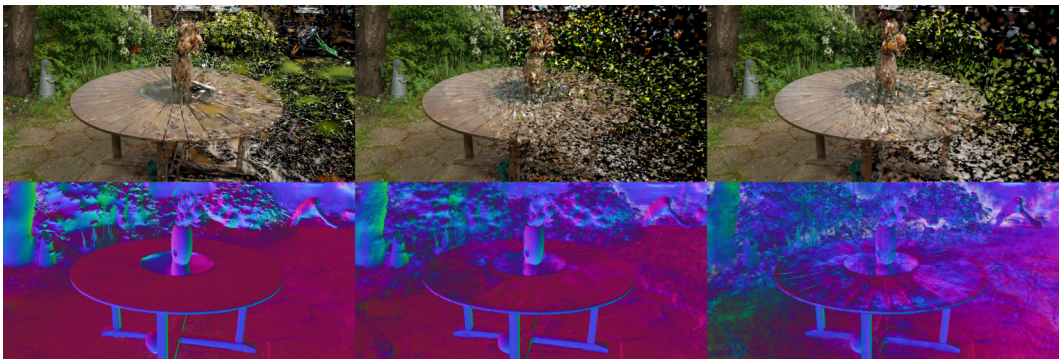
QUADRILATERAL SPLATTING FOR REAL-TIME RADIANCE FIELD RENDERING

Anonymous authors
Paper under double-blind review



2DGS Triangle Splatting **Quadrilateral Splatting (Ours)**

Figure 1: **Primitive and normal visualization on Garden.** Columns show 2DGS, Triangle Splatting (TS), and Quadrilateral Splatting (QS) from the same viewpoint. Top: rendered images gradually transition to sparser subsets of the learned primitives, revealing their shapes and arrangement. Bottom: corresponding surface normals visualize the orientation structure of each representation.

ABSTRACT

Triangle Splatting (TS) has recently emerged as an effective explicit, non-volumetric surface representation for novel-view synthesis (NVS) and real-time rendering, but each triangular primitive is defined over a three-vertex surface domain, which can limit its flexibility for modeling fine-scale local geometry and spatial appearance variation. This limited geometric flexibility can result in less clearly resolved surface-normal structure in regions containing fine geometric detail. We propose **Quadrilateral Splatting (QS)**, which represents scenes using independently optimized planar quadrilateral elements with a continuous bilinear appearance field. The quadrilateral domain provides additional flexibility for modeling local surface geometry, while the continuous bilinear appearance field represents spatial color variation within each primitive. QS jointly optimizes quadrilateral geometry, per-primitive opacity, and view-dependent appearance through differentiable polygon rasterization and alpha compositing, with planar-convex retraction maintaining valid quadrilateral geometry during optimization. Experiments on Mip-NeRF 360 and NeRF Synthetic show improved PSNR, SSIM, and LPIPS over TS and 2DGS on indoor scenes, with clearer fine-scale geometry in rendered normal maps and comparable outdoor quality while preserving real-time rendering. These results demonstrate the effectiveness of quadrilateral primitives with continuous bilinear appearance fields for real-time radiance field rendering.

1 INTRODUCTION

Novel-view synthesis (NVS) renders photorealistic images from previously unseen viewpoints and is a fundamental problem in computer vision and graphics, with applications in VR/AR, medical imaging, autonomous driving, and 3D content creation. Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) achieve high-quality novel-view synthesis using implicit scene representations, but rely on costly neural evaluation and volume rendering. More recently, 3D Gaussian Splatting

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

(3DGS) (Kerbl et al., 2023) significantly advanced the field with explicit volumetric Gaussian primitives, enabling high-quality real-time rendering. Subsequent work further improves rendering quality, scalability, and training efficiency through implicit and volumetric representations (Barron et al., 2022; 2023; Sara Fridovich-Keil and Alex Yu et al., 2022; Müller et al., 2022; Sabour et al., 2023; Ren et al., 2024; Kheradmand et al., 2024; Liu et al., 2025; Sabour et al., 2024; Held et al., 2025b). Despite these advances, volumetric representations parameterize scene content in three-dimensional space, motivating the exploration of lower-dimensional, non-volumetric scene representations.

Lower-dimensional representations offer a natural alternative to volumetric scene modeling, as much of the geometry and appearance observed in real-world scenes is concentrated near surfaces. Representing these surfaces directly with two-dimensional primitives avoids allocating representational capacity throughout the surrounding 3D volume. Such surface-based representations combine explicit geometry with differentiable optimization and efficient rasterization, making them particularly attractive for real-time NVS.

Recent approaches explore this direction using compact two-dimensional primitives embedded in 3D. 2D Gaussian Splatting (2DGS) (Huang et al., 2024a) replaces volumetric Gaussians with oriented two-dimensional primitives, while BillBoard Splatting (Svitov et al., 2024) uses learnable planar elements with spatially varying appearance and opacity. More recently, Triangle Splatting (TS) (Held et al., 2025a) introduced independently optimized triangular surface elements with opacity and view-dependent appearance, enabling high-quality NVS at real-time speeds. Within this class of lower-dimensional, non-volumetric representations, however, each triangle is restricted to a three-vertex domain, limiting its flexibility for modeling fine-scale local geometry and spatial appearance variation within a single primitive. Complex local structure may therefore require refinement of the primitive layout, making fine geometric detail harder to represent compactly and potentially less resolved in surface normals. This motivates a surface primitive with greater geometric flexibility and a continuous appearance field defined over its domain.

To address this limitation, we introduce QS, an explicit, non-volumetric surface representation for real-time radiance field rendering using independently optimized planar quadrilateral elements. Inspired by quadrilateral elements in finite element methods (FEM) (Sifakis & Barbic, 2012), QS combines a four-corner geometric domain with a continuous bilinear appearance field. The quadrilateral domain provides additional flexibility for modeling local surface geometry, while the continuous bilinear appearance field represents spatial color variation within each primitive. QS jointly optimizes quadrilateral geometry, per-primitive opacity, and view-dependent appearance through differentiable polygon rasterization and alpha compositing. Since unconstrained optimization can produce non-planar or non-convex quadrilaterals, we introduce planar-convex retraction to maintain valid quadrilateral geometry throughout optimization. Experiments on Mip-NeRF 360 and NeRF Synthetic show improved PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) over Triangle Splatting (Held et al., 2025a) and 2DGS (Huang et al., 2024a) on indoor scenes, with comparable reconstruction quality on outdoor scenes while preserving real-time rendering. Qualitative normal-map comparisons further reveal clearer fine-scale geometric structures.

In summary, our contributions are: (i) We introduce QS, an explicit non-volumetric surface representation that combines flexible four-corner primitives with continuous bilinear appearance fields for local geometry and appearance modeling. (ii) We develop a differentiable quadrilateral rendering and joint optimization framework with planar-convex retraction to maintain valid geometry throughout training. (iii) We demonstrate improved PSNR, SSIM, and LPIPS over TS and 2DGS on indoor Mip-NeRF 360 and NeRF Synthetic while preserving real-time rendering, together with clearer fine-scale geometric structures in rendered normal maps.

2 Related Works

Implicit and Volumetric Representations. Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) represent scenes using implicit neural functions and synthesize novel views through volume rendering. Subsequent work improves rendering quality, scalability, and training efficiency (Barron et al., 2022; 2023; Sara Fridovich-Keil and Alex Yu et al., 2022; Müller et al., 2022; Sabour et al., 2023; Ren et al., 2024). More recently, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) introduced explicit volumetric Gaussian primitives with differentiable rasterization, enabling high-quality real-time rendering. Follow-up work explores shading and inverse rendering (Jiang et al.,

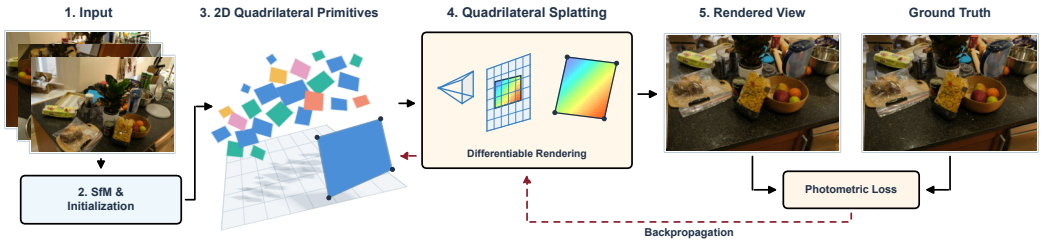


Figure 2: **Overview of Quadrilateral Splatting.** (1) Multi-view images and SfM provide cameras and sparse 3D points for initialization. (2) The scene is represented by planar quadrilateral primitives with learnable geometry, opacity, and view-dependent appearance. (3) Differentiable splatting projects the primitives and evaluates coverage and appearance. (4) Alpha compositing produces the rendered view for image supervision. (5) Gradients optimize geometry and appearance, while planar-convex retraction preserves valid quadrilaterals.

2023; Liang et al., 2023; Sun et al., 2025), anti-aliasing and multi-scale modeling (Yan et al., 2024; Yu et al., 2024a), densification (Cheng et al., 2024; Zhou & Ni, 2025; Nguyen et al., 2025), and efficiency (Navaneet et al., 2023). Other extensions address surface reconstruction (Yu et al., 2024b; Dai et al., 2024; Huang et al., 2024a; 2025), dynamic scenes (Wu et al., 2024; Yang et al., 2023; Cai et al., 2024), generative modeling (Chen et al., 2024; Tang et al., 2024), and large-scale scenes (Mallick et al., 2024; Lin et al., 2024; Huang et al., 2024b). EGGS (Zhang et al., 2025) combines exchangeable 2D and 3D Gaussians to balance geometry and appearance. In contrast, our work focuses on lower-dimensional, non-volumetric polygonal primitives that represent scene surfaces explicitly.

Compact Non-Volumetric Representations. A complementary line of work represents scenes using two-dimensional primitives embedded in 3D. 2D Gaussian Splatting (2DGS) (Huang et al., 2024a) uses oriented planar Gaussian primitives, while BillBoard Splatting (Svitov et al., 2024) employs planar elements with learnable RGB textures and alpha maps for spatially varying appearance and opacity. More recently, Triangle Splatting (TS) (Held et al., 2025a) introduced independently optimized triangular surface elements with per-primitive opacity and view-dependent appearance for high-quality real-time rendering. Within this class of explicit non-volumetric representations, the three-vertex domain can limit flexibility for modeling fine-scale local geometry and spatial appearance variation within each element. Our QS instead uses a four-corner planar domain with a continuous bilinear appearance field, providing greater geometric flexibility and continuous spatial color variation within each primitive. Planar-convex retraction maintains valid quadrilateral geometry during optimization.

3 Method

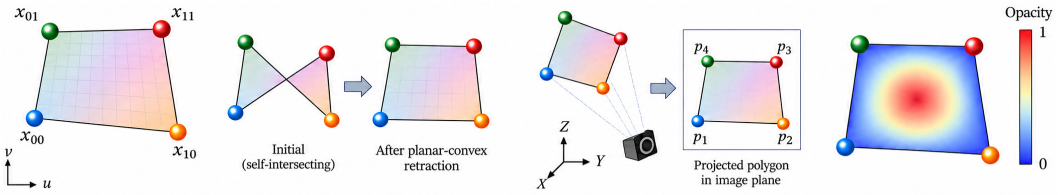
QS reconstructs a scene from calibrated images $\{I_k, \Pi_k\}_{k=1}^{K}$ using independently optimized planar quadrilateral surface elements. Each primitive combines a four-corner geometric domain with a continuous bilinear appearance field, per-primitive opacity, and view-dependent coefficients. Figure 2 provides an overview of QS, from primitive initialization to differentiable rendering and joint geometry–appearance optimization.

3.1 Quadrilateral Surface Representation

A primitive q stores four cyclic vertices $X_q = (x_{00}, x_{10}, x_{11}, x_{01}) \in \mathbb{R}^{4\times 3}$, opacity $o_q \in (0, 1)$, a positive edge-falloff exponent σ_q, degree-three spherical-harmonic (SH) coefficients H_q, and three RGB spatial residual vectors. Its local surface is parameterized as

$$S_q(u, v) = (1-u)(1-v)x_{00} + u(1-v)x_{10} + uvx_{11} + (1-u)vx_{01}, \qquad (u, v) \in [0, 1]^2. \quad (1)$$

Elements have no prescribed connectivity. Opacity and SH coefficients are shared over each primitive, while the three RGB residual vectors define continuous spatial appearance variation over its local domain.



(a) Quad Parameterization (b) Planar-convex retraction (c) Projection to the image plane (d) Polygonal opacity

Figure 3: **Quadrilateral geometry and polygonal opacity.** (a) Four-corner surface coordinates. (b) Retraction restores valid geometry. (c) Projection forms an image-space quadrilateral. (d) Opacity decreases toward its exterior edges.

To maintain planar and convex geometry during optimization, we retract the vertices after each update. With $\xi = 2u - 1$ and $\eta = 2v - 1$, the surface can be written as

$$S_q(\xi, \eta) = c + \xi a + \eta b + \xi\eta d, \tag{2}$$

where c is the center, a and b define a local frame, and d controls the deviation from a parallelogram. We project d into the span of this frame and constrain its coefficients:

$$d \leftarrow \gamma_u a + \gamma_v b, \qquad |\gamma_u| + |\gamma_v| \le \tau, \qquad \tau = 0.95. \tag{3}$$

For linearly independent a and b, this produces a planar, strictly convex quadrilateral with a nonfolding coordinate map. Figure 3 illustrates the resulting geometry and projected footprint. The renderer culls degenerate frames, and Appendix A.1 provides the construction and proof.

3.2 Differentiable Quadrilateral Splatting

For camera k, perspective projection maps each visible quadrilateral to a convex image-space polygon $\mathcal{P}_q$:

$$p_{qi} = \Pi_k(x_{qi}), \qquad i \in \{00, 10, 11, 01\}. \tag{4}$$

Following the polygonal opacity formulation of Triangle Splatting (Held et al., 2025a), let $\ell_{qi}(p)$ denote the signed distance from pixel p to supporting edge i, with negative values inside $\mathcal{P}_q$. We define $\phi_q(p) = \max_i \ell_{qi}(p)$ and let ρ_q denote the projected polygon's inscribed-circle radius. The pixel opacity is

$$w_q(p) = \max\left(0, -\frac{\phi_q(p)}{\rho_q}\right), \qquad \alpha_q(p) = \min(0.99, o_q w_q(p)^{\sigma_q}). \tag{5}$$

Here, o_q controls the overall opacity of the primitive, while σ_q controls how rapidly its opacity decreases toward the polygon boundary. Coverage is evaluated directly from the four projected polygon edges.

For each covered pixel, a perspective-correct plane mapping followed by inversion of Equation 1 recovers its local coordinates (u, v). In centered coordinates, the appearance field is

$$c_q(p, \omega_q) = b_q(\omega_q; H_q) + \xi r_q^u + \eta r_q^v + \xi\eta r_q^{uv}, \tag{6}$$

where b_q is the nonnegative SH base color and ω_q is the viewing direction from the camera center to the primitive center. The residuals $r_q^u, r_q^v, r_q^{uv} \in \mathbb{R}^3$ are view independent. Together, the ξ, η, and $\xi\eta$ terms define a continuous bilinear spatial appearance field over the quadrilateral, while the SH component models view-dependent appearance. Appendix A.2 details coordinate inversion, numerical fallbacks, and basis normalization.

Elements are binned into tiles, sorted by primitive-center depth, and composited front-to-back. Figure 4 illustrates the appearance evaluation and compositing process:

$$\widehat{I}(p) = \sum_q T_q(p)\alpha_q(p)c_q(p, \omega_q) + T_{\text{end}}(p)c_{\text{bg}}, \qquad T_q(p) = \prod_{j<q}(1 - \alpha_j(p)). \tag{7}$$

The CUDA renderer rejects negligible contributions and terminates compositing at low transmittance. The rendering process provides gradients for appearance, opacity, and edge-falloff parameters. In the reported accelerated configuration, geometry receives gradients through polygonal coverage and the SH viewing direction, while the appearance-coordinate map is held fixed. Appendix A.3 details this approximation and the locally fixed visibility decisions.

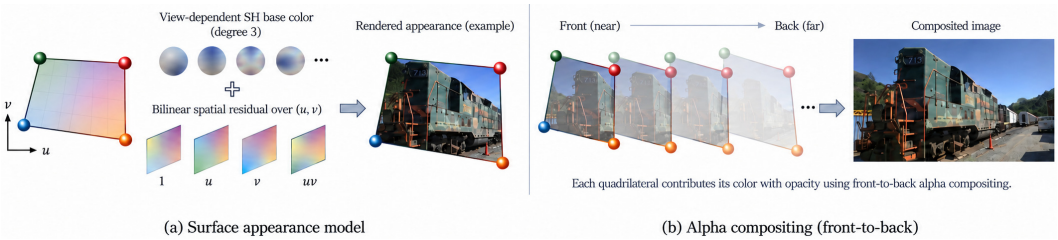


Figure 4: **Appearance modeling and alpha compositing in QS.** (a) Each quadrilateral uses view-dependent SH color with a bilinear spatial residual over local coordinates (u, v). (b) Depth-ordered quadrilaterals are combined using front-to-back alpha compositing to form the final image.

3.3 Joint Optimization and Primitive Adaptation

For real scenes, small quadrilateral elements are initialized at SfM and auxiliary shell points, with scales determined by local point spacing. Synthetic scenes use seeded point initialization and known cameras. With cameras fixed, each sampled training image I supervises the rendered prediction $\widehat{I}$ through

$$\mathcal{L} = (1 - \lambda_s)\mathcal{L}_{\text{pixel}} + \lambda_s\big(1 - \text{SSIM}(\widehat{I}, I)\big) + \lambda_o\mathcal{R}_o + \lambda_n\mathcal{R}_n + \lambda_a\mathcal{R}_a, \tag{8}$$

with $\lambda_s = 0.2$. The regularizers penalize mean opacity, encourage agreement between rendered surface normals and depth-derived normals, and penalize inverse mean primitive area during growth. Each iteration jointly updates quadrilateral vertices, SH coefficients, bilinear spatial residuals, opacity, and edge-falloff exponents with Adam, followed by planar-convex retraction. Main benchmark models receive exactly 30,000 updates. Loss definitions, learning rates, and image-resolution schedules are given in Appendix A.4.

Splitting, cloning, and pruning adapt the primitive population under a scene-specific cap. Figure 5 illustrates the primitive adaptation process. Growth alternates candidate sampling between opacity and inverse edge-falloff exponent. Large selected elements are split into four local child domains, while small selected elements are cloned with in-plane displacement. Low-opacity, low-contribution, or insufficiently observed elements are pruned. Each split adds three net elements and each clone adds one, subject to the remaining capacity. Children inherit the SH base and the parent bilinear appearance field restricted to their local domains. Since adaptation changes primitive support, it need not preserve the rendered image exactly. Appendix A.5 provides the adaptation rules and transfer equations.

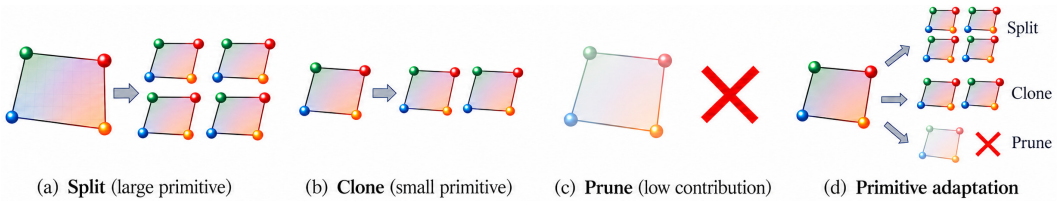


Figure 5: **Primitive adaptation during training.** (a) Large elements split into four children. (b) Small elements are cloned. (c) Weakly supported elements are pruned. (d) These operations adapt the representation within a primitive budget.

4 Experiments and Results

4.1 Experimental Setup

Dataset. We evaluate on Mip-NeRF 360 (Barron et al., 2022), NeRF Synthetic (Mildenhall et al., 2020), and Tanks & Temples. Mip-NeRF 360 contains five outdoor and four indoor scenes spanning unbounded environments and detailed indoor spaces. Following the standard protocol, every eighth image is held out for testing, with outdoor and indoor images downsampled by factors of four

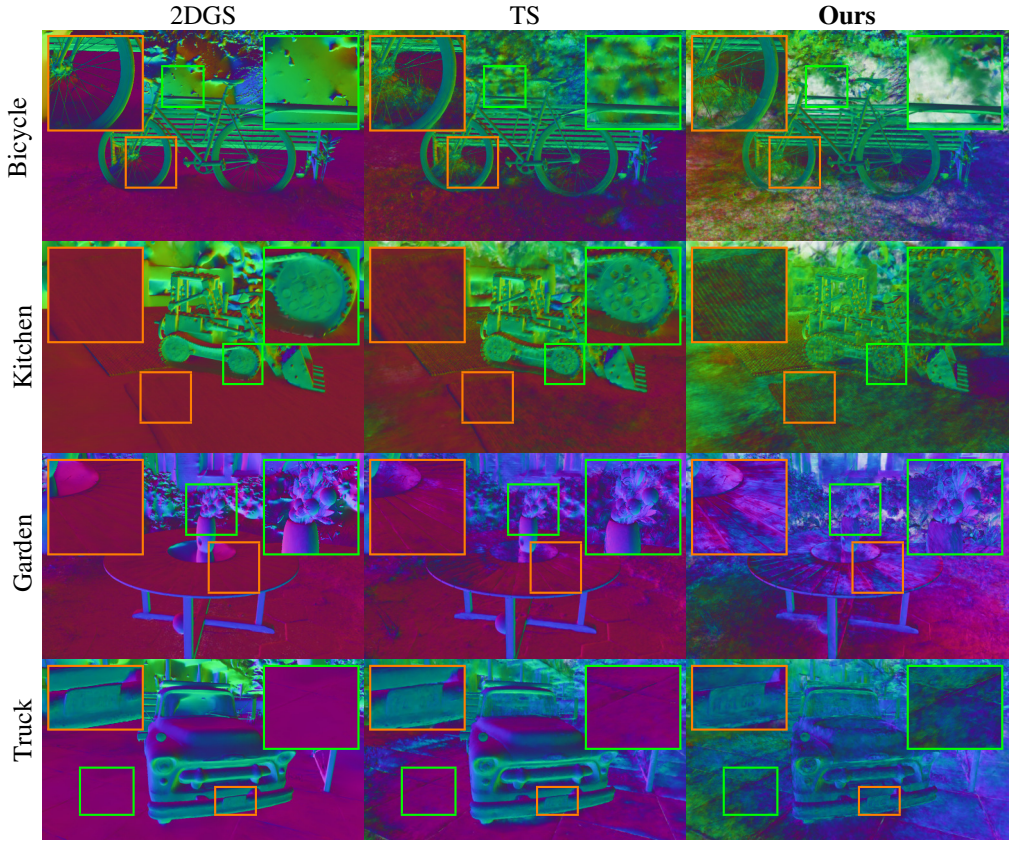


Figure 6: **Qualitative surface-normal comparison on Mip-NeRF 360 and Tanks & Temples.** Rows show Bicycle, Kitchen, Garden, and Truck; columns show 2DGS, TS, and QS. Enlarged regions highlight fine-scale surface structures, where the continuous bilinear quadrilateral parameterization produces more clearly resolved normal details.

and two, respectively. NeRF Synthetic contains eight objects, each with 100 training and 200 test views at 800×800 resolution on a white background. For Tanks & Temples, we evaluate on Train and Truck. Results are averaged over scenes, with indoor and outdoor Mip-NeRF 360 reported separately.

Implementation Details. We train QS and retrain 2DGS (Huang et al., 2024a) and TS (Held et al., 2025a) for 30,000 updates on a single NVIDIA RTX 4090. QS jointly optimizes primitive geometry, opacity, boundary falloff, and appearance using Adam, without an additional refinement stage. Appearance combines degree-three spherical harmonics with bilinear spatial residuals defined within each quadrilateral primitive. Real scenes use SfM-based initialization with auxiliary shell points, while synthetic scenes use a shared initialization of 100,000 random points. Primitive adaptation uses split, clone, and prune operations during training. Because the methods use different primitive parameterizations and adaptation strategies, final primitive counts and model sizes are not explicitly matched and are reported alongside efficiency results. Learning rates, loss weights, optimization schedules, and population-control settings are provided in Appendices A.4 and A.5.

Evaluation Metrics. We measure rendering quality using PSNR, SSIM (Wang et al., 2004), and VGG-based LPIPS (Zhang et al., 2018), where higher PSNR and SSIM and lower LPIPS indicate better quality. Metrics are averaged over test views and then equally across scenes. All retrained methods follow the same image-processing and metric protocol, detailed in Appendix A.6. We additionally report training time, rendering FPS, primitive count, and model storage. Rendering speed is measured over repeated test-view passes after warmup. Model storage includes scene representation and rendering parameters, excluding optimizer states, gradients, and temporary buffers.

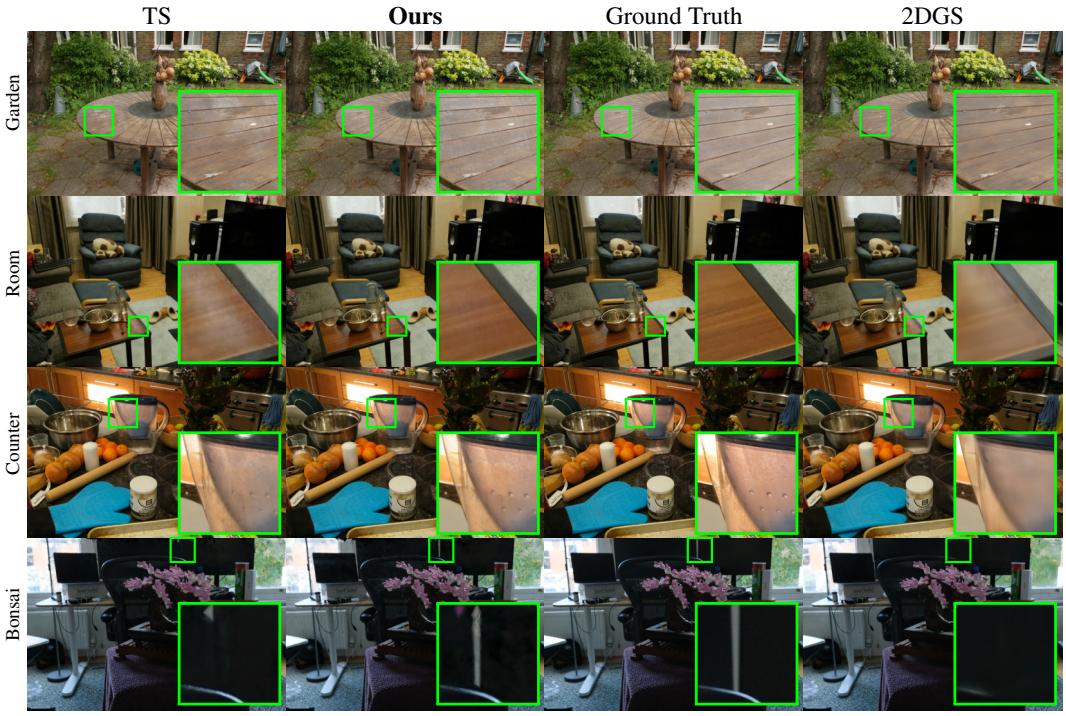


Figure 7: **Qualitative comparison on Mip-NeRF 360.** Garden is an outdoor scene; Room, Counter, and Bonsai are indoor scenes. Columns show TS, QS, ground truth, and 2DGS after 30,000 updates. QS better preserves local appearance variation and fine-scale details in novel views.

Table 1: **Quantitative comparison on Mip-NeRF 360 and Tanks & Temples.** Among non-volumetric methods, first, second, and third place are highlighted in red, orange, and yellow, respectively. Results marked † are adopted from the TS paper (Held et al., 2025a); 2DGS, TS, and QS are retrained for 30,000 updates.

	Outdoor Mip-NeRF 360			Indoor Mip-NeRF 360			Tanks & Temples		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Implicit Methods									
Instant-NGP†	–	–	–	–	–	–	21.92	0.745	0.305
Mip-NeRF 360†	24.47	0.691	0.283	31.72	0.917	0.179	22.22	0.759	0.257
Zip-NeRF†	25.58	0.750	0.207	32.25	0.926	0.167	–	–	–
Volumetric Primitives									
3DGS†	24.64	0.731	0.234	30.41	0.920	0.189	23.14	0.841	0.183
3DGS-MCMC†	25.51	0.760	0.210	31.08	0.917	0.208	24.29	0.860	0.190
DBS†	25.10	0.751	0.246	32.29	0.936	0.220	24.85	0.870	0.140
3DCS†	24.07	0.700	0.238	31.33	0.927	0.166	23.94	0.851	0.156
Non-Volumetric Primitives									
BBSplat†	23.55	0.669	0.281	30.62	0.921	0.178	25.12	0.868	0.172
2DGS (Retrained)	24.23	0.705	0.284	30.13	0.913	0.211	23.19	0.833	0.210
TS (Retrained)	24.19	0.722	0.217	30.88	0.926	0.162	23.07	0.855	0.144
QS (Ours)	24.08	0.713	0.218	31.21	0.927	0.157	23.24	0.854	0.142

4.2 Results

Quantitative Results. Tables 1 and 2 show that QS improves average PSNR, SSIM, and LPIPS over the retrained baselines on indoor Mip-NeRF 360 and NeRF Synthetic, while maintaining comparable outdoor image quality. Improvements across multiple scenes and metrics support the com-

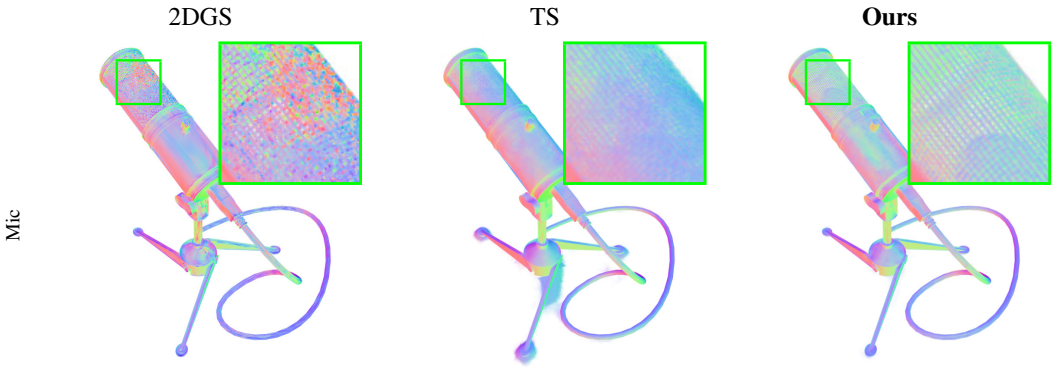


Figure 8: **Qualitative rendered-normal comparison on NeRF Synthetic Mic.** 2DGS, TS, and QS are shown from the same held-out view after 30,000 updates. Enlarged regions highlight fine-scale surface structure, where the continuous bilinear quadrilateral parameterization produces more clearly resolved normal details.

Table 2: **Quantitative comparison on NeRF Synthetic after 30,000 updates.** All methods use the same 200 test views per object; averages weight all eight objects equally. First, second, and third place are highlighted in red, orange, and yellow, respectively.

Metric	Method	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg.
PSNR↑	2DGS	35.262	26.091	35.426	37.396	35.106	29.669	35.035	30.665	33.081
	TS (Retrained)	35.919	25.587	35.259	37.870	35.881	30.348	35.128	31.107	33.387
	QS (Ours)	36.288	26.011	36.464	37.797	36.501	30.804	36.068	31.126	33.882
SSIM↑	2DGS	0.9849	0.9538	0.9881	0.9841	0.9802	0.9577	0.9905	0.9041	0.9679
	TS (Retrained)	0.9873	0.9473	0.9880	0.9851	0.9836	0.9616	0.9907	0.9039	0.9684
	QS (Ours)	0.9874	0.9515	0.9901	0.9852	0.9845	0.9639	0.9922	0.9022	0.9696
LPIPS↓	2DGS	0.0148	0.0398	0.0117	0.0226	0.0198	0.0387	0.0072	0.1105	0.0331
	TS (Retrained)	0.0123	0.0474	0.0116	0.0221	0.0143	0.0353	0.0073	0.1001	0.0313
	QS (Ours)	0.0118	0.0399	0.0093	0.0197	0.0127	0.0301	0.0058	0.0967	0.0283

bination of flexible quadrilateral geometry and continuous bilinear appearance. The appearance field captures local color variation within each primitive, while distinct depth layers and occlusion boundaries still require separate geometric supports. The benchmark evaluates this combined representation; component contributions are examined in the ablation studies. On Tanks & Temples, QS achieves improved average PSNR and LPIPS with a small SSIM trade-off. Table 4 shows fewer elements than TS on average on Mip-NeRF 360 and Tanks & Temples with real-time rendering, although fewer elements do not imply lower storage or faster computation. Table 3 further shows higher PSNR at every tested supervision level on Counter, with the advantage increasing as more views become available. These single-seed results illustrate the benefits of additional image evidence; protocol details appear in Appendix A.9.

Qualitative Results. Figure 7 illustrates how QS preserves local appearance variation and fine-scale structures in novel views. The enlarged regions show more clearly preserved table structure in Room, visible perforations in Counter, and thin background structures in Bonsai. These examples complement the aggregate image-quality metrics by showing where local reconstruction differences occur. Figures 6 and 8 further show more clearly resolved fine-scale structures in selected regions of the rendered normal maps, including Bicycle spokes and the Mic grille. These normal maps visualize the local surface orientations represented by QS. Detailed normal structure does not always translate directly to higher novel-view synthesis scores, since rendered normals describe local surface orientation while RGB reconstruction also depends on appearance, opacity, visibility, and image-space alignment. Together, the RGB and normal-map comparisons provide complementary evidence of

Table 3: **Quantitative comparison under reduced training-view supervision on Counter.** Both methods use 30,000 updates, the same 30 test views, and fixed SfM initialization and calibration. Results use one seed; the better result for each metric and training-view fraction is shown in bold.

	30% (63 views)			50% (105 views)			70% (147 views)			100% (210 views)		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TS (Retrained)	24.71	**0.826**	**0.245**	26.60	**0.872**	0.205	27.79	**0.890**	0.190	28.83	0.907	0.175
QS (Ours)	**24.85**	0.814	0.248	**26.78**	0.867	**0.203**	**28.09**	0.889	**0.184**	**29.27**	**0.910**	**0.166**

Table 4: **Efficiency comparison.** Per-scene averages after 30,000 updates on one RTX 4090. Model size denotes stored parameters, not peak GPU memory.

Dataset	Method	# Elements (M)↓	Model (MB)↓	Train/scene (min)↓	FPS↑
Mip-NeRF 360	2DGS (Retrained)	2.063	478.6	24.4	132.4
	TS (Retrained)	3.604	865.0	36.9	170.8
	QS (Ours)	3.093	894.0	52.9	148.1
Tanks & Temples	2DGS (Retrained)	0.867	201.0	10.9	262.0
	TS (Retrained)	2.100	503.9	19.7	334.5
	QS (Ours)	2.052	593.0	33.6	253.4

how QS represents both local appearance variation and fine-scale surface structure. Appendix A.7 provides further discussion of this distinction.

4.2.1 Ablation Studies

We ablate bilinear appearance and geometric flexibility on Bicycle, Counter, and Lego using four configurations, three paired seeds, 30,000 joint updates, and a 200k primitive cap. Full QS achieves the best average PSNR, SSIM, and LPIPS. Removing the spatial residual or restricting elements to parallelograms lowers average quality, although removing the residual slightly improves Lego PSNR and SSIM. The appearance ablation also reduces parameter count, so it does not isolate the interpolation rule. Increasing the cap to 400k improves all three metrics on every tested scene in seed-zero runs, at the cost of higher storage and lower FPS. Full results appear in Appendix A.8 and Tables 5 to 7. We also examine reduced training-view supervision on Counter with fixed SfM initialization and calibration. Table 3 shows higher PSNR for QS at every tested fraction and lower LPIPS from 50% supervision onward, while TS retains higher SSIM at reduced fractions and lower LPIPS at 30%. These single-seed results illustrate metric-dependent sensitivity to the available image evidence. Protocol details appear in Appendix A.9.

5 Conclusion

We introduced QS, an explicit, non-volumetric representation based on planar four-corner primitives with bilinear spatial appearance, shared opacity, and view-dependent coefficients. Differentiable polygon rasterization and planar-convex retraction enable joint optimization of geometry, opacity, boundary falloff, and appearance, while split, clone, and prune operations adapt the primitive population during training. Experiments on Mip-NeRF 360, NeRF Synthetic, and Tanks & Temples show that QS improves average PSNR, SSIM, and LPIPS over retrained TS and 2DGS on indoor Mip-NeRF 360 and NeRF Synthetic, and improves average PSNR and LPIPS on Tanks & Temples, where TS retains higher SSIM. Outdoor Mip-NeRF 360 remains more challenging, with comparable but not consistently improved rendering quality. Ablations support the combined contribution of spatially varying appearance and adaptable quadrilateral geometry, with scene-dependent effects. QS also uses fewer primitives on average than TS on Mip-NeRF 360 and Tanks & Temples while maintaining real-time rendering, at the cost of higher storage and slower training and rendering. Overall, the results support adaptable quadrilateral primitives as a promising representation for surface-based novel-view synthesis, with remaining challenges in outdoor reconstruction and computational efficiency.

AI USE STATEMENT

In this work, we used generative AI tools for paper editing, proofreading, and improving the clarity and presentation of the manuscript. We did not use generative AI tools for developing the research methodology, designing experiments, generating experimental results, or interpreting the results, and the remaining tasks requiring disclosure are not applicable to this work. We have reviewed all AI-assisted work, and all AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the implementation and evaluation details needed to reproduce our results throughout the main paper and appendix. The experimental setup specifies the datasets, initialization, training protocol, and evaluation metrics, while the appendix provides additional details on optimization, loss functions, primitive adaptation, and evaluation procedures. We also report per-scene quantitative results and efficiency measurements to support verification of the reported results.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023.

Weiwei Cai, Weicai Ye, Peng Ye, Tong He, and Tao Chen. DynaSurfGS: Dynamic Surface Reconstruction with Planar-based Gaussian Splatting. 2024.

Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting, 2024. URL https://arxiv.org/abs/2309.16585.

Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. *arXiv preprint arXiv:2402.14650*, 2024.

Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.

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

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

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

Han Huang, Yulun Wu, Chao Deng, Ge Gao, Ming Gu, and Yu-Shen Liu. Fatesgs: Fast and accurate sparse-view surface reconstruction using gaussian splatting with depth-feature consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes, 2024b. URL https://arxiv.org/abs/2312.14937.

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023.

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

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. Spotlight Presentation.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. *arXiv preprint arXiv:2311.16473*, 2023.

Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. Vastgaussian: Vast 3d gaussians for large scene reconstruction, 2024. URL https://arxiv.org/abs/2402.17427.

Rong Liu, Dylan Sun, Meida Chen, Yue Wang, and Andrew Feng. Deformable beta splatting, 2025. URL https://arxiv.org/abs/2501.18630.

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312. doi: 10.1145/3680528.3687694. URL https://doi.org/10.1145/3680528.3687694.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compact3d: Smaller and faster gaussian splatting with vector quantization. *arXiv preprint arXiv:2311.18159*, 2023.

Hoang Chuong Nguyen, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. IBGS: Image-based gaussian splatting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=AZLj6ObEDF.

Weining Ren, Zihan Zhu, Boyang Sun, Jiaqi Chen, Marc Pollefeys, and Songyou Peng. Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J. Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20626–20636, June 2023.

Sara Sabour, Lily Goli, George Kopanas, Mark Matthews, Dmitry Lagun, Leonidas Guibas, Alec Jacobson, David J. Fleet, and Andrea Tagliasacchi. SpotLessSplats: Ignoring distractors in 3d gaussian splatting. *arXiv:2406.20055*, 2024.

Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450316781. doi: 10.1145/2343483.2343501. URL https://doi.org/10.1145/2343483.2343501.

Hanxiao Sun, Yupeng Gao, Jin Xie, Jian Yang, and Beibei Wang. Svg-ir: Spatially-varying gaussian splatting for inverse rendering. *Proceedings of CVPR 2025*, 2025.

David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. Billboard splatting (bbsplat): Learnable textured primitives for novel view synthesis. *arXiv preprint arXiv:2411.08508*, 2024.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation, 2024. URL https://arxiv.org/abs/2309.16653.

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

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320, June 2024.

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

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

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19447–19456, June 2024a.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Yancheng Zhang, Guangyu Sun, and Chen Chen. Eggs: Exchangeable 2d/3d gaussian splatting for geometry-appearance balanced novel view synthesis, 2025. URL https://arxiv.org/abs/2512.02932.

Hongbi Zhou and Zhangkai Ni. Perceptual-gs: Scene-adaptive perceptual densification for gaussian splatting, 2025. URL https://arxiv.org/abs/2506.12400.

A Appendix

This appendix supplements Sections 3 and 4 with geometric derivations, implementation details, and the protocols supporting our quantitative, qualitative, ablation, and sensitivity studies. QS jointly optimizes geometry, opacity, boundary falloff, and appearance for 30,000 updates, with population adaptation performed within the same training loop. The experiments examine local appearance, surface structure, and the quality and resource trade-offs of the complete representation.

A.1 Planar-Convex Geometry and Retraction

For cyclic vertices $(p_0, p_1, p_2, p_3) = (x_{00}, x_{10}, x_{11}, x_{01})$, we recover the center, local frame, and deformation variables as

$$\begin{aligned} c &= \tfrac{1}{4}(p_0 + p_1 + p_2 + p_3), \qquad a = \tfrac{1}{4}(-p_0 + p_1 + p_2 - p_3), \\ b &= \tfrac{1}{4}(-p_0 - p_1 + p_2 + p_3), \quad d = \tfrac{1}{4}(p_0 - p_1 + p_2 - p_3). \end{aligned} \tag{9}$$

Using $\xi = 2u - 1$ and $\eta = 2v - 1$, Equation 1 becomes $S(\xi, \eta) = c + \xi a + \eta b + \xi\eta d$. After each unconstrained Adam update, we obtain the least-squares coordinates of d in the frame (a, b),

$$\begin{bmatrix} a^\top a & a^\top b \\ a^\top b & b^\top b \end{bmatrix} \begin{bmatrix} \widetilde{\gamma}_u \\ \widetilde{\gamma}_v \end{bmatrix} = \begin{bmatrix} a^\top d \\ b^\top d \end{bmatrix}, \qquad \gamma = \widetilde{\gamma} \min\left(1, \frac{\tau}{\|\widetilde{\gamma}\|_1}\right), \tag{10}$$

with $\tau = 0.95$ and $\gamma = 0$ when $\widetilde{\gamma} = 0$. We set $d' = \gamma_u a + \gamma_v b$ and reconstruct the four vertices from c, a, b, d'. Numerically singular frames use $d' = 0$ and are subsequently handled by renderer culling.

When a and b are linearly independent and $\|\gamma\|_1 \le \tau < 1$, the retracted surface lies in $c + \mathrm{span}(a, b)$. Its oriented area derivative is

$$\partial_\xi S \times \partial_\eta S = (1 + \gamma_u \eta + \gamma_v \xi)(a \times b), \qquad 1 + \gamma_u \eta + \gamma_v \xi \ge 1 - \tau > 0. \tag{11}$$

The map therefore has no interior fold. The four oriented corner turns have positive factors $4(1 \pm \gamma_u \pm \gamma_v)$ relative to $a \times b$, so the boundary is strictly convex, providing the valid surface footprint and projected support shown in Figure 3. Retraction is applied outside automatic differentiation after the optimizer step; Adam moments are retained.

A.2 Surface Coordinates and Appearance

The implementation evaluates the spatial appearance field in the centered basis $\xi = 2u - 1$ and $\eta = 2v - 1$,

$$c_q(u, v, \omega) = b_q(\omega) + r_q^u \xi + r_q^v \eta + r_q^{uv} \xi\eta, \tag{12}$$

where $b_q(\omega) = \max(0, \frac{1}{2} + \mathrm{SH}(\omega; H_q))$ componentwise. This is the centered field in Equation 6; residuals are added after clamping the SH base. All four corners share one view-dependent SH block, while the three RGB residual vectors are view independent. The field varies continuously within each quadrilateral; independently optimized elements do not enforce appearance continuity across their boundaries. This spatial variation describes color within an element and does not introduce surface curvature.

For optimization, we condition $\psi = (\xi, \eta, \xi\eta)^\top$ using normalized physical-area moments:

$$\mu_q = \mathbb{E}_A[\psi], \qquad C_q = \mathbb{E}_A[(\psi - \mu_q)(\psi - \mu_q)^\top], \qquad W_q = C_{q,\mathrm{reg}}^{-1/2}. \tag{13}$$

A 2×2 Gauss rule integrates these planar moments; covariance eigenvalues are floored at $\max(10^{-8}\lambda_{\max}, 10^{-12})$. Invalid caches use $\mu_q = 0$ and $W_q = I$. With residual vectors stored as rows, the learned coefficients decode as

$$D_q = W_q \widetilde{D}_q, \qquad H_{q,\mathrm{DC}} = \widetilde{H}_{q,\mathrm{DC}} - \frac{\mu_q^\top D_q}{C_0}, \qquad C_0 = \frac{1}{\sqrt{4\pi}}. \tag{14}$$

The DC correction precedes the SH clamp, preserving the field in Equation 12. Caches are constructed at initialization and for split children, held fixed between topology changes, and excluded from the learned checkpoint payload, which stores decoded coefficients. They add no learned appearance parameters.

To recover (u, v), we select the cyclic rotation with the largest adjacent-edge cross product in 3D. For the rotated world-space vertices $\bar{x}_0, \ldots, \bar{x}_3$, let $e = \bar{x}_1 - \bar{x}_0$, $f = \bar{x}_3 - \bar{x}_0$, and write $\bar{x}_2 - \bar{x}_0 = \kappa e + \nu f$. Projected corners and reciprocal homogeneous depths map each pixel to the physical plane coordinates (s, t) of $\bar{x}_0 + se + tf$. The inverse satisfies

$$s = u + (\kappa - 1)uv, \qquad t = v + (\nu - 1)uv. \tag{15}$$

Eliminating v gives

$$(\nu - 1)u^2 + [1 + (\kappa - 1)t - (\nu - 1)s]u - s = 0. \tag{16}$$

We use the linear limit when the quadratic coefficient is negligible, select an admissible root using the residual of Equation 15, and apply two Newton corrections. The cyclic rotation is undone when assigning corner weights. Small boundary excursions are clamped. If conditioning or inversion checks fail, the renderer uses equal nodal weights while retaining the primitive's opacity contribution.

A.3 Rasterization and Differentiation

The normalization radius ρ_q in Equation 5 is obtained from the largest inscribed circle of the projected convex polygon. We solve the small linear program

$$\max_{z,r} r \quad \text{subject to} \quad \ell_i(z) + r \le 0, \qquad i = 0, \dots, 3, \tag{17}$$

by enumerating triples of active constraints. Primitive admission requires opacity above 0.01, an inscribed radius of at least one pixel, and overlap with the image bounds. Pixel contributions below $1/255$ are rejected, and front-to-back compositing terminates when the remaining transmittance reaches 10^{-4}. Opacity is capped at 0.99. These contributions blend the local appearance fields into the rendered image, as shown in Figure 4.

The accelerated benchmark backward pass differentiates nodal appearance, opacity, boundary exponent, projected edges, the selected inradius solution, and the SH viewing direction. It evaluates the exact bilinear coordinate map in the forward pass, but omits that map's derivative with respect to geometry. Writing $z_q = (\xi, \eta)$, the complete color derivative contains two paths:

$$\frac{\partial c_q}{\partial X_q} = \frac{\partial b_q}{\partial \omega_q}\frac{\partial \omega_q}{\partial X_q} + \frac{\partial c_q}{\partial z_q}\frac{\partial z_q}{\partial X_q}. \tag{18}$$

The benchmark implementation retains the first path and holds z_q fixed for the second. Geometry still receives the separate opacity-coverage and geometric-regularization gradients. This is an approximate geometry gradient within a loop that jointly updates every parameter group. Sorting, culling, active-set choices, edge ties, and the accepted compositing prefix remain fixed within each local backward pass. Camera parameters remain fixed. Thus, the renderer provides piecewise derivatives of the executed rendering path rather than derivatives through discrete visibility decisions.

A.4 Initialization and Joint Optimization

For Mip-NeRF 360 and Tanks & Temples, we augment the supplied SfM cloud of N points with $\lfloor 0.05N \rfloor$ auxiliary points on an origin-centered Fibonacci sphere of radius $\max_{i,j} |p_{ij}|$. Each center receives a randomly oriented square scaled by local point spacing. SH DC is initialized from point colors, with white assigned to the auxiliary shell. Synthetic NeRF uses the same 100,000 seed-zero random points in $[-1.3, 1.3]^3$ for all compared methods, without the auxiliary shell. Initial opacity and boundary exponent are 0.28 and 1.16, respectively; higher-order SH and spatial-residual coefficients start at zero.

Each update samples one training camera and minimizes

$$\begin{aligned}\mathcal{L}_t = {} & 0.8\mathcal{L}_{\text{pixel}} + 0.2\big(1 - \text{SSIM}(\widehat{I}, I)\big) + \lambda_o \overline{o} \\ & + \mathbf{1}_{t>5000}\, 10^{-4}\mathcal{L}_{\text{normal}} + \mathbf{1}_{t<25000}\frac{10^{-8}}{\max(\overline{A}, 10^{-12})},\end{aligned} \tag{19}$$

Here $\mathcal{L}_{\text{pixel}}$ is mean absolute RGB error; the inherited large-outdoor-scene branch uses mean squared error when the initialized scene extent exceeds 300. That branch is inactive for all five Mip-NeRF 360 outdoor scenes. The normal loss averages $1 - \langle n_{\text{render}}, n_{\text{depth}} \rangle$ over pixels. The quantities $\overline{o}$ and $\overline{A}$ are mean primitive opacity and area, and $\lambda_o = 0.0055$. Training performs 30,000 Adam updates jointly over vertices, SH coefficients, spatial appearance, opacity, and boundary falloff. SH degree starts at zero and increases every 1,000 updates until degree three. There is no subsequent appearance-only or falloff-only stage.

Parameter group	Learning rate
Vertices	$1.8\times10^{-3} \rightarrow 1.8\times10^{-5}$
SH DC / remaining SH	2.5×10^{-3} / 1.25×10^{-4}
Spatial residual	7×10^{-4}
Opacity / boundary falloff	0.014 / 8×10^{-4}

The vertex rate decays exponentially over the 30,000-update horizon; the other listed rates remain constant. Opacity uses a sigmoid and the boundary exponent uses $\sigma = 0.01 + \exp(s)$. Adam

precedes topology operations, including on split/prune iterations, so parameter replacement does not discard that iteration's update.

For accelerated 30K Mip-NeRF 360 and Synthetic runs, the first 10K updates interleave half-resolution images with a full-resolution image every fourth update; the first update is also full resolution. The remaining 20K updates are full resolution, completing the same 30,000 joint updates. The implemented schedule therefore contains 7,499 half-resolution and 22,501 full-resolution updates, with every parameter group updated at both resolutions. Reported Tanks & Temples runs use full resolution throughout all 30,000 updates. The component ablation protocol is specified separately in Appendix A.8.

A.5 Primitive Population Adaptation

Population adaptation combines splitting, cloning, and pruning every 500 updates, as shown in Figure 5. Growth is active for $500 < t < 25000$, followed by final pruning for $t > 25000$. A growth event targets $\min(N_{\max}, \lfloor 1.3N \rfloor)$ primitives. Candidate selection alternates between weights proportional to opacity and inverse boundary exponent. Selected elements with a recorded projected size above 24 pixels split into four children using edge midpoints and the center; smaller selected elements are cloned with random in-plane displacement.

For a parent residual $\mathbf{a}x + \mathbf{b}y + \mathbf{m}xy$, a child centered at $(x_0, y_0) \in \{-\frac{1}{2}, \frac{1}{2}\}^2$ receives

$$\mathbf{a}' = \tfrac{1}{2}\mathbf{a} + \tfrac{1}{2}y_0\mathbf{m}, \qquad \mathbf{b}' = \tfrac{1}{2}\mathbf{b} + \tfrac{1}{2}x_0\mathbf{m}, \qquad \mathbf{m}' = \tfrac{1}{4}\mathbf{m}. \tag{20}$$

The constant offset $x_0\mathbf{a}+y_0\mathbf{b}+x_0y_0\mathbf{m}$ is absorbed into SH DC, and residuals are re-encoded in each child's normalized basis. This polynomial restriction is exact within an unchanged unclipped SH branch. Changes in opacity support and clipping mean that splitting is not globally image preserving. Clones inherit appearance and assign $o' = 1 - \sqrt{1 - o}$ to each of two copies, with a finite-logit safeguard. New child moments start at zero; unchanged elements retain optimizer state.

An element is pruned if any enabled criterion holds: opacity at most 0.014, maximum blending contribution below 0.022, or fewer than two qualifying views with projected inradius above one pixel. The contribution test is enabled when there are fewer than 250 training views or no view-support collection round is active. The view-count test starts after update 1,000 and is suspended during collection rounds. These rounds traverse the training cameras after population changes to accumulate support counts. Under the same post-1,000 condition, indoor and synthetic growth also removes elements whose recorded projected size exceeds 1,400 pixels. Growth ceases at 25K; later events only prune. Splitting costs three net elements and cloning one; sampled parents are restricted to the prefix fitting the remaining capacity. The benchmark configuration does not use merging, directional splitting, or appearance-gradient-based candidate selection.

A.6 Evaluation Protocol

For each test view, predictions and targets are quantized to the same 8-bit RGB representation used by the Triangle Splatting evaluation pipeline. PSNR uses mean squared pixel error, SSIM uses the shared implementation, and LPIPS uses the VGG backbone from `lpipsPyTorch` with inputs in $[0, 1]$. This input convention differs from implementations that normalize LPIPS inputs to $[-1, 1]$. The shared protocol applies to our locally retrained 2DGS, TS, and QS results; published entries retain their source protocols and are marked separately in Table 1. Metrics are averaged over all test views of each scene and then equally over scenes. Outdoor and indoor Mip-NeRF 360 means contain five and four scenes, respectively; Synthetic NeRF contains eight objects, each with 200 test views. The Tanks & Temples average weights Train and Truck equally. Benchmark training does not enforce equal final primitive counts or model storage.

Rendering speed is measured on one RTX 4090 after ten warmup frames and five complete passes through the test cameras. Timing includes camera-dependent projection, appearance evaluation, binning, sorting, and compositing. Model size reports the uncompressed learned tensor payload and excludes optimizer state, allocator overhead, transient rasterization buffers, and runtime caches. For inference, we cache the camera-independent part of the quadrilateral coordinate map, omit output writes for culled elements, evaluate the compact SH-plus-bilinear appearance basis without materializing four corner colors, and cooperatively stage 32 primitives per compositor block. These changes

do not modify the learned representation. All held-out views on the three evaluated benchmarks pass a same-checkpoint rendering audit; the largest absolute raw RGB difference from the training renderer is 2.57×10^{-6}, and the reported quality metrics are unchanged. In the native TS clone rule, a parent opacity o is transferred to two children using $o' = 1 - \sqrt{1 - o}$ followed by the inverse sigmoid. On Ship, 51 parent opacities rounded to exactly one in FP32, which would produce 102 positive-infinite child logits. We cap only these rounded values at $\mathrm{nextafter}(1, 0)$ before evaluating the same rule; every finite native output is retained unchanged. The 30,000-update raw model is entirely finite, and all 200 held-out views pass the live-versus-reloaded rendering audit. The stabilized 30,000-update Ship endpoint is included in the eight-object TS average.

A.7 Interpreting Surface Normals and Image Fidelity

Figures 6 and 8 complement the RGB comparisons in Figure 7 by visualizing local surface orientation. The clearly resolved patterns in selected QS regions describe the structure visible in these normal maps. They address a different aspect of the representation from the final RGB agreement measured by PSNR, SSIM, and LPIPS, as also reflected in the separate appearance and geometry evaluations of 2DGS (Huang et al., 2024a).

For a nondegenerate planar-convex element, Equation 11 implies the unit normal

$$n_q(u, v) = \frac{\partial_u S_q \times \partial_v S_q}{\|\partial_u S_q \times \partial_v S_q\|} = \frac{a_q \times b_q}{\|a_q \times b_q\|}. \tag{21}$$

It is constant over the primitive despite the bilinear surface coordinates. The normal visualization transforms normals into a common world frame, uses the renderer's camera-facing sign convention, and normalizes their front-to-back accumulation. At foreground pixels with nonzero accumulation,

$$\widehat{n}(p) = \mathrm{normalize}\left(\sum_q T_q(p)\alpha_q(p)n_q\right). \tag{22}$$

The bilinear RGB coefficients do not enter this expression. With geometry, opacity, and boundary falloff fixed, changing appearance leaves the normal map unchanged while changing the rendered image. Fine normal patterns therefore arise from the learned element orientations and their blending, rather than from bilinear color interpolation alone. Appearance and geometry interact during joint training, but the qualitative comparisons do not isolate this indirect interaction.

For RGB images scaled to $[0, 1]$, PSNR is $-10 \log_{10}(\mathrm{MSE})$. RGB agreement depends on appearance, opacity, visibility, and image-space alignment as well as geometry. A displaced contour can increase RGB error without changing the orientations displayed on either side. Conversely, appearance changes can improve RGB agreement without changing the normal map. Opacity blending can also combine several orientations within one pixel. Thus, visibly finer normal patterns do not imply a particular ordering of NVS scores. The outdoor examples illustrate how resolved normal structure can coexist with comparable overall rendering quality and remaining RGB errors. We use these visualizations to characterize local orientation patterns, not as a quantitative measure of geometric accuracy or an explanation of the outdoor metric differences.

A.8 Component Ablations and Primitive Capacity

Configurations and protocol. We evaluate four configurations on Bicycle, Counter, and Lego: full QS; QS without the spatial residual; QS restricted to parallelograms; and QS with both restrictions. The appearance control fixes the nine spatial RGB coefficients to zero, retaining the shared SH appearance. The geometry control sets the retraction margin to zero, enforcing $d = 0$ in Equation 9. All other settings are shared within each scene. Seeds 0, 1, and 2 are paired across configurations using identical initial fields and training-camera sequences. Each of the 36 runs performs 30,000 updates of its active parameter groups under a 200k primitive cap, using the benchmark image resolution throughout. Three additional seed-zero runs evaluate full QS at a 400k cap. These component studies use their common ablation trainer; their timing measurements are separate from the accelerated full-benchmark results in Table 4.

Appearance and geometry. Table 5 summarizes all 36 runs as means and sample standard deviations over three paired seeds. Full QS achieves the best three-scene average on all metrics. Removing spatial appearance or restricting geometry reduces average quality, with the largest PSNR

loss when both restrictions are applied. This supports their combined use in QS, while the slight Lego PSNR and SSIM gains without the residual show that their benefits depend on scene content. The appearance control has 48 active color coefficients per element rather than 57. It therefore tests the spatial appearance component, including its additional capacity, rather than isolating the interpolation rule. Standard deviations describe variation across the three seeds and are not confidence intervals.

Table 5: **Quantitative comparison of appearance and geometry ablations.** All configurations use a 200k primitive cap and 30,000 updates. Values are means ± sample standard deviations over three paired seeds; Avg. first averages the three scenes within each seed. Within each scene or average, the top three mean values are highlighted in red, orange, and yellow, respectively.

Scene	Configuration	PSNR↑	SSIM↑	LPIPS↓
Bicycle	Full QS	23.080 ± 0.047	0.62567 ± 0.00068	0.35114 ± 0.00050
	No spatial residual	22.750 ± 0.006	0.60378 ± 0.00021	0.38145 ± 0.00036
	Parallelogram	22.911 ± 0.012	0.61586 ± 0.00091	0.35744 ± 0.00086
	Both restrictions	22.480 ± 0.008	0.58929 ± 0.00006	0.39039 ± 0.00027
Counter	Full QS	28.501 ± 0.054	0.89259 ± 0.00017	0.21203 ± 0.00032
	No spatial residual	28.328 ± 0.020	0.88773 ± 0.00015	0.22237 ± 0.00021
	Parallelogram	28.384 ± 0.035	0.88991 ± 0.00041	0.21564 ± 0.00028
	Both restrictions	28.166 ± 0.023	0.88364 ± 0.00016	0.22796 ± 0.00013
Lego	Full QS	35.830 ± 0.010	0.98272 ± 0.00004	0.01464 ± 0.00004
	No spatial residual	35.862 ± 0.015	0.98307 ± 0.00004	0.01482 ± 0.00005
	Parallelogram	35.610 ± 0.006	0.98202 ± 0.00002	0.01560 ± 0.00004
	Both restrictions	35.538 ± 0.006	0.98200 ± 0.00002	0.01646 ± 0.00003
Avg.	Full QS	29.137 ± 0.029	0.83366 ± 0.00025	0.19260 ± 0.00014
	No spatial residual	28.980 ± 0.002	0.82486 ± 0.00005	0.20621 ± 0.00018
	Parallelogram	28.968 ± 0.012	0.82926 ± 0.00018	0.19623 ± 0.00024
	Both restrictions	28.728 ± 0.012	0.81831 ± 0.00003	0.21161 ± 0.00012

Efficiency and capacity. Table 6 reports the realized element counts, model sizes, training times, and rendering rates for the same runs. All configurations store four vertices and retain the nine spatial coefficient slots, including zero padding in the appearance controls. Consequently, stored size is not the minimum deployable size of each control, and equal caps do not imply equal final counts. Training times include the control-specific validation checks. Inference uses the same validated tile-16, batch-64 renderer across these ablations, with ten warmup frames and five complete test-view passes. Table 7 shows that increasing the cap to 400k improves all three image metrics on each tested scene, with increased storage and lower FPS. The representation therefore benefits from additional capacity as well as from its within-element appearance and geometry. These capacity comparisons use seed zero and characterize the quality and resource trade-off for those runs.

Table 6: **Efficiency comparison of appearance and geometry ablations.** Values are means ± sample standard deviations over three seeds at a 200k primitive cap on one RTX 4090. Model size includes padded zero residuals in the appearance controls; training times include their validation overhead.

Scene	Configuration	Elements (M)	Model (MB)	Train (min)	FPS
Bicycle	Full QS	0.1992 ± 0.0000	57.56 ± 0.00	18.14 ± 0.36	865.33 ± 37.26
	No spatial residual	0.1991 ± 0.0000	57.55 ± 0.01	16.87 ± 0.65	900.33 ± 100.31
	Parallelogram	0.1991 ± 0.0000	57.55 ± 0.01	18.10 ± 0.43	900.86 ± 53.78
	Both restrictions	0.1991 ± 0.0000	57.53 ± 0.01	16.78 ± 0.57	964.75 ± 35.97
Counter	Full QS	0.1983 ± 0.0000	57.32 ± 0.01	41.35 ± 0.60	474.60 ± 17.89
	No spatial residual	0.1989 ± 0.0000	57.49 ± 0.00	39.61 ± 1.46	472.32 ± 15.27
	Parallelogram	0.1983 ± 0.0001	57.32 ± 0.02	41.73 ± 0.59	514.82 ± 8.02
	Both restrictions	0.1989 ± 0.0001	57.48 ± 0.02	40.88 ± 1.91	475.25 ± 27.10
Lego	Full QS	0.1895 ± 0.0000	54.76 ± 0.00	19.89 ± 0.99	628.78 ± 48.38
	No spatial residual	0.1893 ± 0.0000	54.71 ± 0.01	18.41 ± 0.89	675.97 ± 22.16
	Parallelogram	0.1889 ± 0.0001	54.58 ± 0.02	19.50 ± 1.00	682.54 ± 35.76
	Both restrictions	0.1881 ± 0.0002	54.35 ± 0.07	18.18 ± 1.12	719.31 ± 38.03

Table 7: **Quantitative comparison at different primitive budgets.** Full QS is trained from scratch for 30,000 updates with seed 0. The 200k entries reuse the corresponding component-ablation runs, while the 400k entries are additional runs. Image-quality metrics, model size, and FPS show the quality and resource trade-offs.

Scene	Cap	PSNR↑	SSIM↑	LPIPS↓	Elements (M)	Model (MB)	FPS↑
Bicycle	200k	23.091	0.62551	0.35105	0.1991	57.55	822.32
	400k	23.527	0.66632	0.29915	0.3982	115.09	640.69
Counter	200k	28.476	0.89242	0.21239	0.1983	57.32	493.45
	400k	28.899	0.90295	0.19077	0.3938	113.81	342.76
Lego	200k	35.839	0.98275	0.01462	0.1895	54.75	631.23
	400k	36.281	0.98386	0.01332	0.3259	94.18	466.93

Table 8: **Quantitative comparison on individual Tanks & Temples scenes.** Results for Train and Truck use locally retrained models after 30,000 updates under the same evaluation protocol. First, second, and third place are highlighted in red, orange, and yellow, respectively, using unrounded values.

	LPIPS↓		PSNR↑		SSIM↑	
Method	Truck	Train	Truck	Train	Truck	Train
2DGS (Retrained)	0.174	0.247	25.07	21.31	0.873	0.794
TS (Retrained)	0.108	0.181	24.91	21.23	0.888	0.822
QS (Ours)	0.107	0.177	25.05	21.43	0.886	0.822

A.9 Sensitivity to Training-View Supervision

Table 3 evaluates nested subsets of Counter's 210 training images: 63, 105, 147, and 210 views, corresponding to 30%, 50%, 70%, and 100% supervision. Every run uses the same 30 test views and 30,000 updates. The full-supervision setting reuses the completed benchmark endpoints. Counter was selected because QS improves all three metrics over TS under full supervision, allowing us to examine how those benefits change as image evidence is reduced.

Full-scene SfM points, camera calibration, and scene extent remain fixed across fractions. The methods share the sampled training-camera sequence at each fraction, while retaining their respective benchmark training configurations and resolution schedules. This is a supervision-sensitivity study with fixed initialization, not an end-to-end sparse-view reconstruction experiment or a component ablation.

QS maintains higher PSNR at every tested fraction, and its advantage increases as more views become available. LPIPS also improves relative to TS at 50%, 70%, and 100% supervision. SSIM favors TS at the reduced fractions, and LPIPS favors TS at 30%. The results therefore show that QS remains effective with reduced supervision, with metric-dependent benefits from additional image evidence. The single-scene, single-seed protocol characterizes this setting and does not establish lower sensitivity to removing views across datasets.

A.10 Per-Scene Quantitative Results

Tables 8 to 11 report our locally retrained 2DGS, TS, and QS results after 30,000 updates using the same evaluation protocol. Red, orange, and yellow cells denote first, second, and third place, respectively, ranked using unrounded metrics.

A.11 Limitations and Future Directions

Scene-dependent benefits. QS improves average image quality on indoor Mip-NeRF 360 and NeRF Synthetic while maintaining comparable outdoor quality. Its largest gains occur in the evaluated indoor and object-centric settings. Quadrilaterals provide flexible local coverage and within-element color variation, while distinct depth layers still require separate surface elements. Adapting primitive allocation to local scene complexity is a natural direction for extending these benefits to vegetation

Table 9: **Quantitative comparison of LPIPS on individual Mip-NeRF 360 scenes.** All methods are locally retrained for 30,000 updates and evaluated under the same protocol. First, second, and third place are highlighted in red, orange, and yellow, respectively, using unrounded values.

Method	Bicycle	Flowers	Garden	Stump	Treehill	Room	Counter	Kitchen	Bonsai
2DGS (Retrained)	0.269	0.375	0.146	0.258	0.374	0.244	0.229	0.146	0.227
TS (Retrained)	0.189	0.285	0.106	0.216	0.290	0.188	0.175	0.117	0.169
QS (Ours)	0.206	0.271	0.111	0.216	0.288	0.185	0.166	0.113	0.162

Table 10: **Quantitative comparison of PSNR on individual Mip-NeRF 360 scenes.** All methods are locally retrained for 30,000 updates and evaluated under the same protocol. First, second, and third place are highlighted in red, orange, and yellow, respectively, using unrounded values.

Method	Bicycle	Flowers	Garden	Stump	Treehill	Room	Counter	Kitchen	Bonsai
2DGS (Retrained)	24.78	21.15	26.69	26.13	22.40	30.83	28.13	30.35	31.22
TS (Retrained)	25.00	20.76	27.16	26.13	21.90	31.19	28.83	31.39	32.11
QS (Ours)	24.85	20.74	27.01	25.82	21.97	31.37	29.27	31.64	32.56

and occlusion boundaries. The present benchmark does not isolate the causes of the different indoor and outdoor outcomes.

Representation and computation. QS maintains real-time rendering and uses fewer elements than TS on average on Mip-NeRF 360 and Tanks & Temples. However, its additional per-element parameters and rendering work lead to greater storage and longer training and rendering times in these comparisons, as shown in Table 4. The benchmark therefore evaluates a different allocation of representation capacity rather than quality at matched storage. The accelerated backward pass also holds appearance coordinates fixed when differentiating geometry, as detailed in Appendix A.3. Compact parameter storage and more efficient gradient evaluation remain opportunities for improving the balance between rendering quality and computational cost.

Scope of the evidence. Our normal visualizations characterize visible orientation patterns, complementing the RGB evaluation without serving as a quantitative geometry benchmark. The component ablations support the combined geometry and appearance design, while comparisons against equally parameterized appearance alternatives would further clarify the role of bilinear interpolation. The Counter experiment measures sensitivity to image supervision with fixed initialization on one scene and one seed. Extending this study across scenes and seeds would clarify how these benefits vary with the amount and distribution of training views.

Table 11: **Quantitative comparison of SSIM on individual Mip-NeRF 360 scenes.** All methods are locally retrained for 30,000 updates and evaluated under the same protocol. First, second, and third place are highlighted in red, orange, and yellow, respectively, using unrounded values.

Method	Bicycle	Flowers	Garden	Stump	Treehill	Room	Counter	Kitchen	Bonsai
2DGS (Retrained)	0.732	0.574	0.844	0.756	0.619	0.909	0.894	0.918	0.932
TS (Retrained)	0.767	0.613	0.863	0.756	0.609	0.924	0.907	0.927	0.944
QS (Ours)	0.749	0.612	0.856	0.747	0.602	0.924	0.910	0.929	0.946