaussian optimization and rendering. The four rows for each scene share the final count, 40k steps, and pruning rules.

Scene	Frontend	DoG	PSNR	SSIM	LPIPS
bicycle	PWRS	off	24.912464	0.711723	0.322201
bicycle	PWRS	on	24.924702	0.718900	0.309296
bicycle	CAED	off	25.115911	0.722167	0.312862
bicycle	CAED	on	24.977793	0.727866	0.299972
drjohnson	PWRS	off	29.618927	0.906632	0.252378
drjohnson	PWRS	on	29.572353	0.907649	0.250630
drjohnson	CAED	off	29.654438	0.905854	0.254122
drjohnson	CAED	on	29.650642	0.907479	0.251655
playroom	PWRS	off	30.154819	0.907454	0.264630
playroom	PWRS	on	30.107569	0.907637	0.262443
playroom	CAED	off	29.636814	0.902655	0.263233
playroom	CAED	on	30.228838	0.909558	0.261037

Table 16: DoG interaction: $(\mathrm{CAED}_{on} - \mathrm{PWRS}_{on}) - (\mathrm{CAED}_{off} - \mathrm{PWRS}_{off})$. Positive PSNR/SSIM and negative LPIPS indicate that DoG increases the relative frontend gain.

Scene	ΔPSNR	ΔSSIM	ΔLPIPS
bicycle	-0.150356	-0.001478	+0.000016
drjohnson	+0.042778	+0.000609	-0.000720
playroom	+0.639275	+0.006720	-0.000009

Table 17: CAED–PWRS differences at three shared capacities, all with seed 0. The two methods have identical final counts in every row. Low-capacity runs are new; reference and larger capacities come from historical paired anchors.

Scene	Capacity	Count	ΔPSNR	ΔSSIM	ΔLPIPS
bicycle	Low	341,187	+0.063419	+0.009173	-0.007272
bicycle	Reference	487,410	+0.053091	+0.008966	-0.009323
bicycle	Larger	633,633	+0.188419	+0.009419	-0.009657
drjohnson	Low	217,613	+0.056618	+0.000046	+0.000739
drjohnson	Reference	310,876	+0.078289	-0.000169	+0.001025
drjohnson	Larger	404,139	+0.117233	-0.001473	+0.002022
playroom	Low	128,676	-0.004007	+0.000452	-0.001417
playroom	Reference	183,823	+0.121269	+0.001921	-0.001406
playroom	Larger	238,970	-0.206474	-0.001391	+0.000307

Table 18: Budget control at fixed final capacity, seed 0. Values are variant minus the default square-root $B_{\max}$; only the training-time growth budget changes, while final counts remain at the reference capacity.

Scene	Schedule and amplitude	ΔPSNR	ΔSSIM	ΔLPIPS
bicycle	Square-root, $0.7B_{\max}$	+0.070904	+0.000223	-0.000729
bicycle	Square-root, $1.3B_{\max}$	+0.054920	-0.002471	+0.003520
bicycle	Linear, $B_{\max}$	-0.014534	+0.000142	+0.001026
drjohnson	Square-root, $0.7B_{\max}$	-0.086607	-0.001078	+0.000910
drjohnson	Square-root, $1.3B_{\max}$	+0.004856	-0.000940	-0.000128
drjohnson	Linear, $B_{\max}$	-0.123716	-0.001781	+0.001224
playroom	Square-root, $0.7B_{\max}$	-0.018282	+0.000369	-0.000057
playroom	Square-root, $1.3B_{\max}$	-0.237526	-0.007722	+0.001042
playroom	Linear, $B_{\max}$	-0.622658	-0.011129	+0.002930
Three-scene mean	Square-root, $0.7B_{\max}$	-0.011328	-0.000162	+0.000041
Three-scene mean	Square-root, $1.3B_{\max}$	-0.059250	-0.003711	+0.001478
Three-scene mean	Linear, $B_{\max}$	-0.253636	-0.004256	+0.001727

D CONTEXTUAL COMPARISON ACROSS MODEL CAPACITIES

Table 19 supplements the dataset-level results with equally weighted averages over all 13 scenes. Each method retains its evaluated capacity; only PWRS and CAED have identical per-scene Gaussian counts and PLY sizes. The table therefore compares quality and resource use at different operating points, rather than ranking methods at a common capacity. Its footnote distinguishes the three training-time definitions.

Table 19: Overall averages over 13 scenes for six methods. 3DGS is listed first and is excluded from the ranking. Bold and underlined values denote the observed best and second-best results among the five non-3DGS methods; capacities and timing definitions differ, so matched comparisons use PWRS/CAED. PUP-GS training time is unavailable and is shown as "–".

Method	Gaussians	PLY (MB)	PSNR	SSIM	LPIPS	Train (s)†	FPS	Render Mem. (MiB)
3DGS	2,525,587	626.35	27.3160	0.8359	0.2111	1,683.4^{W}	119.73	2,156.0
ImprovedGS	1,605,092	398.06	**27.9160**	**0.8534**	**0.1853**	<u>696.5</u>^{I}	149.66	1,585.7
PUP-GS	<u>302,678</u>	<u>75.07</u>	26.4831	0.8073	0.2622	–	**391.88**	**424.0**
SteepGS	1,892,704	469.39	26.7634	0.8131	<u>0.2451</u>	11,779.8^{C}	173.21	1,608.2
PWRS	**252,165**	**66.57**	27.0741	0.8146	0.2682	783.2^{W}	<u>271.29</u>	<u>879.1</u>
CAED (ours)	**252,165**	**66.57**	<u>27.3125</u>	<u>0.8191</u>	0.2633	**655.2**^{W}	248.49	975.1

† W: training-command wall clock (including initialization and saving); I: project-internal training time (selected repeat); C: cumulative CUDA time. Markers rank reported values only and do not imply a shared timing protocol.