GAUSS WHAT YOU NEED: COMPACT GAUSSIAN SPLATTING ACROSS SCENE SCALES

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting reconstructs a scene as a collection of Gaussian primitives from a set of posed photographs called the capture. The number of primitives used to represent the scene affects reconstruction quality, storage, and rendering cost. How to select this number automatically across capture scales remains unresolved: configurations effective on standard benchmarks can leave larger captures with too few Gaussians to reconstruct fine details. We observe that the *surface* to represent, given by the capture's extent and resolution, is known before training, whereas its *content complexity* becomes apparent during training, through the reconstruction quality on the training views. We introduce TangoGS, which combines capture-derived model sizing with training-based adaptation: the capture determines the scale of the model, and training feedback determines its final size within that scale. Before training, TangoGS derives a learning allowance for model growth from the capture's total pixels after discounting views that re-observe the same scene points. During training, reconstruction quality guides how many Gaussians to add and remove. On 13 standard benchmark scenes, TangoGS matches the mean PSNR of the best-performing evaluated baseline, LeGS, with 48% fewer Gaussians. On eight large captures, the same configuration automatically scales to larger models when necessary, achieving the highest mean PSNR among evaluated methods: 0.54 dB above the runner-up with $2.3\times$ as many Gaussians. Together, capture-derived learning allowances and training-quality guided density control enable a state-of-the-art quality–size compromise across scene scales without retuning.

1 INTRODUCTION

3D Gaussian Splatting (Kerbl et al., 2023) represents a scene as Gaussians learned from photographs. During training, *density control* determines the number of Gaussians through *densification* (adding Gaussians) and *pruning* (removing them). On a single large capture, our evaluated models range from 87K to 7.4M Gaussians and span 5 dB in PSNR. Selecting the Gaussian count therefore requires balancing reconstruction quality against storage within this broad search space of model sizes.

We group existing density controllers by how they determine Gaussian count: *Fixed-threshold methods* add and remove Gaussians by comparing per-Gaussian signals, such as gradient magnitude and opacity, against preset cutoffs (Hanson et al., 2025). *Preset-budget methods* use a target count or cap specified per scene or derived from the capture using a dataset-specific multiplier (Mallick et al., 2024; Kheradmand et al., 2024; Liao et al., 2025). *Automatic methods* estimate a budget from observed growth or let the count emerge from learned masks and per-Gaussian density-control actions (Chen et al., 2025; Lee et al., 2024; Ning et al., 2026). Applied unchanged to larger captures, settings released for standard benchmarks can leave too few Gaussians to reconstruct fine detail, as our transfer experiments show (Section 3.3). The challenge is to balance reconstruction quality and model size across capture scales without the repeated training runs that manual retuning requires.

Two kinds of information determine how many Gaussians a capture requires. The first is *surface*: the extent and resolution of the observed scene indicate how much of it the model must represent and are measurable before training. The second is *content complexity*: equally sized captures may depict a uniform surface or intricate vegetation, requiring different numbers of Gaussians. How well the model fits this content during training provides an indication of its complexity.

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.

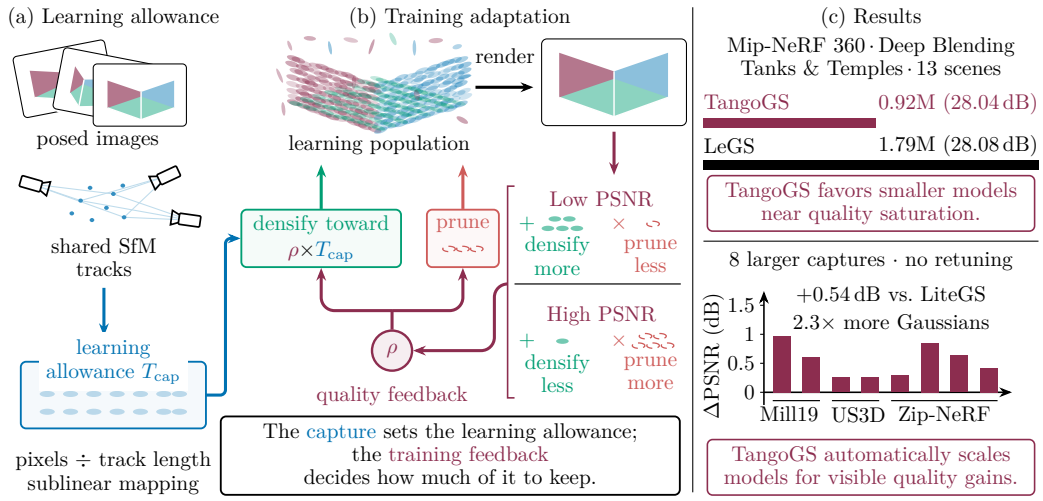


Figure 1: **TangoGS overview.** The capture sets a learning allowance. Quality feedback controls growth, pruning and the final population retained for continued optimization.

We therefore combine *capture-derived model sizing* with *training-based adaptation*. Before training, TangoGS divides the capture's total pixel count by the average number of images observing each scene point, discounting overlapping coverage. A fixed conversion shared across captures maps this adjusted pixel count to a *learning allowance*, a Gaussian budget for model growth (Fig. 1a). During training, a factor computed from mean training-view PSNR sets the densification target and the fraction of Gaussians retained during pruning, adapting model size within this allowance (Fig. 1b).

Our contributions are threefold:

- **A design principle.** We show that combining a capture-derived learning allowance with density control guided by training quality yields a state-of-the-art quality–size compromise across scene scales. A single configuration calibrated on standard benchmarks serves both standard and large captures.
- **A density controller.** TangoGS derives a learning allowance from the capture's total pixel count adjusted for overlapping coverage, and uses training-view PSNR to set the densification target and the fraction of Gaussians retained during pruning. To the best of our knowledge, this is the first density controller to couple a learning allowance derived from pixel count and overlapping coverage with PSNR-guided densification and pruning.
- **An empirical survey across capture scales.** We organize density controllers by how they set and adapt model size, and rerun 13 released implementations under a common evaluation protocol on 13 standard scenes from Mip-NeRF 360 (Barron et al., 2022), Tanks & Temples (Knapitsch et al., 2017) and Deep Blending (Hedman et al., 2018), and on eight large captures from Mill19 (Turki et al., 2022), UrbanScene3D (Lin et al., 2022) and Zip-NeRF (Barron et al., 2023). We examine how quality–size trade-offs shift from standard to large captures, giving practitioners a common basis for choosing a controller suited to their capture scale and quality–size priorities.

On the 13 standard scenes, TangoGS matches LeGS's (Ning et al., 2026) mean PSNR with roughly half as many Gaussians and lies on the PSNR–count frontier for every dataset (Section 3.2). Transferred unchanged to eight large captures, it achieves the highest mean PSNR and lowest mean learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018) among evaluated methods. Compared with the perceptual-quality runner-up, reference 3DGS (Kerbl et al., 2023), TangoGS uses $1.06\times$ as many Gaussians, achieves 0.63 dB higher mean PSNR, and lowers LPIPS on seven of eight captures. In neither regime does an evaluated baseline achieve lower mean LPIPS with fewer Gaussians. Ablations confirm both components' contributions: replacing the capture-derived allowance with a fixed one reduces large-capture PSNR by at least 0.43 dB, while disabling feedback adaptation inflates the Gaussian count by at least $1.9\times$ without improving PSNR.

2 METHOD

Standard 3DGS initializes one Gaussian per sparse 3D point reconstructed by structure-from-motion (SfM) with COLMAP (Schönberger & Frahm, 2016), which estimates camera poses and triangulates points from features matched across images. Training fits each Gaussian's position, shape, opacity, and color to the capture's images (Kerbl et al., 2023). To recover fine detail beyond the sparse initialization, scheduled densification clones or splits Gaussians, while pruning removes those contributing least to rendered images. After density control ends, the final population is optimized for the remaining training iterations.

TangoGS intervenes at three stages (Fig. 1): the capture sets the learning allowance before training (Section 2.1); training quality adjusts growth and pruning within that allowance (Section 2.2); and the same signal selects the final population retained for continued optimization (Section 2.3).

2.1 SETTING THE LEARNING ALLOWANCE

An SfM point's *track length* is the number of images observing it, indicating how often a scene point is re-observed. Summing track lengths over the M tracked points gives O image–point observations, so the mean track length $\bar{t}_{\mathrm{SfM}} = O/M$ is the average number of images observing a point. For n_{cap} capture images at loaded resolution $W \times H$, the total pixel count is $X = n_{\mathrm{cap}} WH$. Dividing by the mean track length discounts repeated observations, giving the track-adjusted pixel count:

$$\tilde{X} = \frac{X}{\bar{t}_{\mathrm{SfM}}} = \frac{XM}{O} = \frac{M}{O/X}. \tag{1}$$

Read this way, $\tilde{X}$ grows with the number of distinct scene points and shrinks when the same points are observed more often, so it approximates observed surface rather than the amount of imagery. Duplicating every image and its observations doubles both X and O, leaving $\tilde{X}$ unchanged, so redundant views do not inflate the allowance. Extracting more features from the same images yields more points; if their track-length distribution stays similar, M and O grow in proportion and $\tilde{X}$ stays unchanged, so the allowance depends on the capture rather than on how finely it was reconstructed. With the same points and observations, halving both image dimensions divides $\tilde{X}$ by four: coarser images carry less detail to fit. We then convert $\tilde{X}$ into the learning allowance T_{cap} (Fig. 1a) in two steps:

$$T_{\mathrm{lin}} = g\,\tilde{X}, \qquad T_{\mathrm{cap}} = \sqrt{T_{\mathrm{ref}}\,T_{\mathrm{lin}}}. \tag{2}$$

The first step converts track-adjusted pixels to Gaussians at a fixed rate g, measured in Gaussians per track-adjusted pixel. The second takes the geometric mean of T_{lin} and a pivot T_{ref}, making the allowance grow with the square root of the track-adjusted pixel count: four times the pixels gives twice the allowance. The pivot sets the overall level; a capture with $T_{\mathrm{lin}} = T_{\mathrm{ref}}$ receives an allowance of exactly T_{ref} Gaussians.

Deriving the constants. We set T_{ref} to the geometric mean of T_{lin} over the 13 standard scenes detailed in Section 3.1, about 2.46M Gaussians. The square-root mapping therefore preserves the geometric mean allowance while reducing the spread between scenes. Because T_{ref} also scales with g, this rate sets the overall allowance scale without changing the ratios between captures. We use $g \simeq 0.0634$ Gaussians per track-adjusted pixel, placing that scale near reference 3DGS's mean final count of about 2.5M Gaussians on these scenes.

2.2 QUALITY FEEDBACK DURING LEARNING

At each growth or pruning event, let P be the mean per-image PSNR over the just-completed training epoch. From P, we compute a keep ratio ρ that scales the learning allowance T_{cap} into the densification target T_{adapt} (Fig. 1b):

$$\rho = \frac{1}{s} - c\,(P - P^*), \qquad T_{\mathrm{adapt}} = \rho\,T_{\mathrm{cap}}, \qquad s = 1.7, \quad c = 0.0075, \quad P^* = 20\,\mathrm{dB}. \tag{3}$$

The floor P^* is the PSNR of an image reconstructed with RMSE 0.1; a run that has not reached it still fits the scene only coarsely, so at or below the floor the ratio is at least $1/s$ and the final selection

in Section 2.3 removes nothing. Above the floor the ratio falls by c per dB: lower training quality permits more capacity, while higher quality allows more aggressive pruning; Section 3.5 shows the ratios this produces for every benchmark capture. We set c and s on the standard scenes and apply them unchanged to the large captures.

At each growth event, budgeted densification (Mallick et al., 2024; Liao et al., 2025) adds Gaussians up to a target that ramps linearly from the initial point count to T_{adapt} over the growth window. Some events are followed by a *soft prune*, which ranks Gaussians present before growth using a sensitivity score (Hanson et al., 2025) and keeps the top fraction ρ. Newly created Gaussians are exempt until they have trained, and removed capacity can be regrown at later events.

Scaling the growth schedule with the iteration count. Each training iteration uses one view, and an epoch covers all training views. Under the fixed iteration count that 3DGS implementations share, captures with more views complete fewer epochs, so fixed spacing in epochs would give them fewer growth events. We therefore space growth events five epochs apart for runs of at least 150 epochs, typical of standard scenes. For shorter runs, we reduce the spacing proportionally, to a minimum of one epoch, keeping the number of growth events roughly constant. Soft prunes follow a separate, sparser schedule, allowing growth frequency to increase without additional pruning.

2.3 Selecting Retained Capacity

After growth ends, we select the retained count in a single prune. Its strength matches the effect of continuing periodic pruning until training ends. We consider a reference schedule with pruning events spaced 8% of the total training iterations apart, and let h count its remaining events. For a population of N_{pre} Gaussians entering this event, the retained count N_{G} is obtained by applying the keep fraction $f = \min(1, s\rho)$, h times:

$$N_{\text{G}} = N_{\text{pre}} \cdot f^h \tag{4}$$

Because removed Gaussians in this stage cannot be regrown, the multiplier $s > 1$ makes each reference prune retain a larger fraction than a soft prune at ρ. At or below P^*, $f = 1$, so the final selection removes nothing (Eq. (3)). We rank Gaussians by their sensitivity score, the accumulated, opacity-weighted effect on training views, keep the N_{G} highest, and optimize that population for the rest of training. We implement the three stages in LiteGS (Liao et al., 2025), keeping its renderer and optimizer.

3 Experiments

Quality differences between 3DGS models cannot be attributed to size alone: Gaussian placement, optimization trajectories, training schedules, and hyperparameters also differ across implementations. We therefore make two comparisons on each capture set. *Against LiteGS*: its released default provides the closest comparison, since TangoGS builds on it, differing only in density control and training profile. *Against state-of-the-art baselines*: comparisons across released implementations assess which methods best suit different captures.

3.1 Setup

We rerun thirteen released baselines on two sets of captures. The 13 *standard scenes* are nine Mip-NeRF 360 scenes (Barron et al., 2022), *truck* / *train* from Tanks & Temples (Knapitsch et al., 2017), and *drjohnson* / *playroom* from Deep Blending (Hedman et al., 2018). These standard 3DGS benchmarks serve as TangoGS's development regime and as a reproduction check against published baseline results. We use released COLMAP reconstructions, every-eighth-frame test splits, and standard resolutions. The eight *large captures* are Mill19 *rubble* / *building* (Turki et al., 2022), UrbanScene3D *residence* / *sci-art* (Lin et al., 2022), and Zip-NeRF *alameda* / *london* / *nyc* / *berlin* (Barron et al., 2023). We report PSNR and SSIM (Wang et al., 2004) (higher is better), VGG LPIPS (Zhang et al., 2018) (lower is better), and final Gaussian count. All re-runs use one Nvidia GH200.

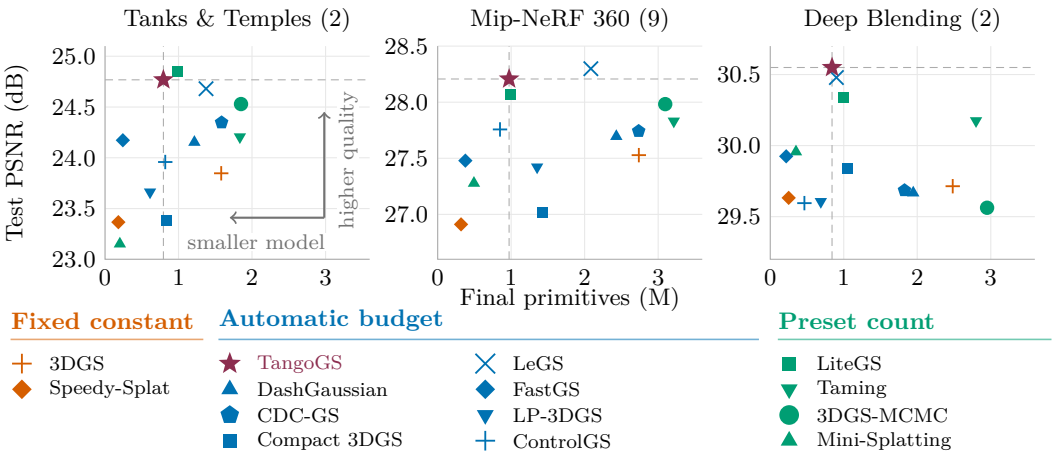


Figure 2: **Benchmark quality–size trade-offs.** Points are dataset means. Color identifies the scale mechanism and shape the method. Full results: Supplementary §C.

3.2 Evaluation on Standard Benchmarks

LiteGS uses the same default target of 1M Gaussians for every scene. TangoGS adapts capacity per scene (Fig. 2): it uses 18% more Gaussians on *bicycle* and 14% fewer on *stump*, improving PSNR by 0.85 dB and 0.33 dB, respectively, and lowering LPIPS on both (Supplementary §C). The largest quality cost of downsizing occurs on *truck*: at 0.68× LiteGS's count, TangoGS stays within 0.1 dB in PSNR but has 0.026 higher LPIPS. At matched count on *truck*, TangoGS leads by 0.07 dB, while LiteGS retains a 0.013 LPIPS advantage, showing that the perceptual gap cannot be explained by size alone. Across all 13 scenes, LiteGS trained to TangoGS's per-scene counts trails by 0.21 dB mean PSNR with worse LPIPS (Supplementary §E).

State-of-the-art baselines.

- *Preset-budget methods.* Taming 3DGS (Mallick et al., 2024) and 3DGS-MCMC (Kheradmand et al., 2024) fill their per-scene presets, using 2.3× to 3.6× as many Gaussians as TangoGS without higher mean PSNR. Taming 3DGS assigns *bonsai* and *counter* the same released budget of 1.2M Gaussians, reaching 32.4 dB and 29.0 dB, respectively. From allowances within 6% of each other, TangoGS's quality feedback retains 0.78M on *bonsai* and 1.06M on the harder *counter*, exceeding Taming 3DGS's PSNR by 0.6 dB and 0.7 dB, respectively.
- *Automatic methods* remain influenced by inherited thresholds: DashGaussian (Chen et al., 2025), LP-3DGS (Zhang et al., 2024) and Compact 3DGS (Lee et al., 2024) use nearly constant fractions of reference 3DGS's count across scenes (0.85×, 0.43× and 0.52×, respectively). LP-3DGS uses 2.41M Gaussians on *garden* versus 0.47M on *bonsai*, a 5× difference, yet trails TangoGS on both; TangoGS retains 0.82M on *garden* and 0.78M on *bonsai*. ControlGS (Zhang et al., 2025) selects smaller models with lower mean quality. Local growth and pruning follow image gradients with uneven PSNR gains: LeGS uses 2.1× to 4.1× our count on the five outdoor scenes, gaining 0.46 dB on *garden* at 4.1× but losing 0.31 dB on *treehill* at 2.4×. On the remaining eight scenes, it matches TangoGS within 0.04 dB at 1.3× the count.

3.3 Transfer to Larger Captures

All methods use standard-scene settings and a requested 60k iterations.

LiteGS. Every aerial initialization exceeds LiteGS's default 1M target, restricting growth and leaving final counts near initialization, above the requested budget. On these scenes, quality feedback retains 37% to 54% of TangoGS's 5.3M to 8.5M learning allowances, yielding 2.0M to 4.6M Gaussians. Across all eight captures, TangoGS improves PSNR by 0.26 dB to 0.97 dB and lowers LPIPS by 0.029 to 0.067, at 2.32× LiteGS's mean count. At TangoGS's selected counts, LiteGS still trails by 0.28 dB mean PSNR with worse LPIPS.

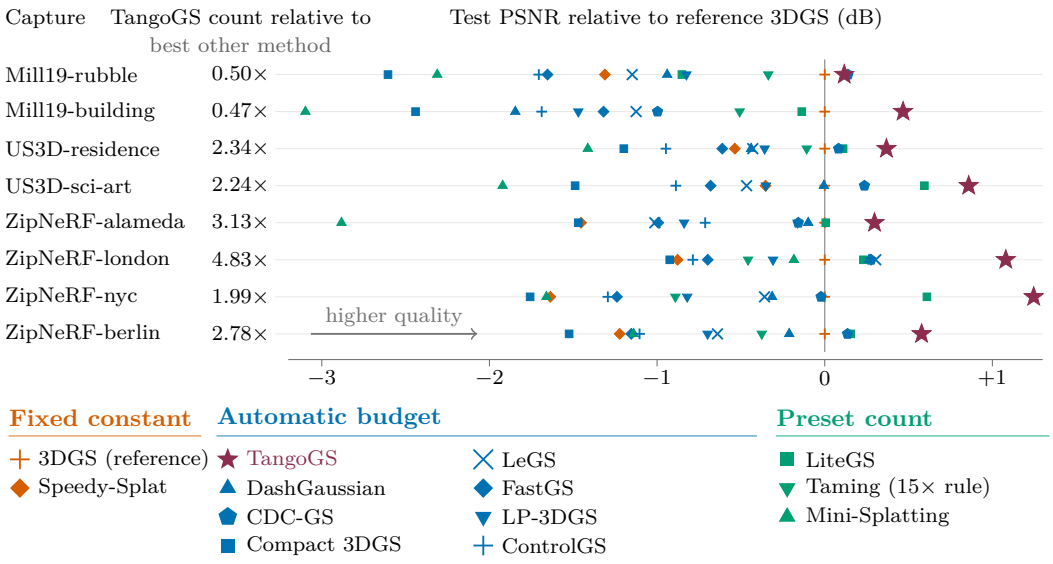


Figure 3: **Transfer of released settings.** PSNR relative to reference 3DGS at 60k iterations; The count column gives TangoGS's count relative to the highest-PSNR method. Speedy-Splat on *building* (−4.43 dB) and 3DGS-MCMC are omitted; full results and diagnostics: Supplementary §D.

State-of-the-art baselines.

- A *preset budget* helps only if growth can reach it. Taming 3DGS caps growth at its budget but only densifies Gaussians whose accumulated gradient passes reference 3DGS's threshold. It reaches its released standard-scene budgets, near reference 3DGS's counts. On aerial captures, its released 15× SfM-point multiplier requests 24M to 32M Gaussians, but few qualify, filling only 7% to 12% of the budget. Zip-NeRF's interiors lack a released Taming 3DGS capture class; all three multipliers (2×, 5×, 15× SfM points) yield 0.30M to 0.44M Gaussians. The gradient threshold therefore dominates model size across these captures. Taming 3DGS trails TangoGS by 0.45 dB to 2.14 dB on seven captures, matches it within 0.01 dB on *sci-art*, and has higher LPIPS on all eight.
- *Automatic methods* remain sensitive to standard growth and pruning settings on larger captures. LP-3DGS, Compact 3DGS and ControlGS retain only 6% to 46% of reference 3DGS's population and trail TangoGS by 0.7 dB to 3.0 dB. Automatic growth also remains constrained by inherited thresholds: DashGaussian stays near reference 3DGS's counts, while LeGS ends below the SfM initialization on every aerial capture. FastGS (Ren et al., 2026) likewise produces small models with lower PSNR. CDC-GS (Dong et al., 2025) grows larger aerial models, but still trails TangoGS except on *rubble*, where it matches PSNR at twice our count. Automatic count selection therefore does not ensure quality–size transfer (Supplementary §D).
- *Fixed-threshold* Speedy-Splat (Hanson et al., 2025) prunes a released 80% of the population grown by reference 3DGS, ending at 0.06M to 0.58M Gaussians, 0.9 dB to 4.9 dB behind TangoGS.

How much effort does retuning require? A difficulty in retuning these baselines to match TangoGS's quality–size compromise is that growth and pruning thresholds have no direct interpretation in terms of final model size or reconstruction quality. For example, 3DGS, Taming 3DGS, DashGaussian, and Speedy-Splat control densification using the inherited gradient threshold; LeGS likewise uses ordinary and absolute-gradient thresholds to select growth candidates. The effect of a cutoff depends on the capture's gradient distribution, which evolves during training, and changing it alters subsequent growth and optimization. Consequently, finding an effective setting can require repeated training runs, alongside adjustments to pruning thresholds or regularization strengths. This empirical parameter selection is the additional deployment effort that TangoGS avoids through capture conditioning and training feedback (Supplementary §G).

Contribution Portability. To test whether the controller transfers across implementations, we run it on reference 3DGS through Taming 3DGS's released code. It matches Taming's released settings

within 0.1 dB on the 13-scene mean with 3.2× fewer Gaussians, and exceeds them on the large captures by 0.8 dB at 2.2× the count (Supplementary §H).

3.4 Ablations: Capture Conditioning and Adaptation

We evaluate all four combinations of TangoGS's two components on all 21 captures, holding the training profile, growth and pruning schedules, views and iteration count fixed (Fig. 4). Disabling the *learning allowance* replaces capture-derived allowances with LiteGS's fixed default of 1M Gaussians. Disabling *count adaptation* removes quality feedback and sensitivity ranking, reverting to LiteGS's density control, which grows to the allowance and prunes only transparent Gaussians or those never rendered.

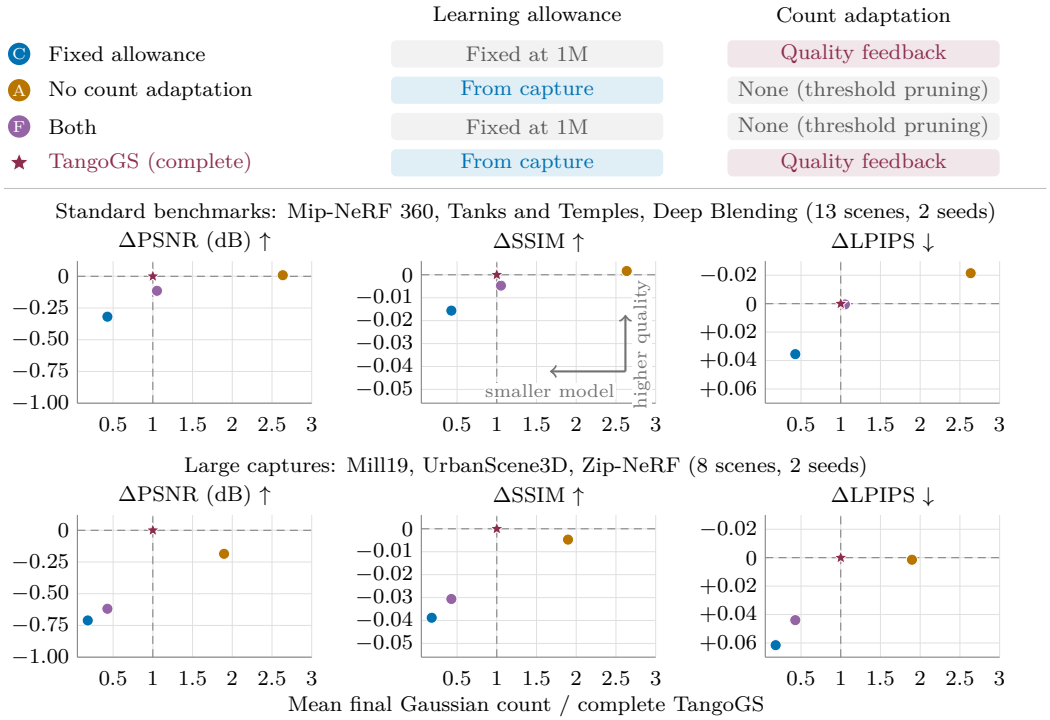


Figure 4: **Capture scale and adaptation ablation.** Points show quality changes and count ratios relative to complete TangoGS (star). Complete results: Supplementary §E.

Without adaptation, the allowance is fully spent: final counts are 2.6× TangoGS's on standard scenes and 1.9× on large captures (Fig. 4). On standard scenes, this lowers LPIPS by 0.021 without improving PSNR. Quality feedback depends on training PSNR alone, so it compresses a fixed 1M allowance as strongly as a capture-derived one. The components are therefore complementary: only the complete method reaches the highest PSNR of the combinations below the fully spent allowance's count in both regimes, at about half that count.

3.5 Quality Feedback Variation and Its Effect on Capacity

Figure 5 plots the keep ratio of every capture over training; since the ratio is a fixed function of training PSNR, the right axis reads back the PSNR behind each value, for example 31.8 dB for a ratio of 0.5, and the dotted line marks the 20 dB floor below which the final selection removes nothing. The terminal reading (dots) is taken after the last opacity decay, which briefly lowers training PSNR, so it sits above the last learning reading.

Scenes that end training above 33 dB (Kitchen, Bonsai, Room, Dr Johnson, Playroom) read ρ between 0.46 and 0.49 and keep 56% to 64% of the grown population at the terminal step. Those between 27 dB and 31 dB read 0.51 to 0.53 and keep 70% to 82%. Those below 27 dB, among them five of the eight large captures, read 0.54 to 0.58 and keep 82% to 97%; on Sci-Art and Residence

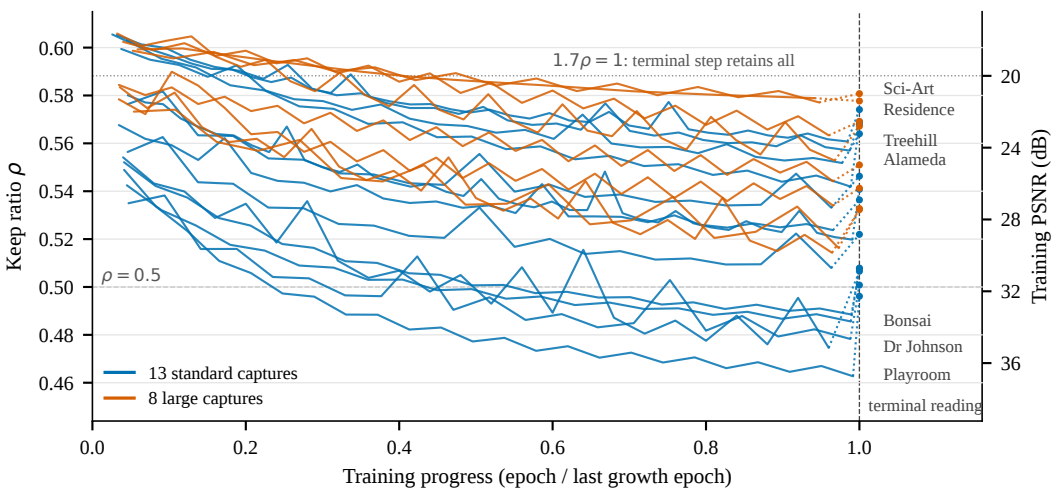


Figure 5: Keep ratio of every capture during training.

the retained fraction $s\rho$ reaches 0.98, so the terminal step removes almost nothing and the growth target alone sets the final size.

Whether these differences matter depends on where a capture sits. Holding $\rho = 0.5$ throughout changes Room and Dr Johnson by at most 4% in count and 0.08 dB, since their adaptive ratio already ends near 0.5. On Garden the adaptive ratio of 0.54 retains 1.29× the Gaussians for 0.15 dB higher PSNR and 0.014 lower LPIPS; on Building, Rubble and Sci-Art the ratios of 0.55 to 0.58 retain 1.45× to 1.66× for 0.13 dB to 0.16 dB and 0.006 to 0.019 lower LPIPS. The fixed 0.5 rule approximates the adaptive rule on the captures with the highest training PSNR and selects smaller, lower-quality models on the others.

3.6 Ablation: The Capture Statistic

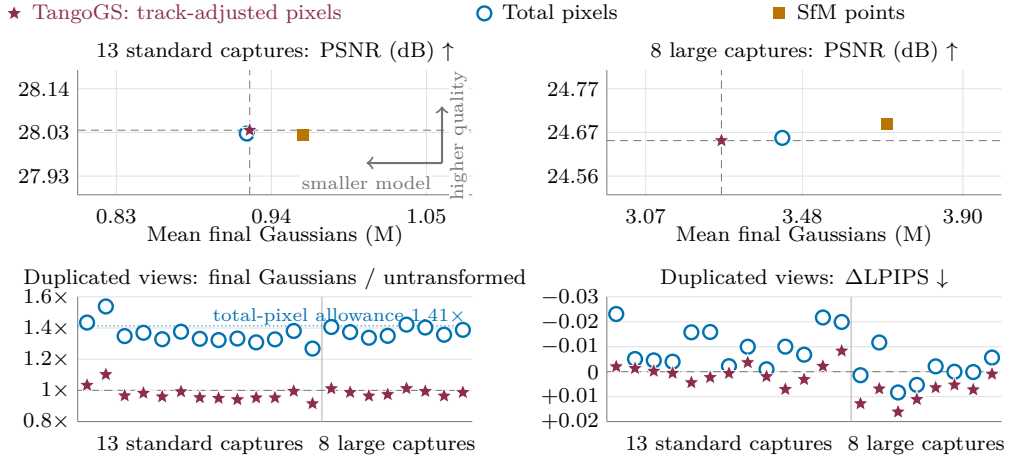


Figure 6: **Capture statistic ablation.** Top: standard and large-capture PSNR versus model size. Bottom: count and LPIPS changes after duplicating every view, held-out evaluation unchanged.

We compare three capture statistics for the learning allowance: track-adjusted pixels (ours), total pixels, and SfM point count, each normalized on the same 13 standard captures (Fig. 6). On the benchmarks all three reach the same mean quality, within 0.04 dB and 0.003 LPIPS, but SfM points spend more Gaussians for it: track-adjusted pixels use 4% fewer on standard scenes and 12% fewer on large captures, with total pixels in between. SfM-point allowances follow the matched-observation density per image pixel (Eq. (1)) rather than the capture itself: 1.3× to 1.7× ours on

aerial captures and $0.5\times$ to $0.8\times$ on Zip-NeRF interiors, with the larger model gaining at most 0.3 dB in each case (Supplementary §F).

Redundant views then separate the two pixel statistics. Duplicating every image adds no information but multiplies the total-pixel allowance by $\sqrt{2}$ and increases the mean final count by 36% to 38%. The track-adjusted allowance stays unchanged, with final counts within about 10% of the original. Both nevertheless lose large-capture PSNR when duplication halves the number of training epochs: size invariance does not ensure rendering invariance.

All three statistics reach similar mean quality, but SfM points allocate more Gaussians for it, and of the two pixel statistics only track adjustment keeps the allowance invariant to redundant views.

4 Related Work

We group prior density controllers by the three decisions TangoGS automates:

Choosing a Gaussian budget. Budgeted methods expose a target Gaussian count or cap. Taming 3DGS (Mallick et al., 2024) derives its target from the number of points reconstructed by SfM, with multipliers chosen by dataset class. LiteGS (Liao et al., 2025) grows toward a supplied target, 3DGS-MCMC (Kheradmand et al., 2024) samples under a preset cap, and Mini-Splatting (Fang & Wang, 2024) reorganizes Gaussians within a constrained count. TangoGS also uses an explicit learning allowance, computed before training from image count, resolution, and repeated observations of scene points using one global conversion.

Adapting model size during training. Compact 3DGS (Lee et al., 2024) and LP-3DGS (Zhang et al., 2024) learn retention masks; ControlGS (Zhang et al., 2025) balances uniform splitting and opacity regularization with one global parameter. DashGaussian (Chen et al., 2025) estimates budgets from observed growth and coordinates densification with image resolution; SIG (Xue et al., 2026) coordinates both using frequency-based measures of reconstruction progress. GS^2 (Yang et al., 2026) weighs reconstruction gains against model complexity to stop densification, with opacity-regularized pruning. RLGS (Li et al., 2025) learns to adjust learning rates and densification settings, with evaluation on budgeted backbones; Prune Wisely (Wang et al., 2026a) uses reconstruction loss to schedule pruning toward a preset target. Smart Target Point Control (Bisht & Kolb, 2026) tracks a supplied count along a prescribed training trajectory; BlitzGS (Wang et al., 2026b) combines importance-based pruning, a spawn gate, and density-control feedback. Our contribution is the policy coupling a capture-derived learning allowance to quality feedback during growth and final retention, using established budgeted growth and score-based pruning.

Choosing where to grow and prune. Local refinement guides growth using image errors (Rota Bulò et al., 2024; Baranowski et al., 2026), gradient magnitude and direction (Ye et al., 2024; Zhou et al., 2026), second-order information (Wang et al., 2025), or perceptual importance (Zhou & Ni, 2025). FastGS (Ren et al., 2026), CDC-GS (Dong et al., 2025), and LeGS (Ning et al., 2026) guide growth and pruning through cross-view errors, local image complexity relative to Gaussian density, and learned per-Gaussian actions, respectively. SPARE-GS (Chen et al., 2026) allocates regional growth and pruning quotas from a budget based on the current Gaussian count. TangoGS decides only how much to grow and retain, setting targets for LiteGS's budgeted growth and keep fractions for sensitivity pruning (Hanson et al., 2025).

5 Conclusion

Our central contribution is to distinguish capture scale from content complexity when selecting the capacity of a Gaussian scene representation. The extent and resolution of the observed surface inform how much capacity to make available before training; reconstruction quality provides feedback on how much to retain as the content is learned. Our experiments support treating these as complementary decisions: adaptation within an unsuitable allowance can restrict reconstruction quality, while setting capacity from capture scale alone can retain unnecessary Gaussians, reducing compactness. This motivates a design principle for density control across scales: use the capture to establish the capacity scale and learning feedback to adapt its use.

AI USE STATEMENT

In this work, we used generative AI tools to design and give feedback on methodology and experiments, to implement methods, to clean and reformat datasets, and to interpret results. Additionally, we used generative AI tools to generate tables and figures from experiment logs and to edit the text of this paper for readability. All AI-generated code was read and tested by the authors, every reported number is extracted from training and evaluation logs by scripts that are checked against the tables in this paper, and all experimental conclusions were examined and confirmed by the authors.

REFERENCES

Bartłomiej Baranowski, Stefano Esposito, Patricia Gschoßmann, Anpei Chen, and Andreas Geiger. Conegs: Error-guided densification using pixel cones for improved reconstruction with fewer primitives. In *International Conference on 3D Vision (3DV)*, pp. 989–999, 2026. doi: 10.1109/3DV69130.2026.00099. URL https://baranowskibrt.github.io/conegs/.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469, 2022. URL https://jonbarron.info/mipnerf360/.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19697–19705, 2023. URL https://jonbarron.info/zipnerf/.

Pratik Singh Bisht and Andreas Kolb. Smart target point control for gaussian splatting methods. *arXiv preprint arXiv:2605.16158*, 2026. URL https://arxiv.org/abs/2605.16158.

Youyu Chen, Junjun Jiang, Kui Jiang, Xiao Tang, Zhihao Li, Xianming Liu, and Yinyu Nie. Dashgaussian: Optimizing 3d gaussian splatting in 200 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11146–11155, 2025. URL https://arxiv.org/abs/2503.18402.

Zhang Chen, Shuai Wan, Fuzheng Yang, Jiazhi Xia, Weiyao Lin, and Junhui Hou. SPARE-GS: Structural parsimony and resource efficiency for 3D Gaussian splatting. arXiv preprint arXiv:2607.16624, 2026. URL https://arxiv.org/abs/2607.16624.

Zhemeng Dong, Junjun Jiang, Youyu Chen, Jiaxin Zhang, Kui Jiang, and Xianming Liu. Reframing gaussian splatting densification with complexity-density consistency of primitives. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://cdc-gs.github.io/.

Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision (ECCV)*, pp. 165–181, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/9866_ECCV_2024_paper.php.

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 Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 21537–21546, 6 2025. URL https://arxiv.org/abs/2412.00578.

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. URL https://repo-sam.inria.fr/fungraph/deep-blending/.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 7 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

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 (NeurIPS)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/93be245fce00a9bb2333c17ceae4b732-Paper-Conference.pdf.

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://doi.org/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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21719–21728, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Lee_Compact_3D_Gaussian_Representation_for_Radiance_Field_CVPR_2024_paper.html.

Zhan Li, Huangying Zhan, Changyang Li, Qingan Yan, and Yi Xu. Rlgs: Reinforcement learning-based adaptive hyperparameter tuning for gaussian splatting. arXiv preprint arXiv:2508.04078, 2025. URL https://arxiv.org/abs/2508.04078.

Kaimin Liao, Hua Wang, Zhi Chen, Luchao Wang, and Yaohua Tang. Litegs: A high-performance framework to train 3dgs in subminutes via system and algorithm codesign. arXiv preprint arXiv:2503.01199, 2025. URL https://arxiv.org/abs/2503.01199v3.

Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, reconstructing, and simulating: The UrbanScene3D dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/3218_ECCV_2022_paper.php.

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De la Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 2:1–2:11, 2024. doi: 10.1145/3680528.3687694. URL https://arxiv.org/abs/2406.15643.

Zhenhua Ning, Xin Li, Jun Yu, Guangming Lu, Yaowei Wang, and Wenjie Pei. Beyond heuristics: Learnable density control for 3d gaussian splatting. In *International Conference on Machine Learning (ICML)*, 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26094–26103, 2026. URL https://arxiv.org/abs/2511.04283.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in gaussian splatting. In *European Conference on Computer Vision (ECCV)*, pp. 347–362, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/8041_ECCV_2024_paper.php.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016. doi: 10.1109/CVPR.2016.445. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Schonberger_Structure-From-Motion_Revisited_CVPR_2016_paper.html.

Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable construction of large-scale NeRFs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Turki_Mega-NERF_Scalable_Construction_of_Large-Scale_NeRFs_for_Virtual_Fly-Throughs_CVPR_2022_paper.html.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a. URL https://arxiv.org/abs/2602.24136.

Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, and Rakesh Ranjan. Steepest descent density control for compact 3D Gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2505.05587.

Zhongtao Wang, Huishan Au, Yilong Li, Mai Su, Haojie Jin, Yisong Chen, Meng Gai, Fei Zhu, and Guoping Wang. BlitzGS: City-scale gaussian splatting at lightning speed. *arXiv preprint arXiv:2605.13794*, 2026b. URL https://arxiv.org/abs/2605.13794v2.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://ece.uwaterloo.ca/~z70wang/publications/ssim.pdf.

Weiyi Xue, Fan Lu, Chi Zhang, Tianhang Wang, Sanqing Qu, Zehan Zheng, Boyuan Zheng, Junqiao Zhao, and Guang Chen. Signal structure-aware Gaussian splatting for large-scale scene reconstruction. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/f67449c7ab72f441d3a713b046c6818c-Abstract-Conference.html.

Xianben Yang, Tao Wang, Yuxuan Li, Yi Jin, and Haibin Ling. Gs^2: Graph-based spatial distribution optimization for compact 3d gaussian splatting. arXiv preprint arXiv:2604.01884, 2026. URL https://arxiv.org/abs/2604.01884.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *ACM International Conference on Multimedia (ACM MM)*, pp. 1053–1061, 2024. URL https://doi.org/10.1145/3664647.3681361.

Fengdi Zhang, Yibao Sun, Hongkun Cao, and Ruqi Huang. Controlgs: Consistent structural compression control for deployment-aware gaussian splatting. arXiv preprint arXiv:2505.10473, 2025. URL https://arxiv.org/abs/2505.10473v3.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF 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.

Zhaoliang Zhang, Tianchen Song, Yongjae Lee, Li Yang, Cheng Peng, Rama Chellappa, and Deliang Fan. Lp-3dgs: Learning to prune 3d gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/dd51dbce305433cd60910dc5b0147be4-Abstract-Conference.html.

Hongbi Zhou and Zhangkai Ni. Perceptual-gs: Scene-adaptive perceptual densification for gaussian splatting. In *International Conference on Machine Learning (ICML)*, 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 (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/24402f5a0df4ab461aa759ca2ecb8df6-Abstract-Conference.html.