Provable Pruning for Efficient 3D Gaussian Splatting via Coresets

Waseem Mousa **Alaa Maalouf**
University of Haifa

Abstract

3D Gaussian Splatting (3DGS) enables high-quality real-time novel-view synthesis, but practical scenes often contain millions of Gaussians, making compression essential for deployment on limited hardware. Existing reduction methods are effective but mostly heuristic: they provide no multiplicative approximation guarantee for the rendered objective, and thus rely heavily on costly post-pruning finetuning to recover quality. We ask a basic question: *can a 3DGS scene be provably replaced by a much smaller weighted subset (coreset) while preserving the objective of interest?* We first show that, in the unrestricted setting, no non-trivial multiplicative 3DGS coreset exists. We then show that multiplicative guarantees are not impossible, but resolution-dependent. For a prescribed rendering resolution, such as representative views or grids of views/rays, we provide the first weighted coreset construction theorem for 3DGS. The construction samples Gaussians by sensitivity: provable importance scores measuring each Gaussian's role in the full-scene objective. Finally, under explicit validity and log-transmittance stability assumptions, we turn this objective guarantee into a rendering guarantee. Empirically, our method is strongest where deployment needs it most: aggressive compression with no or minimal recovery compute. In prune-only and very short finetuning regimes, it achieves state-of-the-art performance, showing that principled importance estimation can be both theoretically meaningful and practically useful.

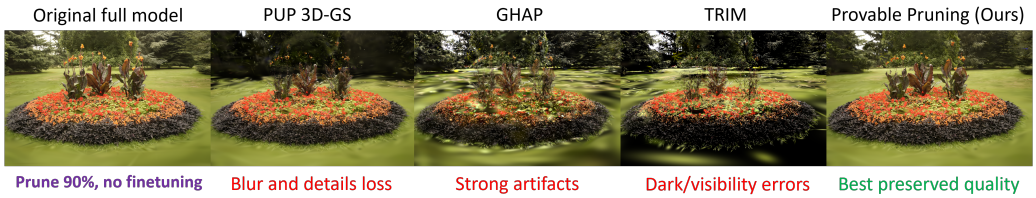

Figure 1: Our method prunes a pretrained 3D Gaussian scene while provably preserving the quality on a set of queries. In the regime where no finetuning is used, our approach preserves rendering quality better than competing pruning heuristics, especially at aggressive pruning ratios.

1 Introduction

3D Gaussian Splatting (3DGS) has become a central representation for real-time novel-view synthesis, combining explicit Gaussian scene models with visibility-aware differentiable splatting and strong fidelity–speed tradeoffs [1–8]. Yet, high-quality reconstructions often require large Gaussian sets, creating significant memory, storage, bandwidth, and runtime costs, especially on resource-limited hardware. This has motivated growing work on 3DGS pruning and global compaction [9–22].

The gap. While these methods demonstrate that strong compression is often possible, they leave two key bottlenecks. First, the compressed Gaussian set is often followed by a costly recovery stage, requiring substantial hardware resources and many fine-tuning iterations. Second, existing approaches are largely heuristic, providing no provable guarantees on approximation error or subset size.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Our approach: Coresets for 3DGS. To this end, we study 3DGS pruning through the lens of coresets: small weighted subsets that approximately preserve a target objective over a prescribed family of queries. This leads to our central question: *Can a full 3DGS scene be provably replaced by a much smaller weighted subset of the gaussian while preserving the target rendering objective, and under what assumptions is such a guarantee possible?* This question is especially important in low-compute settings. When only a prune-only evaluation or a very short recovery phase is feasible, the selected subset directly determines performance. A principled compression method should therefore identify which Gaussians are most important for preserving the rendered objective, while providing explicit control over the approximation error, not requiring intense fine-tuning steps.

The challenge. While coreset construction frameworks provide such guarantees for many objectives [23–26], including subset-selection approaches for neural-network compression [27–29], the main obstacle in 3DGS is visibility: front-to-back compositing couples Gaussians through transmittance, so changing or removing one early Gaussian can affect the contribution of many later Gaussians. We therefore ask whether 3DGS admits any non-trivial multiplicative coreset guarantee at all, and if not in full generality, what restrictions recover a useful theorem in practice.

A glimpse into our findings. Our answer is twofold: In the unrestricted setting, no general multiplicative coreset theorem is possible: if the query family is rich enough to isolate individual Gaussians, then every Gaussian can become indispensable. However, this unrestricted view is stronger than what is needed for rendering. In practice, approximation is required only over a finite set of queries (views) induced by the desired rendering resolution, e.g., desired grid/set of views or rays. Under this finite-query formulation, together with a fixed full-scene objective, frozen itemwise contributions, and stable transmittance/occlusion, we prove a weighted coreset construction theorem. The resulting Gaussian subset size scales only logarithmically with the number of representative queries/resolution. The construction is based on assigning each Gaussian a sensitivity scores, which quantify the importance of it with respect to the objective and guide the sampling procedure. Thus, our main theoretical message is: *non-trivial 3DGS coresets are impossible in full generality, but provable pruning is possible under practically meaningful restrictions.*

Our Contribution. This work develops a theory-guided framework for 3DGS compression/pruning and evaluates it in low-compute deployment regimes. Our main contributions are:

1. **Impossibility in full generality.** We prove that no non-trivial multiplicative coreset guarantee can hold for unrestricted 3DGS rendering: when the query family is rich enough to isolate individual Gaussians, every Gaussian may become necessary; see theorem 1.
2. **Resolution-dependent provable 3DGS pruning (coreset).** First, for a desired rendering resolution (a given set or a grid of views/rays), and a fixed full-scene reference decomposition, we provide the first provable pruning (coreset construction) algorithm for 3DGS; the remaining Gaussian subset size scales logarithmically by the size of the desired views set (resolution); see section 3.2 and theorem 2. Then, we prove that transfer from a fixed-objective guarantee for true 3DGS re-rendering is possible under explicit validity and log-transmittance stability assumptions; see section 3.3 and theorem 3.
3. **Extension beyond finite resolutions (query sets).** We extend the representative-query theorem from finite query families to compact query regions using representative covers and Lipschitz control; see section 3.4 and theorem 4.

Extensive experiments indicate that our theoretically inspired method delivers strong performance across the evaluated settings, with particularly clear gains in prune-only/no-finetuning evaluation and very short finetuning budgets, where costly recovery is not feasible; see sections 3.5 and 4.

2 Related Work

3D Gaussian Splatting. 3DGS introduced an explicit radiance-field representation built from anisotropic Gaussians, interleaved optimization and density control, and visibility-aware splatting, enabling fast training and real-time rendering [1]. A large follow-up literature refined this framework in several directions, including anti-aliasing and scale robustness [2, 5], structured or view-adaptive parameterizations [3], stronger geometric fidelity [4], and alternative optimization or densification dynamics [6, 30, 31]. Broader surveys document how quickly 3DGS has expanded across optimization, systems, and applications [7, 8]. For our purposes, the key point is that once 3DGS became a strong rendering primitive, Gaussian-set size and cost became central research problems [9, 10].

3DGS compaction and pruning. A growing literature targets the memory, storage, and runtime costs of large 3d Scenes, including budgeted training or densification methods [9, 10]. Others compress pretrained models through pruning, compact parameterizations, or global reduction [11–18]. A third group emphasizes coding and storage efficiency through progressive or quantized representations [19–21]. Compression surveys now provide a broader taxonomy of this landscape [8, 22]. Pruning [15, 16] is especially attractive because it is post hoc and representation-preserving: it outputs a smaller standard 3DGS scene that can be rendered by existing pipelines, without specialized runtime libraries or separate decoding, while directly reducing memory and rendering cost.

Coresets and provable pruning. Coresets are small weighted subsets that approximate an objective over a query family [32–34]. Coresets have been applied to many problems, including regression, subspace approximation, k-means, and even neural-network training [35–40]. Sensitivity sampling and related frameworks are a standard route to obtaining such coresets [23, 25, 41, 42]. Specifically, provable subset-selection results were developed for neural-network pruning, including data-dependent and data-independent neuron pruning [27–29, 43]. These results are closest in spirit to our goal, but they assume objectives with a fixed itemwise decomposition once the input is fixed.

3 3DGS Pruning via Coresets

3DGS scene. 3DGS represents a scene by a set of Gaussians. Let $N > 0$ be an integer, a 3DGS scene is a set $G = \{g_1, \ldots, g_N\}$ of N Gaussians, where for every $i \in [N]$: $g_i = (\mu_i, \Sigma_i, \alpha_i, c_i)$ such that $\mu_i \in \mathbb{R}^3$ is the mean, $\Sigma_i \in \mathbb{R}^{3\times 3}$ is the positive definite covariance matrix, $\alpha_i \in [0, 1]$ is the opacity, and c_i is the view-dependent intensity of a color channel.

Our goal. For a given query (camera pose and a pixel to render), to render the desired pixel, Gaussians are composited through front-to-back order: each Gaussian contributes according to its projected footprint, color, and opacity, scaled by the transmittance left by earlier Gaussians. We seek a multiplicative approximation guarantee over a family of rendering queries. To mathematically formulate this in a standard relative-error form, each query (we wish to approximate) specifies the camera parameters (pose) together with a specific output channel in a pixel to approximate, producing a nonnegative *scalar* objective for one RGB channel. Thus, a guarantee over all queries implies approximation for all selected views, pixels, and channels, and therefore for the full RGB render.

Rendering query. Formally, let $\Lambda = \{R, G, B\}$ denote the output channels. A rendering query is $q = (C, x, \lambda) \in \mathcal{Q}$, where $C = (R, t)$ are the camera extrinsics[1] with $R \in SO(3)$ and $t \in \mathbb{R}^3$, defining the world-to-camera map $p \mapsto Rp + t$, x is an image-plane location (a pixel), and $\lambda \in \Lambda$ is an output channel. Thus, a query q corresponds to one color channel of one pixel from a camera pose.

Scalar rendering function. For $g_i \in G$, let $k(g_i, q)$ be the projected Gaussian-kernel value of Gaussian g_i at pixel x under the camera pose C, let $\rho(g_i, q) := \alpha_i k(g_i, q)$ be its attenuation factor, and let $\prec_{G,q}$ denote the front-to-back order induced by the view depths of the Gaussians in G under pose C. The full-scene prefix transmittance of Gaussian g_i is $T(G, g_i, q) := \prod_{j \prec_{G,q} i}(1 - \rho(g_j, q))$, and its full-scene contribution is $a(G, g_i, q) := c_i(q)\rho(g_i, q)T(G, g_i, q)$. The resulting per-channel scalar rendered value of scene G is $A(G, q) := \sum_{g_i \in G} a(G, g_i, q)$. A note is given here[2].

[1] For simplicity, we assume fixed camera intrinsics and fixed image domain throughout, and omit them from the notation.

[2] Note that the quantities $k(g_i, q)$, $\rho(g_i, q)$, $T(G, g_i, q)$, and the ordering $\prec_{G,q}$, depend only on the camera extrinsics C, the pixel x, and the Gaussian model G, and are invariant to the output channel λ.

A pruned 3DGS scene. A weighted pruned scene is represented by a vector $w \in [0,\infty)^N$, where $w_i = 0$ removes Gaussian g_i and $w_i > 0$ reweights it. $\|w\|_0$ is the number of nonzero entries of w (number of retained Gaussians). Its true reduced-scene render is $A_w(G,q) := \sum_{g_i \in G} w_i c_i(q) \rho(g_i,q) T_w(G,g_i,q)$, where $T_w(G,g_i,q) := \prod_{j \prec_{G,q} i}(1 - w_j \rho(g_j,q))$. Note that, in this paper, a 3DGS coreset is a weighted pruned scene whose induced objective approximates the full-scene objective over a family of queries/views.

Definition 1 (Multiplicative ε-coreset for 3DGS)**.** Let $N > 0$, $G = \{g_1, \ldots, g_N\}$ be a 3DGS scene, and let $\mathcal{U} \subseteq \mathcal{Q}$ be a family of rendering queries. Let $w \in [0,\infty)^N$ represent a weighted pruned scene. Recall A, the nonnegative scalar objective of the full scene (per-channel scalar rendering function), and let A_w be the corresponding objective induced by w. We call w an *ε-coreset* for G on $\mathcal{U}$ if $|A(G,q) - A_w(G,q)| \le \varepsilon A(G,q)$, for every $q \in \mathcal{U}$.

Algorithm 1 Sensitivity-based 3DGS coreset pruning

Require: Pretrained 3DGS scene $G = \{g_i\}_{i=1}^n$, and a budget (pruned model size) m
Ensure: Pruned weighted 3DGS scene G'

1: Build finite queries $\mathcal{Q}'$ from representative cameras, pixels/tiles, and RGB channels.
2: For each $g_i \in G$ and $q \in \mathcal{Q}'$, compute $\mathrm{imp}(g_i,q) := a(G,g_i,q)/A(G,q)$.
3: Set $s(g_i) := \max_{q \in \mathcal{Q}'} \mathrm{imp}(g_i,q), \quad S := \sum_{j=1}^n s(g_j), \quad p(g_i) := s(g_i)/S$.
4: Sample m Gaussians i.i.d. from G according to p; let N_i be the number of times g_i is sampled.
5: Assign weight $w_i := N_i/(m\,p(g_i))$ for every sampled Gaussian; $w = (w_1, \cdots, w_n)$
6: **return** w

3.1 The Unrestricted Case: No General 3DGS Coreset

We first ask whether unrestricted 3DGS rendering admits a non-trivial multiplicative coreset guarantee.

Theorem 1 (General-case impossibility)**.** *For every $N \ge 1$ and every $0 < \varepsilon < 1$, there exist a 3DGS scene $G = \{g_1, \ldots, g_N\}$ and a query family $\mathcal{Q}$ such that no weighted reduced scene specified $w \in [0,\infty)^N$ with $\|w\|_0 < N$ is an ε-coreset for the true rendered objective A_w on $\mathcal{Q}$.*

Proof overview. We construct a counterexample. Let $G = \{g_1, \ldots, g_N\}$ be any scene for which the query family is rich enough to isolate each Gaussian individually: for every $i \in [N]$, there exists a query $q_i \in \mathcal{Q}$ such that Gaussian g_i is the unique positive contributor at q_i. Then $A(G,q_i) = a(G,g_i,q_i) > 0$ and all other Gaussians contribute zero at q_i. Now let $w \in [0,\infty)^N$ satisfy $\|w\|_0 < N$. Some index $i \in [N]$ must satisfy $w_i = 0$. At the corresponding isolating query q_i, the reduced scene therefore satisfies $A_w(G,q_i) = 0$, while the full scene satisfies $A(G,q_i) > 0$. Hence $|A(G,q_i) - A_w(G,q_i)| = A(G,q_i)$, so the relative error at q_i is exactly 1. Therefore w cannot be an ε-coreset for any $0 < \varepsilon < 1$. The full proof, including an explicit footprint-isolation construction for finite-support rasterized 3DGS splats, is given in appendix J. □

The interpretation: if the query family is rich enough to isolate individual Gaussians, then every Gaussian is indispensable. A coreset theorem must restrict the query family, the objective, or both.

3.2 Coresets for 3DGS Under Representative Queries

The unrestricted impossibility result shows that non-trivial guarantees require structure. We therefore restrict attention to a finite representative query family induced by the target views together with a nonnegative scalar objective and a stability condition on prefix transmittance perturbations. We then define the sampling rule and state the fixed-objective and rendered guarantees.

Setting: Finite query set Let $\mathcal{Q}' \subseteq \mathcal{Q}$ be a finite family of representative queries: guarantees are required over a controlled set of camera poses and image samples, not over a continuum of all possible cameras and rays. In 3DGS terms, $\mathcal{Q}'$ defines the view-space discretization of the guarantee: its density determines how finely the approximation covers camera translations and rotations, while its image samples define the pixels, tiles, or channels being scored.

Our approach: Gaussian sensitivity. We seek to quantify the importance of each Gaussian. Intuitively, a Gaussian may contribute very differently across queries: it may be highly influential for some views, rays, or pixels, and nearly irrelevant for others. We therefore first define the contribution of each Gaussian to each query. We then define its global importance, or sensitivity, as the maximum relative contribution over all queries. This identifies Gaussians whose removal could cause large approximation errors, and distinguishes them from Gaussians that are consistently negligible. First, we disregard queries where $A(G, q) = 0$ since no Gaussian is required for them. For each Gaussian $g_i \in G$ and each query $q \in \mathcal{Q}'$, we introduce the *importance* of g_i with respect to q: $imp(g_i, q) := \frac{a(G,g_i,q)}{A(G,q)}$, We then define the Gaussian's *sensitivity* score as:

$$s(g_i) := \max_{q \in \mathcal{Q}'} imp(g_i, q) = \max_{q \in \mathcal{Q}'} a(G, g_i, q)/A(G, q), \tag{1}$$

and the total sensitivity of the scene is $S := \sum_{g_i \in G} s(g_i)$. Thus $s(g_i)$ is the largest normalized contribution of Gaussian i over the representative family.

Pruning pipeline: 3DGS Coreset via Sensitivity sampling. For every Gaussian g_i, define $p(g_i) := s(g_i)/S$. For a target sample size m, sample (i.i.d) m Gaussian $G' = \{g'_1, \ldots, g'_m\}$ from the scene G, where each g'_i is sampled with probability $p(g'_i)$. Define n_i to be the number of times g_i was sampled. The output weight vector $w \in [0, \infty)^N$ is then $w_i := \dfrac{n_i}{mp(g_i)}$. Equivalently, each sampled occurrence of Gaussian g_i contributes an inverse-probability weight $(mp(g_i))^{-1}$. The resulting weighted reduced scene is supported on at most m Gaussians and is unbiased for the fixed-objective relaxation on every fixed query. Algorithm 1 summarizes the pipeline.

Relaxed objective. For the coreset theorem, we also introduce the auxiliary fixed-objective relaxation $A^{\mathrm{th}}_w(G, q) := \sum_{g_i \in G} w_i a(G, g_i, q)$. Here each itemwise term $a(G, g_i, q)$ is frozen under the transmittance of the full scene G. This is a deliberate theoretical relaxation: once the subset changes, the true rendered objective A_w no longer decomposes into fixed per-Gaussian terms, whereas A^{th}_w does. The later rendered theorem shows how a guarantee for A^{th}_w can be transferred back to A_w under explicit transmittance-stability assumptions.

Theorem 2 (Query size dependent 3DGS coreset)**.** *Let G be a 3DGS scene, and let $\mathcal{Q}'$ be a desired finite set of queries. Let $\varepsilon_c, \delta \in (0, 1)$, and let w be the weight vector produced by the pruning pipeline above (e.g., the output of a call to Alg. 1 with G and m as inputs). If $m \geq \frac{3S}{\varepsilon_c^2} \log\left(\frac{2|\mathcal{Q}'|}{\delta}\right)$, then, with probability at least $1 - \delta$, w is an ε_c-coreset for the fixed-objective relaxation A^{th} on $\mathcal{Q}'$, i.e., $\left|A(G, q) - A^{\mathrm{th}}_w(G, q)\right| \leq \varepsilon_c A(G, q)$ for all $q \in \mathcal{Q}'$. Moreover, inherently $\|w\|_0 \leq m$.*

Proof overview. For any fixed $q \in \mathcal{Q}'$, the estimator $A^{\mathrm{th}}_w(G, q)$ is an average of i.i.d. nonnegative random variables with mean $A(G, q)$. The sensitivity bound implies that each sampled term is at most $SA(G, q)$ after normalization. A standard multiplicative concentration bound therefore gives a relative-error guarantee for one query, and a union bound over the finite family $\mathcal{Q}'$ yields the simultaneous statement. The full proof is given in appendix K. □

The fixed-objective theorem is not yet a theorem about true re-rendering. The reason is structural: once the reduced scene changes, the prefix transmittances change as well, so later Gaussian contributions are no longer fixed functions of the full scene.

3.3 Transfer to true rendering under mild transmittance stability

To transfer the fixed-objective guarantee back to true rendering, we add one more ingredient: a stability requirement saying that the reduced scene preserves the relevant prefix transmittances up to controlled multiplicative distortion.

Definition 2 (Weighted rendering validity)**.** A weight vector $w \in [0, \infty)^N$ is called *valid* on the query set $\mathcal{Q}'$ if $0 \leq w_i \rho(g_i, q) < 1$ for every $q \in \mathcal{Q}'$ and $i \in [N]$. This ensures that every factor in $T_w(G, g_i, q)$ is nonnegative, so the reduced-scene transmittances are well-defined.

Assumption 1 (Log-transmittance stability)**.** There exists $\gamma \geq 0$ such that for every $q \in \mathcal{Q}'$ and $i \in [N]$ with $w_i > 0$, one has $|\log T(G, g_i, q) - \log T_w(G, g_i, q)| \leq \gamma$. This assumption captures the regime in which pruning and reweighting do not drastically alter the occlusion and transmittance structure along the representative queries.

Theorem 3 (Finite-query rendered coreset under mild assumptions)**.** *Let w be the weight vector produced by the sensitivity-based sampling rule, and suppose that w is valid on $\mathcal{Q}'$ and satisfies Assumption 1. Under the same sample-size condition as in theorem 2, with probability at least $1 - \delta$, w is an ε_r-coreset for the true rendered objective A on $\mathcal{Q}'$, where $\varepsilon_r := \max\Big\{1 - e^{-\gamma}(1 - \varepsilon_c),\ e^{\gamma}(1 + \varepsilon_c) - 1\Big\}$. Equivalently, $|A(G, q) - A_w(G, q)| \leq \varepsilon_r A(G, q)$ for all $q \in \mathcal{Q}'$.*

Proof overview. Theorem 2 controls $A_w^{\mathrm{th}}(G, q)$ relative to $A(G, q)$. The log-transmittance stability assumption then implies that, for each retained Gaussian and representative query, the reduced-scene transmittance differs from the full-scene transmittance by at most a factor $e^{\pm\gamma}$. Summing these termwise bounds yields a multiplicative sandwich between $A_w(G, q)$ and $A_w^{\mathrm{th}}(G, q)$, and combining the two estimates gives the rendered guarantee. The full proof is given in appendix L. □

The complementary per-query and qualifying-subset forms, and their distinction from empirical coverage diagnostics, are given in appendices D and L.1.

3.4 Compact-region extension

The finite-query theorem controls a prescribed representative $\mathcal{Q}'$. It is useful to ask when such a guarantee transfers from finitely many representatives to a continuous region of nearby queries. We state a standard representative-cover extension: if every query in a compact region is close to some representative query, and both the full and reduced rendered objectives vary smoothly inside the region, then the guarantee extends to the whole region with an additional cover-radius error.

Assumption 2 (Compact nondegenerate query region)**.** Let $\mathcal{Q}_r \subseteq \mathcal{Q}$ be a compact query region equipped with a metric $d_{\mathcal{Q}}$ (think of $\mathcal{Q}_r$ as a local region of camera poses, pixels, and channels around a representative set). On $\mathcal{Q}_r$, assume: (i) $\mathcal{Q}_r$ is compact, (ii) the full rendered objective $q \mapsto A(G, q)$ is Lipschitz on $\mathcal{Q}_r$ with constant L_{full}, (iii) the reduced rendered objective $q \mapsto A_w(G, q)$ is Lipschitz on $\mathcal{Q}_r$ with constant L_w, and (iv) the full rendered objective is bounded away from zero: $A(G, q) \geq A_{\min} > 0$, for every $q \in \mathcal{Q}_r$.

Definition 3 (ρ-net)**.** A finite set $\mathcal{Q}_r' \subseteq \mathcal{Q}_r$ is a ρ-net of $\mathcal{Q}_r$ if for every $q \in \mathcal{Q}_r$ there exists $q' \in \mathcal{Q}_r'$ such that $d_{\mathcal{Q}}(q, q') \leq \rho$.

The Lipschitz requirement is a local regularity condition. It is expected to hold on regions where the active set of Gaussians is finite, the front-to-back order does not change discontinuously, and the camera-to-ray map, color functions, and projected kernel values vary smoothly. In other words, the region should not cross a sharp visibility or ordering discontinuity. The appendix gives the formal smoothness-to-Lipschitz argument; see appendix M.

Theorem 4 (Compact-region extension from representatives)**.** *Assume Assumption 2. Let $\mathcal{Q}_r' \subseteq \mathcal{Q}_r$ be a finite ρ-net. Suppose that the reduced scene satisfies the multiplicative guarantee on the representative set: $|A(G, q') - A_w(G, q')| \leq \varepsilon_0 A(G, q')$, for every $q' \in \mathcal{Q}_r'$. Then for every $q \in \mathcal{Q}_r$, $|A(G, q) - A_w(G, q)| \leq \varepsilon_{\mathrm{reg}} A(G, q)$, where $\varepsilon_{\mathrm{reg}} := \varepsilon_0 + \frac{\big((1+\varepsilon_0)L_{\mathrm{full}} + L_w\big)\rho}{A_{\min}}$.*

Proof overview. For any query $q \in \mathcal{Q}_r$, choose a representative $q' \in \mathcal{Q}_r'$ within distance ρ. The error at q is decomposed into three terms: the change of the full render $A(G, \cdot)$ from q to q', the representative-set approximation error at q', and the change of the reduced render $A_w(G, \cdot)$ from q' back to q. The two change terms are controlled by the Lipschitz constants, while the middle term is controlled by the representative-set coreset guarantee. The lower bound $A(G, q) \geq A_{\min}$ converts the resulting additive cover-radius term into relative error. The full proof is given in appendix M. □

When theorem 3 holds on $\mathcal{Q}_r'$, we may take $\mathcal{Q}' = \mathcal{Q}_r'$ and instantiate ε_0 by the finite-query rendered coreset error ε_r. Thus, theorem 4 extends the finite rendered guarantee from a representative query set to a compact region, at the cost of an additional term proportional to the cover radius ρ.

Regional and distance-dependent forms, including the positive-signal scope, are discussed in appendices D and M.

3.5 Sensitivity at Multiple Resolutions

The finite-query coreset theorem is stated for per-channel objectives, but the same argument applies to any finite collection of nonnegative objectives obtained by aggregating these channel contributions. In particular, one may define the query space at different rendering resolutions: individual color channels, pixels, tiles, or scenes. Each choice simply changes what is treated as a single query objective before applying the same sensitivity-sampling argument. The Chernoff analysis is unchanged: the sampled random variables now correspond to the chosen aggregated objective rather than to a single color channel, and the resulting guarantee preserves that objective over the finite query family.

Per-pixel construction. This variant treats each camera and image-plane location as a query, aggregates the RGB-channel contributions of each Gaussian into a single nonnegative pixel-level objective (luminance or total color contribution); Sensitivities are computed based on this objective. Rather than preserving each channel separately, the coreset preserves the aggregated pixel objective.

Per-tile and Per-scene construction. The per-tile variant further coarsens the query resolution. Each rendered image is partitioned into non-overlapping tiles, for example matching the tile granularity used by the 3DGS rasterization kernel. For each tile, a Gaussian's contribution is accumulated over the pixels and channels inside that tile, and sensitivities are computed with respect to the resulting tile-level objective. This gives a finite-query coreset guarantee for tile-aggregated rendering objectives.

Table 1: Per-dataset average prune-only results at prune ratios 0.90 and 0.99. Rows are prune ratios and methods; columns are grouped by dataset and metric. Bold indicates best per dataset/metric/ratio; colored cells mark the top three methods.

Prune	Method	Mip-NeRF 360			Deep Blending			Tanks & Temples		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
0.90	GHAP	16.76	0.458	0.494	20.53	0.744	0.437	14.83	0.488	0.483
	PUP	15.79	0.535	0.429	19.52	0.765	0.393	13.38	0.577	**0.406**
	Trim	14.45	0.413	0.480	16.52	0.642	0.429	12.46	0.471	0.458
	Unif	13.16	0.391	0.516	15.69	0.654	0.501	10.22	0.437	0.482
	Ours	**18.16**	**0.580**	**0.417**	**22.16**	**0.790**	**0.390**	**16.35**	**0.599**	0.407
0.99	GHAP	10.19	0.177	0.664	10.14	0.401	0.653	7.94	0.224	0.675
	PUP	10.67	0.227	0.624	10.13	0.428	0.621	8.94	0.320	0.627
	Trim	9.58	0.136	0.661	7.66	0.132	0.683	6.23	0.104	0.668
	Unif	10.27	0.174	0.642	7.84	0.189	0.696	6.92	0.163	0.658
	Ours	**13.97**	**0.396**	**0.600**	**15.62**	**0.660**	**0.526**	**11.98**	**0.435**	**0.606**

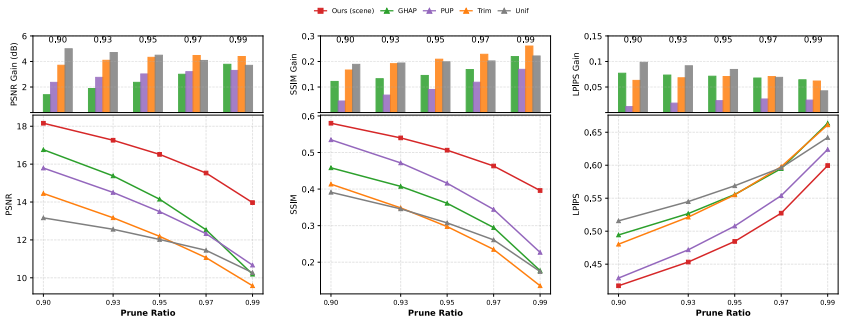

Figure 2: Combined prune-only curves and deltas for mip-NeRF 360: prune-only deltas (top row), and the prune-only curves (bottom row). For prune-only results, our method is significantly stronger than the competing methods across all metrics.

The per-scene variant uses the coarsest aggregation level. Here, each Gaussian's contribution is accumulated over the entire representative rendering set: all selected views, pixels, and channels. Sensitivities are therefore computed with respect to the full scene-level objective used for scoring. This is the variant used in our experiments.

Additional variants. The coreset framework motivates sensitivity-based selection, but the exact score aggregation can be engineered for practical robustness. In addition to the core color-aware L_1-style contribution score, our ablations evaluate L_2 variants, namely computing the sensitivities based on $a^{L_2}(G, g_i, q) := c_i^2(q)\rho^2(g_i, q)T^2(G, g_i, q)$, which emphasize large contribution discrepancies, and nocolor variants, namely $a^{nc}(G, g_i, q) := \rho(g_i, q)T(G, g_i, q)$, which remove the color factor to focus the score more strongly on geometry, opacity, visibility, and coverage. A^{L_2} and A^{nc} are also defined accordingly. These variants are empirical engineering extensions of the core method: they preserve the same sensitivity-driven selection pipeline.

4 Experimental Results

Our experimental study targets scarce-compute deployment: aggressive compression at high prune ratios, prune-only evaluation, and short finetuning budgets where long post-pruning recovery is not affordable. The results report how much fidelity each reduction rule preserves immediately after pruning and how the reduced models change under small recovery budgets. The prune-only setting corresponds to zero post-pruning optimization. The short-recovery setting uses low finetuning budgets of 10, 50, 100, and 200 iterations. This emphasizes the regime in which the retained subset must already be strong, rather than relying on a long retraining phase. Experiments were conducted on a PC equipped with one NVIDIA A100-SXM4-40GB GPU and AMD EPYC 7742 64-Core CPUs.

Datasets. We use the same 13 scenes used by the 3DGS work [1] and other competing methods: Nine scenes from the Mip-NeRF 360 dataset [44], two scenes (truck and train) from the Tanks & Temples dataset [45], and two scenes (drjohnson and playroom) from the Deep Blending dataset [46].

Reported results. We report PSNR, SSIM, and LPIPS as the primary image-quality metrics. We also report the pruned-Gaussian ratio and the post-pruning optimization budget where applicable.

Competing methods are: (1) GHAP: Gaussian Herding across Pens [17]; (2) PUP 3D-GS: Principled Uncertainty Pruning for 3D Gaussian Splatting [16]; (3) Trimming the Fat [15]; (4) Uniform sampling is used as a query-agnostic baseline. Our method is denoted *Ours*. Unless stated otherwise, in all experiments, we use the per-scene sensitivity variant. We also evaluate all variants later in this section.

Prune-only results at aggressive compression. We evaluate prune-only fidelity at the strongest compression levels where at most 20% of the Gaussians are retained. As shown in table 1, for prune ratios 0.9 and 0.99, our method variants consistently outperform the competing methods across all test datasets, achieving the best PSNR, SSIM, and LPIPS scores in the prune-only setting. The same trend persists across the remaining compression ratios: $\{0.80, 0.85, 0.90\}$ and $\{0.95, 0.97, 0.99\}$ which are reported, respectively, in table 26 and table 27 in the appendix due to space.

Prune-only curves and delta trends. figure 2 sweeps the high prune-ratio regime on mip-NeRF 360 and reports PSNR, SSIM, and LPIPS. The upper panels report delta plots that isolate the gap between our method and each competing method: positive deltas indicate an advantage for our method, with PSNR and SSIM computed as ours minus competitor, and LPIPS computed as competitor minus ours. It clearly shows how each method degrades as compression becomes more aggressive. While our method already shows a large advantage, this gap widens at higher prune ratios for PSNR and SSIM, and remains consistently strong for LPIPS. The only exception is the delta to uniform pruning, whose gap can shrink at the highest prune ratios because uniform pruning already starts from very poor fidelity. The same trends are observed on all of the other datasets in figure 25 of the appendix.

Short-recovery results. The short-recovery experiments separate two effects: the fidelity preserved immediately after pruning, and the amount each method can recover under a small finetuning budget. Matched comparisons against competing methods at prune ratios 0.95, 0.97, and 0.99 are shown in figure 3 for mip-NeRF 360. Our methods maintain a clear advantage throughout the low-budget recovery regime, largely because they start from a stronger reduced model. The gap is most pronounced at shorter finetuning budgets and higher prune ratios, precisely where recovery compute is most limited. The main exception is GHAP, which begins to approach our performance after 200 finetuning iterations. This is expected, since GHAP is a global compaction method rather than standard pruning: by merging and adapting Gaussians, it has more opportunity to recover quality during finetuning. However, this also makes it less suitable for the strict prune-only or very-short-recovery setting, where the goal is to obtain an immediately strong vanilla-compatible subset with minimal post-processing. The same phenomenon is observed on the other tested datasets; see figures 30 and 31.

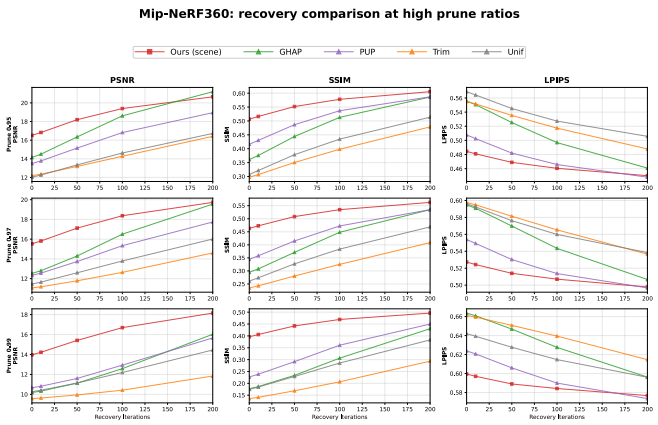

Figure 3: Short-recovery comparison against other methods on mip-NeRF 360 at prune ratios 0.95, 0.97, and 0.99.

Ablations: Effect of sensitivity resolution and engineering variant. We compare the different sensitivity resolution variants, varying the pruning ratio and reporting PSNR, LPIPS, and SSIM, both in the prune-only setting (figure 5) and after the short recovery stage (figure 4). We evaluate the per-channel, per-pixel, per-tile, and per-scene variants introduced in Section 3.5, together with simple engineering variants of the scene-level score, such as removing the color term or using an ℓ_2 aggregation instead of ℓ_1. Across both settings, the same trend is observed: per-scene sensitivity and its engineered variant perform best, followed by per-tile, per-pixel, and per-channel scoring. This suggests that aggregating contributions over larger rendering objectives yields a more stable importance signal, especially for identifying Gaussians that are useful across many pixels and views rather than only for isolated channel-level queries. The fact that the trend persists after recovery further shows that the aggregated sensitivities provide a better initialization for finetuning, not only better immediate pruning. Finally, the strong performance of the engineered variant indicates that the sensitivity framework is robust but not saturated: the theoretical principle already gives high-quality subsets, while simple implementation choices in the aggregation rule can still provide additional gains.

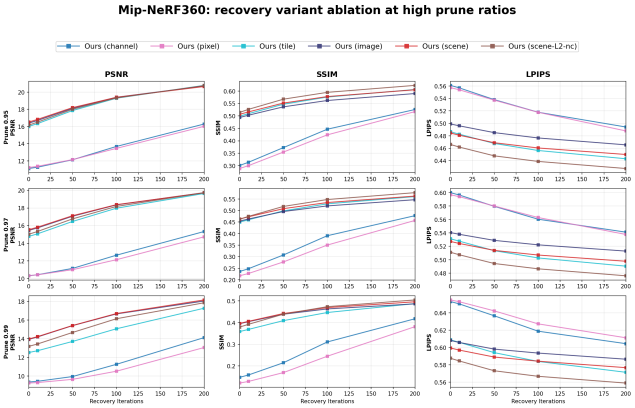

Figure 4: Short-recovery comparison of our methods variants on mip-NeRF 360 at prune ratios 0.95, 0.97, and 0.99.

Additional sampling-variability results, the separate CUDA-scored quality/efficiency comparison, and qualitative comparisons are reported in appendices B, C and E.

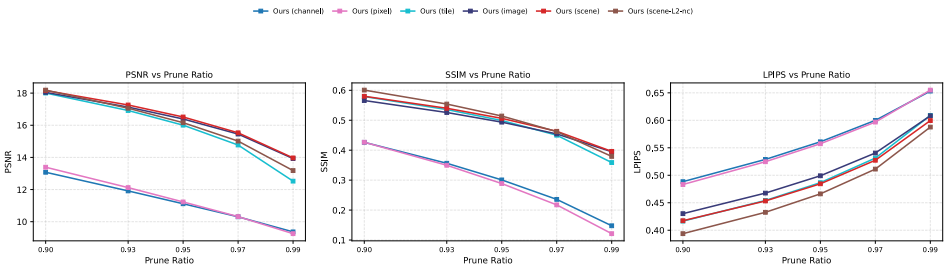

Figure 5: Ablation: prune-only on Mip-NeRF 360 (prune ratios 0.90–0.99) across our variants.

5 Conclusion, Limitations, and Scope

We studied 3D Gaussian Splatting pruning from a provable coreset perspective. Our results show that the right question is not whether unrestricted 3DGS admits a universal multiplicative coreset: it does not. Rather, meaningful guarantees become possible once the target rendering resolution is specified through a representative family of views, rays, pixels, or aggregated objectives. Under this structured regime, we proved a sensitivity-based coreset construction for 3DGS and showed how its fixed-objective guarantee can be transferred to true re-rendering under explicit validity and log-transmittance stability assumptions. Beyond the theory, the same sensitivity principle leads to a practical pruning rule that is especially effective when recovery compute is scarce. Across prune-only and very short finetuning regimes, our method preserves rendering quality better than competing pruning heuristics at aggressive compression ratios. This suggests that coreset-based importance estimation is not only a useful theoretical lens for understanding when 3DGS compression is possible, but also a practical tool for deployment settings where one cannot rely on long post-pruning optimization. We hope this work provides a foundation for future compression methods that combine the efficiency of vanilla-compatible pruning with explicit approximation guarantees.

Limitations. Our guarantees apply to the structured regime formalized in the paper, not to unrestricted continuous rendering, where we prove that non-trivial multiplicative coresets are impossible. The positive results hold for representative query families induced by a target rendering resolution, such as grids of camera poses, rays, pixels, or tiles, and transfer to rendering under validity and log-transmittance stability assumptions. Empirically, we focus on prune-only evaluation and very short finetuning budgets. This low-compute regime is where subset quality matters most, since little recovery compute is available to repair poor pruning decisions. Methods that rely on longer recovery or specialized compression pipelines may become more competitive under larger optimization budgets, but they are largely orthogonal to our contribution: our pruning rule can provide a strong vanilla-compatible initialization before additional refinement. See appendix N for the extended discussion.

Code availability. Public implementation: https://github.com/waseem-m/3dgs_provable_coresets.

Acknowledgments and Disclosure of Funding

Funding. The authors received no funding for this work.

Competing interests. The authors declare no competing interests.

References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, 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.

[2] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, 2024. doi: 10.1109/CVPR52733.2024.01839.

[3] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20654–20664, 2024. doi: 10.1109/CVPR52733.2024.01952. URL https://openaccess.thecvf.com/content/CVPR2024/html/Lu_Scaffold-GS_Structured_3D_Gaussians_for_View-Adaptive_Rendering_CVPR_2024_paper.html.

[4] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*, pages 32:1–32:11. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428.

[5] Zhihao Liang, Qi Zhang, Wenbo Hu, Lei Zhu, Ying Feng, and Kui Jia. Analytic-splatting: Anti-aliased 3D gaussian splatting via analytic integration. In *Computer Vision – ECCV 2024*, pages 281–297. Springer, 2024. doi: 10.1007/978-3-031-72643-9_17.

[6] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D gaussian splatting as markov chain monte carlo. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2573. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/93be245fce00a9bb2333c17ceae4b732-Abstract-Conference.html.

[7] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 3D gaussian splatting: Survey, technologies, challenges, and opportunities. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(7):6832–6852, 2025. doi: 10.1109/TCSVT.2025.3538684.

[8] Milena T. Bagdasarian, Paul Knoll, Yi-Hsin Li, Florian Barthel, Anna Hilsmann, Peter Eisert, and Wieland Morgenstern. 3DGS.zip: A survey on 3D gaussian splatting compression methods. *Computer Graphics Forum*, 44(2):e70078, 2025. doi: 10.1111/cgf.70078.

[9] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *Computer Vision – ECCV 2024*, pages 165–181. Springer, 2024. doi: 10.1007/978-3-031-72980-5_10.

[10] 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 *Proceedings of SIGGRAPH Asia 2024 Conference Papers*, pages 2:1–2:11. Association for Computing Machinery, 2024. doi: 10.1145/3680528.3687694.

[11] 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. doi: 10.52202/079017-4447. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/fd881d3b625437354d4421818f81058f-Abstract-Conference.html.

[12] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. EAGLES: Efficient accelerated 3D gaussians with lightweight encodings. In *Computer Vision – ECCV 2024*, pages 54–71. Springer, 2024. doi: 10.1007/978-3-031-73036-8_4.

[13] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3D gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21719–21728, 2024. doi: 10.1109/CVPR52733.2024.02052.

[14] Shuzhao Xie, Weixiang Zhang, Chen Tang, Yunpeng Bai, Rongwei Lu, Shijia Ge, and Zhi Wang. MesonGS: Post-training compression of 3D gaussians via efficient attribute transformation. In *Computer Vision – ECCV 2024*, pages 434–452. Springer, 2024. doi: 10.1007/978-3-031-73414-4_25.

[15] Muhammad Salman Ali, Maryam Qamar, Sung-Ho Bae, and Enzo Tartaglione. Trimming the fat: Efficient compression of 3D gaussian splats through pruning. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25–28, 2024*. BMVA, 2024. URL https://papers.bmvc2024.org/0358.pdf.

[16] Alex Hanson, Allen Tu, Vasu Singla, Bethmage 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 (CVPR)*, pages 5949–5958, 2025. doi: 10.1109/CVPR52734.2025.00558.

[17] Tao Wang, Mengyu Li, Geduo Zeng, Cheng Meng, and Qiong Zhang. Gaussian herding across pens: An optimal transport perspective on global gaussian reduction for 3DGS. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=j1QkrVjNVF.

[18] 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 (CVPR)*, pages 26673–26682, 2025. doi: 10.1109/CVPR52734.2025.02484.

[19] Yihang Chen, Mengyao Li, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. PCGS: Progressive compression of 3D gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 3111–3119, 2026. doi: 10.1609/aaai.v40i4.37304.

[20] Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Optimized minimal 3D gaussian splatting. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=GMiC4ccyHn.

[21] Haishan Wang, Mohammad Hassan Vali, and Arno Solin. Compressing 3D gaussian splatting by noise-substituted vector quantization. In *Image Analysis – 23rd Scandinavian Conference, SCIA 2025, Reykjavik, Iceland, June 23–25, 2025, Proceedings, Part I*, volume 15725 of *Lecture Notes in Computer Science*, pages 338–352. Springer, 2025. doi: 10.1007/978-3-031-95911-0_24.

[22] Muhammad Salman Ali, Chaoning Zhang, Marco Cagnazzo, Giuseppe Valenzise, Enzo Tartaglione, and Sung Ho Bae. Compression in 3D gaussian splatting: A survey of methods, trends, and future directions. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026. doi: 10.1109/TCSVT.2026.3676048. Accepted/In press.

[23] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 569–578, 2011. doi: 10.1145/1993636.1993712.

[24] Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *CoRR, abs/1612.00889*, 2016. URL https://arxiv.org/abs/1612.00889.

[25] Jeff M. Phillips. Coresets and sketches. In Csaba D. Toth, Joseph O'Rourke, and Jacob E. Goodman, editors, *Handbook of Discrete and Computational Geometry*, pages 1269–1288. Chapman and Hall/CRC, 3 edition, 2017.

[26] Dan Feldman. Core-sets: An updated survey. *WIREs Data Mining and Knowledge Discovery*, 10(1):e1335, 2020. doi: 10.1002/WIDM.1335.

[27] Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela Rus. Data-dependent coresets for compressing neural networks with applications to generalization bounds. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJfwJ2A5KX.

[28] Murad Tukan, Loay Mualem, and Alaa Maalouf. Pruning neural networks via coresets and convex geometry: Towards no assumptions. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

[29] Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. Provable filter pruning for efficient neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJxkOlSYDH.

[30] Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 326–342. Springer, 2024.

[31] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 1053–1061, 2024.

[32] Pankaj K Agarwal, Sariel Har-Peled, Kasturi R Varadarajan, et al. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52(1):1–30, 2005.

[33] Mihai Badoiu and Kenneth L Clarkson. Smaller core-sets for balls. In *SODA*, volume 3, pages 801–802, 2003.

[34] Piyush Kumar and E Alper Yildirim. Minimum-volume enclosing ellipsoids and core sets. *Journal of Optimization Theory and applications*, 126(1):1–21, 2005.

[35] Baharan Mirzasoleiman, Jeff A. Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, pages 6950–6960, 2020.

[36] Alaa Maalouf, Adiel Statman, and Dan Feldman. Tight sensitivity bounds for smaller coresets. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2051–2061, 2020.

[37] Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable k -means clustering via lightweight coresets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1119–1127, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219973. URL https://doi.org/10.1145/3219819.3219973.

[38] Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*, 2017.

[39] Ibrahim Jubran, Murad Tukan, Alaa Maalouf, and Dan Feldman. Sets clustering. In *International Conference on Machine Learning*, pages 4994–5005. PMLR, 2020.

[40] Murad Tukan, Samson Zhou, Alaa Maalouf, Daniela Rus, Vladimir Braverman, and Dan Feldman. Provable data subset selection for efficient neural networks training. In *International Conference on Machine Learning*, pages 34533–34555. PMLR, 2023.

[41] Alaa Maalouf, Gilad Eini, Ben Mussay, Dan Feldman, and Margarita Osadchy. A unified approach to coreset learning. 2021.

[42] Morad Tukan, Alaa Maalouf, and Dan Feldman. Coresets for near-convex functions. *Advances in Neural Information Processing Systems*, 33, 2020.

[43] Ben Mussay, Margarita Osadchy, Vladimir Braverman, Samson Zhou, and Dan Feldman. Data-independent neural pruning via coresets. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1gmHaEKwB.

[44] 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*, pages 5470–5479, 2022.

[45] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[46] 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 (ToG)*, 37(6):1–15, 2018.

A Appendix Overview

The appendix contains the full formal statements and proofs underlying the main text, together with additional experimental figures. For readability, each major theorem is restated here together with the relevant definitions. The proof structure follows the same logic as the paper: first the impossibility theorem, then the finite-query weighted coreset theorem for the fixed objective, then the precise obstruction for true rendering, then the transfer under log-transmittance stability, and finally the regionwise extension. The experimental appendix expands the main plots into per-scene graphs or to a bigger pruning and training budget regime and reports full ablations and recovery curves.

B Historical-score Sampling Variability

Design and scope. The historical color-aware per-scene ℓ_1-maximum sensitivity tensor is reused without rescoring. Each of 13 scenes is pruned at 70, 80, 85, 90, 95, 97, and 99%, with ten selection seeds (0–9): $13 \times 7 \times 10 = 910$ prune-only runs. Sensitivity-proportional sampling is without replacement, retains exactly $\max(1, \mathrm{round}((1-r)N))$ distinct source rows, and uses unit weights with no recovery or quality-based resampling. Only the random selected identities vary; the source model, scoring tensor, ordered held-out cameras and metric protocol are fixed. This study measures practical randomized selection variability; it does not claim that its unit-weight, without-replacement rule is the weighted sampling construction in the theorem.

Replicates and aggregation. Metrics are first averaged over the ordered held-out views of a scene. For each seed, the scene metrics are averaged equally within a dataset (nine Mip-NeRF 360, two Deep Blending, two Tanks & Temples scenes). The ten resulting seed-level dataset means are the replicates. Table 2 reproduces their saved means and sample SDs, using the $n-1$ denominator with $n = 10$. Neither camera counts nor scene counts are used as the number of seed replicates. The largest dataset-level SDs at 99% are 0.0671 dB PSNR, 0.002487 SSIM and 0.001220 LPIPS (Deep Blending). These small SDs do not mean that all variants have equal mean quality; the deterministic historical top-k controls remain distinct from the random runs and from the CUDA-scored models below.

Table 2: Historical-score sampling variability: saved dataset means ± sample standard deviations over ten seed-level dataset means, with equal-scene averaging within each seed. All 910 runs are prune-only. The SD measures selection variability, not uncertainty across cameras or scenes.

Pruning	Dataset	PSNR	SSIM	LPIPS
70%	Mip-NeRF 360	21.40780 ± 0.009270	0.70361 ± 0.000071	0.30770 ± 0.000090
70%	Deep Blending	25.89576 ± 0.011028	0.85512 ± 0.000108	0.30361 ± 0.000178
70%	Tanks & Temples	19.36492 ± 0.014080	0.72362 ± 0.000167	0.27948 ± 0.000319
80%	Mip-NeRF 360	20.06992 ± 0.011854	0.65346 ± 0.000172	0.35008 ± 0.000098
80%	Deep Blending	24.53053 ± 0.016865	0.82986 ± 0.000174	0.33466 ± 0.000258
80%	Tanks & Temples	18.18367 ± 0.018965	0.67369 ± 0.000279	0.32672 ± 0.000278
85%	Mip-NeRF 360	19.13387 ± 0.011949	0.61501 ± 0.000204	0.38231 ± 0.000131
85%	Deep Blending	23.53414 ± 0.020090	0.81104 ± 0.000153	0.35731 ± 0.000231
85%	Tanks & Temples	17.37512 ± 0.016214	0.63867 ± 0.000172	0.36085 ± 0.000164
90%	Mip-NeRF 360	17.91407 ± 0.010113	0.56162 ± 0.000188	0.42669 ± 0.000119
90%	Deep Blending	22.13864 ± 0.018852	0.78445 ± 0.000274	0.38859 ± 0.000352
90%	Tanks & Temples	16.28784 ± 0.017903	0.59154 ± 0.000338	0.40714 ± 0.000210
95%	Mip-NeRF 360	16.14886 ± 0.005735	0.48184 ± 0.000177	0.49263 ± 0.000099
95%	Deep Blending	19.78576 ± 0.018452	0.73980 ± 0.000396	0.43771 ± 0.000248
95%	Tanks & Temples	14.60748 ± 0.016952	0.52060 ± 0.000624	0.47762 ± 0.000441
97%	Mip-NeRF 360	15.07743 ± 0.007577	0.43421 ± 0.000291	0.53199 ± 0.000084
97%	Deep Blending	18.08517 ± 0.041463	0.70629 ± 0.000996	0.47007 ± 0.000441
97%	Tanks & Temples	13.50214 ± 0.035145	0.47694 ± 0.001029	0.52135 ± 0.000554
99%	Mip-NeRF 360	13.17115 ± 0.017745	0.35131 ± 0.000783	0.59690 ± 0.000293
99%	Deep Blending	14.61487 ± 0.067126	0.62119 ± 0.002487	0.53635 ± 0.001220
99%	Tanks & Temples	11.51036 ± 0.054628	0.40599 ± 0.001351	0.59258 ± 0.000725

C Supplemental CUDA-scored Matched Quality and Efficiency

This section reports the completed matched prune-only evaluation of Full 3DGS, PUP-3DGS, GHAP, Trimming-the-Fat and our CUDA-scored per-scene method. It is a *separate saved evaluation*: the main paper's historical-score quality tables and recovery curves are retained unchanged. The CUDA-scored models are reported only in this appendix, not substituted into historical tables or recovery curves.

C.1 Protocol and column definitions

All methods start from the same scene-specific pretrained iteration-30000 baseline, use no post-pruning recovery, and are evaluated with the stock GraphDECO renderer on identical ordered held-out cameras, black background and the standard 1600-pixel width policy. The saved construction and evaluation manifests join every quality and efficiency condition by its exact serialized PLY SHA-256. The all-13 experiment has 325 scene conditions including unavailable conditions, 1,615 rendering repetitions and 9,386 per-view metric rows. Measurements use one unshared NVIDIA A100 GPU; timing is synchronized at GPU phase boundaries.

The matched grid contains exactly **30, 50, 70, 80, 90 and 99% pruning**. Historical 95/97% quality, recovery and variability results are available separately; no complete matched cost/memory/FPS rows are asserted for those two ratios. Every table below contains one pruning ratio, with Full 3DGS repeated as the 0%-pruned reference.

- **PSNR/SSIM/LPIPS**: means over held-out views within a scene; PSNR and SSIM are higher-is-better, LPIPS lower-is-better. Dataset values equally average nine/two/two scenes, not cameras pooled across scenes; nothing is averaged across datasets.
- **G (M)**: actual retained Gaussian count divided by 10^6, not just the requested target. **PLY (MiB)**: serialized point-cloud bytes divided by 2^{20}. Counts and sizes are equal-scene means, so displayed counts may be fractional averages. Earlier rebuttal tables used an ambiguous "MB" label for these binary units; the units here are explicit.

- **Score (s)**: native scoring time, CUDA color-aware per-scene ℓ_1 maximum for ours and native Fisher computation for PUP. GHAP's `incl.` means scoring and reduction are inseparable and included in cold time. Trim's `ckpt.` means it uses checkpoint-carried statistics, not zero-cost scoring.
- **Cold (s)**: fresh-process wall time through creation of a serialized, render-ready model, including initialization, scoring, selection, construction and serialization. Five complete-process repetitions are averaged within scene, then scenes equally within dataset. This is not merely the sum of selected GPU-kernel times.
- **Prune MiB**: whole-process sampled NVML peak during native model construction, in binary MiB; it is not a live-model tensor size or an allocator-only counter.
- **FPS**: total frames divided by elapsed time over complete ordered camera passes. **Mean ms/p95 ms**: mean and 95th-percentile steady-state frame latency, excluding image writing and metric computation. Repetition statistics are averaged within scene and then equally across scenes. Dataset mean FPS is therefore not 1000 divided by dataset mean latency.
- **Render MiB**: condition-isolated whole-process sampled NVML rendering/evaluation peak, in binary MiB. `-` means not applicable, including Full-model pruning cost. `N/A` means unavailable: GHAP's unmodified native objective guard rejected the truck reduction at 30/50%, so the corresponding two-scene Tanks & Temples aggregates are conservatively unavailable, rather than averaging only train.

Tradeoff and comparison scope. The saved six-ratio, three-dataset table gives score times of 9.003/9.274/10.745 s for our Mip-NeRF 360/Deep Blending/Tanks & Temples models, 3.6–7.9× faster than native PUP Fisher scoring. Our method has the lowest pruning memory in 18/18 dataset–ratio settings and the lowest cold wall time in 13/18. It is weaker at 30–50%, mixed at 70%, and strongest from 80% onward: best in 26/27 PSNR/SSIM/LPIPS cells at 80/90/99%, including all 18 cells at 90/99%. Trim generally renders fastest in this grid but has lower aggressive-pruning fidelity. These are quality–cost tradeoffs, not a claim to dominate every metric or pruning regime. They concern these displayed CUDA-scored models only.

Table 3: CUDA-scored matched prune-only quality and model size at 30% pruning. Full 3DGS is the unpruned reference; every other row removes 30%. Bold marks the best available pruned method, not the full model.

Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓	G (M)	PLY (MiB)
Mip-NeRF 360	Full 3DGS	27.547	0.8133	0.2207	2.640	624.5
	PUP-3DGS	**27.196**	**0.8082**	**0.2250**	1.848	437.1
	GHAP	25.599	0.7552	0.2765	1.840	435.1
	Trimming-the-Fat	26.667	0.8013	0.2297	1.848	437.1
	Ours (CUDA)	24.889	0.7860	0.2418	1.848	437.1
Deep Blending	Full 3DGS	29.737	0.9068	0.2385	2.482	586.9
	PUP-3DGS	**29.635**	**0.9056**	**0.2398**	1.737	410.9
	GHAP	29.158	0.8946	0.2621	1.732	409.7
	Trimming-the-Fat	29.521	0.9044	0.2420	1.737	410.9
	Ours (CUDA)	28.229	0.8954	0.2511	1.737	410.9
Tanks & Temples	Full 3DGS	23.823	0.8526	0.1690	1.573	372.0
	PUP-3DGS	22.538	0.8340	0.1840	1.101	260.4
	GHAP	N/A	N/A	N/A	N/A	N/A
	Trimming-the-Fat	**23.422**	**0.8448**	**0.1772**	1.101	260.4
	Ours (CUDA)	21.394	0.8082	0.2019	1.101	260.4

Table 4: Matched computational efficiency at 30% pruning, with Full 3DGS repeated as the unpruned reference. All columns, memory units and scoring annotations are defined in appendix C.1.

Dataset	Method	Score (s)	Cold (s)	Prune MiB	FPS	Mean ms	p95 ms	Render MiB
Mip-NeRF 360	Full 3DGS	–	–	–	115.6	9.329	10.717	2787
	PUP-3DGS	70.817	78.499	9428	138.0	7.650	8.782	2120
	GHAP	incl.	104.418	9372	155.2	6.855	7.980	2128
	Trimming-the-Fat	ckpt.	75.192	9716	160.0	6.481	7.490	2129
	Ours (CUDA)	9.003	32.681	3780	136.9	7.754	8.840	2162
Deep Blending	Full 3DGS	–	–	–	122.4	8.499	13.172	2747
	PUP-3DGS	33.130	39.472	8190	158.1	6.484	10.022	2203
	GHAP	incl.	63.603	8256	166.4	6.218	9.336	2090
	Trimming-the-Fat	ckpt.	36.353	8459	164.2	6.256	9.789	2176
	Ours (CUDA)	9.274	30.046	3196	150.3	6.846	10.485	2269
Tanks & Temples	Full 3DGS	–	–	–	155.2	6.432	8.464	1991
	PUP-3DGS	38.826	43.560	5044	185.2	5.401	7.226	1607
	GHAP	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Trimming-the-Fat	ckpt.	25.819	5127	218.7	4.564	5.776	1614
	Ours (CUDA)	10.745	25.364	2268	192.4	5.202	6.794	1590

Table 5: CUDA-scored matched prune-only quality and model size at 50% pruning. Full 3DGS is the unpruned reference; every other row removes 50%. Bold marks the best available pruned method, not the full model.

Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓	G (M)	PLY (MiB)
Mip-NeRF 360	Full 3DGS	27.547	0.8133	0.2207	2.640	624.5
	PUP-3DGS	**25.692**	**0.7902**	**0.2402**	1.320	312.2
	GHAP	24.036	0.7050	0.3196	1.314	310.8
	Trimming-the-Fat	24.638	0.7621	0.2570	1.320	312.2
	Ours (CUDA)	24.196	0.7744	0.2530	1.320	312.2
Deep Blending	Full 3DGS	29.737	0.9068	0.2385	2.482	586.9
	PUP-3DGS	**29.150**	**0.9000**	**0.2474**	1.241	293.5
	GHAP	28.381	0.8806	0.2854	1.238	292.7
	Trimming-the-Fat	28.742	0.8938	0.2529	1.241	293.5
	Ours (CUDA)	27.478	0.8870	0.2626	1.241	293.5
Tanks & Temples	Full 3DGS	23.823	0.8526	0.1690	1.573	372.0
	PUP-3DGS	20.635	0.8050	0.2079	0.786	186.0
	GHAP	N/A	N/A	N/A	N/A	N/A
	Trimming-the-Fat	**21.982**	**0.8176**	**0.1999**	0.786	186.0
	Ours (CUDA)	20.658	0.7869	0.2205	0.786	186.0

Table 6: Matched computational efficiency at 50% pruning, with Full 3DGS repeated as the unpruned reference. All columns, memory units and scoring annotations are defined in appendix C.1.

Dataset	Method	Score (s)	Cold (s)	Prune MiB	FPS	Mean ms	p95 ms	Render MiB
Mip-NeRF 360	Full 3DGS	–	–	–	115.6	9.329	10.717	2787
	PUP-3DGS	70.817	77.055	9428	164.2	6.378	7.413	1767
	GHAP	incl.	98.337	8971	199.1	5.272	6.202	1716
	Trimming-the-Fat	ckpt.	69.701	8958	215.3	4.756	5.688	1726
	Ours (CUDA)	9.003	29.336	3780	156.8	6.688	7.697	1850
Deep Blending	Full 3DGS	–	–	–	122.4	8.499	13.172	2747
	PUP-3DGS	33.130	38.315	8190	199.2	5.107	7.863	1758
	GHAP	incl.	56.714	7876	216.4	4.753	7.060	1721
	Trimming-the-Fat	ckpt.	29.866	7659	219.5	4.650	7.205	1758
	Ours (CUDA)	9.274	26.591	3196	176.2	5.831	8.648	1783
Tanks & Temples	Full 3DGS	–	–	–	155.2	6.432	8.464	1991
	PUP-3DGS	38.826	42.726	5044	224.6	4.489	5.881	1568
	GHAP	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Trimming-the-Fat	ckpt.	22.973	4707	296.5	3.367	4.193	1293
	Ours (CUDA)	10.745	23.054	2268	229.8	4.370	5.605	1406

Table 7: CUDA-scored matched prune-only quality and model size at 70% pruning. Full 3DGS is the unpruned reference; every other row removes 70%. Bold marks the best available pruned method, not the full model.

Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓	G (M)	PLY (MiB)
Mip-NeRF 360	Full 3DGS	27.547	0.8133	0.2207	2.640	624.5
	PUP-3DGS	22.153	0.7324	**0.2859**	0.792	187.4
	GHAP	21.673	0.6269	0.3800	0.790	186.9
	Trimming-the-Fat	20.773	0.6630	0.3229	0.792	187.4
	Ours (CUDA)	**22.751**	**0.7398**	0.2862	0.792	187.4
Deep Blending	Full 3DGS	29.737	0.9068	0.2385	2.482	586.9
	PUP-3DGS	**27.039**	**0.8755**	**0.2784**	0.744	176.1
	GHAP	26.596	0.8511	0.3269	0.744	175.9
	Trimming-the-Fat	25.480	0.8489	0.2898	0.744	176.1
	Ours (CUDA)	26.035	0.8642	0.2925	0.744	176.1
Tanks & Temples	Full 3DGS	23.823	0.8526	0.1690	1.573	372.0
	PUP-3DGS	17.922	**0.7454**	**0.2588**	0.472	111.6
	GHAP	19.262	0.6748	0.3357	0.470	111.1
	Trimming-the-Fat	18.566	0.7265	0.2717	0.472	111.6
	Ours (CUDA)	**19.485**	0.7434	0.2620	0.472	111.6

Table 8: Matched computational efficiency at 70% pruning, with Full 3DGS repeated as the unpruned reference. All columns, memory units and scoring annotations are defined in appendix C.1.

Dataset	Method	Score (s)	Cold (s)	Prune MiB	FPS	Mean ms	p95 ms	Render MiB
Mip-NeRF 360	Full 3DGS	–	–	–	115.6	9.329	10.717	2787
	PUP-3DGS	70.817	75.767	9428	215.7	4.844	5.762	1459
	GHAP	incl.	92.348	8922	283.0	3.660	4.432	1299
	Trimming-the-Fat	ckpt.	65.279	8222	327.1	3.098	3.800	1276
	Ours (CUDA)	9.003	23.828	3780	191.7	5.390	6.385	1560
Deep Blending	Full 3DGS	–	–	–	122.4	8.499	13.172	2747
	PUP-3DGS	33.130	37.178	8190	276.6	3.644	5.437	1316
	GHAP	incl.	53.258	7877	317.8	3.218	4.707	1276
	Trimming-the-Fat	ckpt.	25.306	7014	337.6	3.000	4.617	1279
	Ours (CUDA)	9.274	22.962	3196	218.9	4.684	6.755	1398
Tanks & Temples	Full 3DGS	–	–	–	155.2	6.432	8.464	1991
	PUP-3DGS	38.826	41.978	5044	296.9	3.427	4.427	1309
	GHAP	incl.	41.484	4565	386.4	2.576	3.371	996
	Trimming-the-Fat	ckpt.	20.329	4272	452.6	2.202	2.709	916
	Ours (CUDA)	10.745	20.837	2268	293.2	3.432	4.234	1366

Table 9: CUDA-scored matched prune-only quality and model size at 80% pruning. Full 3DGS is the unpruned reference; every other row removes 80%. Bold marks the best available pruned method, not the full model.

Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓	G (M)	PLY (MiB)
Mip-NeRF 360	Full 3DGS	27.547	0.8133	0.2207	2.640	624.5
	PUP-3DGS	19.401	0.6655	0.3366	0.528	124.9
	GHAP	19.778	0.5639	0.4247	0.527	124.7
	Trimming-the-Fat	18.006	0.5688	0.3836	0.528	124.9
	Ours (CUDA)	**21.399**	**0.6972**	**0.3241**	0.528	124.9
Deep Blending	Full 3DGS	29.737	0.9068	0.2385	2.482	586.9
	PUP-3DGS	24.289	0.8412	**0.3172**	0.496	117.4
	GHAP	24.667	0.8198	0.3650	0.496	117.3
	Trimming-the-Fat	22.017	0.7864	0.3350	0.496	117.4
	Ours (CUDA)	**24.817**	**0.8419**	0.3222	0.496	117.4
Tanks & Temples	Full 3DGS	23.823	0.8526	0.1690	1.573	372.0
	PUP-3DGS	16.052	0.6872	0.3097	0.315	74.4
	GHAP	17.611	0.6056	0.3912	0.314	74.2
	Trimming-the-Fat	16.114	0.6315	0.3431	0.315	74.4
	Ours (CUDA)	**18.507**	**0.6999**	**0.3030**	0.315	74.4

Table 10: Matched computational efficiency at 80% pruning, with Full 3DGS repeated as the unpruned reference. All columns, memory units and scoring annotations are defined in appendix C.1.

Dataset	Method	Score (s)	Cold (s)	Prune MiB	FPS	Mean ms	p95 ms	Render MiB
Mip-NeRF 360	Full 3DGS	–	–	–	115.6	9.329	10.717	2787
	PUP-3DGS	70.817	75.112	9428	265.0	3.934	4.751	1236
	GHAP	incl.	90.319	8923	364.2	2.811	3.469	1073
	Trimming-the-Fat	ckpt.	62.641	7871	435.0	2.317	2.854	1060
	Ours (CUDA)	9.003	21.854	3780	222.8	4.604	5.478	1328
Deep Blending	Full 3DGS	–	–	–	122.4	8.499	13.172	2747
	PUP-3DGS	33.130	36.605	8190	353.0	2.840	4.155	1108
	GHAP	incl.	51.781	7876	417.7	2.439	3.553	1037
	Trimming-the-Fat	ckpt.	23.093	6674	461.0	2.184	3.258	1003
	Ours (CUDA)	9.274	21.453	3196	257.8	3.970	5.644	1264
Tanks & Temples	Full 3DGS	–	–	–	155.2	6.432	8.464	1991
	PUP-3DGS	38.826	41.592	5044	370.6	2.753	3.437	1124
	GHAP	incl.	40.632	4569	497.7	1.997	2.562	856
	Trimming-the-Fat	ckpt.	18.711	4062	595.1	1.669	2.047	793
	Ours (CUDA)	10.745	19.764	2268	348.8	2.888	3.507	1188

Table 11: CUDA-scored matched prune-only quality and model size at 90% pruning. Full 3DGS is the unpruned reference; every other row removes 90%. Bold marks the best available pruned method, not the full model.

Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓	G (M)	PLY (MiB)
Mip-NeRF 360	Full 3DGS	27.547	0.8133	0.2207	2.640	624.5
	PUP-3DGS	15.794	0.5348	0.4289	0.264	62.5
	GHAP	16.761	0.4581	0.4943	0.264	62.4
	Trimming-the-Fat	14.453	0.4135	0.4803	0.264	62.5
	Ours (CUDA)	**19.068**	**0.6080**	**0.4001**	0.264	62.5
Deep Blending	Full 3DGS	29.737	0.9068	0.2385	2.482	586.9
	PUP-3DGS	19.516	0.7647	0.3930	0.248	58.7
	GHAP	20.525	0.7436	0.4368	0.248	58.7
	Trimming-the-Fat	16.519	0.6417	0.4290	0.248	58.7
	Ours (CUDA)	**22.751**	**0.8029**	**0.3753**	0.248	58.7
Tanks & Temples	Full 3DGS	23.823	0.8526	0.1690	1.573	372.0
	PUP-3DGS	13.379	0.5773	0.4062	0.157	37.2
	GHAP	14.831	0.4882	0.4831	0.157	37.2
	Trimming-the-Fat	12.465	0.4710	0.4579	0.157	37.2
	Ours (CUDA)	**16.712**	**0.6216**	**0.3806**	0.157	37.2

Table 12: Matched computational efficiency at 90% pruning, with Full 3DGS repeated as the unpruned reference. All columns, memory units and scoring annotations are defined in appendix C.1.

Dataset	Method	Score (s)	Cold (s)	Prune MiB	FPS	Mean ms	p95 ms	Render MiB
Mip-NeRF 360	Full 3DGS	–	–	–	115.6	9.329	10.717	2787
	PUP-3DGS	70.817	74.462	9428	373.1	2.797	3.450	977
	GHAP	incl.	88.319	8924	530.9	1.900	2.304	851
	Trimming-the-Fat	ckpt.	60.400	7500	654.1	1.540	1.876	841
	Ours (CUDA)	9.003	19.857	3780	279.3	3.635	4.422	1116
Deep Blending	Full 3DGS	–	–	–	122.4	8.499	13.172	2747
	PUP-3DGS	33.130	35.993	8190	510.3	1.954	2.756	892
	GHAP	incl.	50.272	7877	605.2	1.655	2.262	830
	Trimming-the-Fat	ckpt.	20.696	6353	706.5	1.412	1.946	801
	Ours (CUDA)	9.274	19.661	3196	323.8	3.132	4.484	1041
Tanks & Temples	Full 3DGS	–	–	–	155.2	6.432	8.464	1991
	PUP-3DGS	38.826	41.199	5044	511.1	2.001	2.448	899
	GHAP	incl.	40.750	4566	700.5	1.416	1.729	712
	Trimming-the-Fat	ckpt.	17.221	3850	842.6	1.175	1.369	700
	Ours (CUDA)	10.745	18.614	2268	443.8	2.265	2.756	938

Table 13: CUDA-scored matched prune-only quality and model size at 99% pruning. Full 3DGS is the unpruned reference; every other row removes 99%. Bold marks the best available pruned method, not the full model.

Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓	G (M)	PLY (MiB)
Mip-NeRF 360	Full 3DGS	27.547	0.8133	0.2207	2.640	624.5
	PUP-3DGS	10.669	0.2268	0.6239	0.026	6.2
	GHAP	10.189	0.1769	0.6637	0.026	6.2
	Trimming-the-Fat	9.580	0.1356	0.6612	0.026	6.2
	Ours (CUDA)	**14.333**	**0.4149**	**0.5970**	0.026	6.2
Deep Blending	Full 3DGS	29.737	0.9068	0.2385	2.482	586.9
	PUP-3DGS	10.133	0.4283	0.6209	0.025	5.9
	GHAP	10.137	0.4007	0.6535	0.025	5.9
	Trimming-the-Fat	7.656	0.1315	0.6834	0.025	5.9
	Ours (CUDA)	**16.088**	**0.6753**	**0.5234**	0.025	5.9
Tanks & Temples	Full 3DGS	23.823	0.8526	0.1690	1.573	372.0
	PUP-3DGS	8.942	0.3203	0.6266	0.016	3.7
	GHAP	7.942	0.2245	0.6749	0.016	3.7
	Trimming-the-Fat	6.228	0.1042	0.6680	0.016	3.7
	Ours (CUDA)	**12.070**	**0.4395**	**0.6008**	0.016	3.7

Table 14: Matched computational efficiency at 99% pruning, with Full 3DGS repeated as the unpruned reference. All columns, memory units and scoring annotations are defined in appendix C.1.

Dataset	Method	Score (s)	Cold (s)	Prune MiB	FPS	Mean ms	p95 ms	Render MiB
Mip-NeRF 360	Full 3DGS	–	–	–	115.6	9.329	10.717	2787
	PUP-3DGS	70.817	73.849	9428	814.2	1.254	1.457	673
	GHAP	incl.	90.474	8925	1086.3	0.912	1.041	648
	Trimming-the-Fat	ckpt.	58.498	7163	1242.9	0.804	0.936	646
	Ours (CUDA)	9.003	18.124	3780	485.8	2.066	2.559	794
Deep Blending	Full 3DGS	–	–	–	122.4	8.499	13.172	2747
	PUP-3DGS	33.130	35.435	8190	1008.7	0.981	1.178	641
	GHAP	incl.	52.621	7877	1261.9	0.780	0.914	635
	Trimming-the-Fat	ckpt.	18.612	6023	1409.2	0.699	0.783	621
	Ours (CUDA)	9.274	18.011	3196	625.6	1.594	2.248	695
Tanks & Temples	Full 3DGS	–	–	–	155.2	6.432	8.464	1991
	PUP-3DGS	38.826	40.858	5044	968.2	1.051	1.169	618
	GHAP	incl.	42.837	4566	1327.5	0.741	0.809	588
	Trimming-the-Fat	ckpt.	16.261	3651	1449.3	0.679	0.726	588
	Ours (CUDA)	10.745	17.631	2268	732.4	1.374	1.588	681

D Transmittance, Local Regularity and Coverage Diagnostics

The following saved diagnostics address the practical quantities appearing in the rendering and regional theorems. Their model families and sampled domains are stated separately; none is substituted for a continuous-domain numerical certificate.

D.1 Retained-identity prefix-transmittance coverage

For a retained original Gaussian i and pixel query q, the diagnostic is $\delta_i(q) = |\log T_{\mathrm{full},i}(q) - \log T_{\mathrm{pruned},i}(q)|$. The strict finite-domain maximum is retained in the scene exports. Eligible target events pass the stock float32 alpha threshold at the pixel; prefix loss is accumulated directly in float64 without renderer early termination. The train/scoring and held-out camera sets are measured separately across 13 scenes at 80/90/99% pruning.

Nested thresholds. The query-centric construction takes a Gaussian-event quantile at each active pixel, then a pixel quantile within each camera, then a camera quantile. The Gaussian-centric construction instead takes a pixel-event quantile for each Gaussian and camera, then a camera quantile for each Gaussian, then a Gaussian quantile. A 95/95/95 row uses 0.95 at each successive level; it does *not* mean that 95% of all scalar queries satisfy the theorem's maximum-over-Gaussians condition. Entries below are equal-scene means of these nested thresholds, not dataset-wide quantile thresholds or additional coverage guarantees.

PUP and Trim are compared with the same retained-identity protocol after verifying their output rows against the original baseline and their ratio-wise nested subsets. GHAP is not compared for this quantity because its GMR operation changes the Gaussian basis, so an original-row retained-prefix comparison is undefined. It remains included in the quality/efficiency study.

Event distributions and worst-case records. The saved event CDFs average per-scene empirical CDFs equally within each dataset, rather than pooling all ray–Gaussian events. Event p80/p99 are the conservative upper edges of their authoritative histogram bins. Both held-out thresholds are lower for ours in all 9/9 dataset–ratio settings against PUP and Trim, whereas exact finite-domain maxima are mixed (3/9 lower against PUP and 6/9 against Trim). The event histogram, exact maximum and worst-case records therefore answer different questions; a percentile cannot replace the global maximum in the theorem.

Table 15: Saved matched event-distribution summaries at 80% pruning. Event p80/p99 are conservative upper-bin edges from equal-scene CDF averages, not quantiles of pooled events. The maximum column is the saved equal-scene summary of exact finite-domain maxima, not a uniform dataset certificate.

Dataset	Split	Method	Event p80	Event p99	Max summary
Mip-NeRF 360	train	Ours	2.461	14.407	101.135
Mip-NeRF 360	train	PUP-3DGS	3.311	15.450	95.542
Mip-NeRF 360	train	Trimming-the-Fat	5.986	23.728	111.781
Deep Blending	train	Ours	1.097	10.651	129.121
Deep Blending	train	PUP-3DGS	2.524	13.589	132.437
Deep Blending	train	Trimming-the-Fat	3.370	14.913	94.877
Tanks & Temples	train	Ours	5.109	23.313	233.419
Tanks & Temples	train	PUP-3DGS	6.039	28.662	232.346
Tanks & Temples	train	Trimming-the-Fat	12.783	45.733	314.657
Mip-NeRF 360	test	Ours	2.479	14.559	83.684
Mip-NeRF 360	test	PUP-3DGS	3.352	15.793	85.616
Mip-NeRF 360	test	Trimming-the-Fat	6.396	24.713	107.677
Deep Blending	test	Ours	1.069	10.893	114.773
Deep Blending	test	PUP-3DGS	2.705	14.938	97.609
Deep Blending	test	Trimming-the-Fat	3.565	16.637	87.559
Tanks & Temples	test	Ours	5.052	23.144	181.339
Tanks & Temples	test	PUP-3DGS	6.098	29.450	149.520
Tanks & Temples	test	Trimming-the-Fat	12.766	46.040	199.040

Table 16: Saved matched event-distribution summaries at 90% pruning. Event p80/p99 are conservative upper-bin edges from equal-scene CDF averages, not quantiles of pooled events. The maximum column is the saved equal-scene summary of exact finite-domain maxima, not a uniform dataset certificate.

Dataset	Split	Method	Event p80	Event p99	Max summary
Mip-NeRF 360	train	Ours	3.756	18.828	113.705
Mip-NeRF 360	train	PUP-3DGS	5.233	21.267	115.018
Mip-NeRF 360	train	Trimming-the-Fat	7.793	27.654	120.377
Deep Blending	train	Ours	1.844	13.411	142.259
Deep Blending	train	PUP-3DGS	3.716	17.625	154.368
Deep Blending	train	Trimming-the-Fat	4.875	18.885	110.443
Tanks & Temples	train	Ours	7.295	29.942	306.917
Tanks & Temples	train	PUP-3DGS	9.194	38.375	320.647
Tanks & Temples	train	Trimming-the-Fat	15.651	53.169	373.602
Mip-NeRF 360	test	Ours	3.788	19.137	96.284
Mip-NeRF 360	test	PUP-3DGS	5.231	21.670	102.106
Mip-NeRF 360	test	Trimming-the-Fat	8.318	28.882	116.182
Deep Blending	test	Ours	1.835	13.871	118.380
Deep Blending	test	PUP-3DGS	3.928	19.439	112.194
Deep Blending	test	Trimming-the-Fat	5.044	21.107	101.639
Tanks & Temples	test	Ours	7.213	29.525	216.856
Tanks & Temples	test	PUP-3DGS	9.222	39.007	198.287
Tanks & Temples	test	Trimming-the-Fat	15.728	53.532	241.326

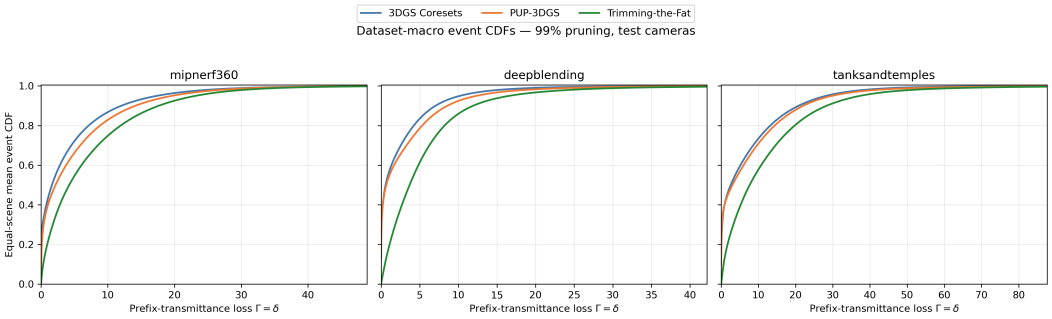

Figure 6: Existing equal-scene event CDF comparison on held-out cameras at 99% pruning. Dataset event CDFs are averaged over scenes; they are not CDFs of the maximum-over-Gaussians per-query certificate.

Table 17: Saved matched event-distribution summaries at 99% pruning. Event p80/p99 are conservative upper-bin edges from equal-scene CDF averages, not quantiles of pooled events. The maximum column is the saved equal-scene summary of exact finite-domain maxima, not a uniform dataset certificate.

Dataset	Split	Method	Event p80	Event p99	Max summary
Mip-NeRF 360	train	Ours	7.131	28.770	141.352
Mip-NeRF 360	train	PUP-3DGS	8.823	30.669	138.349
Mip-NeRF 360	train	Trimming-the-Fat	11.338	33.869	135.079
Deep Blending	train	Ours	4.105	18.897	168.210
Deep Blending	train	PUP-3DGS	5.014	22.230	151.073
Deep Blending	train	Trimming-the-Fat	8.087	28.221	155.781
Tanks & Temples	train	Ours	13.457	45.765	400.742
Tanks & Temples	train	PUP-3DGS	14.538	51.530	408.183
Tanks & Temples	train	Trimming-the-Fat	19.849	63.366	427.196
Mip-NeRF 360	test	Ours	7.185	29.199	122.901
Mip-NeRF 360	test	PUP-3DGS	8.765	31.065	122.759
Mip-NeRF 360	test	Trimming-the-Fat	11.770	35.413	124.580
Deep Blending	test	Ours	4.181	19.744	135.222
Deep Blending	test	PUP-3DGS	5.252	24.598	129.173
Deep Blending	test	Trimming-the-Fat	8.276	31.421	125.533
Tanks & Temples	test	Ours	13.158	46.065	253.898
Tanks & Temples	test	PUP-3DGS	14.538	52.836	278.331
Tanks & Temples	test	Trimming-the-Fat	19.722	64.969	289.490

At fixed coverage, lower thresholds indicate more stable retained prefixes. On held-out cameras, the query-centric 90/90/90, 95/95/95 and 99/99/99 thresholds for ours are lower than PUP in 8/9, 9/9 and 9/9 dataset–ratio settings, and lower than Trim in all 9/9 settings at each level. The Gaussian-centric comparisons are 8/9, 9/9 and 9/9 against PUP, and 8/9 at each level against Trim. Large finite-domain maxima and large exponentiated thresholds can still make a numerical global rendering bound loose. These comparative measurements quantify where the practical retained set changes visibility less; without the fixed-objective and validity conditions, they are not error certificates for that practical selection rule.

Table 18: Nested retained-identity transmittance thresholds at 80% pruning on held-out cameras. Each entry is an equal-scene mean of thresholds; lower is more stable at matched nested coverage. These are not maxima of $\gamma(q)$ or numerical rendered-error certificates.

Dataset	Family	Coverage	Ours	PUP	Trim
Mip-NeRF 360	query	90/90/90	6.296	7.548	11.949
Mip-NeRF 360	query	95/95/95	9.830	11.542	17.686
Mip-NeRF 360	query	99/99/99	19.264	21.781	30.435
Mip-NeRF 360	gaussian	90/90/90	13.491	14.703	20.959
Mip-NeRF 360	gaussian	95/95/95	23.284	24.955	32.137
Mip-NeRF 360	gaussian	99/99/99	42.152	42.866	52.045
Deep Blending	query	90/90/90	6.299	8.367	9.475
Deep Blending	query	95/95/95	10.504	13.141	14.674
Deep Blending	query	99/99/99	23.140	32.248	33.398
Deep Blending	gaussian	90/90/90	12.802	19.284	19.921
Deep Blending	gaussian	95/95/95	19.993	28.015	29.020
Deep Blending	gaussian	99/99/99	40.926	46.258	47.040
Tanks & Temples	query	90/90/90	12.522	14.577	24.529
Tanks & Temples	query	95/95/95	18.953	24.100	37.564
Tanks & Temples	query	99/99/99	34.467	46.532	67.526
Tanks & Temples	gaussian	90/90/90	29.529	39.509	51.110
Tanks & Temples	gaussian	95/95/95	47.746	60.258	76.584
Tanks & Temples	gaussian	99/99/99	96.390	104.359	128.123

Table 19: Nested retained-identity transmittance thresholds at 90% pruning on held-out cameras. Each entry is an equal-scene mean of thresholds; lower is more stable at matched nested coverage. These are not maxima of $\gamma(q)$ or numerical rendered-error certificates.

Dataset	Family	Coverage	Ours	PUP	Trim
Mip-NeRF 360	query	90/90/90	8.421	10.594	14.909
Mip-NeRF 360	query	95/95/95	12.901	15.596	21.631
Mip-NeRF 360	query	99/99/99	25.015	28.651	36.002
Mip-NeRF 360	gaussian	90/90/90	18.839	20.800	24.519
Mip-NeRF 360	gaussian	95/95/95	31.032	34.342	37.054
Mip-NeRF 360	gaussian	99/99/99	52.582	56.252	58.679
Deep Blending	query	90/90/90	8.314	11.128	11.811
Deep Blending	query	95/95/95	13.246	17.294	18.758
Deep Blending	query	99/99/99	30.569	41.289	41.183
Deep Blending	gaussian	90/90/90	17.391	25.958	25.841
Deep Blending	gaussian	95/95/95	26.503	36.907	36.071
Deep Blending	gaussian	99/99/99	51.916	58.677	55.695
Tanks & Temples	query	90/90/90	17.789	20.204	29.793
Tanks & Temples	query	95/95/95	25.666	32.320	44.762
Tanks & Temples	query	99/99/99	45.735	61.561	81.483
Tanks & Temples	gaussian	90/90/90	37.787	54.938	61.270
Tanks & Temples	gaussian	95/95/95	60.506	82.190	90.752
Tanks & Temples	gaussian	99/99/99	119.677	141.258	148.886

Table 20: Nested retained-identity transmittance thresholds at 99% pruning on held-out cameras. Each entry is an equal-scene mean of thresholds; lower is more stable at matched nested coverage. These are not maxima of $\gamma(q)$ or numerical rendered-error certificates.

Dataset	Family	Coverage	Ours	PUP	Trim
Mip-NeRF 360	query	90/90/90	14.025	16.374	21.713
Mip-NeRF 360	query	95/95/95	20.654	23.861	30.405
Mip-NeRF 360	query	99/99/99	37.783	40.909	46.556
Mip-NeRF 360	gaussian	90/90/90	30.961	30.784	30.936
Mip-NeRF 360	gaussian	95/95/95	47.788	49.062	45.808
Mip-NeRF 360	gaussian	99/99/99	76.571	78.575	72.947
Deep Blending	query	90/90/90	13.164	15.607	16.613
Deep Blending	query	95/95/95	20.419	24.170	28.907
Deep Blending	query	99/99/99	47.351	51.063	55.260
Deep Blending	gaussian	90/90/90	27.701	37.793	38.614
Deep Blending	gaussian	95/95/95	42.307	52.829	51.496
Deep Blending	gaussian	99/99/99	73.991	76.963	74.000
Tanks & Temples	query	90/90/90	32.195	31.536	38.623
Tanks & Temples	query	95/95/95	44.160	48.676	58.382
Tanks & Temples	query	99/99/99	82.169	88.375	101.109
Tanks & Temples	gaussian	90/90/90	55.424	84.238	78.789
Tanks & Temples	gaussian	95/95/95	88.417	117.444	113.584
Tanks & Temples	gaussian	99/99/99	167.079	202.553	185.679

Table 21: Nested retained-identity transmittance thresholds at 80% pruning on scoring/training cameras. Each entry is an equal-scene mean of thresholds; lower is more stable at matched nested coverage. These are not maxima of $\gamma(q)$ or numerical rendered-error certificates.

Dataset	Family	Coverage	Ours	PUP	Trim
Mip-NeRF 360	query	90/90/90	6.480	7.711	11.975
Mip-NeRF 360	query	95/95/95	10.033	11.525	17.235
Mip-NeRF 360	query	99/99/99	19.839	22.195	31.241
Mip-NeRF 360	gaussian	90/90/90	11.799	12.767	18.283
Mip-NeRF 360	gaussian	95/95/95	20.251	21.309	28.606
Mip-NeRF 360	gaussian	99/99/99	47.727	47.051	57.120
Deep Blending	query	90/90/90	6.601	8.378	8.957
Deep Blending	query	95/95/95	10.507	13.185	14.346
Deep Blending	query	99/99/99	23.023	28.593	28.890
Deep Blending	gaussian	90/90/90	11.550	16.247	17.233
Deep Blending	gaussian	95/95/95	22.243	27.049	28.138
Deep Blending	gaussian	99/99/99	52.657	53.244	52.088
Tanks & Temples	query	90/90/90	12.952	14.952	25.133
Tanks & Temples	query	95/95/95	18.641	23.971	37.708
Tanks & Temples	query	99/99/99	35.750	53.516	74.287
Tanks & Temples	gaussian	90/90/90	29.450	39.854	49.548
Tanks & Temples	gaussian	95/95/95	48.139	65.544	81.228
Tanks & Temples	gaussian	99/99/99	116.791	134.842	174.703

Table 22: Nested retained-identity transmittance thresholds at 90% pruning on scoring/training cameras. Each entry is an equal-scene mean of thresholds; lower is more stable at matched nested coverage. These are not maxima of $\gamma(q)$ or numerical rendered-error certificates.

Dataset	Family	Coverage	Ours	PUP	Trim
Mip-NeRF 360	query	90/90/90	8.506	10.613	14.687
Mip-NeRF 360	query	95/95/95	12.923	15.546	20.914
Mip-NeRF 360	query	99/99/99	25.315	29.319	36.897
Mip-NeRF 360	gaussian	90/90/90	16.531	18.326	21.641
Mip-NeRF 360	gaussian	95/95/95	27.291	29.140	33.159
Mip-NeRF 360	gaussian	99/99/99	59.051	60.510	64.310
Deep Blending	query	90/90/90	8.669	11.227	11.346
Deep Blending	query	95/95/95	13.674	17.661	17.715
Deep Blending	query	99/99/99	29.396	36.651	34.742
Deep Blending	gaussian	90/90/90	15.213	21.722	22.122
Deep Blending	gaussian	95/95/95	28.324	35.037	34.846
Deep Blending	gaussian	99/99/99	63.039	65.123	61.780
Tanks & Temples	query	90/90/90	17.881	20.675	30.289
Tanks & Temples	query	95/95/95	25.338	32.856	44.701
Tanks & Temples	query	99/99/99	47.558	67.683	86.871
Tanks & Temples	gaussian	90/90/90	37.604	55.549	59.748
Tanks & Temples	gaussian	95/95/95	62.523	87.406	95.005
Tanks & Temples	gaussian	99/99/99	147.024	174.941	202.902

Table 23: Nested retained-identity transmittance thresholds at 99% pruning on scoring/training cameras. Each entry is an equal-scene mean of thresholds; lower is more stable at matched nested coverage. These are not maxima of $\gamma(q)$ or numerical rendered-error certificates.

Dataset	Family	Coverage	Ours	PUP	Trim
Mip-NeRF 360	query	90/90/90	13.888	16.302	20.909
Mip-NeRF 360	query	95/95/95	20.346	23.537	28.441
Mip-NeRF 360	query	99/99/99	38.098	41.730	47.051
Mip-NeRF 360	gaussian	90/90/90	27.798	28.108	27.915
Mip-NeRF 360	gaussian	95/95/95	43.662	42.180	41.802
Mip-NeRF 360	gaussian	99/99/99	83.922	80.924	78.476
Deep Blending	query	90/90/90	13.447	15.073	16.396
Deep Blending	query	95/95/95	20.503	23.248	26.664
Deep Blending	query	99/99/99	42.315	44.569	49.280
Deep Blending	gaussian	90/90/90	23.015	30.474	33.281
Deep Blending	gaussian	95/95/95	39.326	46.573	50.048
Deep Blending	gaussian	99/99/99	82.500	82.744	85.062
Tanks & Temples	query	90/90/90	31.654	31.674	38.128
Tanks & Temples	query	95/95/95	45.313	49.217	56.632
Tanks & Temples	query	99/99/99	80.908	94.334	110.027
Tanks & Temples	gaussian	90/90/90	54.695	84.286	77.557
Tanks & Temples	gaussian	95/95/95	89.529	124.882	118.829
Tanks & Temples	gaussian	99/99/99	201.913	241.712	247.233

D.2 Six-scene local-pose diagnostics

The scenes are bicycle, garden, drjohnson, playroom, truck and train. Around a deterministic training-camera medoid, translations are normalized by $\tau = 0.01R_{\mathrm{scene}}$ and rotations by $\phi = 1^\circ$. For matching pixels and RGB channels, the query distance reduces to

$$d_C(C, C') = \sqrt{(\|c - c'\|_2/\tau)^2 + (\theta(R^\top R')/\phi)^2}.$$

The 37 representatives comprise the anchor plus positive/negative probes along three translation and three rotation axes at magnitudes 0.25, 0.5 and 1.0. The rendered diagnostic family contains these poses plus 128 Sobol direction pairs at four magnitudes, $37 + 128 \times 4 = 549$ poses *per scene*. Scalar adjacent-pose slopes are $|A(q) - A(q')|/d_C(C, C')$ over matching pixel-channel queries, not RGB-vector norms. Visibility and depth-order changes are retained, not filtered away.

Table 24 separates full-model slopes from the supplemental 99%-pruned top-k slopes. Full-model p99 slopes are 1.045–2.142, while sampled maxima are 5.199–9.023. Pixel Gaussian sequences change for 99.21–99.84% of adjacent-pose pixels. Thus, sequence changes are widespread but usually do not induce an exceptionally large scalar output change; the sampled maximum exposes the tail. The reduced-model p99 range at 99% is 0.352–1.294. These are observed finite-domain distributions, not upper bounds on continuous-region L_{full} or L_w.

Table 24: Six-scene local diagnostics. Full p95/p99/max are scalar pose-pair slopes; Reduced p99 is the 99%-pruned top-k diagnostic model. Changed order is the fraction of adjacent-pose pixel sequences changing; zero count is over all 549 poses, pixels and channels. These are sampled-domain diagnostics, not certified continuous-domain Lipschitz constants or positive lower bounds. The reduced top-k columns belong to the supplemental evaluation, not the historical main-paper quality tables.

Scene	Full p95	Full p99	Full max	Reduced p99	Changed order	Zero count
bicycle	0.948	2.142	7.437	1.294	99.60%	0
garden	0.716	1.290	5.228	0.624	99.81%	24,887
drjohnson	0.474	1.045	5.760	0.359	99.74%	0
playroom	0.618	1.411	5.199	0.352	99.21%	36,488
truck	0.689	1.514	9.023	0.572	99.50%	0
train	1.003	1.989	8.176	0.629	99.84%	2

Signal coverage. The full-model Gaussian-only RGB signal is measured over every pixel/channel at all 549 poses, without foreground masking or zero filtering. The rebuttal reported the following proportions above fixed signal thresholds: At $A_0 = 0.01$, reported coverage is 96.90–99.99%. The

Table 25: Full-model positive-signal coverage reported in the accepted discussion, over the six-scene 549-pose diagnostics. These are empirical pixel-channel proportions, not positive lower bounds over a continuous region. The saved raw histogram and minimum/zero records remain distinct from these displayed coverage summaries.

Scene	$A > 0.01$	$A > 0.05$	$A > 0.10$
bicycle	99.99%	96.85%	90.78%
garden	99.76%	97.92%	94.47%
drjohnson	99.69%	92.95%	79.06%
playroom	99.16%	94.18%	84.59%
truck	99.99%	98.82%	94.04%
train	96.90%	89.50%	85.29%

saved signal records also contain exact zeros in garden, playroom and train (table 24). Any region containing an exact-zero query fails the theorem's $A_{\min} > 0$ hypothesis. Conversely, a region known to satisfy $A(q) \geq A_0$ everywhere can use $A_{\min} = A_0$; a sampled coverage percentage alone does not establish that condition. Increasing A_0 trades empirical coverage for a tighter denominator.

Representative distances are a separate measurement. An independent set of 8,192 fixed-seed Sobol validation poses per scene probes the controlled local region against the 37 representatives. Its sampled maximum nearest-representative distance is $\widehat{\rho} = 1.229444174$; this is not a certified continuous-region cover radius. As stated in the rebuttal, the unseen-view theorem applies to qualifying covered compact regions under its explicit assumptions, not arbitrary camera trajectories.

E Supplemental Qualitative Comparisons and Failure Cases

This gallery uses existing saved renders and is confined to the appendix. Historical main-paper figures are retained. The prune-only panels use the same held-out camera, intrinsics, resolution and crop coordinates across methods, with no recovery. The saved selected-quality and shared-challenging cases cover bicycle, drjohnson and truck at 70/80/90/95/99%. The corresponding 95% models were constructed for this gallery; this does not add a matched 95% timing/memory row to the six-ratio grid.

The first gallery compares decoded $[0, 1]$ images to the *ground-truth photograph*; its error map is mean-channel absolute RGB difference with common range $[0, 0.25]$. The red rectangle fixes the aligned crop. Each scene has a selected qualitative case and a shared challenging case; these selected examples do not estimate performance over all cameras. Crop/camera identities, clipped fractions and source model hashes are preserved in the saved manifests.



Figure 7: bicycle, 70% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras _DSC8832.JPG, _DSC8679.JPG, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

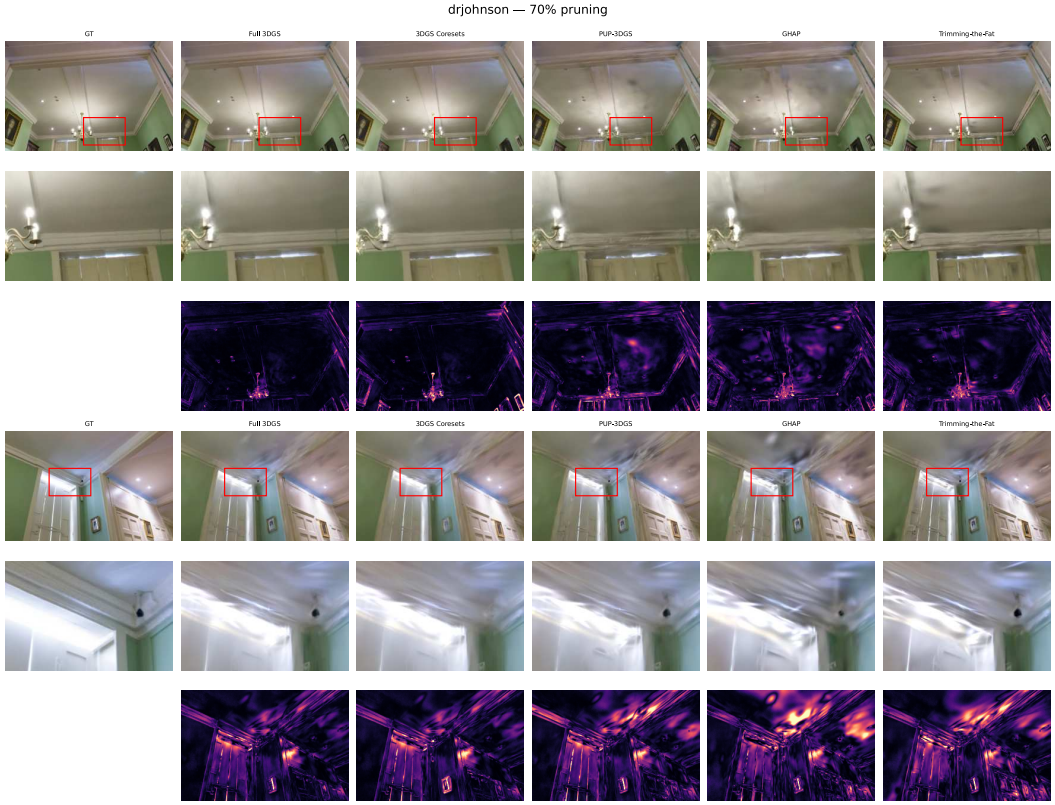

Figure 8: drjohnson, 70% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras IMG_6410.jpg, IMG_6392.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

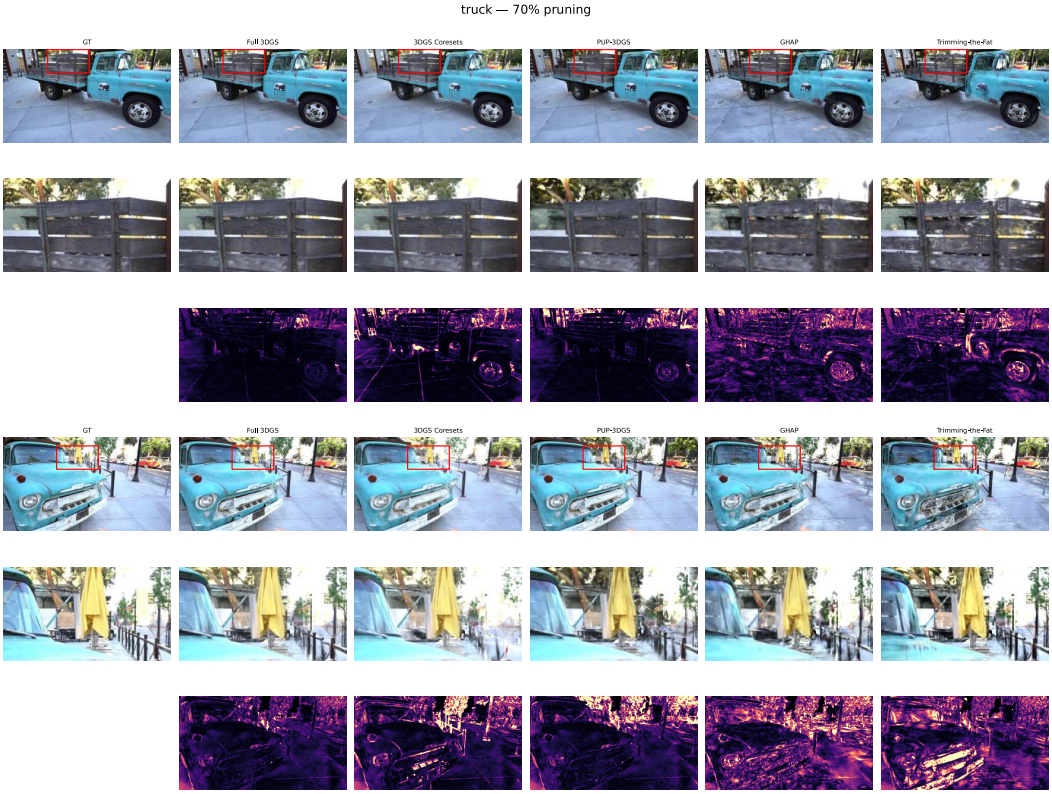

Figure 9: truck, 70% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras 000073.jpg, 000233.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

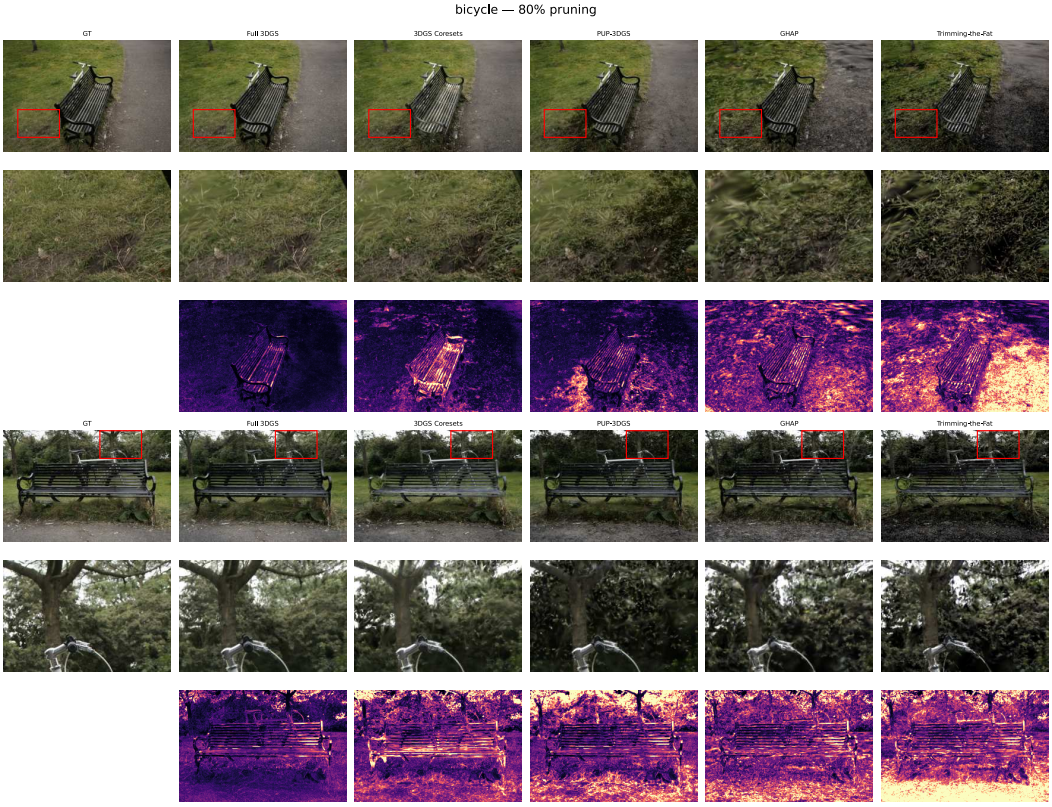

Figure 10: bicycle, 80% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras _DSC8848.JPG, _DSC8703.JPG, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

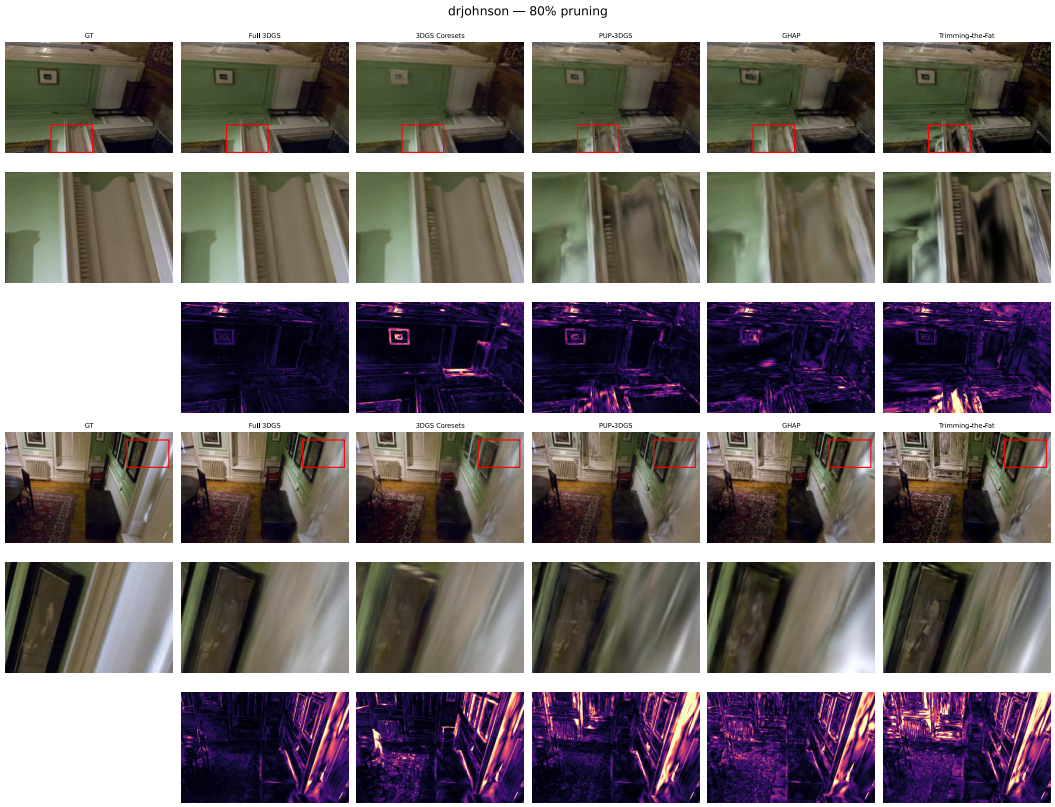

Figure 11: drjohnson, 80% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras IMG_6483.jpg, IMG_6589.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.



Figure 12: truck, 80% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras 000193.jpg, 000105.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

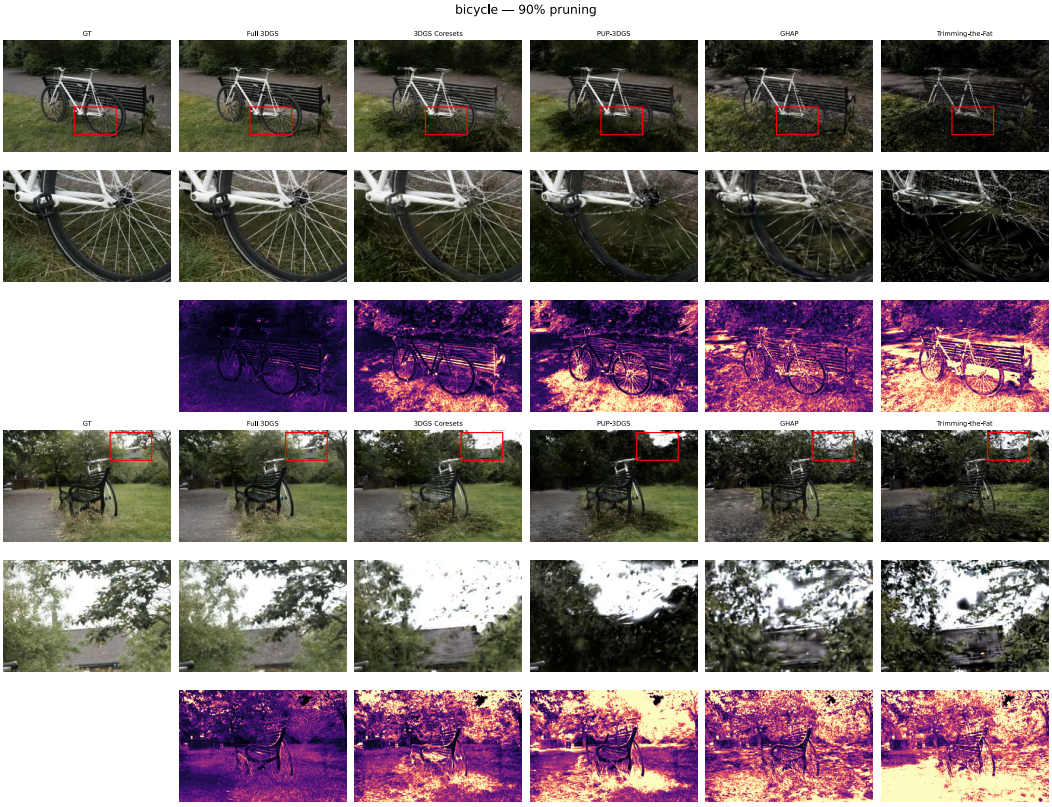

Figure 13: bicycle, 90% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras _DSC8776.JPG, _DSC8695.JPG, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

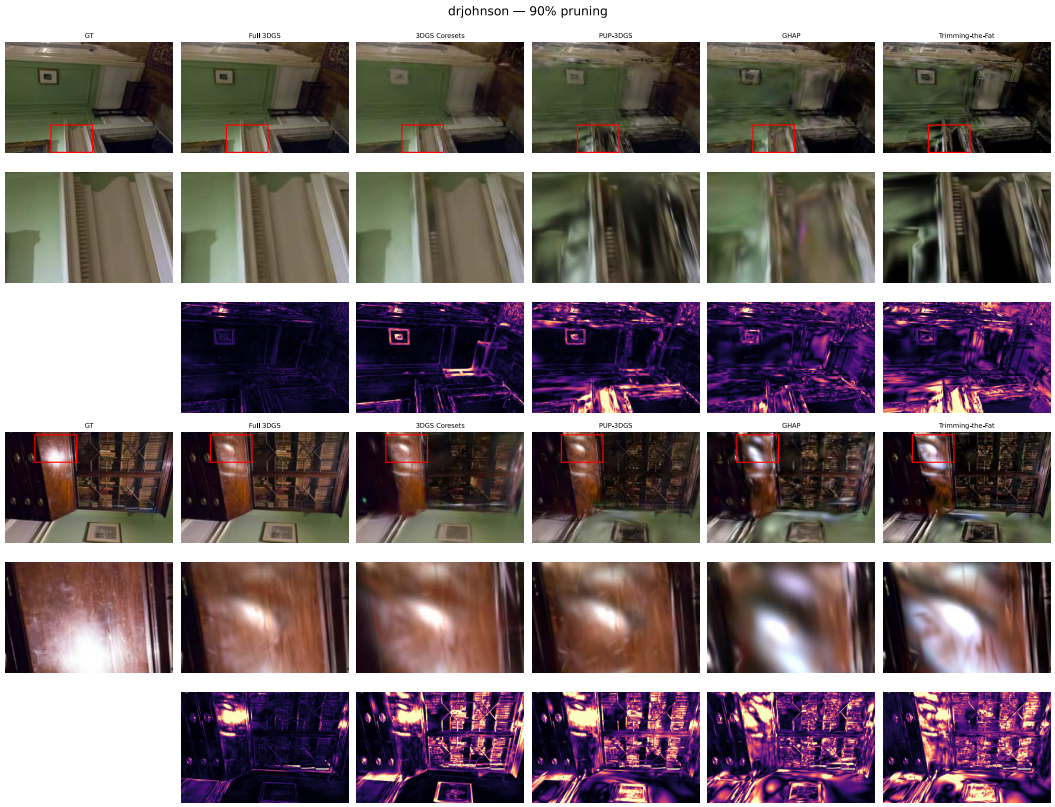

Figure 14: drjohnson, 90% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras IMG_6483.jpg, IMG_6500.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.



Figure 15: truck, 90% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras 000193.jpg, 000121.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

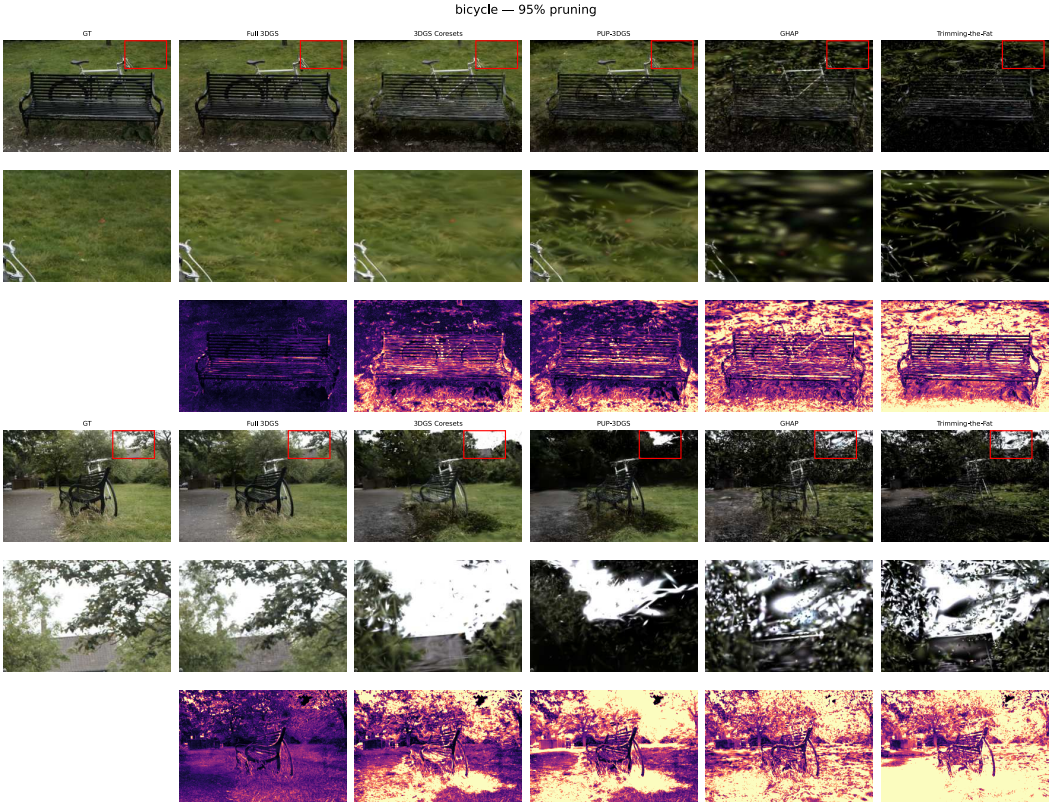

Figure 16: bicycle, 95% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras _DSC8800.JPG, _DSC8695.JPG, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

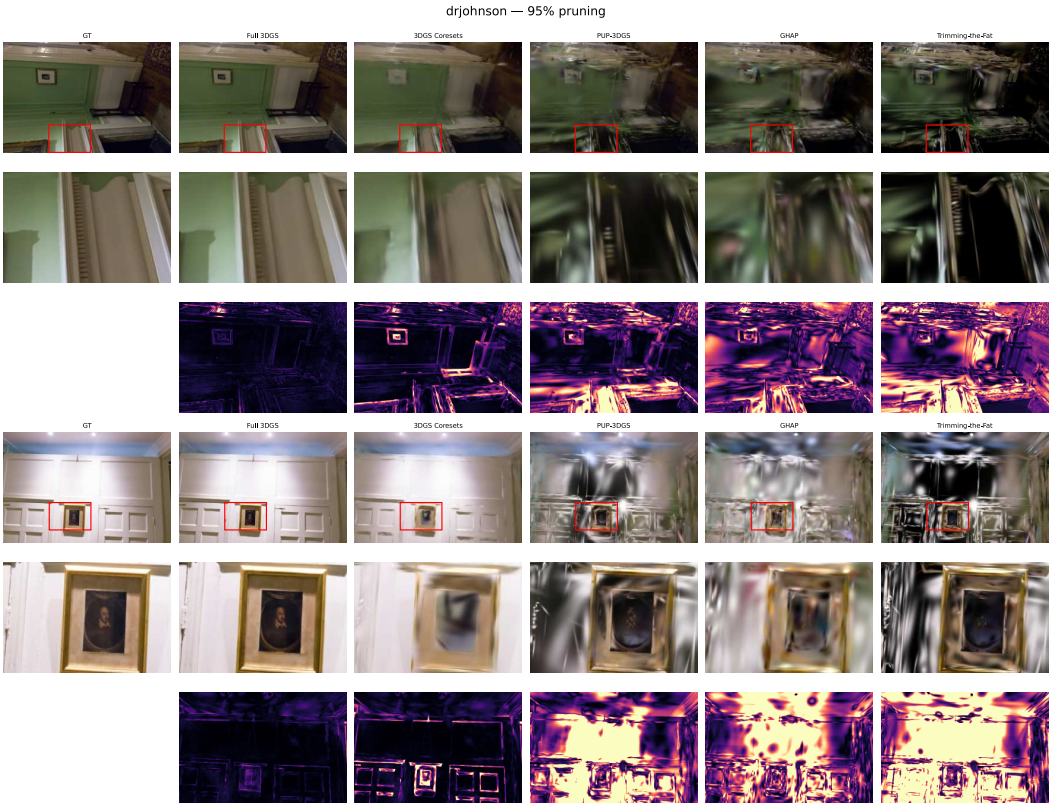

Figure 17: drjohnson, 95% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras IMG_6483.jpg, IMG_6366.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.



Figure 18: truck, 95% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras 000193.jpg, 000121.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

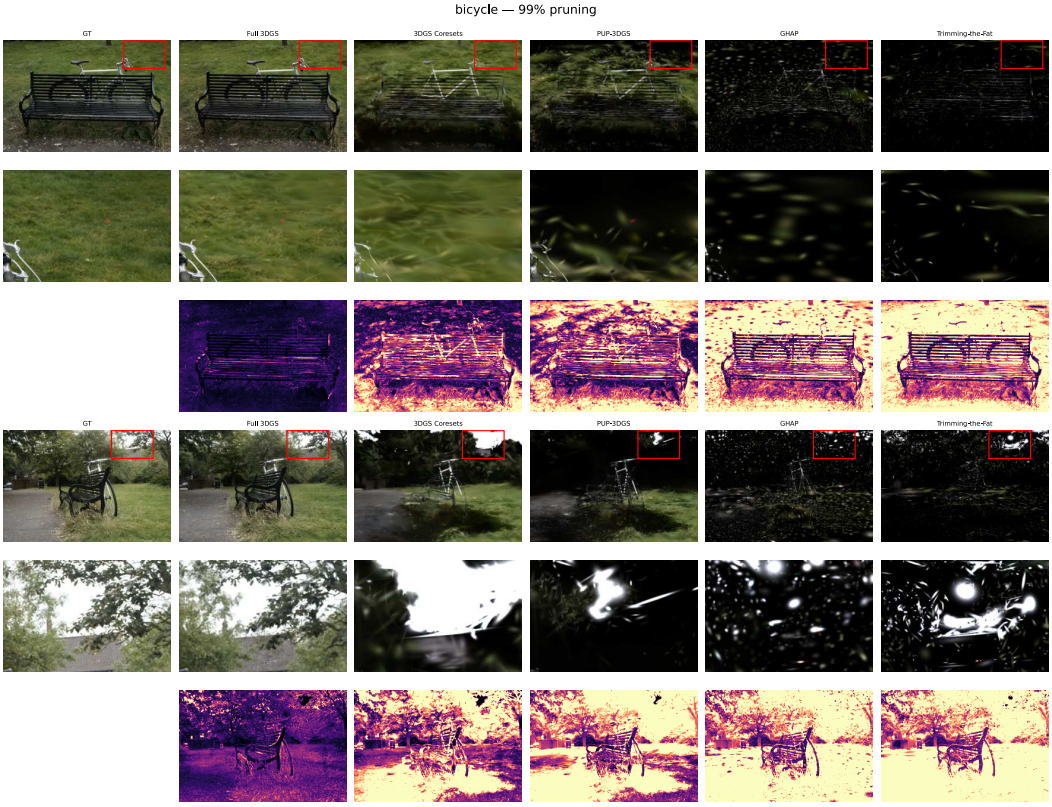

Figure 19: bicycle, 99% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras _DSC8800.JPG, _DSC8695.JPG, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

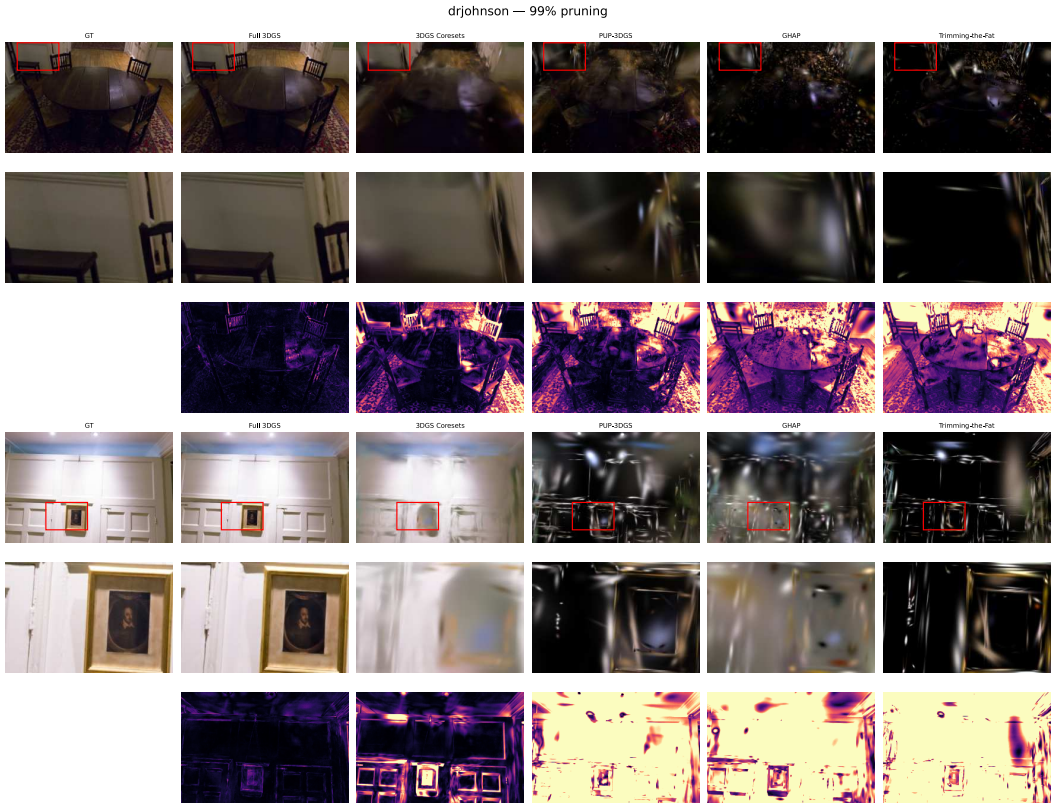

Figure 20: drjohnson, 99% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras IMG_6552.jpg, IMG_6366.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

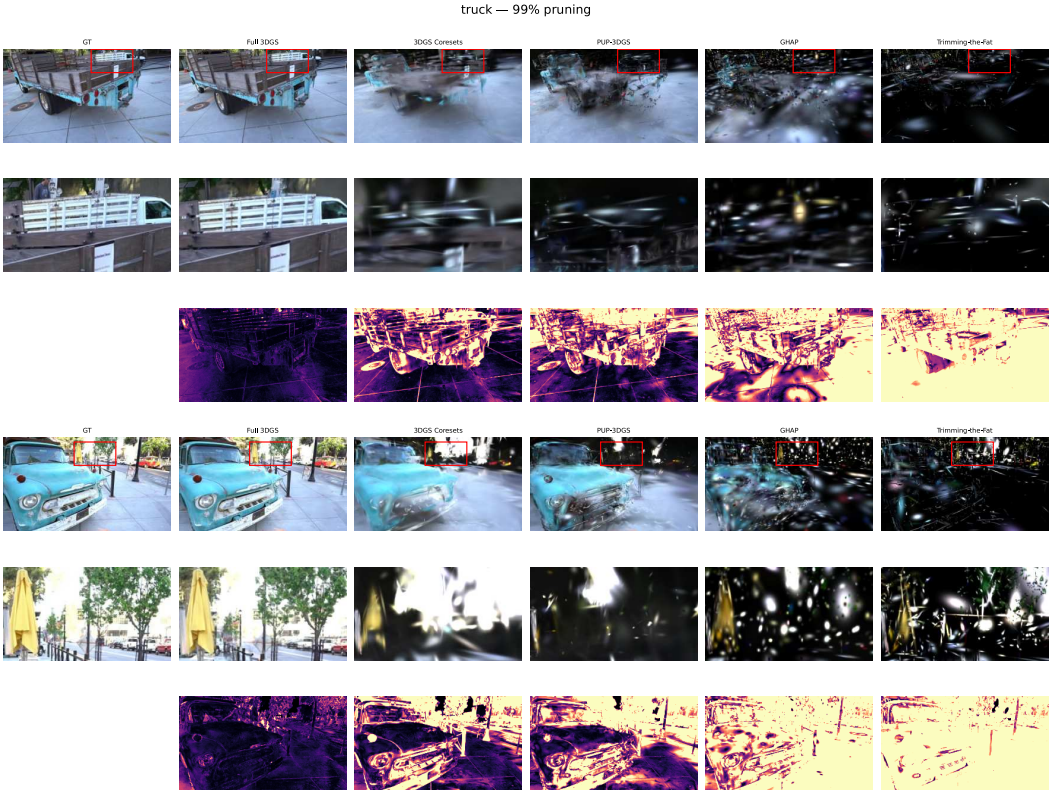

Figure 21: truck, 99% pruning, no recovery. Columns: ground truth, Full 3DGS, ours, PUP, GHAP and Trim. The upper/lower blocks show selected-quality/shared-challenging cameras 000193.jpg, 000233.jpg, each with a full view, aligned crop and mean absolute RGB error to the photograph (common range $[0, 0.25]$). Selection and crop coordinates are fixed across methods.

E.1 Very-short-recovery renders

Figure 22 provides the promised visual comparison between prune-only and very-short-recovery settings at 80, 85 and 90% pruning. It uses existing saved outputs from drjohnson with the same camera and full-frame layout at zero and 100 recovery iterations. The recorded checkpoint, configuration, camera and image identities were checked before inclusion; the numerical recovery curves are unchanged.

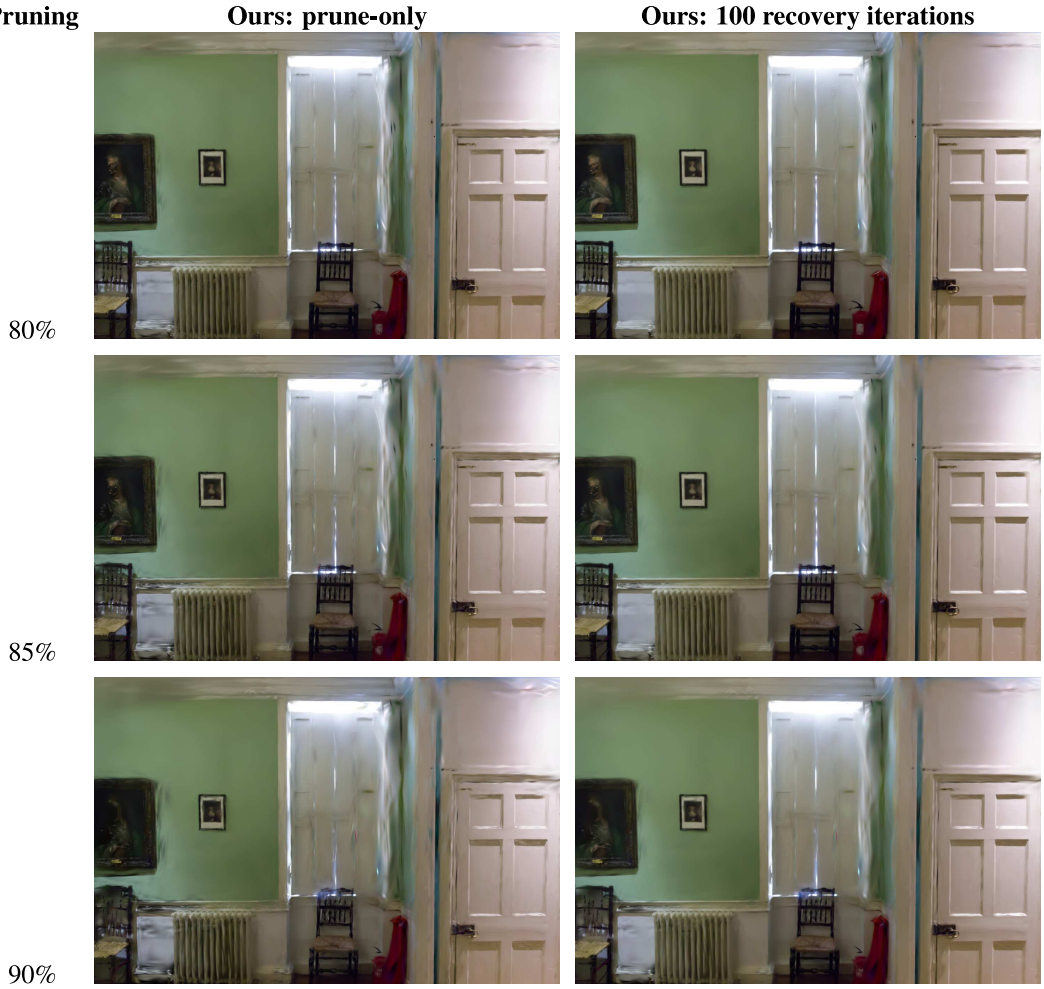

Figure 22: Matched prune-only and very-short-recovery renders on drjohnson across three pruning ratios. The columns show the saved iteration-30000 pruned output and the corresponding iteration-30100 output after 100 recovery iterations. All six full-frame images use the same saved held-out camera (`00000.png`), resolution and framing. The camera is fixed by the first saved filename, not selected by a quality metric. These existing renders complement the quantitative recovery curves; no new rendering or optimization was performed.

E.2 Full-reference worst-competitive views

The corrected full-view supplement provides an independent deterministic selection by error relative to *Full 3DGS*, not the photograph. Its worst-competitive camera minimizes the gap between the best competitor's Full-reference MAE and ours, with no favorable-camera eligibility filter. The two existing 99% sheets below retain all five quarter-image crop rules (best/worst Full similarity, best/worst competitive margin, and nearest-rank median Full error) and their common error-map range $[0, 0.66797388]$. This range is the saved global p99 display limit; values above it are saturated, not silently discarded. GT, Full and every reduced method use aligned full views and crops. These sheets

expose smoothing of fine geometry, loss of low-signal content, and local regions where a competitor can better match Full even when our overall quality is stronger. These are two worst-competitive views, not the rebuttal's specific set of six median-error crops. These sheets are prune-only; the saved very-short-recovery renders are presented separately above.

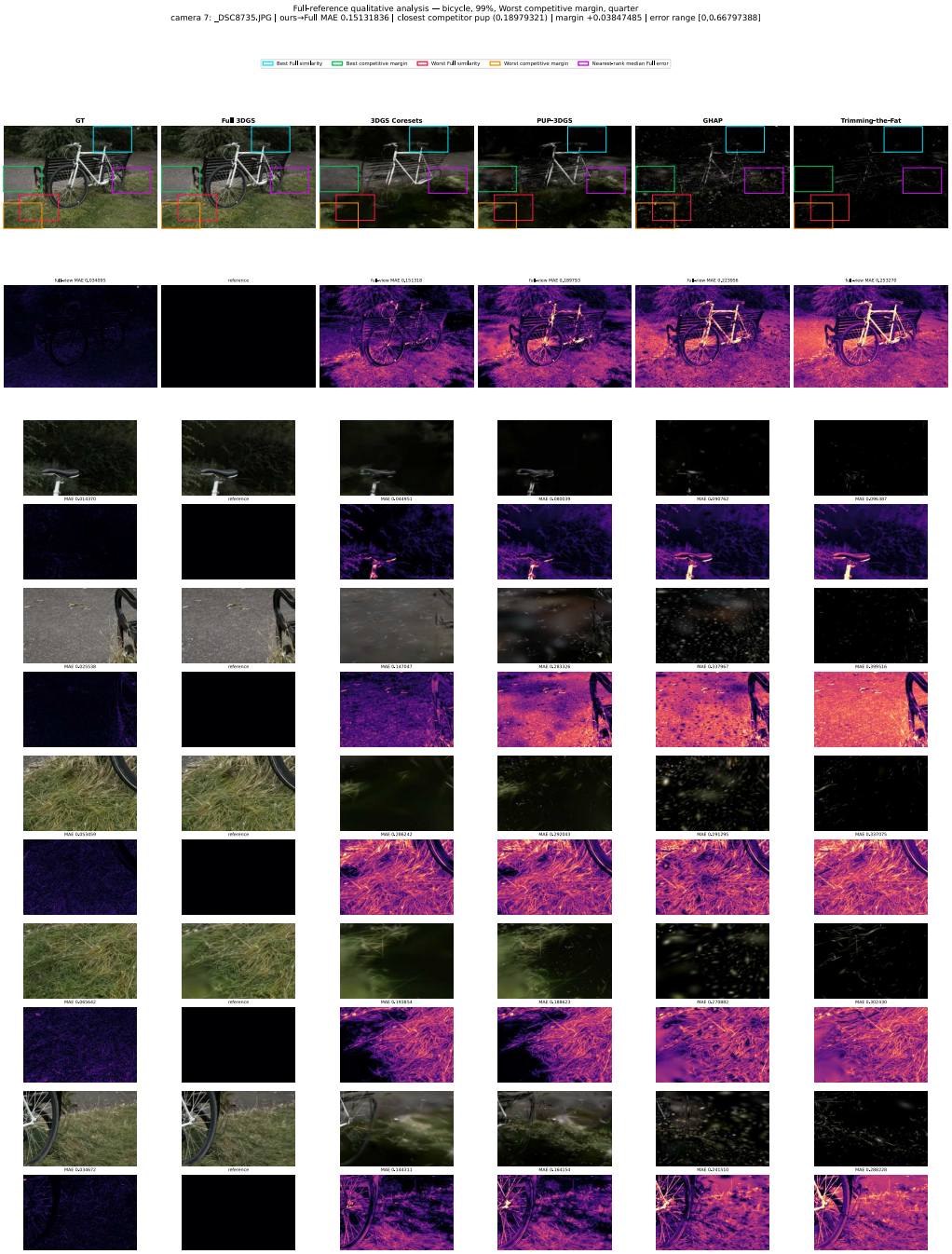

Figure 23: bicycle, 99% pruning, no recovery: existing corrected full-view sheet for the worst competitive-margin camera. Columns are GT, Full 3DGS, ours, PUP, GHAP and Trim. Five aligned quarter-image crop rules and error maps are retained. Here MAE and maps are relative to Full 3DGS, not GT; the common display range is $[0, 0.66797388]$.

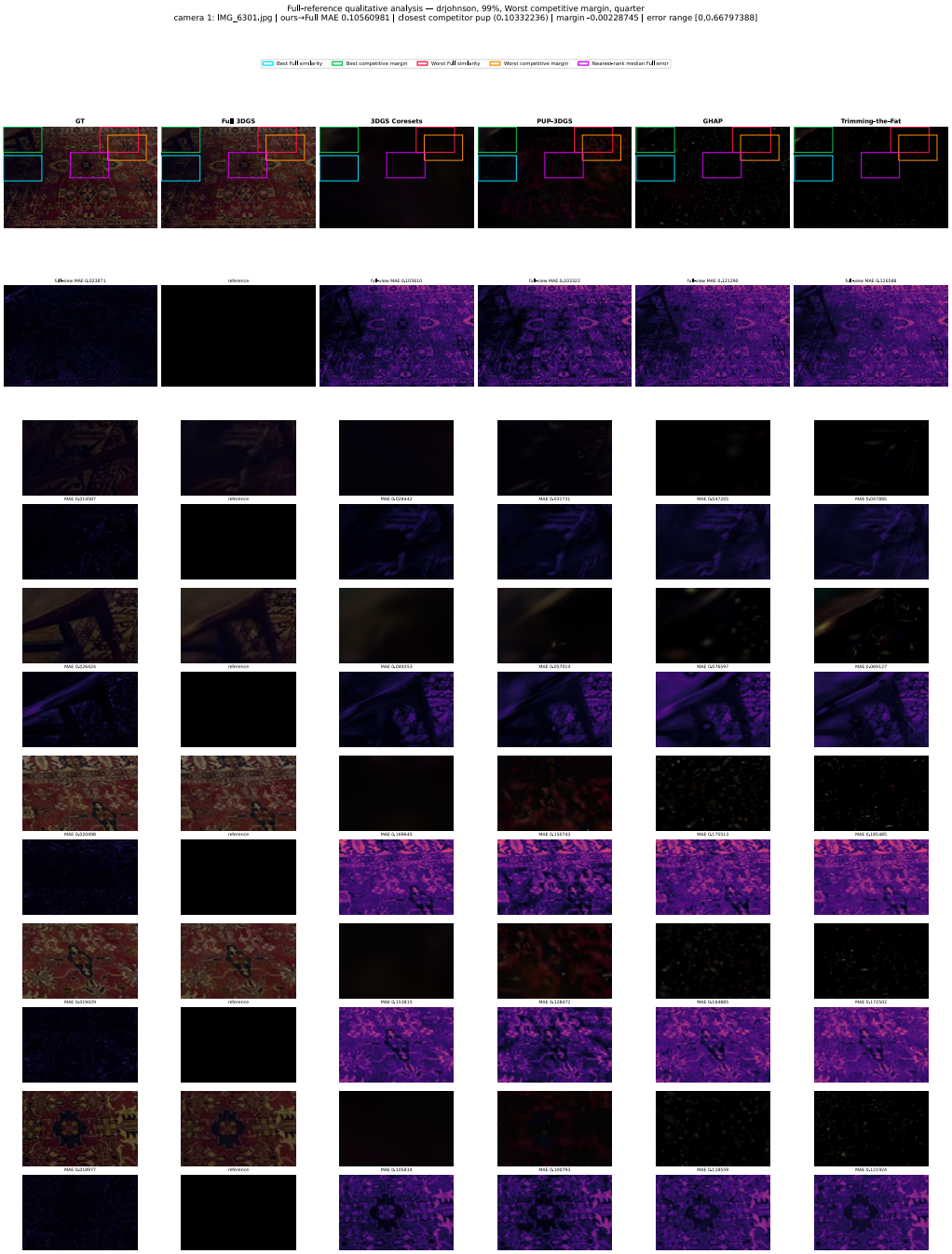

Figure 24: drjohnson, 99% pruning, no recovery: existing corrected full-view sheet for the worst competitive-margin camera. Columns are GT, Full 3DGS, ours, PUP, GHAP and Trim. Five aligned quarter-image crop rules and error maps are retained. Here MAE and maps are relative to Full 3DGS, not GT; the common display range is $[0, 0.66797388]$.

F Additional Experimental Results

This section contains the full experimental figures supporting the main results: prune-only tables and all-dataset curves in tables 26 to 28 and figure 25, short-recovery comparisons in figures 26 to 31, per-scene prune-only graphs in figures 32 to 34, and full ablations/recovery curves in figures 35 to 40 and 42.

Prune-only scene-level table.

Table 26: Prune-only results at prune ratios 0.80, 0.85, and 0.90 over all 13 scenes. Rows are scenes split into PSNR / SSIM / LPIPS, with each bracket followed by a per-dataset average row, and columns are grouped by prune ratio and method. Bold indicates best per scene/metric within each prune-ratio group; colored cells mark the top three methods.

Scene	Metric	Prune 0.80					Prune 0.85					Prune 0.90				
		GHAP	PUP	Trim	Unif	Ours	GHAP	PUP	Trim	Unif	Ours	GHAP	PUP	Trim	Unif	Ours
Flowers	PSNR ↑	16.06	16.89	14.34	10.71	**17.23**	15.18	15.69	12.74	10.39	**16.66**	14.01	14.31	11.12	10.03	**15.91**
	SSIM ↑	0.350	**0.454**	0.369	0.224	0.428	0.314	**0.418**	0.306	0.199	0.404	0.267	0.367	0.227	0.174	**0.372**
	LPIPS ↓	0.525	**0.456**	0.501	0.603	0.481	0.550	**0.483**	0.543	0.626	0.503	0.584	**0.521**	0.600	0.651	0.532
Treehill	PSNR ↑	18.47	18.03	13.00	13.23	**19.21**	17.70	17.08	11.97	12.63	**18.74**	16.55	15.89	10.96	12.39	**18.10**
	SSIM ↑	0.432	0.542	0.281	0.399	**0.576**	0.405	0.509	0.227	0.368	**0.553**	0.367	0.461	0.172	0.349	**0.516**
	LPIPS ↓	0.512	0.448	0.553	0.537	**0.417**	0.531	0.479	0.577	0.558	**0.442**	0.555	0.520	0.602	0.578	**0.479**
Bicycle	PSNR ↑	19.07	18.88	15.41	15.11	**19.93**	17.90	17.82	14.04	14.47	**19.12**	16.19	16.61	12.52	14.07	**18.01**
	SSIM ↑	0.477	0.579	0.444	0.378	**0.588**	0.430	0.530	0.364	0.338	**0.549**	0.363	0.464	0.263	0.309	**0.494**
	LPIPS ↓	0.433	0.360	0.426	0.502	**0.357**	0.460	0.394	0.469	0.531	**0.385**	0.498	0.440	0.522	0.557	**0.427**
Bonsai	PSNR ↑	21.99	19.03	**22.90**	12.42	19.86	20.38	16.68	**21.24**	11.79	19.16	18.09	14.10	**19.17**	11.31	18.24
	SSIM ↑	0.726	**0.774**	0.764	0.473	0.736	0.676	0.701	**0.706**	0.426	0.698	0.602	0.587	0.624	0.359	**0.651**
	LPIPS ↓	0.388	0.322	**0.310**	0.427	0.344	0.422	0.367	**0.351**	0.453	0.377	0.466	0.424	**0.402**	0.487	0.417
Counter	PSNR ↑	21.16	20.14	21.30	16.26	**22.83**	19.83	17.94	19.34	14.85	**21.58**	17.93	15.19	17.03	13.30	**19.92**
	SSIM ↑	0.691	0.780	0.753	0.639	**0.797**	0.650	0.725	0.691	0.586	**0.764**	0.588	0.634	0.604	0.504	**0.717**
	LPIPS ↓	0.381	0.297	**0.295**	0.380	0.301	0.413	0.339	0.336	0.409	**0.334**	0.455	0.396	0.394	0.458	**0.377**
Garden	PSNR ↑	18.86	17.83	14.35	16.75	**19.36**	17.41	16.44	12.77	15.94	**18.26**	15.33	14.75	11.09	15.07	**16.95**
	SSIM ↑	0.491	0.597	0.468	0.510	**0.604**	0.435	0.531	0.383	0.467	**0.546**	0.360	0.446	0.284	0.418	**0.476**
	LPIPS ↓	0.402	**0.304**	0.407	0.409	0.313	0.435	**0.352**	0.462	0.443	0.356	0.483	**0.415**	0.526	0.481	0.416
Kitchen	PSNR ↑	19.24	17.37	18.34	14.27	**19.31**	17.69	15.19	16.09	13.36	**17.96**	15.68	12.89	13.52	11.92	**16.40**
	SSIM ↑	0.659	0.733	0.726	0.615	**0.766**	0.603	0.657	0.651	0.569	**0.717**	0.525	0.550	0.548	0.467	**0.651**
	LPIPS ↓	0.369	**0.257**	0.267	0.327	0.258	0.412	0.310	0.320	0.361	**0.299**	0.466	0.385	0.394	0.437	**0.358**
Room	PSNR ↑	23.17	24.30	**24.34**	18.03	21.51	21.81	**22.44**	22.20	16.76	20.78	19.56	**19.89**	19.63	14.73	19.78
	SSIM ↑	0.762	**0.852**	0.822	0.715	0.796	0.729	**0.819**	0.773	0.669	0.774	0.675	**0.766**	0.696	0.588	0.744
	LPIPS ↓	0.367	**0.272**	0.284	0.386	0.314	0.395	**0.303**	0.324	0.417	0.339	0.437	**0.352**	0.377	0.465	0.376
Stump	PSNR ↑	20.00	22.13	18.06	16.86	**22.34**	18.93	20.49	16.61	16.19	**21.50**	17.50	18.51	15.05	15.65	**20.12**
	SSIM ↑	0.487	0.677	0.492	0.430	**0.683**	0.442	0.623	0.409	0.390	**0.655**	0.377	0.540	0.303	0.353	**0.599**
	LPIPS ↓	0.446	0.314	0.409	0.476	**0.305**	0.471	0.352	0.451	0.502	**0.329**	0.505	0.406	0.505	0.526	**0.374**
Mip-NeRF 360	PSNR ↑	19.78	19.40	18.01	14.85	**20.18**	18.54	17.75	16.33	14.04	**19.30**	16.76	15.79	14.45	13.16	**18.16**
	SSIM ↑	0.564	**0.665**	0.569	0.487	0.664	0.521	0.612	0.501	0.446	**0.629**	0.458	0.535	0.413	0.391	**0.580**
	LPIPS ↓	0.425	**0.337**	0.384	0.450	0.343	0.454	0.375	0.426	0.478	**0.374**	0.494	0.429	0.480	0.516	**0.417**
DrJohnson	PSNR ↑	**24.48**	23.88	21.68	22.17	23.74	22.92	21.84	19.42	19.88	**22.95**	20.69	19.14	16.59	18.00	**21.92**
	SSIM ↑	0.812	**0.832**	0.770	0.776	0.813	0.781	**0.798**	0.711	0.730	0.796	0.730	0.743	0.608	0.688	**0.773**
	LPIPS ↓	0.371	**0.320**	0.333	0.399	0.346	0.401	**0.354**	0.368	0.429	0.367	0.441	0.400	0.418	0.461	**0.396**
Playroom	PSNR ↑	24.85	24.70	22.36	16.85	**25.08**	23.18	22.67	19.84	15.25	**23.95**	20.36	19.90	16.44	13.39	**22.39**
	SSIM ↑	0.827	**0.850**	0.802	0.727	0.849	0.801	0.825	0.755	0.686	**0.831**	0.757	0.786	0.675	0.620	**0.807**
	LPIPS ↓	0.359	**0.314**	0.337	0.454	0.325	0.388	**0.342**	0.377	0.491	0.350	0.432	0.386	0.440	0.541	**0.385**
Deep Blending	PSNR ↑	**24.67**	24.29	22.02	19.51	24.41	23.05	22.25	19.63	17.56	**23.45**	20.53	19.52	16.52	15.69	**22.16**
	SSIM ↑	0.820	**0.841**	0.786	0.752	0.831	0.791	0.812	0.733	0.708	**0.814**	0.744	0.765	0.642	0.654	**0.790**
	LPIPS ↓	0.365	**0.317**	0.335	0.426	0.336	0.395	**0.348**	0.372	0.460	0.359	0.437	0.393	0.429	0.501	**0.390**
Train	PSNR ↑	16.61	15.15	15.47	10.68	**16.88**	15.59	13.79	14.12	10.01	**16.21**	14.20	12.05	12.29	8.72	**15.32**
	SSIM ↑	0.573	**0.629**	0.606	0.462	0.612	0.525	0.572	0.544	0.420	**0.579**	0.463	0.494	0.458	0.338	**0.537**
	LPIPS ↓	0.415	0.373	**0.362**	0.469	0.382	0.455	0.419	**0.408**	0.497	0.417	0.505	0.481	0.473	0.534	**0.461**
Truck	PSNR ↑	18.61	16.96	16.76	12.79	**19.38**	17.25	15.98	14.94	12.32	**18.52**	15.47	14.71	12.64	11.73	**17.37**
	SSIM ↑	0.638	**0.746**	0.657	0.591	0.739	0.584	**0.714**	0.583	0.567	0.706	0.514	0.661	0.484	0.536	**0.662**
	LPIPS ↓	0.367	**0.247**	0.324	0.376	0.272	0.408	**0.278**	0.376	0.399	0.307	0.461	**0.331**	0.443	0.431	0.354
Tanks & Temples	PSNR ↑	17.61	16.05	16.11	11.74	**18.13**	16.42	14.89	14.53	11.17	**17.37**	14.83	13.38	12.46	10.22	**16.35**
	SSIM ↑	0.606	**0.687**	0.632	0.526	0.675	0.555	**0.643**	0.564	0.494	0.643	0.488	0.577	0.471	0.437	**0.599**
	LPIPS ↓	0.391	**0.310**	0.343	0.423	0.327	0.431	**0.348**	0.392	0.448	0.362	0.483	**0.406**	0.458	0.482	0.407

Table 27: Prune-only results at prune ratios 0.95, 0.97, and 0.99 over all 13 scenes. Rows are scenes split into PSNR / SSIM / LPIPS, with each bracket followed by a per-dataset average row, and columns are grouped by prune ratio and method. Bold indicates best per scene/metric within each prune-ratio group; colored cells mark the top three methods.

Scene	Metric	Prune 0.95					Prune 0.97					Prune 0.99				
		GHAP	PUP	Trim	Unif	Ours	GHAP	PUP	Trim	Unif	Ours	GHAP	PUP	Trim	Unif	Ours
Flowers	PSNR ↑	12.08	12.45	9.54	9.65	**14.74**	10.94	11.49	8.85	9.44	**13.92**	8.91	10.31	8.03	9.15	**12.60**
	SSIM ↑	0.198	0.278	0.140	0.146	**0.322**	0.159	0.220	0.101	0.132	**0.289**	0.092	0.141	0.052	0.110	**0.241**
	LPIPS ↓	0.642	0.584	0.671	0.682	**0.581**	0.680	0.625	0.711	0.699	**0.617**	0.761	0.690	0.771	0.727	**0.687**
Treehill	PSNR ↑	14.50	14.18	9.98	11.99	**17.13**	13.01	13.27	9.56	11.72	**16.51**	10.24	11.61	9.05	10.62	**15.51**
	SSIM ↑	0.308	0.384	0.118	0.324	**0.457**	0.265	0.336	0.089	0.306	**0.422**	0.159	0.242	0.059	0.245	**0.380**
	LPIPS ↓	0.594	0.577	0.634	0.604	**0.537**	0.622	0.608	0.655	0.619	**0.571**	0.674	0.654	0.682	0.655	**0.626**
Bicycle	PSNR ↑	13.66	14.91	10.97	13.66	**16.45**	12.13	13.80	10.34	13.46	**15.60**	10.38	11.78	9.70	13.07	**14.40**
	SSIM ↑	0.254	0.365	0.145	0.278	**0.416**	0.178	0.304	0.097	0.263	**0.376**	0.081	0.192	0.042	0.234	**0.318**
	LPIPS ↓	0.560	0.510	0.594	0.587	**0.492**	0.604	0.554	0.635	0.602	**0.534**	0.679	0.627	0.693	0.633	**0.611**
Bonsai	PSNR ↑	14.93	11.75	16.03	10.60	**16.94**	12.94	10.73	14.08	10.11	**15.98**	10.71	9.75	11.42	9.60	**14.36**
	SSIM ↑	0.474	0.415	0.488	0.245	**0.585**	0.380	0.310	0.397	0.186	**0.545**	0.219	0.171	0.231	0.093	**0.474**
	LPIPS ↓	0.527	0.497	0.475	0.534	**0.468**	0.561	0.533	0.514	0.552	**0.499**	0.605	0.568	0.564	0.573	**0.546**
Counter	PSNR ↑	15.24	12.51	13.95	11.62	**17.49**	13.52	11.47	12.42	10.94	**16.07**	11.00	10.09	10.23	9.64	**13.78**
	SSIM ↑	0.498	0.499	0.471	0.363	**0.647**	0.438	0.423	0.390	0.281	**0.606**	0.303	0.281	0.238	0.148	**0.536**
	LPIPS ↓	0.518	0.477	0.477	0.531	**0.443**	0.556	0.521	0.523	0.563	**0.485**	0.616	0.589	0.594	0.611	**0.553**
Garden	PSNR ↑	12.45	12.48	9.21	13.92	**15.25**	10.77	11.26	8.42	13.28	**14.29**	8.48	9.36	7.53	10.13	**12.58**
	SSIM ↑	0.253	0.330	0.165	0.348	**0.387**	0.190	0.266	0.113	0.306	**0.342**	0.091	0.163	0.053	0.182	**0.276**
	LPIPS ↓	0.556	0.509	0.604	0.537	**0.503**	0.607	0.566	0.649	0.573	**0.555**	0.700	0.663	0.724	0.668	**0.647**
Kitchen	PSNR ↑	12.86	10.12	10.60	9.37	**14.44**	11.06	8.81	9.33	7.91	**13.44**	8.31	7.16	7.54	6.48	**12.02**
	SSIM ↑	0.404	0.392	0.402	0.292	**0.556**	0.326	0.304	0.322	0.198	**0.502**	0.205	0.168	0.193	0.080	**0.421**
	LPIPS ↓	0.544	0.506	0.512	0.556	**0.448**	0.589	0.576	0.581	0.618	**0.506**	0.662	0.676	0.680	0.684	**0.608**
Room	PSNR ↑	16.33	16.89	15.97	12.36	**18.18**	14.36	15.27	13.79	11.49	**17.12**	11.29	12.71	10.74	9.76	**15.29**
	SSIM ↑	0.585	0.672	0.568	0.459	**0.695**	0.514	0.607	0.481	0.391	**0.665**	0.352	0.469	0.290	0.244	**0.598**
	LPIPS ↓	0.501	**0.428**	0.454	0.536	0.434	0.543	0.475	0.501	0.567	**0.471**	0.613	0.551	0.583	0.618	**0.523**
Stump	PSNR ↑	15.31	16.14	13.45	15.07	**18.02**	14.06	14.87	12.77	14.69	**16.85**	12.38	13.27	11.99	14.00	**15.19**
	SSIM ↑	0.273	0.409	0.180	0.313	**0.491**	0.205	0.330	0.125	0.286	**0.421**	0.090	0.215	0.064	0.235	**0.321**
	LPIPS ↓	0.558	0.482	0.575	0.553	**0.455**	0.592	0.527	0.611	0.572	**0.508**	0.664	0.598	0.660	0.610	**0.594**
Mip-NeRF 360	PSNR ↑	14.15	13.49	12.19	12.02	**16.52**	12.53	12.33	11.06	11.45	**15.53**	10.19	10.67	9.58	10.27	**13.97**
	SSIM ↑	0.361	0.416	0.297	0.308	**0.506**	0.295	0.344	0.235	0.261	**0.463**	0.177	0.227	0.136	0.174	**0.396**
	LPIPS ↓	0.556	0.508	0.555	0.569	**0.484**	0.595	0.554	0.598	0.596	**0.527**	0.664	0.624	0.661	0.642	**0.600**
DrJohnson	PSNR ↑	16.95	15.64	13.01	15.00	**20.17**	14.77	13.67	11.32	12.72	**18.83**	11.30	10.94	9.39	9.47	**16.20**
	SSIM ↑	0.630	0.645	0.413	0.597	**0.733**	0.557	0.561	0.288	0.492	**0.703**	0.360	0.372	0.111	0.194	**0.630**
	LPIPS ↓	0.502	0.467	0.493	0.510	**0.439**	0.536	0.510	0.532	0.545	**0.468**	0.592	0.570	0.580	0.602	**0.516**
Playroom	PSNR ↑	15.98	15.77	11.66	9.93	**19.94**	13.19	13.34	8.96	8.33	**18.39**	8.98	9.33	5.92	6.22	**15.04**
	SSIM ↑	0.677	0.713	0.529	0.451	**0.771**	0.609	0.656	0.405	0.359	**0.748**	0.442	0.485	0.152	0.184	**0.690**
	LPIPS ↓	0.510	0.468	0.557	0.644	**0.439**	0.573	0.531	0.650	0.700	**0.474**	0.715	0.672	0.787	0.791	**0.537**
Deep Blending	PSNR ↑	16.46	15.70	12.34	12.47	**20.06**	13.98	13.51	10.14	10.53	**18.61**	10.14	10.13	7.66	7.84	**15.62**
	SSIM ↑	0.653	0.679	0.471	0.524	**0.752**	0.583	0.608	0.347	0.426	**0.725**	0.401	0.428	0.132	0.189	**0.660**
	LPIPS ↓	0.506	0.468	0.525	0.577	**0.439**	0.554	0.520	0.591	0.623	**0.471**	0.653	0.621	0.683	0.696	**0.526**
Train	PSNR ↑	11.81	10.18	9.72	7.95	**14.01**	10.32	8.99	8.40	7.38	**13.11**	8.00	7.43	6.84	6.49	**11.41**
	SSIM ↑	0.368	0.390	0.322	0.255	**0.481**	0.310	0.321	0.240	0.201	**0.449**	0.212	0.221	0.135	0.120	**0.400**
	LPIPS ↓	0.579	0.561	0.561	0.594	**0.523**	0.621	0.604	0.609	0.624	**0.561**	0.683	0.665	0.670	0.665	**0.620**
Truck	PSNR ↑	12.71	12.98	9.35	10.93	**15.62**	10.94	11.96	7.74	9.83	**14.54**	7.88	10.46	5.62	7.34	**12.55**
	SSIM ↑	0.414	0.566	0.323	0.490	**0.590**	0.353	0.506	0.228	0.393	**0.544**	0.237	0.420	0.074	0.207	**0.470**
	LPIPS ↓	0.534	**0.432**	0.538	0.483	0.436	0.582	0.495	0.588	0.555	**0.490**	0.667	**0.588**	0.666	0.651	0.592
Tanks & Temples	PSNR ↑	12.26	11.58	9.53	9.44	**14.81**	10.63	10.48	8.07	8.60	**13.83**	7.94	8.94	6.23	6.92	**11.98**
	SSIM ↑	0.391	0.478	0.323	0.372	**0.535**	0.332	0.414	0.234	0.297	**0.496**	0.224	0.320	0.104	0.163	**0.435**
	LPIPS ↓	0.557	0.496	0.550	0.538	**0.480**	0.601	0.549	0.599	0.590	**0.525**	0.675	0.627	0.668	0.658	**0.606**

Table 28: Per-dataset average prune-only results at prune ratios 0.80, 0.85, 0.90, 0.95, 0.97, and 0.99. Rows are prune ratios and methods; columns are grouped by dataset and metric. Bold indicates best per dataset/metric/ratio; colored cells mark the top three methods.

Prune	Method	Mip-NeRF 360			Deep Blending			Tanks & Temples		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
0.80	GHAP	19.78	0.564	0.425	**24.67**	0.820	0.365	17.61	0.606	0.391
	PUP	19.40	**0.665**	**0.337**	24.29	**0.841**	**0.317**	16.05	**0.687**	**0.310**
	Trim	18.01	0.569	0.384	22.02	0.786	0.335	16.11	0.632	0.343
	Unif	14.85	0.487	0.450	19.51	0.752	0.426	11.74	0.526	0.423
	Ours	**20.18**	0.664	0.343	24.41	0.831	0.336	**18.13**	0.675	0.327
0.85	GHAP	18.54	0.521	0.454	23.05	0.791	0.395	16.42	0.555	0.431
	PUP	17.75	0.612	0.375	22.25	0.812	**0.348**	14.89	**0.643**	**0.348**
	Trim	16.33	0.501	0.426	19.63	0.733	0.372	14.53	0.564	0.392
	Unif	14.04	0.446	0.478	17.56	0.708	0.460	11.17	0.494	0.448
	Ours	**19.30**	**0.629**	**0.374**	**23.45**	**0.814**	0.359	**17.37**	0.643	0.362
0.90	GHAP	16.76	0.458	0.494	20.53	0.744	0.437	14.83	0.488	0.483
	PUP	15.79	0.535	0.429	19.52	0.765	0.393	13.38	0.577	**0.406**
	Trim	14.45	0.413	0.480	16.52	0.642	0.429	12.46	0.471	0.458
	Unif	13.16	0.391	0.516	15.69	0.654	0.501	10.22	0.437	0.482
	Ours	**18.16**	**0.580**	**0.417**	**22.16**	**0.790**	**0.390**	**16.35**	**0.599**	0.407
0.95	GHAP	14.15	0.361	0.556	16.46	0.653	0.506	12.26	0.391	0.557
	PUP	13.49	0.416	0.508	15.70	0.679	0.468	11.58	0.478	0.496
	Trim	12.19	0.297	0.555	12.34	0.471	0.525	9.53	0.323	0.550
	Unif	12.02	0.308	0.569	12.47	0.524	0.577	9.44	0.372	0.538
	Ours	**16.52**	**0.506**	**0.484**	**20.06**	**0.752**	**0.439**	**14.81**	**0.535**	**0.480**
0.97	GHAP	12.53	0.295	0.595	13.98	0.583	0.554	10.63	0.332	0.601
	PUP	12.33	0.344	0.554	13.51	0.608	0.520	10.48	0.414	0.549
	Trim	11.06	0.235	0.598	10.14	0.347	0.591	8.07	0.234	0.599
	Unif	11.45	0.261	0.596	10.53	0.426	0.623	8.60	0.297	0.590
	Ours	**15.53**	**0.463**	**0.527**	**18.61**	**0.725**	**0.471**	**13.83**	**0.496**	**0.525**
0.99	GHAP	10.19	0.177	0.664	10.14	0.401	0.653	7.94	0.224	0.675
	PUP	10.67	0.227	0.624	10.13	0.428	0.621	8.94	0.320	0.627
	Trim	9.58	0.136	0.661	7.66	0.132	0.683	6.23	0.104	0.668
	Unif	10.27	0.174	0.642	7.84	0.189	0.696	6.92	0.163	0.658
	Ours	**13.97**	**0.396**	**0.600**	**15.62**	**0.660**	**0.526**	**11.98**	**0.435**	**0.606**

Prune-only curves and delta trends.

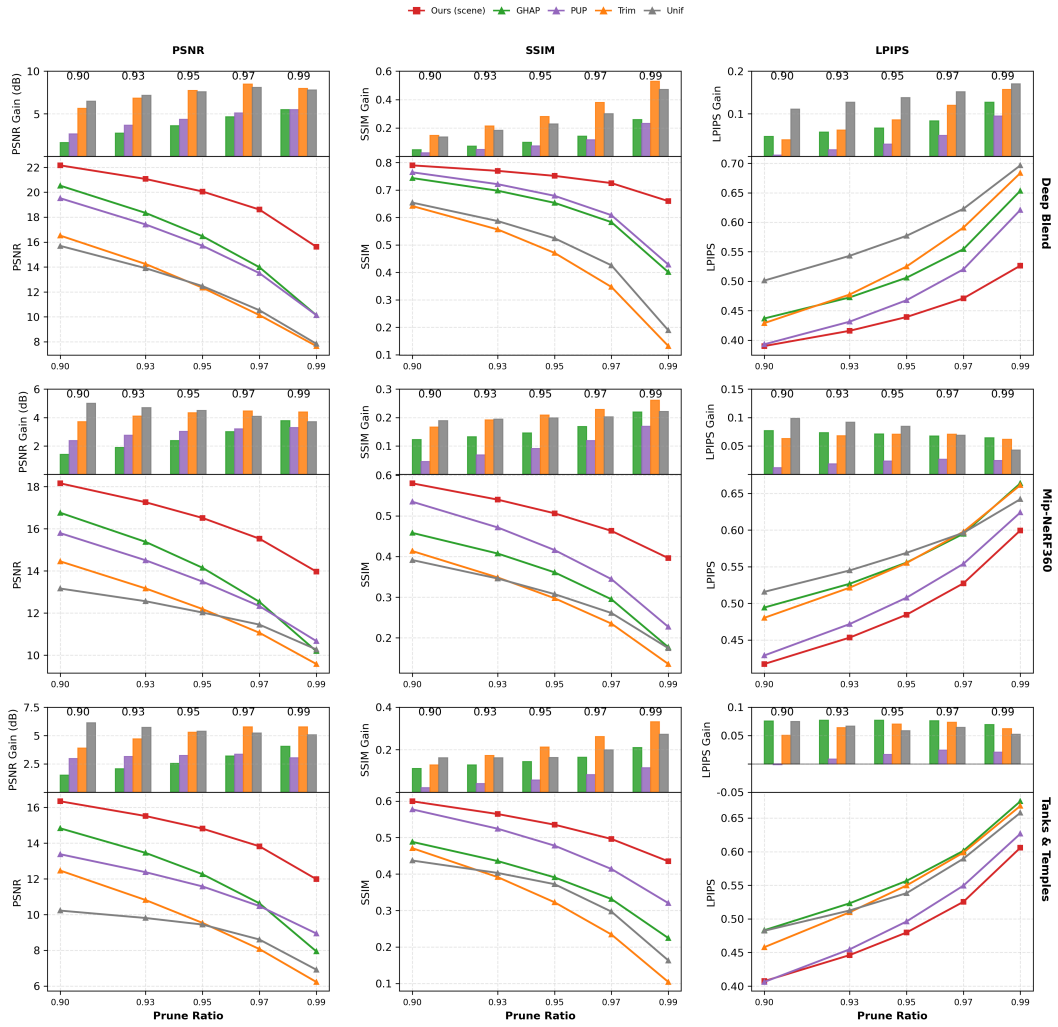

Figure 25: Prune-only averages per dataset.

Short-recovery curves with prune ratio on the x-axis.

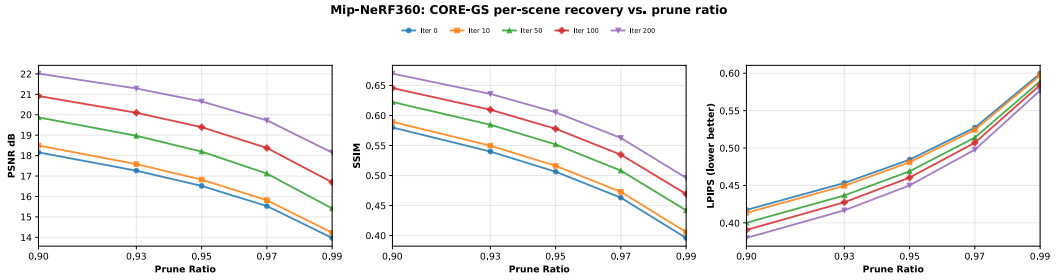

Figure 26: Transposed short-recovery curves for our scene-variant on mip-NeRF 360. The x-axis is prune ratio and each curve corresponds to a recovery iteration.

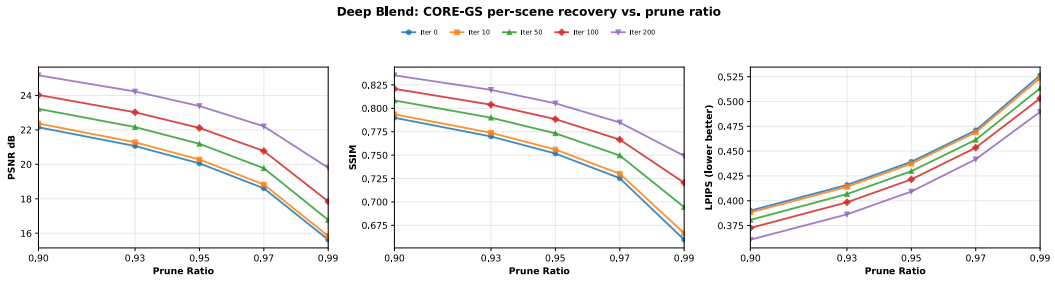

Figure 27: Transposed short-recovery curves for our scene-variant on Deep Blending, using the same layout as figure 26.

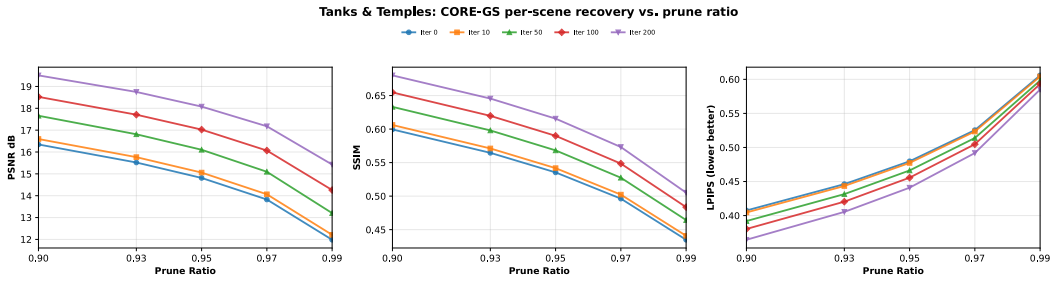

Figure 28: Transposed short-recovery curves for our scene-variant on Tanks & Temples, using the same layout as figure 26.

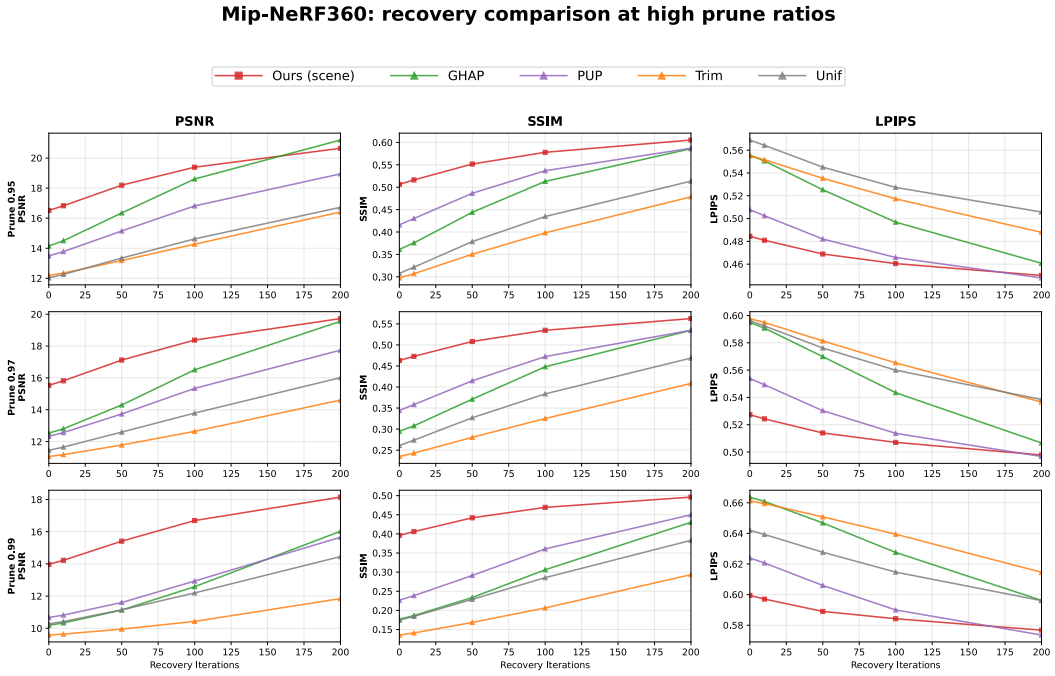

Figure 29: Short-recovery comparison on mip-NeRF 360 at prune ratios 0.95, 0.97, and 0.99. Curves compare our scene-variant against GHAP, PUP 3D-GS, Trimming the Fat, and Uniform.

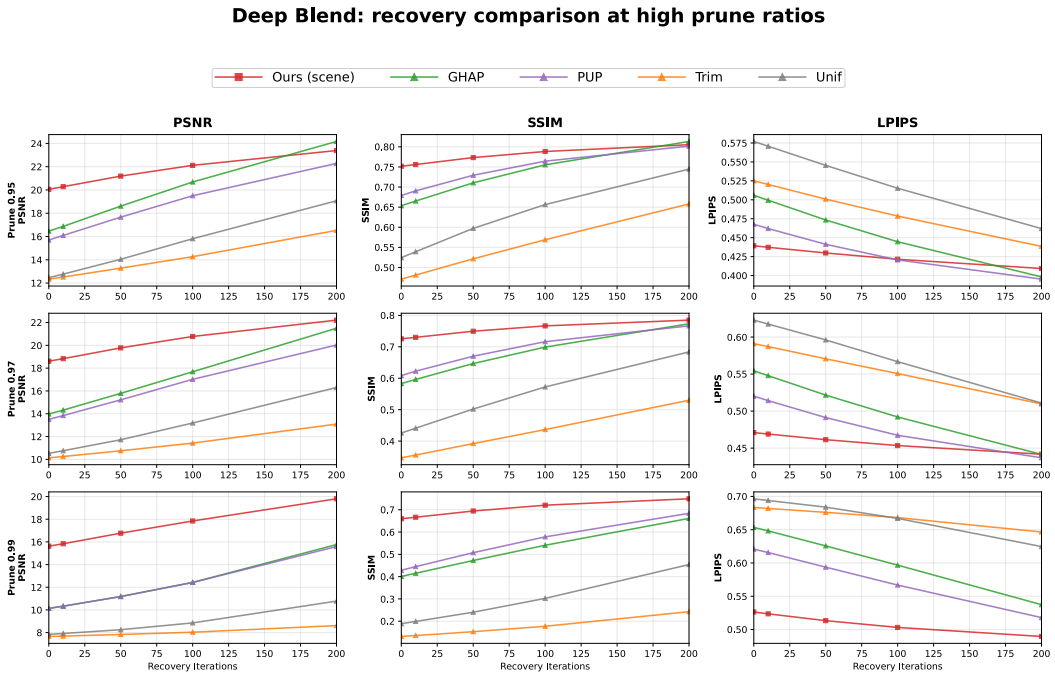

Figure 30: Short-recovery comparison on Deep Blending at prune ratios 0.95, 0.97, and 0.99.

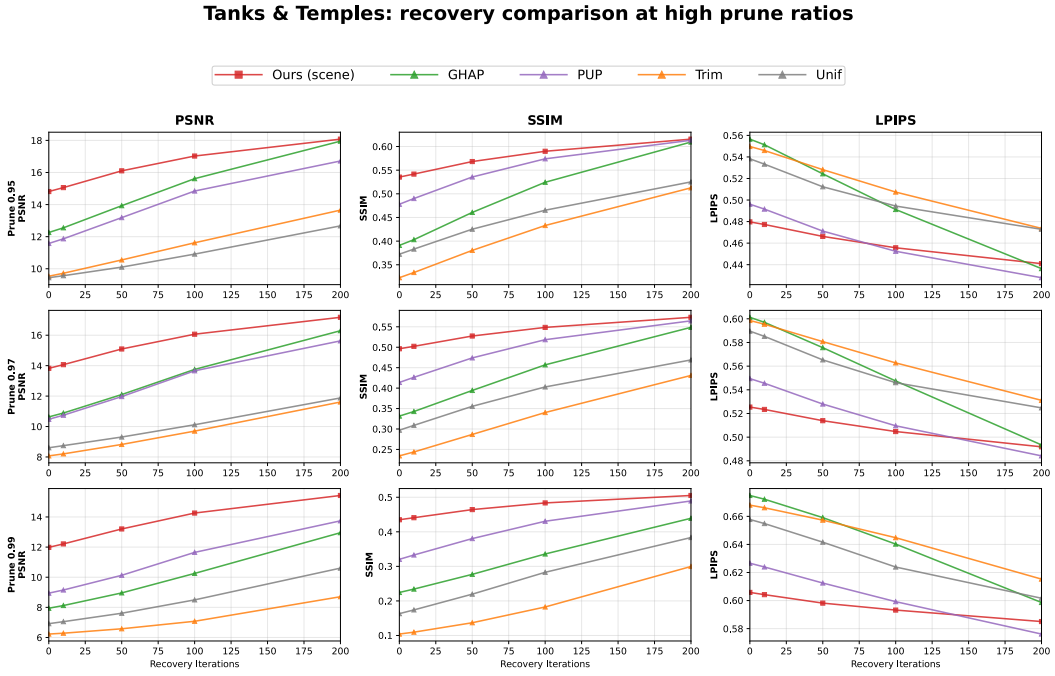

Figure 31: Short-recovery comparison on Tanks & Temples at prune ratios 0.95, 0.97, and 0.99.

G Prune-only Per-scene Graphs

This appendix section expands the main average-over-scenes prune-only graphs into per-scene plots. The three figures below cover 13 scenes $\times$ 3 metrics, for a total of 39 prune-only panels. Each panel plots metric value on the y-axis against prune ratio on the x-axis, with curves for CORE-GS variants and the competing methods. These plots support per-scene variability analysis and show whether averaged trends hide scene-specific failures or strengths.

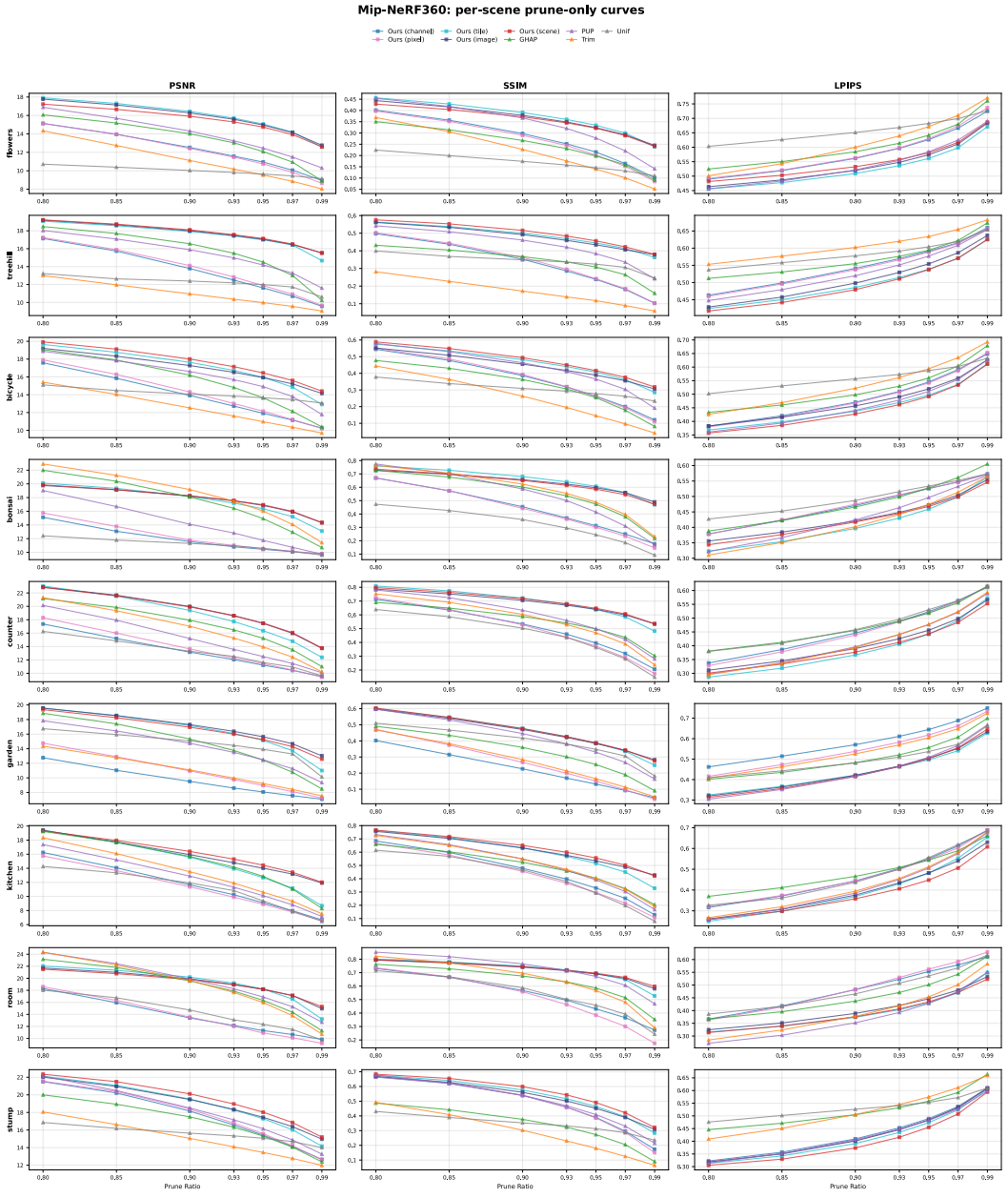

Figure 32: Per-scene prune-only curves for mip-NeRF 360.

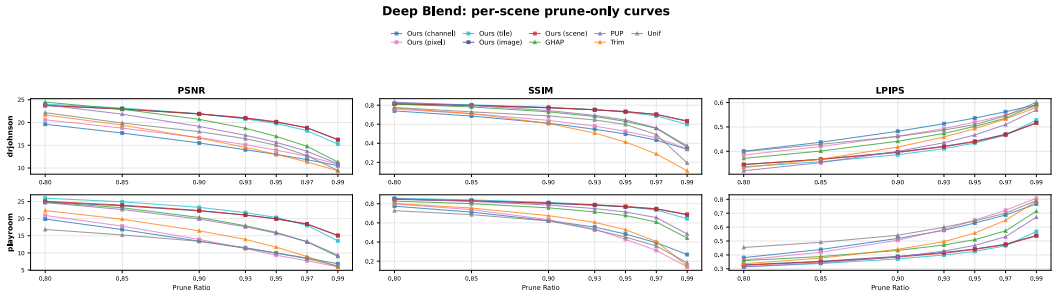

Figure 33: Per-scene prune-only curves for Deep Blending.

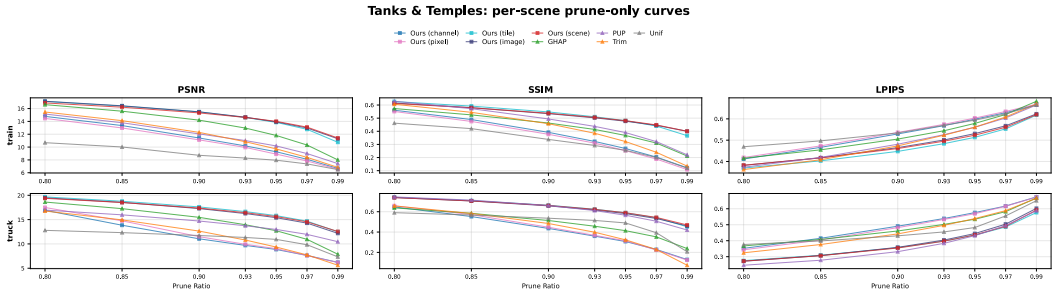

Figure 34: Per-scene prune-only curves for Tanks & Temples.

H Additional Ablations and Recovery Curves

The following figures report the full CORE-GS ablations, including per-channel, per-pixel, per-tile, per-image, and per-scene aggregation levels, as well as implementation-specific L1/L2 and color/nocolor variants where needed for reproducibility. The final three figures give the wider short-recovery comparison across prune ratios 0.90, 0.93, 0.95, 0.97, and 0.99.

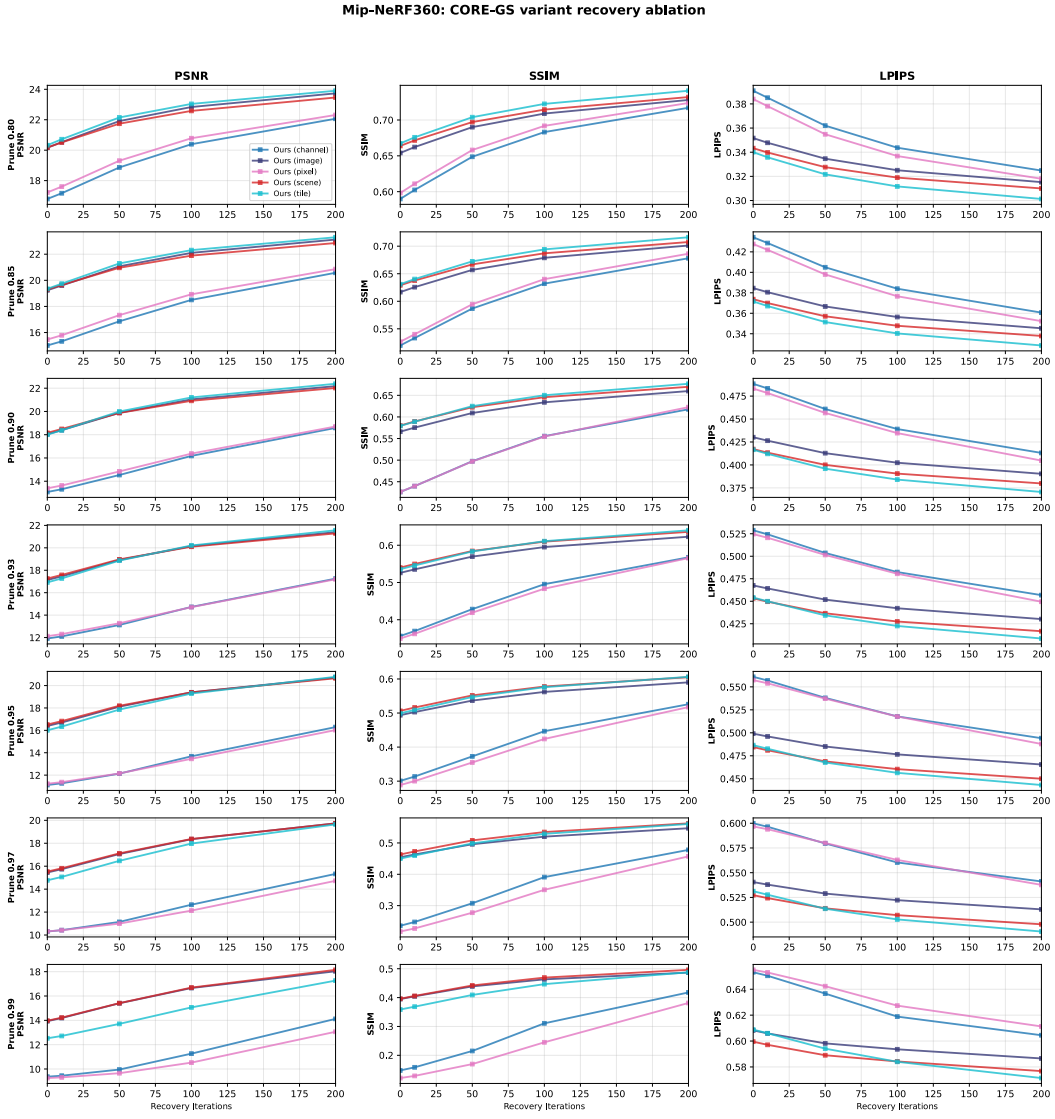

Figure 35: CORE-GS variant recovery ablation for mip-NeRF 360.

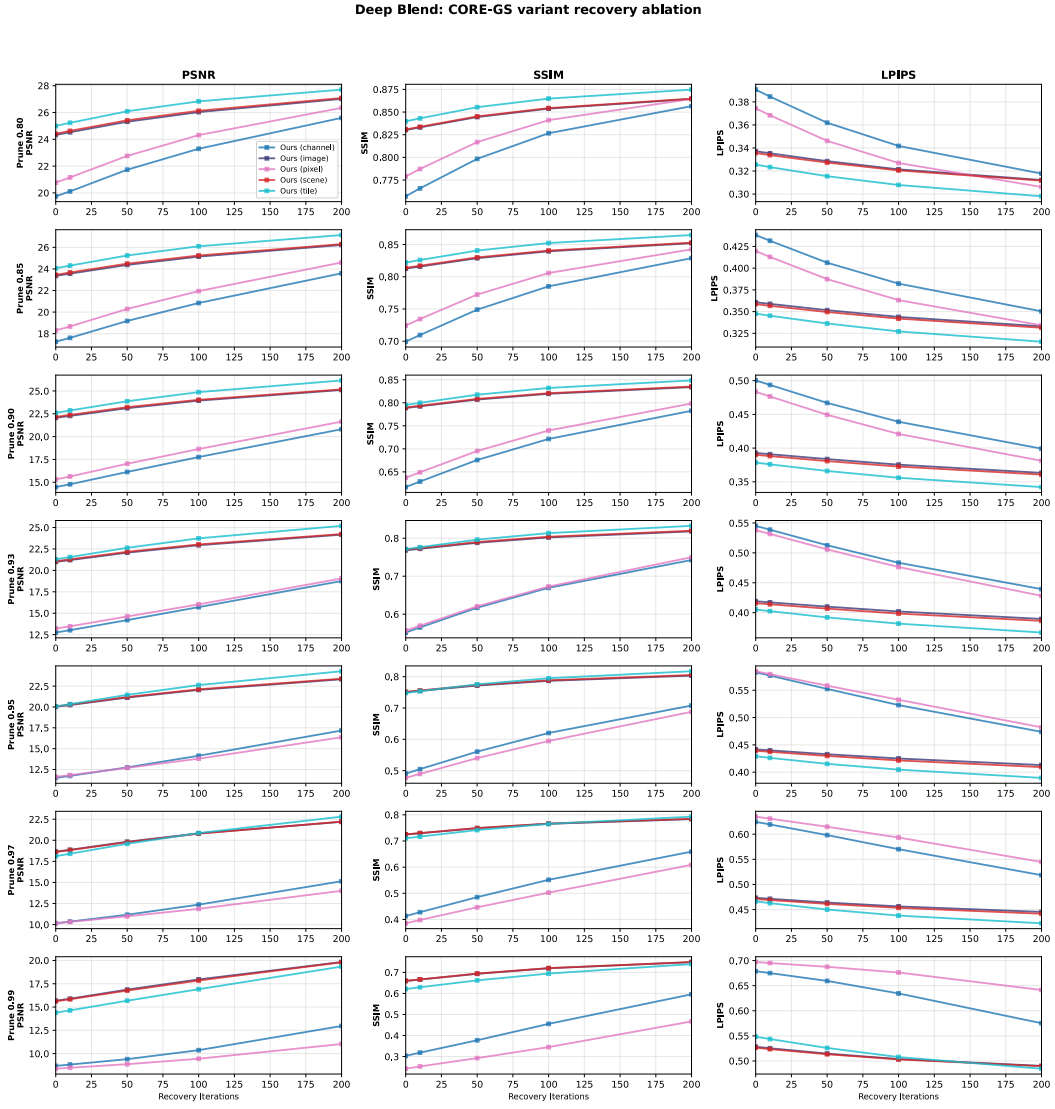

Figure 36: CORE-GS variant recovery ablation for Deep Blending.

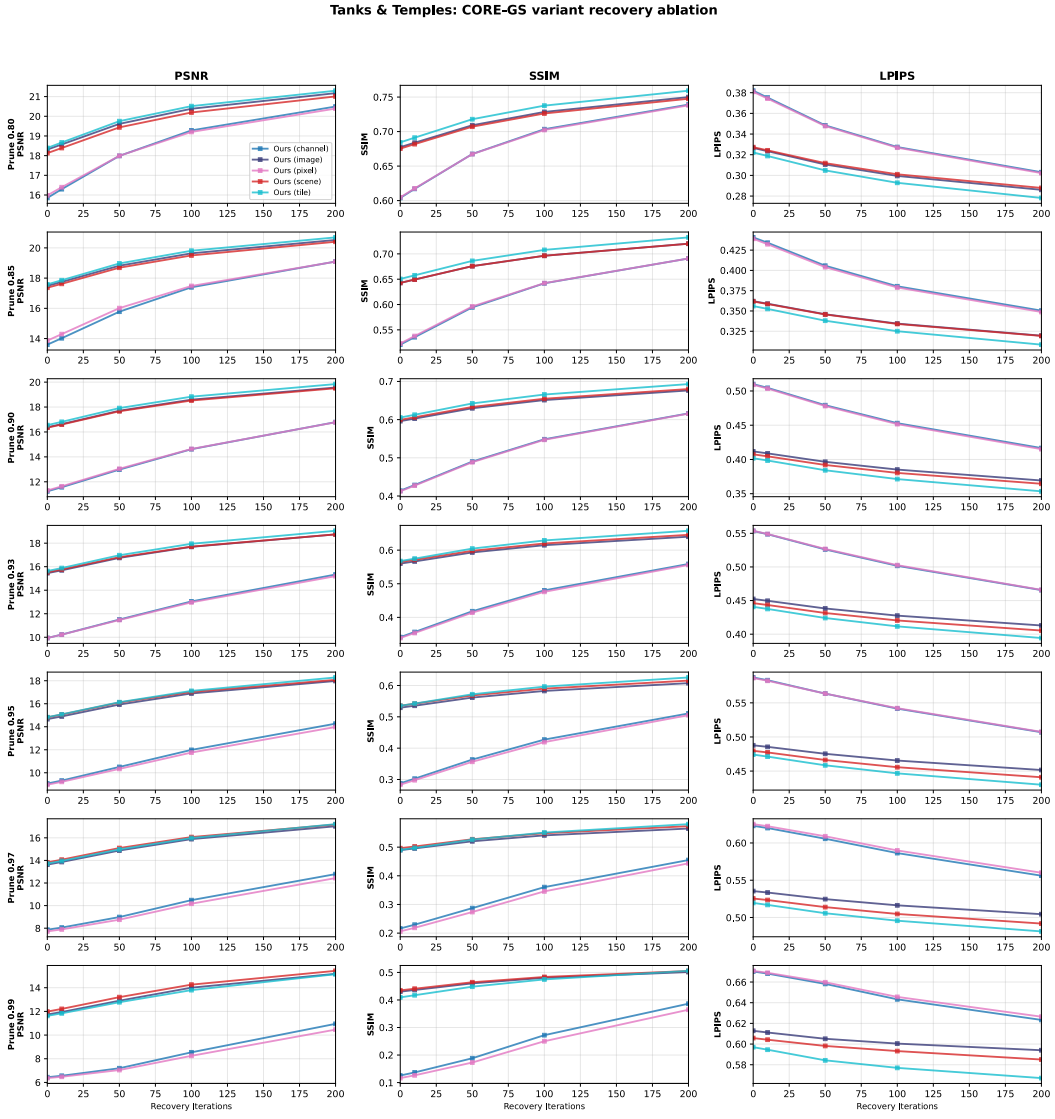

Figure 37: CORE-GS variant recovery ablation for Tanks & Temples.

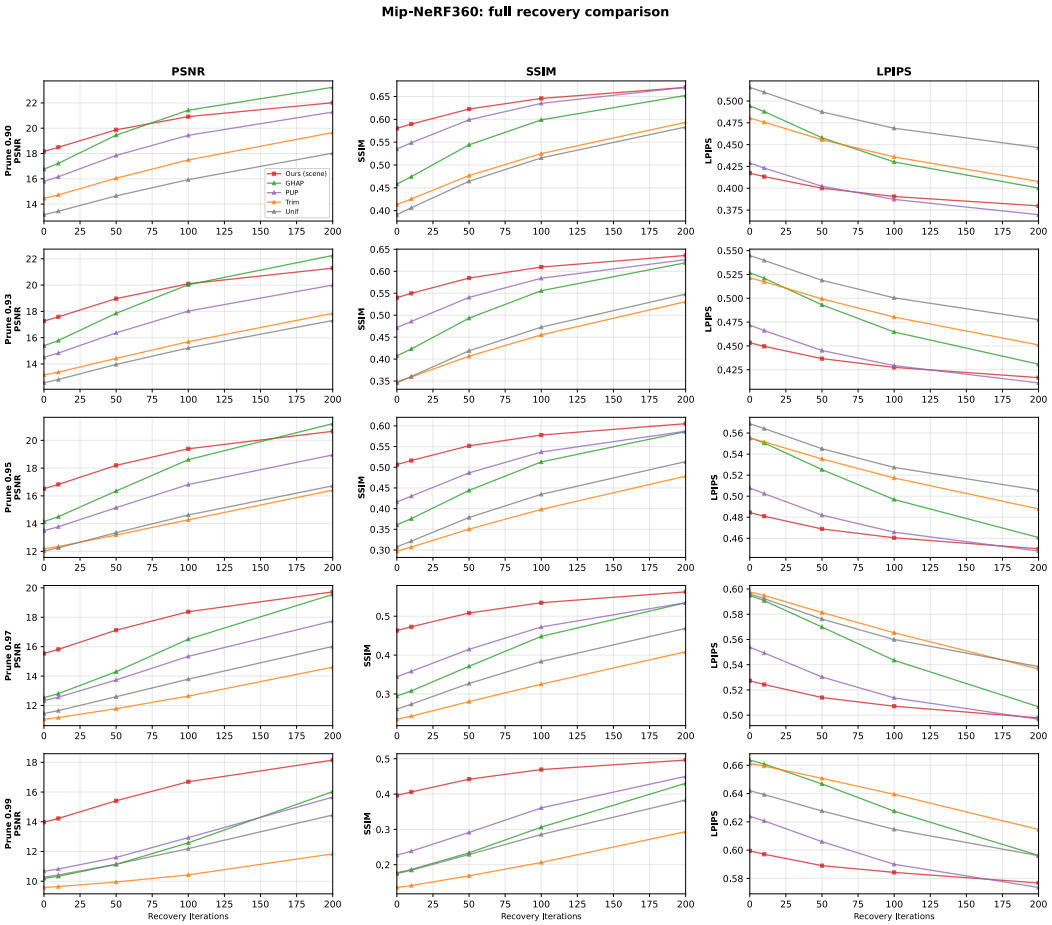

Figure 38: Full short-recovery comparison for mip-NeRF 360 across prune ratios 0.90–0.99.

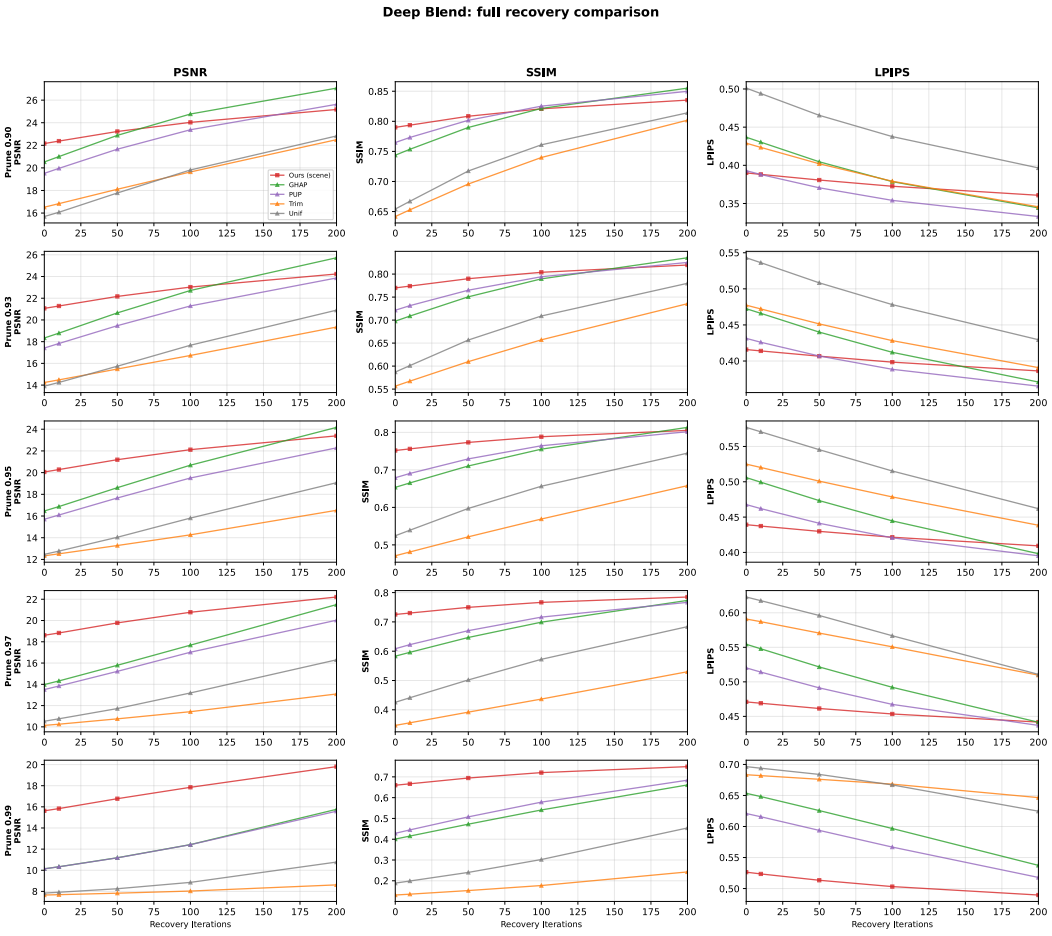

Figure 39: Full short-recovery comparison for Deep Blending across prune ratios 0.90–0.99.

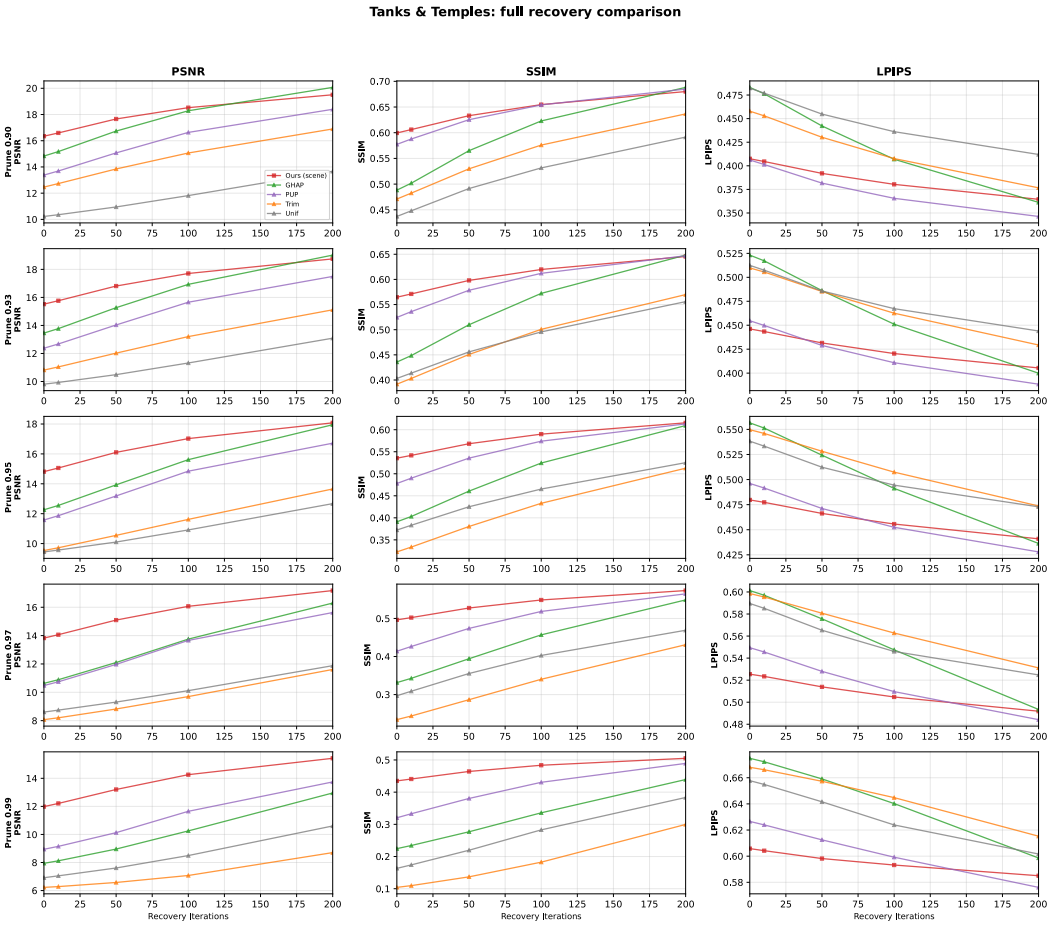

Figure 40: Full short-recovery comparison for Tanks & Temples across prune ratios 0.90–0.99.

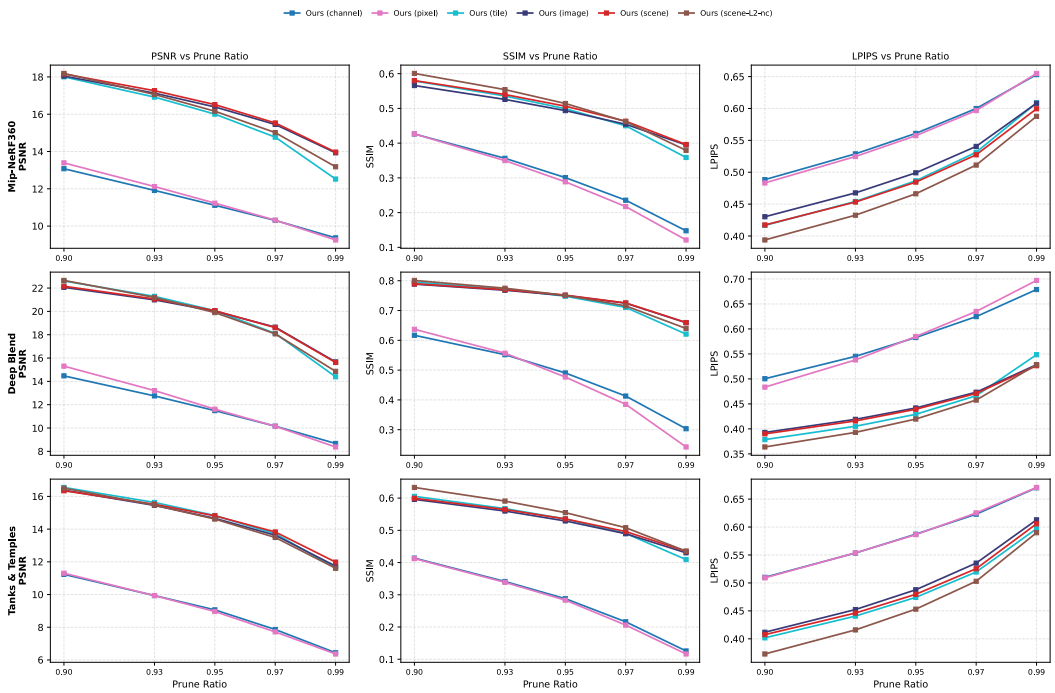

Figure 41: Ablation study.

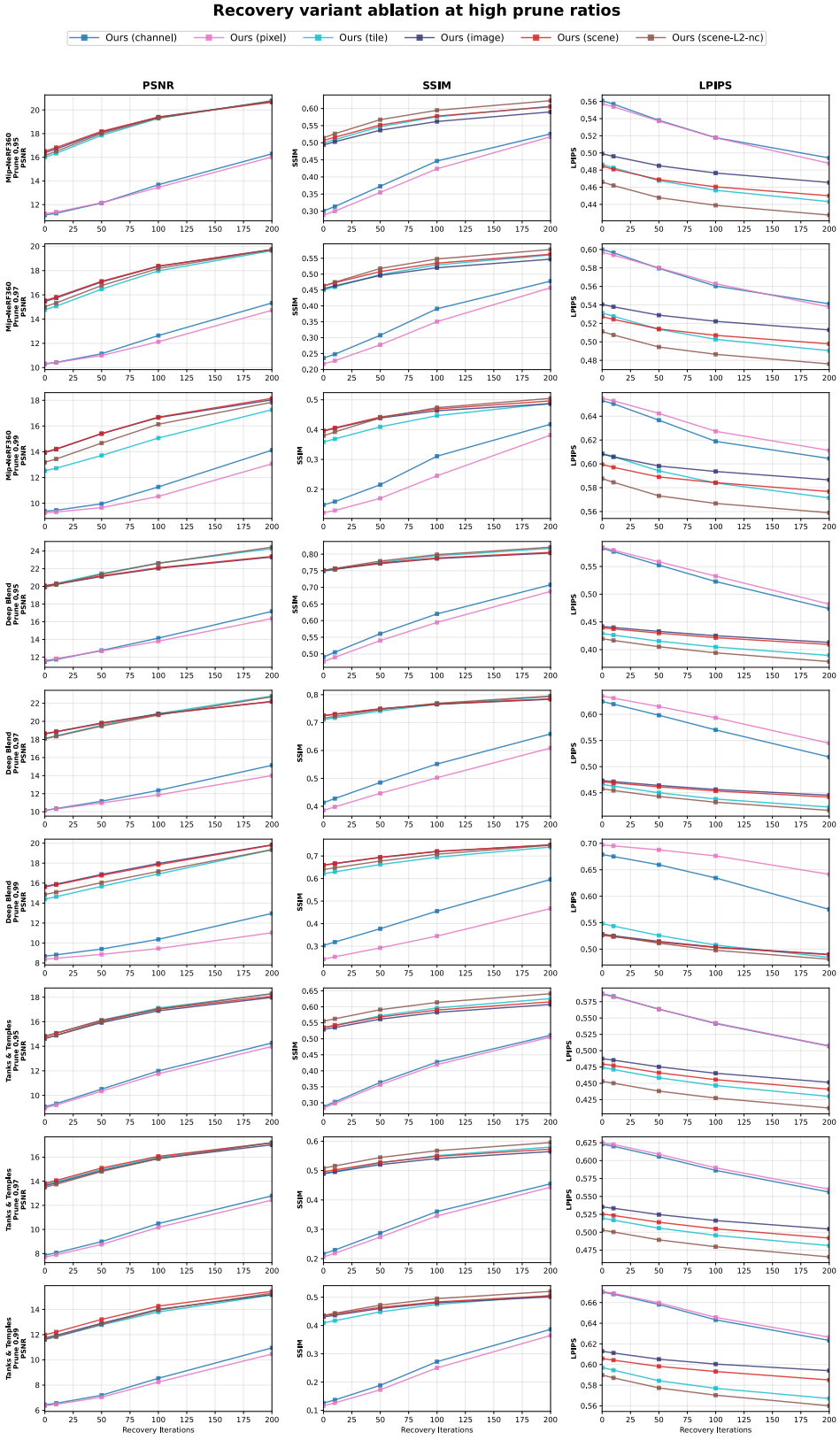

Figure 42: Ablation study.

I Full Setup and Notation

For readability, we restate the two core definitions used throughout the appendix.

Definition 4 (Multiplicative ε-coreset for 3DGS, restated)**.** Let $N > 0$, let $G = \{g_1, \ldots, g_N\}$ be a 3DGS scene, and let $\mathcal{U} \subseteq \mathcal{Q}$ be a family of rendering queries. Let $w \in [0, \infty)^N$ represent a weighted pruned scene. Recall A, the nonnegative scalar objective of the full scene, and let A_w be the corresponding objective induced by w. We call w an *ε-coreset* for G on $\mathcal{U}$ if

$$|A(G, q) - A_w(G, q)| \leq \varepsilon A(G, q) \qquad \text{for every } q \in \mathcal{U}.$$

Definition 5 (Weighted rendering validity, restated)**.** A weight vector $w \in [0, \infty)^N$ is called *valid* on $\mathcal{Q}'$ if

$$0 \leq w_i \rho(g_i, q) < 1 \qquad \text{for every } q \in \mathcal{Q}' \text{ and every } i \in [N].$$

Remark 1. The theoretical objective $A_w^{\text{th}}(G, q)$ is a classical coreset-style weighted sum of fixed itemwise terms $a(G, g_i, q)$. The true rendered objective $A_w(G, q)$ is the physically meaningful quantity of interest but is coupled through the reduced-scene transmittances $T_w(G, g_i, q)$.

J General-case impossibility: restated theorem and proof

We now give the full construction behind theorem 1. The construction uses the finite projected footprints used by rasterized 3DGS implementations. Mathematically, an untruncated Gaussian has infinite support, so exact footprint isolation is not available for the ideal infinite-support kernel. In practical splatting, however, each projected Gaussian is evaluated only on a bounded screen-space footprint, such as a constant-radius Mahalanobis ellipse, commonly described as a 3σ footprint. The following proof is stated for this truncated-footprint rendered objective.

Theorem 5 (General-case impossibility, restated)**.** *For every $N \geq 1$ and every $0 < \varepsilon < 1$, there exist a 3DGS scene $G = \{g_1, \ldots, g_N\}$ and a query family $\mathcal{Q}$ such that no weighted reduced scene specified $w \in [0, \infty)^N$ with $\|w\|_0 < N$ is an ε-coreset for the true rendered objective A_w on $\mathcal{Q}$.*

Proof. Fix $N \geq 1$ and $0 < \varepsilon < 1$. We construct a scene and a finite query family for which each query isolates exactly one Gaussian.

Fix one image-plane location x_0 and one color channel $\lambda \in \Lambda$. Choose N camera poses $C_1, \ldots, C_N$ and construct N Gaussians $g_1, \ldots, g_N$ such that, for every $i \in [N]$, under camera pose C_i, the truncated projected footprint of g_i contains x_0, while the truncated projected footprint of every g_j with $j \neq i$ does not contain x_0. This is possible in the unrestricted setting: place the Gaussian centers far apart in 3D, choose camera pose C_i to view g_i at the selected pixel x_0, and choose the projected covariances sufficiently small so that all other Gaussians lie outside the finite rasterized footprint at that pixel.

For each Gaussian $g_i = (\mu_i, \Sigma_i, \alpha_i, c_i)$, choose Σ_i positive definite and choose $\alpha_i \in (0, 1)$. Decrease the opacities if necessary so that

$$0 \leq \rho(g_j, q_i) < 1 \qquad \text{for every } i, j \in [N].$$

This is possible because the query family constructed below is finite. Choose the scalar appearance in channel λ so that the isolated Gaussian has positive color contribution at its corresponding query.

Define the query family

$$\mathcal{Q} := \{q_1, \ldots, q_N\}, \qquad q_i := (C_i, x_0, \lambda).$$

By construction of the truncated projected footprints, for every $i \in [N]$,

$$k(g_i, q_i) > 0,$$

whereas for every $j \neq i$,

$$k(g_j, q_i) = 0.$$

Therefore

$$\rho(g_i, q_i) = \alpha_i k(g_i, q_i) > 0, \qquad \rho(g_j, q_i) = 0 \quad \text{for every } j \neq i.$$

Since $0 \le \rho(g_j, q_i) < 1$ for all i, j, every prefix transmittance factor $1 - \rho(g_j, q_i)$ is positive. In particular,

$$T(G, g_i, q_i) > 0.$$

Thus, at query q_i, Gaussian g_i is the unique positive contributor:

$$a(G, g_i, q_i) = c_i(q_i)\rho(g_i, q_i)T(G, g_i, q_i) > 0,$$

while for every $j \neq i$,

$$a(G, g_j, q_i) = c_j(q_i)\rho(g_j, q_i)T(G, g_j, q_i) = 0,$$

because $\rho(g_j, q_i) = 0$. Consequently,

$$A(G, q_i) = \sum_{j=1}^{N} a(G, g_j, q_i) = a(G, g_i, q_i) > 0.$$

Now let $w \in [0, \infty)^N$ be any weighted pruned scene with $\|w\|_0 < N$. Then there exists an index $i \in [N]$ such that $w_i = 0$. Consider the corresponding query q_i. For every $j \neq i$, Gaussian g_j has zero truncated-footprint contribution at q_i. The only Gaussian with positive full-scene contribution at q_i is g_i, but its reduced weight is zero. Therefore

$$A_w(G, q_i) = 0.$$

On the other hand,

$$A(G, q_i) > 0.$$

Hence

$$|A(G, q_i) - A_w(G, q_i)| = A(G, q_i) > \varepsilon A(G, q_i),$$

because $0 < \varepsilon < 1$. Thus w fails the multiplicative coreset guarantee at query q_i.

Since the argument holds for every w with $\|w\|_0 < N$, no weighted reduced scene supported on fewer than N Gaussians is an ε-coreset for this scene and the query family $\mathcal{Q}$. □

K Full Proof of the Finite-Query Fixed-Objective Theorem

Lemma 1 (Multiplicative Chernoff for bounded variables)**.** *Let $Z_1, \ldots, Z_m$ be i.i.d. random variables with $0 \le Z_t \le 1$ and $\mathbb{E}[Z_t] = \mu$. Then for every $0 < \varepsilon \le 1$,*

$$\Pr\left[\left|\frac{1}{m}\sum_{t=1}^{m} Z_t - \mu\right| > \varepsilon\mu\right] \le 2\exp\left(-\frac{\varepsilon^2\mu m}{3}\right).$$

Theorem 6 (Query size dependent 3DGS coreset, restated)**.** *Let G be a 3DGS scene, and let $\mathcal{Q}' \subseteq \mathcal{Q}$ be a finite set of representative queries after discarding queries with $A(G, q) = 0$. Let $\varepsilon_c, \delta \in (0, 1)$, and let w be the random weight vector produced by the sensitivity-based sampling rule of section 3.2. If*

$$m \ge \frac{3S}{\varepsilon_c^2}\log\left(\frac{2|\mathcal{Q}'|}{\delta}\right),$$

then, with probability at least $1 - \delta$, w is an ε_c-coreset for the fixed-objective relaxation A^{th} on $\mathcal{Q}'$, i.e.,

$$\left|A(G, q) - A_w^{\mathrm{th}}(G, q)\right| \le \varepsilon_c A(G, q) \qquad \text{for all } q \in \mathcal{Q}'.$$

Moreover, $\|w\|_0 \le m$.

Proof. Let $I_+ := \{i \in [N] : s(g_i) > 0\}$ and, for $i \in I_+$, let $p_i := s(g_i)/S$. Sample indices $J_1, \ldots, J_m$ independently from I_+ with $\Pr[J_t = i] = p_i$. Let n_i be the number of times index $i \in I_+$ is sampled, and set

$$w_i := \begin{cases} \dfrac{n_i}{mp_i}, & i \in I_+, \\ 0, & i \notin I_+. \end{cases}$$

Fix one query $q \in \mathcal{Q}'$. Since $A(G,q) > 0$ by construction of $\mathcal{Q}'$, define

$$Y_t(q) := \frac{a(G, g_{J_t}, q)}{p_{J_t} A(G,q)}.$$

Then

$$\mathbb{E}[Y_t(q)] = \sum_{i \in I_+} p_i \frac{a(G,g_i,q)}{p_i A(G,q)} = \frac{1}{A(G,q)} \sum_{i \in I_+} a(G,g_i,q) = 1,$$

because $a(G,g_i,q) = 0$ whenever $s(g_i) = 0$. By the definition of sensitivity,

$$\frac{a(G,g_i,q)}{A(G,q)} \le s(g_i) \qquad \Rightarrow \qquad Y_t(q) \le \frac{s(g_i)}{p_i} = S \qquad \text{for every } i \in I_+.$$

Hence $0 \le Y_t(q)/S \le 1$ and $\mathbb{E}[Y_t(q)/S] = 1/S$. Applying lemma 1 to $Z_t(q) := Y_t(q)/S$ yields

$$\Pr\left[\left|\frac{1}{m}\sum_{t=1}^{m} Y_t(q) - 1\right| > \varepsilon_c\right] \le 2\exp\left(-\frac{\varepsilon_c^2 m}{3S}\right).$$

But

$$\frac{1}{m}\sum_{t=1}^{m} Y_t(q) = \frac{1}{A(G,q)} \sum_{i \in I_+} \frac{n_i}{m p_i} a(G,g_i,q) = \frac{A_w^{\mathrm{th}}(G,q)}{A(G,q)}.$$

Therefore

$$\Pr\left[\left|A_w^{\mathrm{th}}(G,q) - A(G,q)\right| > \varepsilon_c A(G,q)\right] \le 2\exp\left(-\frac{\varepsilon_c^2 m}{3S}\right).$$

Choosing

$$m \ge \frac{3S}{\varepsilon_c^2} \log\left(\frac{2|\mathcal{Q}'|}{\delta}\right)$$

makes the right-hand side at most $\delta/|\mathcal{Q}'|$. A union bound over all $q \in \mathcal{Q}'$ proves the simultaneous claim. The support bound $\|w\|_0 \le m$ is immediate from the construction. □

L Transfer to True Rendering under Log-Transmittance Stability

Assumption 3 (Log-transmittance stability, restated)**.** There exists $\gamma \ge 0$ such that for every $q \in \mathcal{Q}'$ and every $i \in [N]$ with $w_i > 0$, one has

$$|\log T(G,g_i,q) - \log T_w(G,g_i,q)| \le \gamma.$$

Lemma 2 (Exact transmittance-ratio consequence)**.** *Under Assumption 3, for every $q \in \mathcal{Q}'$ and every $i \in [N]$ with $w_i > 0$,*

$$e^{-\gamma} T(G,g_i,q) \le T_w(G,g_i,q) \le e^{\gamma} T(G,g_i,q).$$

Proof. By Assumption 3,

$$-\gamma \le \log T_w(G,g_i,q) - \log T(G,g_i,q) \le \gamma.$$

Exponentiating gives

$$e^{-\gamma} \le \frac{T_w(G,g_i,q)}{T(G,g_i,q)} \le e^{\gamma}.$$

Multiplying by $T(G,g_i,q)$ proves the claim. □

Lemma 3 (Transfer from the theoretical objective to true rendering)**.** *Assume w is valid on $\mathcal{Q}'$ and satisfies Assumption 3. Then for every $q \in \mathcal{Q}'$,*

$$e^{-\gamma} A_w^{\mathrm{th}}(G,q) \le A_w(G,q) \le e^{\gamma} A_w^{\mathrm{th}}(G,q).$$

Proof. By definition,

$$A_w(G,q) = \sum_{g_i \in G} w_i c_i(q)\rho(g_i,q)T_w(G,g_i,q).$$

Applying lemma 2 termwise to indices with $w_i > 0$ and using nonnegativity gives

$$e^{-\gamma} w_i c_i(q)\rho(g_i,q)T(G,g_i,q) \le w_i c_i(q)\rho(g_i,q)T_w(G,g_i,q) \le e^{\gamma} w_i c_i(q)\rho(g_i,q)T(G,g_i,q).$$

For indices with $w_i = 0$, all terms are zero. Summing over $g_i \in G$ yields the result. □

Theorem 7 (Finite-query rendered coreset under mild assumptions, restated)**.** *Let w be the random weight vector produced by the sensitivity-based sampling rule of section 3.2, and suppose that w is valid on $\mathcal{Q}'$ and satisfies Assumption 3. Under the same sample-size condition as in theorem 6, with probability at least $1-\delta$, w is an ε_r-coreset for the true rendered objective on $\mathcal{Q}'$, where*

$$\varepsilon_r := \max\Big\{1 - e^{-\gamma}(1-\varepsilon_c),\ e^{\gamma}(1+\varepsilon_c) - 1\Big\}.$$

Equivalently,

$$|A(G,q) - A_w(G,q)| \le \varepsilon_r A(G,q) \qquad \textit{for all } q \in \mathcal{Q}'.$$

Moreover, $\|w\|_0 \le m$.

Proof. By theorem 6, with probability at least $1-\delta$,

$$(1-\varepsilon_c)A(G,q) \le A_w^{\mathrm{th}}(G,q) \le (1+\varepsilon_c)A(G,q) \qquad \text{for every } q \in \mathcal{Q}'.$$

On the same event, lemma 3 gives

$$e^{-\gamma} A_w^{\mathrm{th}}(G,q) \le A_w(G,q) \le e^{\gamma} A_w^{\mathrm{th}}(G,q).$$

Combining them yields

$$e^{-\gamma}(1-\varepsilon_c)A(G,q) \le A_w(G,q) \le e^{\gamma}(1+\varepsilon_c)A(G,q).$$

Therefore

$$A(G,q) - A_w(G,q) \le \big(1 - e^{-\gamma}(1-\varepsilon_c)\big)A(G,q),$$

and

$$A_w(G,q) - A(G,q) \le \big(e^{\gamma}(1+\varepsilon_c) - 1\big)A(G,q).$$

Taking the maximum of the two one-sided bounds proves the stated ε_r-coreset guarantee. The support bound follows from theorem 6. □

L.1 Per-query transfer and qualifying subsets

For a query q (camera, pixel, and color channel), define $\gamma(q) = \max_{i:w_i>0} |\log T(G,g_i,q) - \log T_w(G,g_i,q)|$. On the same fixed-objective approximation event as in theorem 3, and under renderer validity, every query with $A(G,q) > 0$ satisfies

$$\frac{|A_w(G,q) - A(G,q)|}{A(G,q)} \le e^{\gamma(q)}(1+\varepsilon_c) - 1. \tag{2}$$

Thus $\mathcal{Q}_\tau = \{q \in \mathcal{Q}' : \gamma(q) \le \tau,\ A(G,q) > 0\}$ receives the uniform bound $e^{\tau}(1+\varepsilon_c) - 1$. This complements, rather than replaces, the global theorem: a rare visibility hole can dominate $\max_q \gamma(q)$ without weakening the bound on qualifying queries.

On the fixed-objective approximation event in the proof above, replace the uniform γ by $\gamma(q)$ for a single valid query. The same termwise comparison gives

$$e^{-\gamma(q)}(1-\varepsilon_c)A(G,q) \le A_w(G,q) \le e^{\gamma(q)}(1+\varepsilon_c)A(G,q).$$

For $A(G,q) > 0$ and $0 < \varepsilon_c < 1$, the upper relative-error term dominates the lower one, since

$$\big[e^{\gamma(q)}(1+\varepsilon_c) - 1\big] - \big[1 - e^{-\gamma(q)}(1-\varepsilon_c)\big] = 2\big(\cosh\gamma(q) - 1 + \varepsilon_c \sinh\gamma(q)\big) \ge 0.$$

This proves equation (2). Monotonicity then gives the uniform bound $e^{\tau}(1+\varepsilon_c) - 1$ on $\mathcal{Q}_\tau$. The event controlling the fixed objective is already simultaneous on $\mathcal{Q}'$, so selecting this qualifying subset after constructing w introduces no new probability event. The global theorem remains the uniform statement with $\gamma \ge \max_{q\in\mathcal{Q}'} \gamma(q)$. Event percentiles, or nested percentiles that omit some Gaussian events, are not automatically maxima over all retained Gaussians and therefore do not themselves certify membership in $\mathcal{Q}_\tau$.

M Compact-region Extension from Representatives: Restated Statements and Proof

Assumption 4 (Compact nondegenerate query region, restated)**.** Let $\mathcal{Q}_r \subseteq \mathcal{Q}$ be a query region endowed with a metric $d_{\mathcal{Q}}$. Assume:

1. $\mathcal{Q}_r$ is compact;
2. the full rendered objective $q \mapsto A(G, q)$ is Lipschitz on $\mathcal{Q}_r$ with constant L_{full};
3. the reduced rendered objective $q \mapsto A_w(G, q)$ is Lipschitz on $\mathcal{Q}_r$ with constant L_w;
4. the full rendered objective is bounded away from zero:

$$A(G, q) \geq A_{\min} > 0 \qquad \text{for every } q \in \mathcal{Q}_r.$$

Proposition 1 (Why the Lipschitz assumption is reasonable)**.** *Suppose on a region $\mathcal{Q}_r$: (i) the relevant set of Gaussians is finite; (ii) the front-to-back order does not jump discontinuously inside $\mathcal{Q}_r$; (iii) the camera-to-ray map is smooth; (iv) each color function $q \mapsto c_i(q)$ is smooth; and (v) each projected kernel contribution $q \mapsto k(g_i, q)$ is smooth. Then the full rendered objective $q \mapsto A(G, q)$ and the reduced rendered objective $q \mapsto A_w(G, q)$ are smooth on $\mathcal{Q}_r$ and hence Lipschitz on $\mathcal{Q}_r$ when $\mathcal{Q}_r$ is compact.*

Proof. Each per-Gaussian contribution is a finite product and composition of smooth functions of q: a color factor, a projected-kernel factor, and a transmittance factor built from finitely many smooth attenuation terms. A finite sum of smooth functions is smooth. Every smooth function on a compact set is Lipschitz. □

Definition 6 (ρ-net, restated)**.** A finite set $\mathcal{Q}'_r \subseteq \mathcal{Q}_r$ is a ρ-net of $\mathcal{Q}_r$ if for every $q \in \mathcal{Q}_r$ there exists $q' \in \mathcal{Q}'_r$ such that

$$d_{\mathcal{Q}}(q, q') \leq \rho.$$

Theorem 8 (Compact-region extension from representatives, restated)**.** *Assume Assumption 4. Let $\mathcal{Q}'_r \subseteq \mathcal{Q}_r$ be a finite ρ-net. Suppose that the reduced scene satisfies the multiplicative guarantee on the representative set:*

$$|A(G, q') - A_w(G, q')| \leq \varepsilon_0 A(G, q') \qquad \textit{for every } q' \in \mathcal{Q}'_r.$$

Then for every $q \in \mathcal{Q}_r$,

$$|A(G, q) - A_w(G, q)| \leq \left(\varepsilon_0 + \frac{\big((1 + \varepsilon_0)L_{\text{full}} + L_w\big)\rho}{A_{\min}} \right) A(G, q).$$

Proof. Fix $q \in \mathcal{Q}_r$ and choose $q' \in \mathcal{Q}'_r$ such that $d_{\mathcal{Q}}(q, q') \leq \rho$. By the triangle inequality,

$$\begin{aligned} |A(G, q) - A_w(G, q)| \leq{}& |A(G, q) - A(G, q')| + |A(G, q') - A_w(G, q')| \\ &+ |A_w(G, q') - A_w(G, q)|. \end{aligned}$$

By Lipschitz continuity,

$$|A(G, q) - A(G, q')| \leq L_{\text{full}}\rho, \qquad |A_w(G, q') - A_w(G, q)| \leq L_w\rho.$$

By the representative-set guarantee,

$$|A(G, q') - A_w(G, q')| \leq \varepsilon_0 A(G, q').$$

Using Lipschitz continuity once more,

$$A(G, q') \leq A(G, q) + L_{\text{full}}\rho.$$

Substituting gives

$$\begin{aligned} |A(G, q) - A_w(G, q)| &\leq L_{\text{full}}\rho + \varepsilon_0\big(A(G, q) + L_{\text{full}}\rho\big) + L_w\rho \\ &= \varepsilon_0 A(G, q) + \big((1 + \varepsilon_0)L_{\text{full}} + L_w\big)\rho. \end{aligned}$$

Finally, since $A(G, q) \geq A_{\min}$,

$$\big((1 + \varepsilon_0)L_{\text{full}} + L_w\big)\rho \leq \frac{\big((1 + \varepsilon_0)L_{\text{full}} + L_w\big)\rho}{A_{\min}} A(G, q).$$

Combining the two displays proves the claim. □

Distance-dependent interpretation. Unseen queries inside a qualifying covered region are included, not only its representatives. Different regions may have different regularity constants and positive-signal lower bounds; theorem 9 gives the finite-union result. A region containing an exact-zero query is outside the stated $A_{\min} > 0$ multiplicative regime; this does not invalidate qualifying positive-signal regions elsewhere.

For a particular q and representative q', the same proof retains $d_{\mathcal{Q}}(q, q')$ in place of ρ and yields

$$\frac{|A(G,q) - A_w(G,q)|}{A(G,q)} \leq \varepsilon_0 + \frac{((1+\varepsilon_0)L_{\text{full}} + L_w)d_{\mathcal{Q}}(q,q')}{A(G,q)}.$$

Replacing $A(G,q)$ by its lower bound $A_{\min}$ and the distance by ρ recovers theorem 8. The region must satisfy the stated assumptions; a collection of sampled distances alone is not a continuous-domain net certificate. Likewise, a finite observed fraction with $A(G,q) > A_0$ describes empirical signal coverage, whereas a deterministic regional guarantee requires $A(G,q) \geq A_0$ everywhere in that region.

Corollary 1 (Regionwise target-error form)**.** *Fix $\bar{\varepsilon} > \varepsilon_0$. If*

$$\rho \leq \frac{(\bar{\varepsilon} - \varepsilon_0)A_{\min}}{(1+\varepsilon_0)L_{\text{full}} + L_w},$$

then

$$|A(G,q) - A_w(G,q)| \leq \bar{\varepsilon} A(G,q) \qquad \textit{for every } q \in \mathcal{Q}_r.$$

Proof. Substitute the chosen upper bound on ρ into theorem 8. □

Corollary 2 (Representative-to-region theorem with explicit coreset parameters)**.** *Assume Assumption 4 on $\mathcal{Q}_r$, let $\mathcal{Q}'_r \subseteq \mathcal{Q}_r$ be a finite ρ-net, and assume the hypotheses of theorem 7 hold on $\mathcal{Q}'_r$ with parameters $(\varepsilon_c, \gamma_r, \delta_r)$ and sensitivity sum S_r. If*

$$m_r \geq \frac{3S_r}{\varepsilon_c^2} \log \frac{2|\mathcal{Q}'_r|}{\delta_r},$$

then with probability at least $1 - \delta_r$,

$$|A(G,q) - A_w(G,q)| \leq \left(\varepsilon_{0,r} + \frac{\big((1+\varepsilon_{0,r})L_{\text{full}} + L_w\big)\rho}{A_{\min}} \right) A(G,q) \qquad \textit{for all } q \in \mathcal{Q}_r,$$

where

$$\varepsilon_{0,r} := \max\Big\{ 1 - e^{-\gamma_r}(1-\varepsilon_c),\ e^{\gamma_r}(1+\varepsilon_c) - 1 \Big\}.$$

Proof. Apply theorem 7 on the representative set $\mathcal{Q}'_r$ and then apply theorem 8 with $\varepsilon_0 = \varepsilon_{0,r}$. □

Theorem 9 (Finite union of compact regions)**.** *Let $\mathcal{Q}_{\text{reg}} = \bigcup_{r=1}^{R} \mathcal{Q}_r$ be a finite union of compact query regions satisfying Assumption 4, and for each r let $\mathcal{Q}'_r \subseteq \mathcal{Q}_r$ be a representative net. Suppose the hypotheses of corollary 2 hold on each region with failure probability δ_r, and that*

$$\sum_{r=1}^{R} \delta_r \leq \delta.$$

Then with probability at least $1 - \delta$, every regional conclusion holds simultaneously on all $\mathcal{Q}_r$, hence on all of $\mathcal{Q}_{\text{reg}}$.

Proof. Apply corollary 2 on each region and then take a union bound over $r = 1, \ldots, R$. □

N Limitations and Scope: Extended

Our results are designed for the structured regime formalized in this paper. The guarantees are not meant to cover unrestricted continuous rendering, where we prove that non-trivial multiplicative coresets are impossible. Instead, our positive results apply to representative query families induced by a target rendering resolution, e.g., a grid of camera poses, rays, pixels, or tiles, and extend to compact query regions under Lipschitz regularity and sufficient query coverage. This matches the practical setting in which 3DGS models are evaluated on finite sets of desired views, or with finite jumps from camera pose to another when deployed on a robot. The rendered guarantee also requires validity and log-transmittance stability. These assumptions are not merely technical: they identify the regime in which a fixed-objective coreset approximation can be faithfully transferred to true re-rendering after pruning. They make explicit how much visibility and occlusion structure must be preserved for provable rendering control.

Empirically, our focus is the low-compute regime: prune-only evaluation and very short finetuning budgets. This is precisely where principled subset selection is most important, since there is little recovery compute available to repair a poor pruning decision. Methods that rely on longer recovery or specialized compression pipelines may become more competitive when large post-pruning optimization budgets are available. These directions are largely orthogonal to our contribution: our pruning rule can serve as a strong initialization before applying post-pruning refinement or other recovery procedures. Our goal is therefore not to replace all 3DGS compression techniques, but to provide a principled, vanilla-compatible pruning approach with strong performance when recovery compute is scarce.

O Societal Impacts

This work may have positive societal impact by making 3D Gaussian Splatting models more efficient to store, transmit, and render. By reducing memory and compute requirements, provable pruning can help deploy high-quality 3D representations on resource-constrained devices, lower energy consumption, and make real-time 3D applications more accessible in areas such as robotics, telepresence, education, cultural heritage, and scientific visualization. The theoretical guarantees may also improve reliability by making compression behavior more predictable than purely heuristic pruning.

At the same time, more efficient 3D scene representations may also have negative societal impacts. Easier deployment of compact, high-quality 3D models could support misuse in surveillance, unauthorized reconstruction of private spaces, or the creation and distribution of realistic synthetic or misleading 3D content. Compression may also make such models easier to share at scale, increasing the risk of misuse. These risks are not introduced uniquely by our method, but our work could lower the computational barriers for both beneficial and harmful applications. Therefore, deployment should respect privacy, consent, dataset licensing, and application-specific safety constraints.

P Broader Perspective

The theory developed here suggests that 3DGS coresets are best understood not as a blanket theorem over arbitrary continuous query spaces, but as a layered statement:

1. impossibility in the unrestricted general case;
2. existence on a finite representative family for a fixed objective;
3. transfer to true rendering under explicit transmittance stability; and
4. extension to compact regions by representative covers.

This formulation is intentionally conservative, but it makes the logical structure of the proof transparent and leaves no hidden gap between the negative and positive parts of the claim.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction, we present our contribution as a provable 3DGS pruning algorithm. The paper provides the theoretical framework, while the experiments validate the method empirically.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See sections 5 and N in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions and proof sketches are provided in the main paper. The full proofs are deferred to the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer:[Yes]

   Justification: In addition to the algorithm's details, the public implementation is available at https://github.com/waseem-m/3dgs_provable_coresets.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The public implementation is available at https://github.com/waseem-m/3dgs_provable_coresets.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We adapted the same setting used in prior work in 3DGS compression. So all is already known, and we state that we followed those settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Appendix B and table 2 report saved means and standard deviations over ten sampling seeds for 13 scenes and seven pruning ratios, totaling 910 prune-only runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: all of the experiments were run on the same hardware and resources, mentioned in detail in the paper

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: the research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Section O

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All papers / assets used as properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our method is detailed in this paper and the public implementation is available at https://github.com/waseem-m/3dgs_provable_coresets.

    Guidelines: The

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.