CONTINUOUS SCENE-SCALE 3DGS REPRESENTATION

Anonymous authors
Paper under double-blind review

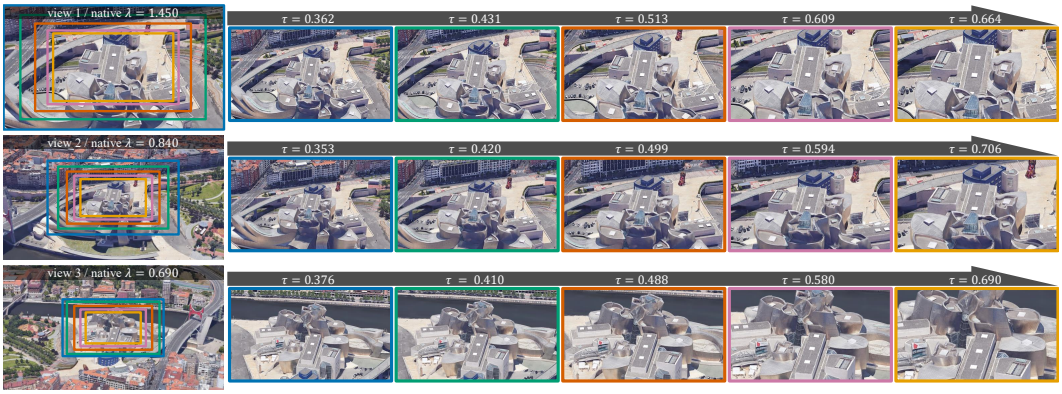


Figure 1: Continuous scene-scale rendering with our ContinuousGS. **Left:** Three native *Bilbao* views have identical resolutions but different sampling densities, reflected by their native scales λ_i. **Right:** One representation renders approximately matched scene scales τ across views. As τ increases, fixed-size center crops cover smaller regions and reveal finer detail. Matching colored boxes indicate the corresponding crop regions in each native view.

ABSTRACT

Continuous scene-scale rendering requires comparable scales across viewing conditions. Common resize-based settings overlook focal-length and depth differences, assigning the same scale to observations with different sampling densities and compromising supervision and evaluation. We establish unified scene scale by combining image resize factors with robust, scene-normalized projection-density estimates, yielding a shared continuous coordinate that approximately aligns sampling density across cameras. Building on this formulation, we introduce ContinuousGS, a 3D Gaussian Splatting (3DGS) framework that adapts one representation across scene scales. Target-first sampling balances supervision over supported log-scale intervals, while anchored appearance and opacity residuals enable continuous adaptation with exact recovery of base attributes at the global finest supported scale. Matched-scale multi-view evidence regularizes appearance residuals and guides detail allocation. Our evaluation protocol aggregates cross-view performance along the shared scale axis, reporting full-range and high-support results separately. Experiments on four datasets demonstrate that ContinuousGS improves rendering quality across scene scales compared with representative methods while largely preserving native-resolution quality.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) supports high-quality novel-view synthesis through an explicit scene representation and differentiable rasterization. For continuous multi-scale rendering, one reconstruction should remain coherent as camera intrinsics, scene depth, and output resolution change the world-space pixel footprint. Representative approaches address these changes through footprint filtering (Yu et al., 2024) or level-of-detail (LoD) selection (Ren et al., 2025). In multi-resolution image supervision and evaluation, scale is often indexed by an image resize factor s, with $s = 1$ denoting native resolution. However, equal resize factors need not correspond to equal scene sampling densities when focal lengths or depths differ. For example, even after two images are downsampled by half ($s = 1/2$), a longer focal length or shallower scene depth can allocate

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.

more pixels to the same scene region (Fig. 1). Grouping these observations under the same resize label therefore does not ensure matched scene sampling scales across views. Footprint filtering and LoD selection adapt rendering detail, but these mechanisms alone do not specify a common scale coordinate for organizing cross-view supervision and evaluation.

We establish *unified scene scale*, a dimensionless measure of projected sampling density shared by cameras within one scene. Here, scale describes image sampling of the scene rather than its physical size. For camera i, a robust native projection-density statistic ρ_i provides a view-level estimate of pixels per unit scene extent from calibrated focal lengths and reconstructed depth. Using a reference density ρ_{ref} from the training cameras, we define the normalized native scale $\lambda_i = \rho_i/\rho_{\text{ref}}$ and the query scale $\tau = s_i\lambda_i$. Normalization cancels any global rescaling of the reconstructed geometry. Each camera realizes target scene scales within its supported interval by resizing according to its native scale, without upsampling. We use one representative scalar per view to support conditioning and cross-view evaluation. This approximation does not equalize spatially varying pixel footprints, and normalized scales from different scenes need not imply equal absolute sampling densities.

Building on this coordinate, we introduce ContinuousGS, which adapts a single 3DGS representation across scene scales. Camera-first sampling induces a scene-scale distribution that depends on the camera sampling policy and the overlap and extent of realizable intervals, potentially overweighting some scale ranges. We instead sample uniformly with respect to log scene scale over the union of realizable intervals before choosing a compatible camera. To model scale-dependent averaging and coverage, anchored residuals adapt base appearance and opacity while sharing geometry and view-dependent appearance. The residuals vanish at the finest scale in the training support, preserving the learned base attributes at this canonical endpoint. Matched-scale multi-view evidence regularizes appearance residuals and guides bounded additional densification. The original rasterizer and density-control procedure remain in place. To evaluate this representation at scales aligned with training, we use the same scene-scale coordinate while accounting for the different intervals supported by test cameras and the actual integer output resolutions. We integrate cross-view performance by log-scale length so that equal multiplicative changes in sampling density receive equal weight, consistent with training. The *full-range* score covers the union of measured scale intervals, including regions supported by only one view. The *high-support* score restricts this range to scales supported by at least five views, and retained coverage reports the fraction that remains. This design connects supervision, representation adaptation, and evaluation through a shared scene-scale coordinate while explicitly accounting for the available view support.

The main contributions are as follows.

- We establish unified scene scale, a continuous coordinate for comparing representative sampling densities across cameras within a scene, invariant to the global scale of reconstruction.
- We introduce ContinuousGS, which combines target-first log-scale sampling with anchored attribute adaptation and multi-view evidence to learn a continuous scene-scale representation.
- We design a unified continuous-scale evaluation protocol that aligns dense scale samples across test views and integrates their performance along a shared scene-scale axis.
- Experiments across 21 scenes from four datasets validate the rendering advantages of ContinuousGS across scene scales.

2 Related Work

Neural Scene Representations. Radiance-field methods (Mildenhall et al., 2022; Liao et al., 2025) learn scene density and view-dependent appearance from posed images, with neural networks providing a continuous representation for novel-view synthesis. Explicit Gaussian representations (Kerbl et al., 2023; Chen & Wang, 2026) encode scenes as collections of anisotropic primitives and combine differentiable rasterization with adaptive density control to support efficient scene reconstruction and rendering. Extensions to primitive geometry (Huang et al., 2024) and spatial organization (Lu et al., 2024) improve surface reconstruction and view adaptation. Complementary optimization strategies use pixel coverage (Zhang et al., 2024), stochastic sampling (Kheradmand et al., 2024), and image structure (Lyu et al., 2026) to guide primitive allocation and refinement. Spatial continuity and view adaptation do not by themselves make sampling density an explicit variable of a scene representation. We condition a single reconstruction on unified scene scale to represent continuous changes in sampled appearance at comparable densities across views.

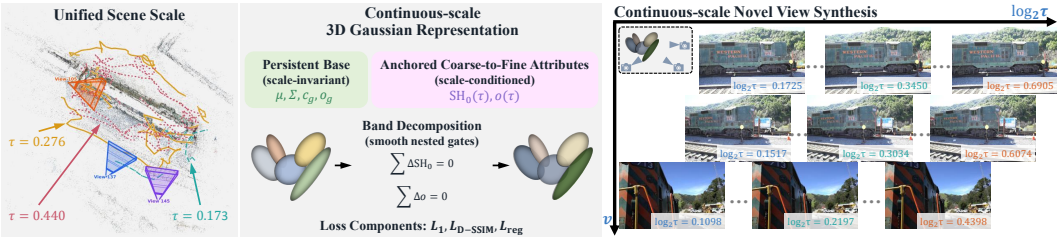


Figure 2: Overview of ContinuousGS. **Left:** Camera intrinsics and reconstructed geometry define a shared scene scale τ. Schematic curves connect camera-axis positions for the endpoints and logarithmic midpoint of the scale interval shared by highlighted views. **Center:** A persistent Gaussian base is augmented with scale-conditioned appearance and opacity residuals controlled by smooth nested gates. Anchoring recovers the base attributes at the finest supported scale. **Right:** One representation renders novel views across viewpoint v (rows) and scene scale τ (columns).

Multi-Scale Rendering. Changes in viewing distance and output resolution alter projected pixel footprints, requiring rendering methods to preserve image quality across sampling densities. The plenoptic sampling analysis of Chai et al. (2000) relates scene depth, image sampling density, and output resolution in image-based rendering. Reconstruction filtering in splatting (Zwicker et al., 2002) and footprint-aware integration in neural radiance fields (Barron et al., 2021; 2023) account for finite pixel footprints as sampling scales vary. Within 3DGS, Yu et al. (2024) combine a 3D smoothing filter with a screen-space Mip filter to address changes in sampling rate. Related approaches improve pixel-window integration (Liang et al., 2024), scale-adaptive filtering (Song et al., 2024; Yang et al., 2025), and primitive responses under viewing changes (Steiner et al., 2025; Tu et al., 2025). Complementing filtering, MSGS (Yan et al., 2024) organizes primitives into discrete scale levels, while Li et al. (2025) learn attribute and position changes for the queried scale. Discrete hierarchies and continuously queried rendering mechanisms offer different forms of scale adaptation, but neither by itself specifies how training observations should be distributed along a shared scene-scale continuum. We sample target scene scale continuously before selecting a compatible view and condition a shared Gaussian representation on the same coordinate, coupling scale coverage during learning with continuous appearance adaptation.

LoD Rendering. Progressive geometric and neural representations use mesh refinement (Hoppe, 1996) and staged network growth (Xiangli et al., 2022) to organize detail across resolutions or viewing distances. Hierarchical Gaussian methods select cumulative detail levels (Ren et al., 2025) or coarsened primitive sets (Kulhanek et al., 2025) by viewing distance, reducing computation on details with small projected footprints. To address memory constraints, Kulhanek et al. (2025) precompute active sets for chunks and blend their opacity contributions across regions. Continuous LoD uses geometric transitions (Hoppe, 1996) or learned distance-dependent attenuation of Gaussian contributions (Cheng et al., 2026) to reduce abrupt changes in detail. LoD guides anti-aliasing through learned sampling-rate-dependent filters (Yang et al., 2025) that balance aliasing suppression against detail loss. For distance-driven LoD, viewpoint proximity alone does not determine sampling density, which also depends on focal length and output resolution. ContinuousGS conditions appearance and opacity residuals on scene sampling density, retaining shared geometry and anchoring the residuals to recover base attributes at the finest training scale.

3 Methodology

3.1 3D Gaussian Splatting Preliminaries

Given calibrated images $\{(\mathbf{I}_i, \mathbf{K}_i, \mathbf{W}_i)\}$ of a static scene, we seek a single 3DGS representation that supports continuous scene-scale queries. Here, $\mathbf{K}_i$ contains the camera intrinsics and $\mathbf{W}_i$ maps world coordinates to camera coordinates. The standard 3DGS formulation (Kerbl et al., 2023; Chen & Wang, 2026) represents the scene with N anisotropic Gaussian primitives $\mathcal{G} = \{(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \mathbf{c}_g, o_g)\}_{g=1}^{N}$. Each primitive has a center $\boldsymbol{\mu}_g$, a covariance $\boldsymbol{\Sigma}_g$, spherical-harmonic coefficients $\mathbf{c}_g$ for view-dependent color, and an opacity logit o_g. Its opacity is $\sigma(o_g)$, where σ denotes the sigmoid function. Local affine projection gives $\boldsymbol{\Sigma}_{ig}^{\mathrm{2D}} = \mathbf{J}_{ig}\mathbf{R}_i\boldsymbol{\Sigma}_g\mathbf{R}_i^\top\mathbf{J}_{ig}^\top$, where

$\mathbf{R}_i$ is the rotation component of $\mathbf{W}_i$ and $\mathbf{J}_{ig}$ is the projection Jacobian at the camera-space Gaussian center. With Gaussians ordered by depth, front-to-back compositing yields $\widehat{\mathbf{I}}_i(\mathbf{x}) = \sum_g \mathbf{c}_g(\mathbf{v}_{ig})\widetilde{\alpha}_{ig}(\mathbf{x}) \prod_{j<g}(1-\widetilde{\alpha}_{ij}(\mathbf{x}))$. Here, $\mathbf{v}_{ig}$ is the viewing direction and $\widetilde{\alpha}_{ig}(\mathbf{x})$ multiplies the opacity $\sigma(o_g)$ by the projected Gaussian footprint at pixel $\mathbf{x}$. Our method keeps geometry and higher-order spherical-harmonic coefficients shared across scales and retains the standard rasterization and density-control procedures. Fig. 2 provides an overview of ContinuousGS.

3.2 Continuous Scene-Scale Representation

Unified Scene Scale. For camera i, we estimate native projection density from the sparse structure-from-motion (SfM) reconstruction produced by COLMAP and used to initialize 3DGS. For cameras with sufficient observations, we use directly associated COLMAP points with positive camera-space depth (see § A.1 and Alg. 1 for the complete procedure, including validity checks and fallbacks for insufficient point support). Let $\mathcal{V}_i$, $z_{ip} > 0$, and $f_i = \sqrt{f_{x,i} f_{y,i}}$ denote the valid points, their camera-space depths, and the effective focal length in pixels, respectively. Using training cameras with sufficient directly associated points as the reference set $\mathcal{A}_{\mathrm{tr}}$, we define the view-level density ρ_i, the training-set reference ρ_{ref}, and the normalized native scale λ_i as:

$$\rho_i = \underset{p\in\mathcal{V}_i}{\mathrm{median}} \frac{\sqrt{f_{x,i} f_{y,i}}}{z_{ip}}, \quad \rho_{\mathrm{ref}} = \underset{i\in\mathcal{A}_{\mathrm{tr}}}{\mathrm{median}}\, \rho_i, \quad \lambda_i = \frac{\rho_i}{\rho_{\mathrm{ref}}}. \tag{1}$$

The geometric-mean focal length treats both image axes symmetrically. The median limits the effect of depth outliers, and normalization makes λ_i dimensionless and invariant to a global change of SfM units. For an image resize factor $s_i \in [s_{\min}, 1]$, we disallow upsampling and define the unified scene scale and the realizable interval of camera i as:

$$\tau = s_i \lambda_i, \quad s_i(\tau) = \frac{\tau}{\lambda_i}, \quad \mathcal{T}_i = [s_{\min}\lambda_i, \lambda_i]. \tag{2}$$

Equal values of τ therefore correspond approximately to equal scene-level sampling density across cameras, while the original 3DGS projection still accounts for local depth. We use $s_{\min} = 0.25$. This provides two octaves of scale coverage per camera while limiting downsampling to one quarter of the native width and height to preserve spatial detail. The value $\tau = 1$ corresponds to the reference projection density, whereas each camera reaches its native resolution at $\tau = \lambda_i$. § A.3.1 analyzes the resulting camera-scale distributions and view support across the four datasets.

Anchored Coarse-to-Fine Attributes. Changing the scene sampling density does not require a separate geometric reconstruction. At a coarser scale, however, each output pixel averages a larger scene region, which changes its mean color and the effective coverage of nearby Gaussians. We therefore keep the centers, covariances, higher-order spherical-harmonic coefficients, and base opacity logits shared across scales, and condition only SH_0 coefficients and effective opacity. An unconstrained scale-conditioned branch could absorb information from the base representation and alter its finest-scale prediction. We designate the finest supported scale as the canonical endpoint and require the summed scale-conditioned corrections to both appearance and opacity logits to vanish there. Let $\ell = \log_2 \tau$, with $[\ell_{\min}, \ell_{\max}]$ denoting the range covered by the training cameras. We map a query to $u(\tau) = \frac{\ell - \ell_{\min}}{\ell_{\max} - \ell_{\min}}$, so $u = 0$ and $u = 1$ denote the coarsest and finest supported scales. The use of log scale assigns equal coordinate length to equal octave intervals. Hard scale boundaries would make the rendered attributes discontinuous. We instead use the clipped cubic smoothstep $h(t) = \bar{t}^2(3 - 2\bar{t})$, where $\bar{t} = \mathrm{clip}(t, 0, 1)$. This function satisfies $h(0) = 0$, $h(1) = 1$, and $h'(0) = h'(1) = 0$, providing a continuous transition with zero slope at each boundary. We place coarse, middle, and fine control stages at $u = 0$, $u = 1/2$, and $u = 1$. Their cumulative gates are:

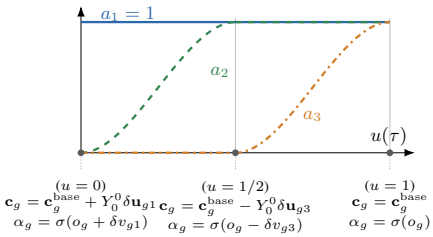


Figure 3: Continuous-scale activation.

$$a_1(\tau) = 1, \qquad a_2(\tau) = h(2u(\tau)), \qquad a_3(\tau) = h(2u(\tau) - 1). \tag{3}$$

The gates in Eq. (3) are nested rather than mutually exclusive. Fig. 3 plots a_1, a_2, and a_3 against the normalized log-scale coordinate $u(\tau)$. The first gate remains active, the second enters over the

coarse-to-middle interval, and the third enters over the middle-to-fine interval. The corresponding memberships are $m_1 = 1 - a_2$, $m_2 = a_2 - a_3$, and $m_3 = a_3$. The memberships are nonnegative and sum to one, so a query moves smoothly from coarse to middle and then to fine support without switching Gaussian sets. The term *band* refers to an interval of the scale coordinate rather than a spatial-frequency decomposition. For Gaussian g and band $b \in \{1, 2, 3\}$, let $\mathbf{u}_{gb} \in \mathbb{R}^3$ and $v_{gb} \in \mathbb{R}$ be raw residuals of the SH_0 coefficients and opacity logit. All three gates equal one at the finest endpoint, so preserving the base color and opacity requires the residuals to sum to zero across bands. We enforce this constraint by subtracting the mean across bands from each raw residual:

$$\delta\mathbf{u}_{gb} = \mathbf{u}_{gb} - \frac{1}{3}\sum_{b'=1}^{3} \mathbf{u}_{gb'}, \quad \delta v_{gb} = v_{gb} - \frac{1}{3}\sum_{b'=1}^{3} v_{gb'}. \tag{4}$$

We weight the centered residuals by the cumulative gates to obtain:

$$\mathbf{c}_g(\mathbf{v}, \tau) = \mathbf{c}_g^{\mathrm{base}}(\mathbf{v}) + Y_0^0 \sum_{b=1}^{3} a_b(\tau)\delta\mathbf{u}_{gb}, \quad \alpha_g(\tau) = \sigma\left(o_g + \sum_{b=1}^{3} a_b(\tau)\delta v_{gb}\right). \tag{5}$$

Here $\mathbf{c}_g^{\mathrm{base}}(\mathbf{v})$ is the original view-dependent color and o_g is the original opacity logit. Colors in Eq. (5) are defined before clamping, and $Y_0^0 = 1/\sqrt{4\pi}$ converts SH_0 coefficient residuals to RGB offsets. At the coarse stage, only the first residual is active. At the middle stage, the first two residuals are active. At the finest stage, all gates equal one and Eq. (4) makes their sum zero. The model therefore recovers $\mathbf{c}_g^{\mathrm{base}}(\mathbf{v})$ and $\sigma(o_g)$ exactly at $\ell_{\max}$. At lower scales, color residuals model scale-dependent averaging and opacity residuals model soft coverage changes. The parameterization stores twelve raw residual scalars per Gaussian without changing projection, sorting, or CUDA rasterization. The structural guarantee applies to the global endpoint $\ell_{\max}$. A native query for camera i occurs at $\tau = \lambda_i$ and need not coincide with that endpoint.

Evidence-Calibrated Residual Learning. We assess support for each residual using camera coverage and reconstruction-gradient agreement. For each Gaussian-band pair (g, b), n_{gb} counts distinct training cameras for which the rasterizer returns a positive projected radius for g and band membership satisfies $m_b > 0.25$. A compact bitset for each pair records camera identities so repeated observations do not increase the count. We maintain exponential moving averages (EMAs) $\bar{\mathbf{g}}_{gb}$ and q_{gb} of the reconstruction gradient and its squared norm. We detach these statistics from backpropagation and exclude regularization gradients from the reliability estimates. We combine count-based coverage C_{gb} with gradient agreement:

$$\eta_{gb} = C_{gb}\,\mathrm{clip}\left(\frac{\|\bar{\mathbf{g}}_{gb}\|_2^2}{q_{gb} + \epsilon}, 0, 1\right), \qquad C_{gb} = \begin{cases} 0, & n_{gb} = 0, \\ 1 - 1/n_{gb}, & n_{gb} \geq 1. \end{cases} \tag{6}$$

where ϵ prevents division by zero. The coverage weight is zero for a single observed camera and approaches one with diminishing gains as distinct cameras accumulate. For $n_{gb} \geq 1$, it equals the probability that two independent uniform draws from the observed camera set select different cameras, providing a count-based measure of cross-view support. The gradient agreement ratio decreases when opposing gradients cancel in $\bar{\mathbf{g}}_{gb}$ while contributing to q_{gb}, and clipping bounds the ratio in $[0, 1]$. Their product remains low when either coverage or agreement is weak. The weight $1 - \eta_{gb}$ relaxes quadratic shrinkage toward the base color:

$$\mathcal{L}_{\mathrm{reg}} = \frac{1}{3N}\sum_{g,b}\left[(1 - \eta_{gb})\,\frac{\|\delta\mathbf{u}_{gb}\|_2^2}{3} + (\delta v_{gb})^2\right]. \tag{7}$$

The factors $1/3$ and $1/(3N)$ average over color channels and Gaussian-band pairs, respectively. Opacity retains a fixed penalty because reliability is estimated from color-residual gradients. § 3.3 details the objective and updates.

Reliability-Aware Detail Allocation. A large centered color residual indicates that a Gaussian requires a substantial appearance correction in a particular scale band. We use this learned variation as a heuristic signal for allocating additional spatial capacity, even when the instantaneous reconstruction gradient is small. The root mean square averages the squared residual over color channels without cancellation between opposite signs and returns an amplitude in residual units. Multiplication by η_{gb} requires this amplitude to be supported by camera coverage and gradient agreement. For

Gaussian g, the maximum over bands prioritizes the strongest supported correction without diluting demand confined to one scale interval:

$$D_g = \max_b \left[\eta_{gb} \sqrt{\frac{\|\delta \mathbf{u}_{gb}\|_2^2}{3}} \right]. \tag{8}$$

The final selection is the union of the vanilla clone or split candidates and up to 1000 additional parent Gaussians below the vanilla gradient threshold, selected in descending order of D_g among candidates with $D_g > 0$. This evidence-guided ranking controls only the bounded candidate expansion and cannot suppress vanilla densification.

3.3 Training and Evaluation

Target-First Training. Sampling a camera before its resize factor makes the induced distribution of $\tau = s_i \lambda_i$ depend on the camera distribution. We define the feasible log-scale set of view i as $\mathcal{L}_i = \{\log_2 t | t \in \mathcal{T}_i\}$. We sample a target scale τ uniformly in log space, with $\ell = \log_2 \tau$ drawn from $\bigcup_i \mathcal{L}_i$, and construct $\mathcal{E}(\tau) = \{i | \tau \in \mathcal{T}_i\}$. The set $\mathcal{E}(\tau)$ contains the cameras that can realize the sampled scale without upsampling. We then draw i uniformly from $\mathcal{E}(\tau)$, resize $\mathbf{I}_i$ using $s_i(\tau) = \tau/\lambda_i$, and adjust $\mathbf{K}_i$ to the realized integer image dimensions. Downsampling uses PyTorch area interpolation. After rounding, we recompute the realized scene scale from the integer image dimensions and use it for conditioning, following the same rule as evaluation in Eq. (10). Fig. 4 illustrates this selection. Target-first sampling gives equal probability to equal log-scale intervals instead of assigning more weight to scales covered by many cameras. For each sampled camera-scale pair, we render $\widehat{\mathbf{I}}$ at the target scale and jointly minimize reconstruction error and residual regularization:

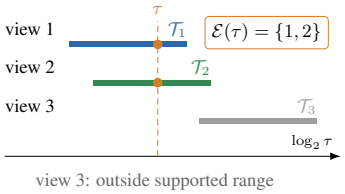


Figure 4: Target-first sampling.

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + 10^{-4} \mathcal{L}_{\text{reg}}, \qquad \mathcal{L}_{\text{rec}} = 0.8 \|\widehat{\mathbf{I}} - \mathbf{I}\|_1 + 0.2 \mathcal{L}_{\text{D-SSIM}}, \tag{9}$$

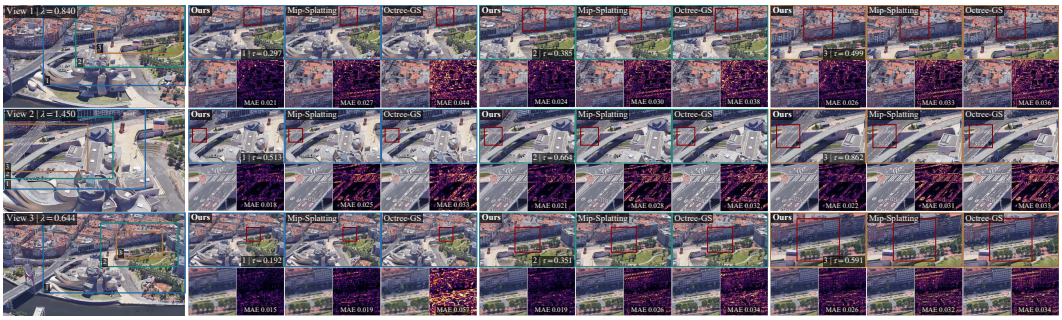
where $\mathcal{L}_{\text{reg}}$ is defined in Eq. (7), and $\mathcal{L}_{\text{rec}}$ combines an ℓ_1 loss with structural dissimilarity (D-SSIM). To update the evidence statistics, we use $\mathbf{g}_{gb} = \partial \mathcal{L}_{\text{rec}} / \partial \mathbf{u}_{gb}$ for each residual $\mathbf{u}_{gb}$. Gradients from residual or opacity regularization are excluded from these updates, and the accumulators remain detached from the optimization graph. At each densification event, we expand the vanilla candidate set as described in § 3.2, then apply the vanilla clone and split operations. New Gaussians inherit canonical attributes and the per-band parameters of the scale-conditioned residuals but reset distinct-view counts and gradient EMAs to zero, so inherited parameters serve only as initialization.

Continuous-Scale Evaluation. For each test camera, a frozen method-independent manifest requests 17 log-uniform targets within $\mathcal{T}_i$, providing a dense discretization of the supported log-scale interval for approximating continuous evaluation. Since integer output dimensions can change the requested resize factor, we record the realized scale of each test node. For native image dimensions (H_i, W_i) and realized dimensions (H_{ij}, W_{ij}) at node j, the realized coordinates are:

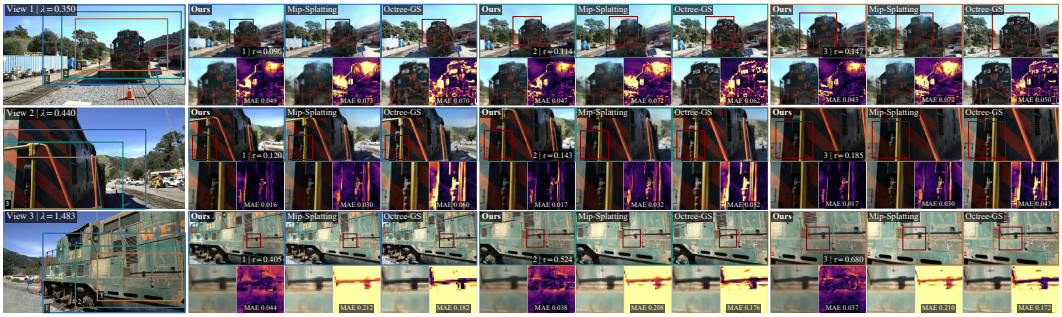
$$s_{ij}^{\text{eff}} = \sqrt{\frac{H_{ij} W_{ij}}{H_i W_i}}, \quad \tau_{ij}^{\text{eff}} = \lambda_i s_{ij}^{\text{eff}}, \quad \ell_{ij} = \log_2 \tau_{ij}^{\text{eff}}. \tag{10}$$

Both conditioning and evaluation use τ_{ij}^{eff} so that each metric value is associated with the scale actually rendered. The manifest merges duplicate realized sizes and records $\Delta_{\max}$, *i.e.*, the largest gap between adjacent realized ℓ_{ij} values. Integer image dimensions introduce a relative deviation between the requested and realized resize factors, which remains below 1% at every evaluated node. At each retained node, we measure Y_{ij}^M, where M denotes PSNR, SSIM (Wang et al., 2004), or 1-LPIPS (Zhang et al., 2018). Using 1-LPIPS makes larger values indicate better quality for every metric. We linearly interpolate between adjacent valid nodes within each supported interval, without extrapolation or interpolation across unsupported gaps. The union of these interval boundaries partitions the log-scale axis into atomic cells $B_r = [\ell_r, \ell_{r+1}]$ of width $\Delta\ell_r = \ell_{r+1} - \ell_r$. For cell r, N_r counts the views supporting the entire cell, and V_r^M is the interpolated metric averaged over the cell and those views. The full-range score, the score restricted to cells supported by at least five views, and the retained scale fraction are:

$$S_{\text{Full}}^M = \frac{\sum_{r:N_r>0} \Delta\ell_r V_r^M}{\sum_{r:N_r>0} \Delta\ell_r}, \quad S_{(5)}^M = \frac{\sum_{r:N_r\geq 5} \Delta\ell_r V_r^M}{\sum_{r:N_r\geq 5} \Delta\ell_r}, \quad Q_5 = \frac{\sum_{r:N_r\geq 5} \Delta\ell_r}{\sum_{r:N_r>0} \Delta\ell_r}. \tag{11}$$



(a) Scene *Bilbao* from BungeeNeRF (Xiangli et al., 2022).



(b) Scene *Train* from Tanks and Temples (Knapitsch et al., 2017).

Figure 5: Qualitative comparisons at three selected unified scales for three views per scene. Numbered frames on the native images identify the regions shown in the corresponding scale groups. Local panels show enlarged predictions and absolute-error maps relative to the scale-matched ground truth (GT), with local mean absolute error (MAE) annotated.

Weighting by $\Delta\ell_r$ gives equal influence to equal log-scale lengths, while N_r exposes the available cross-view support. We construct the atomic cells and compute these integrals independently for each scene. § A.2 derives the local interpolants and cell averages V_r^M from the realized sampling nodes and explains the scale-length aggregation and view-support criterion. Fig. 6 illustrates the atomic cells and their view support.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate 21 scenes from Tanks and Temples (Knapitsch et al., 2017), Deep Blending (Hedman et al., 2018), Mip-NeRF 360 (Barron et al., 2022), and BungeeNeRF (Xiangli et al., 2022) with the standard train-test splits and canonical input resolutions used by 3DGS (Kerbl et al., 2023). We use input resolution factor 1 for Tanks and Temples, Deep Blending, and BungeeNeRF, and factors 2 and 4 for indoor and outdoor scenes in Mip-NeRF 360, respectively. Following § 3.3, table columns labeled PSNR, SSIM, and 1-LPIPS report native-resolution results, while S^M_{Full} and $S^M_{(5)}$ report full-range and high-support continuous-scale scores, respectively, with M replaced by the corresponding metric name. Dataset-level results are arithmetic means of the per-scene scores, giving each scene equal weight regardless of its view count or scale coverage.

Training. We train ContinuousGS and all ablation variants from the same 3DGS initialization for 30,000 iterations. All ablation variants share the same split, initialization, training budget, and evaluation manifest. Both residual tensors use a learning rate of 2.5×10^{-3}, and the gradient evidence statistics use an EMA decay of 0.95. We compare against 3DGS and representative scale-aware methods, including Mip-Splatting, CLoD-GS, Octree-GS, MSGS, and Mipmap-GS. We retrain these methods with their released implementations and original training strategies, and evaluate all checkpoints with the same frozen manifest. All training runs use a single NVIDIA RTX 5090 GPU.

Table 1: Quantitative comparison. Dataset statistics and Q_5 are provided in Tab. 3. Bold marks the best result for each metric within each dataset, including ties at the reported precision.

Tanks and Temples	PSNR↑	SSIM↑	1-LPIPS↑	$S^{\text{PSNR}}_{\text{Full}}$ ↑	$S^{\text{SSIM}}_{\text{Full}}$ ↑	$S^{\text{1-LPIPS}}_{\text{Full}}$ ↑	$S^{\text{PSNR}}_{(5)}$ ↑	$S^{\text{SSIM}}_{(5)}$ ↑	$S^{\text{1-LPIPS}}_{(5)}$ ↑
3DGS (Kerbl et al., 2023)	23.747	0.853	0.876	21.669	0.819	0.893	22.064	0.833	0.898
Mip-Splatting (Yu et al., 2024)	23.423	0.842	0.864	23.152	0.850	0.903	23.354	0.858	0.906
CLoD-GS (Cheng et al., 2026)	23.671	0.844	0.871	22.141	0.823	0.883	22.621	0.832	0.894
Octree-GS (Ren et al., 2025)	24.448	**0.863**	0.881	22.178	0.826	0.886	22.678	0.844	0.898
MSGS (Yan et al., 2024)	23.718	0.844	0.861	22.082	0.821	0.880	22.325	0.830	0.886
Mipmap-GS (Li et al., 2025)	23.726	0.848	0.870	21.884	0.815	0.891	22.228	0.827	0.895
ContinuousGS	**24.536**	0.861	**0.882**	**23.343**	**0.852**	**0.915**	**23.519**	**0.869**	**0.926**
Deep Blending	PSNR↑	SSIM↑	1-LPIPS↑	$S^{\text{PSNR}}_{\text{Full}}$ ↑	$S^{\text{SSIM}}_{\text{Full}}$ ↑	$S^{\text{1-LPIPS}}_{\text{Full}}$ ↑	$S^{\text{PSNR}}_{(5)}$ ↑	$S^{\text{SSIM}}_{(5)}$ ↑	$S^{\text{1-LPIPS}}_{(5)}$ ↑
3DGS (Kerbl et al., 2023)	29.785	**0.908**	0.837	28.605	0.901	0.894	29.336	0.914	0.906
Mip-Splatting (Yu et al., 2024)	29.297	0.903	0.830	28.990	0.908	0.895	29.735	0.918	0.906
CLoD-GS (Cheng et al., 2026)	29.931	0.904	0.821	28.905	0.896	0.885	29.625	0.910	0.901
Octree-GS (Ren et al., 2025)	29.974	0.903	0.823	28.912	0.903	0.887	29.667	0.916	0.903
MSGS (Yan et al., 2024)	29.461	0.903	0.826	28.739	0.902	0.887	29.416	0.913	0.899
Mipmap-GS (Li et al., 2025)	29.561	0.904	0.833	28.340	0.893	0.890	29.117	0.907	0.901
ContinuousGS	**29.991**	0.905	**0.840**	**29.161**	**0.915**	**0.900**	**29.965**	**0.926**	**0.913**
Mip-NeRF 360	PSNR↑	SSIM↑	1-LPIPS↑	$S^{\text{PSNR}}_{\text{Full}}$ ↑	$S^{\text{SSIM}}_{\text{Full}}$ ↑	$S^{\text{1-LPIPS}}_{\text{Full}}$ ↑	$S^{\text{PSNR}}_{(5)}$ ↑	$S^{\text{SSIM}}_{(5)}$ ↑	$S^{\text{1-LPIPS}}_{(5)}$ ↑
3DGS (Kerbl et al., 2023)	27.773	0.826	0.831	25.193	0.804	0.857	25.353	0.812	0.863
Mip-Splatting (Yu et al., 2024)	27.770	**0.829**	0.847	27.192	0.849	0.894	27.996	0.864	0.908
CLoD-GS (Cheng et al., 2026)	27.752	0.814	0.826	25.273	0.805	0.859	25.466	0.811	0.862
Octree-GS (Ren et al., 2025)	**27.845**	0.826	0.837	25.449	0.811	0.864	25.758	0.817	0.871
MSGS (Yan et al., 2024)	27.562	0.818	0.818	26.428	0.816	0.857	26.622	0.824	0.864
Mipmap-GS (Li et al., 2025)	27.703	0.826	0.832	25.494	0.803	0.860	25.660	0.811	0.866
ContinuousGS	27.837	**0.829**	**0.851**	**27.539**	**0.868**	**0.911**	**28.219**	**0.873**	**0.919**
BungeeNeRF	PSNR↑	SSIM↑	1-LPIPS↑	$S^{\text{PSNR}}_{\text{Full}}$ ↑	$S^{\text{SSIM}}_{\text{Full}}$ ↑	$S^{\text{1-LPIPS}}_{\text{Full}}$ ↑	$S^{\text{PSNR}}_{(5)}$ ↑	$S^{\text{SSIM}}_{(5)}$ ↑	$S^{\text{1-LPIPS}}_{(5)}$ ↑
3DGS (Kerbl et al., 2023)	26.752	0.896	0.908	25.315	0.849	0.905	25.352	0.870	0.916
Mip-Splatting (Yu et al., 2024)	26.894	0.895	0.898	26.892	0.883	0.921	27.036	0.891	0.926
CLoD-GS (Cheng et al., 2026)	26.931	0.898	0.911	25.572	0.858	0.900	25.602	0.869	0.911
Octree-GS (Ren et al., 2025)	27.085	0.901	0.918	25.588	0.868	0.908	25.729	0.871	0.913
MSGS (Yan et al., 2024)	26.456	0.889	0.897	25.505	0.857	0.902	25.594	0.869	0.913
Mipmap-GS (Li et al., 2025)	26.587	0.894	0.905	25.364	0.846	0.905	25.480	0.866	0.915
ContinuousGS	**27.104**	**0.904**	**0.922**	**27.030**	**0.887**	**0.927**	**27.156**	**0.895**	**0.932**

4.2 Main Comparison

Quantitative Comparison. ContinuousGS achieves the highest full-range scores for all three metrics in Tab. 1. Mip-Splatting is the strongest baseline throughout this setting. ContinuousGS retains the best or tied-best scores when evaluation is restricted to intervals supported by at least five views. Although the margins vary across datasets, the gains over Mip-Splatting extend to scales with broader view support. At native resolution, ContinuousGS largely preserves the reconstruction quality of 3DGS while providing modest improvements across all four datasets. These results demonstrate that ContinuousGS extends 3DGS to continuous scene-scale rendering while retaining its native-resolution reconstruction quality. The full-range and high-support gains support learning a shared representation that adapts to scene sampling density.

Qualitative Comparison. Fig. 5 compares three sampled scene scales for each of multiple views to examine reconstruction quality as sampling density changes. Within each view, the crops follow the same scene region across scales, allowing direct inspection of how local structures appear at different output resolutions. The building and road-marking crops from *Bilbao* show lower reconstruction error for ContinuousGS across the displayed scales. In *Train*, the locomotive front and the narrow handrail provide further examples of detail and boundary reconstruction across scales. The displayed baselines exhibit stronger error responses along these edges and within the painted surfaces. The appendix comparisons in Figs. 7 to 9 extend this inspection to six additional scenes, illustrating detail reconstruction across sampled scales in urban structures, vegetation, and bicycle spokes. These examples illustrate how ContinuousGS adapts local appearance to changes in sampling density across different views. For each view-scale pair, all methods use identical output dimensions and crop coordinates to ensure comparison on the same sampling grid.

4.3 Ablation Study

We conduct ablation studies on Mip-NeRF 360 to assess how the proposed components contribute to reconstruction quality across scales, as summarized in Tab. 2. Our ablation baseline (Cfg. 0) uses the original 3DGS representation with target-first multi-scale training. Cfg. 0.1 replaces this training protocol with native-resolution training, matching the original 3DGS baseline in Tab. 1. § A.3 pro-

Table 2: Component ablations on Mip-NeRF 360.

Cfg.	Variant	PSNR↑	SSIM↑	1-LPIPS↑	$S^{\mathrm{PSNR}}_{\mathrm{Full}}\uparrow$	$S^{\mathrm{SSIM}}_{\mathrm{Full}}\uparrow$	$S^{\text{1-LPIPS}}_{\mathrm{Full}}\uparrow$	$S^{\mathrm{PSNR}}_{(5)}\uparrow$	$S^{\mathrm{SSIM}}_{(5)}\uparrow$	$S^{\text{1-LPIPS}}_{(5)}\uparrow$
0	3DGS + Target-first Training	27.424	0.810	0.804	26.008	0.825	0.866	26.170	0.832	0.871
0.1	Native-only Training	27.773	0.826	0.831	25.193	0.804	0.857	25.353	0.812	0.863
1	+Joint Residuals	27.786	0.828	0.840	27.072	0.851	0.888	27.347	0.856	0.891
1.1	Appearance Only	27.784	0.827	0.833	26.364	0.836	0.879	26.569	0.841	0.882
1.2	Opacity Only	27.768	0.826	0.841	26.395	0.833	0.882	26.529	0.843	0.885
1.3	w/o Anchoring	27.756	0.824	0.839	26.625	0.844	0.885	27.087	0.847	0.886
1.4	Camera-first	27.810	0.829	0.844	26.812	0.840	0.884	26.942	0.850	0.889
1.5	$\tau \to s$	27.679	0.819	0.824	26.610	0.845	0.878	26.618	0.848	0.886
2	+Evidence Calibration	27.814	0.828	0.843	27.347	0.857	0.905	28.112	0.864	0.907
2.1	Coverage only	27.804	0.824	0.841	27.210	0.851	0.893	27.878	0.855	0.894
2.2	Agreement only	27.808	0.824	0.840	27.304	0.855	0.899	27.972	0.858	0.901
3	+Detail Allocation (Ours)	27.837	0.829	0.851	27.539	0.868	0.911	28.219	0.873	0.919

vides complementary experiments and analyses covering dataset scale distributions, representation cost and efficiency, the extension of target-first training to Mip-Splatting, and additional qualitative comparisons.

Scale-Conditioned Attributes. Appearance-only (Cfg. 1.1) and opacity-only (Cfg. 1.2) variants each add one anchored residual branch to the multi-scale baseline (Cfg. 0). Cfg. 1 and its controls retain fixed quadratic penalties on their active residual branches, using $\eta_{gb} = 0$ in Eq. (7). Both branches recover native-resolution quality and improve continuous-scale scores. The two branches offer complementary benefits, with neither outperforming the other across settings. Combining them in Cfg. 1 improves all six continuous-scale metrics over either branch alone.

Scale Coordinate, Anchoring, and Sampling. Anchoring in Cfg. 1 improves all nine metrics over the unanchored control (Cfg. 1.3). The camera-first control (Cfg. 1.4) samples a camera uniformly, then samples τ log-uniformly within its supported interval, retaining τ as the model's scale coordinate. Cfg. 1.5 instead samples s log-uniformly from $[0.25, 1]$, selects a camera uniformly, and uses s inside the model. These controls share the same camera-resize sampling distribution, so their comparison isolates the model's scale coordinate. Using τ gives higher scores in eight of nine metrics, with $S^{\mathrm{SSIM}}_{\mathrm{Full}}$ favoring s. With τ conditioning fixed, target-first sampling (Cfg. 1) improves both S^{M}_{Full} and $S^{M}_{(5)}$ over camera-first sampling for all three metrics. These comparisons support scene-scale conditioning and a more balanced allocation of supervision along the scale axis.

Evidence and Allocation. Evidence calibration (Cfg. 2) weights the fixed appearance penalty with $1-\eta_{gb}$, while retaining the fixed opacity penalty. Coverage only (Cfg. 2.1) uses $\eta_{gb} = C_{gb}$, whereas Agreement only (Cfg. 2.2) uses the clipped gradient agreement ratio in Eq. (6). Both improve $S^{\mathrm{PSNR}}_{\mathrm{Full}}$, $S^{\mathrm{PSNR}}_{(5)}$, $S^{\text{1-LPIPS}}_{\mathrm{Full}}$, and $S^{\text{1-LPIPS}}_{(5)}$ over fixed regularization. Combining the two factors gives higher scores than either alone across all nine metrics, supporting their complementary roles in calibrating residual freedom. Adding detail allocation (Cfg. 3) further improves all metrics, with gains of 0.192 in $S^{\mathrm{PSNR}}_{\mathrm{Full}}$ and 0.107 in $S^{\mathrm{PSNR}}_{(5)}$ over Cfg. 2. These gains support expanding densification candidates using evidence-weighted residual demand.

5 Conclusion

We presented ContinuousGS, a framework for continuous scene-scale rendering with a single 3DGS representation. A unified scene-scale coordinate approximately aligns sampling density across cameras, providing a common basis for supervision, attribute adaptation, and evaluation. Target-first sampling and anchored appearance and opacity residuals enable learning across this shared scale range, while multi-view evidence guides residual learning and detail allocation. Experiments on four datasets show improved continuous-scale rendering quality over representative methods while preserving native-resolution quality, with gains in both full-range and high-support evaluations. Ablation studies support the complementary roles of appearance and opacity adaptation and the contribution of scene-scale conditioning, anchoring, and evidence-guided refinement to balanced performance across scales. These findings support organizing the learning and evaluation of scale-adaptive scene representations around a shared scene-scale coordinate.

AI USE STATEMENT

In this work, we used generative AI tools to polish the language, check typographical and grammatical errors, and adjust LaTeX formatting. We have reviewed all AI-assisted work, including the language revisions and LaTeX formatting changes. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

§ 3 and 4 describe the method, training settings, datasets, and evaluation protocol. § A.1 specifies the projection-density estimator and its fallback procedure, and § A.2 derives the continuous-scale evaluation scores.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *IEEE International Conference on Computer Vision*, pp. 5835–5844, 2021. doi: 10.1109/ICCV48922.2021.00580. 3

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5460–5469, 2022. doi: 10.1109/CVPR52688.2022.00539. 7, 19

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *IEEE International Conference on Computer Vision*, pp. 19640–19648, 2023. doi: 10.1109/ICCV51070.2023.01804. 3

Jinxiang Chai, Shing-Chow Chan, Heung-Yeung Shum, and Xin Tong. Plenoptic sampling. In *ACM SIGGRAPH*, pp. 307–318, 2000. doi: 10.1145/344779.344932. 3

Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *ACM Computing Surveys*, 58 (12), May 2026. ISSN 0360-0300. doi: 10.1145/3807511. 2, 3

Zhigang Cheng, Mingchao Sun, Yu Liu, Zengye Ge, Luyang Tang, Mu Xu, Yangyan Li, and Peng Pan. CLod-GS: Continuous level-of-detail via 3d gaussian splatting. In *International Conference on Learning Representations*, 2026. 3, 8

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel J. Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics*, 37(6):257, 2018. doi: 10.1145/3272127.3275084. 7

Hugues Hoppe. Progressive meshes. In *ACM SIGGRAPH*, pp. 99–108, 1996. 3

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

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. 1, 2, 3, 7, 8

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 *International Conference on Neural Information Processing Systems*, 2024. 2

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):78:1–78:13, 2017. doi: 10.1145/3072959.3073599. 7

Jonas Kulhanek, Marie-Julie Rakotosaona, Fabian Manhardt, Christina Tsalicoglou, Michael Niemeyer, Torsten Sattler, Songyou Peng, and Federico Tombari. LODGE: level-of-detail large-scale gaussian splatting with efficient rendering. In *International Conference on Neural Information Processing Systems*, 2025. 3

Jiameng Li, Yue Shi, Jiezhang Cao, Bingbing Ni, Wenjun Zhang, Kai Zhang, and Luc Van Gool. Mipmap-gs: Let gaussians deform with scale-specific mipmap for anti-aliasing rendering. In *International Conference on 3D Vision*, pp. 935–945, 2025. doi: 10.1109/3DV66043.2025.00091. 3, 8

Zhihao Liang, Qi Zhang, Wenbo Hu, Lei Zhu, Ying Feng, and Kui Jia. Analytic-splatting: Anti-aliased 3d gaussian splatting via analytic integration. In *European Conference on Computer Vision*, volume 15075, pp. 281–297, 2024. doi: 10.1007/978-3-031-72643-9_17. 3

Yun Liao, Yide Di, Hao Zhou, Kaijun Zhu, Mingyu Lu, Qing Duan, and Junhui Liu. A survey on neural radiance fields. *ACM Computing Surveys*, 58(2), September 2025. ISSN 0360-0300. doi: 10.1145/3758085. 2

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 *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2024. doi: 10.1109/CVPR52733.2024.01952. 2

Linjie Lyu, Ayush Tewari, Jianchun Chen, Thomas Leimkühler, and Christian Theobalt. Faster 3d gaussian splatting convergence via structure-aware densification. In *ACM SIGGRAPH*, pp. 24:1–24:10, 2026. doi: 10.1145/3799902.3811212. 2

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. *Communications of the ACM*, 65(1):99–106, 2022. doi: 10.1145/3503250. 2

Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–15, 2025. ISSN 1939-3539. doi: 10.1109/TPAMI.2025.3568201. 1, 3, 8

Xiaowei Song, Jv Zheng, Shiran Yuan, Huan-ang Gao, Jingwei Zhao, Xiang He, Weihao Gu, and Hao Zhao. SA-GS: scale-adaptive gaussian splatting for training-free anti-aliasing. *CoRR*, abs/2403.19615, 2024. doi: 10.48550/ARXIV.2403.19615. 3

Michael Steiner, Thomas Köhler, Lukas Radl, Felix Windisch, Dieter Schmalstieg, and Markus Steinberger. Aaa-gaussians: Anti-aliased and artifact-free 3d gaussian rendering. In *IEEE International Conference on Computer Vision*, pp. 27650–27659, 2025. doi: 10.1109/ICCV51701.2025.02567. 3

Xuechang Tu, Lukas Radl, Michael Steiner, Markus Steinberger, Bernhard Kerbl, and Fernando De la Torre. Vrsplat: Fast and robust gaussian splatting for virtual reality. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 8(1):1:1–1:22, 2025. doi: 10.1145/3728311. 3

Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3:257–295, 2016. doi: 10.1146/annurev-statistics-041715-033624. 15

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. 6

Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European Conference on Computer Vision*, volume 13692, pp. 106–122, 2022. doi: 10.1007/978-3-031-19824-3_7. 3, 7, 19, 20

Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3d gaussian splatting for anti-aliased rendering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 20923–20931, 2024. doi: 10.1109/CVPR52733.2024.01977. 3, 8

Zhenya Yang, Bingchen Gong, and Kai Chen. LOD-GS: level-of-detail-sensitive 3d gaussian splatting for detail conserved anti-aliasing. *CoRR*, abs/2507.00554, 2025. doi: 10.48550/ARXIV.2507.00554. 3

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 19447–19456, 2024. doi: 10.1109/CVPR52733.2024.01839. 1, 3, 8, 18

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068. 6

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*, volume 15077, pp. 326–342, 2024. doi: 10.1007/978-3-031-72655-2_19. 2

Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus H. Gross. EWA splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. doi: 10.1109/TVCG.2002.1021576. 3

A Appendix

A.1 Projection-Density Estimation 13
A.2 A Functional-Data View of the Evaluation Protocol 15
A.2.1 Shared Curves from Sparse and Irregular Observations 15
A.2.2 Global Atomic-Cell Construction 16
A.2.3 Scale-Length Aggregation and View Support 17
A.3 Additional Experiments 17
A.3.1 Dataset Scale Distributions 17
A.3.2 Representation Cost and Efficiency 17
A.3.3 Extending Target-First Training to Mip-Splatting 18
A.3.4 Additional Qualitative Comparisons 18
A.4 Limitations and Future Directions 18

A.1 Projection-Density Estimation

Notation and Scale Coordinate. We follow the definition of unified scene scale in § 3.2 and restate the notation needed for the estimator below. For each training or test camera i, let W_i and H_i be its native image width and height after dataset resolution normalization, and let $f_{x,i}$ and $f_{y,i}$ be the corresponding focal lengths in pixels. The world-to-camera transform $\mathbf{W}_i$ maps a scene point p to camera-space coordinates (x_{ip}, y_{ip}, z_{ip}), where z_{ip} is its depth. The effective focal length is $f_i = \sqrt{f_{x,i} f_{y,i}}$, so f_i/z_{ip} measures projected sampling density in pixels per unit of scene length for positive-depth points. We denote the valid point set selected for camera i by $\mathcal{V}_i$ and its estimated native projection density by ρ_i. The reference set $\mathcal{A}_{\text{tr}}$ contains training cameras with sufficient valid directly associated points (Level 1 below). Its median density $\rho_{\text{ref}} = \text{median}_{i \in \mathcal{A}_{\text{tr}}} \rho_i$ defines the dimensionless native scale $\lambda_i = \rho_i / \rho_{\text{ref}}$. For an image resize factor $s_i \in [0.25, 1]$ applied to both native image dimensions, the unified scene scale is $\tau = s_i \lambda_i$, and camera i supports the interval $\mathcal{T}_i = [0.25\lambda_i, \lambda_i]$. Thus $\tau = \lambda_i$ denotes the native resolution of camera i, while $\tau = 1$ denotes the reference projection density.

Inputs and Point Support. Alg. 1 estimates density from calibrated cameras and their associated COLMAP point sets $\mathcal{P}_i$. We use the full-scene COLMAP reconstruction supplied with the benchmarks or obtained through standard scene calibration, including test-image SfM associations. Test RGB images are not used to optimize the representation, and test-view λ_i values only specify evaluation scales under a frozen manifest shared by all compared methods. Only training views determine ρ_{ref}. The pooled set $\mathcal{P}_{\text{tr}}$ contains finite points associated with at least one training camera, deduplicated by point ID. Level 1 uses directly associated points with finite camera-space coordinates and positive depth. If these provide insufficient support, Level 2 selects points from $\mathcal{P}_{\text{tr}}$ that also project inside the native image, without considering occlusion. For pinhole intrinsics $\mathbf{K}_i$ with principal point $(c_{x,i}, c_{y,i})$, the projected pixel coordinates are $(f_{x,i} x_{ip}/z_{ip} + c_{x,i}, f_{y,i} y_{ip}/z_{ip} + c_{y,i})$. The current implementation fixes $c_{x,i} = W_i/2$ and $c_{y,i} = H_i/2$, consistently with the centered projection used by the renderer, rather than retaining principal-point offsets from COLMAP. This is a camera-model assumption, not a consequence of image undistortion or resolution normalization. The same convention applies to the extended-image-bound check in Level 3. Principal-point offsets do not enter f_i/z_{ip} directly, but can change point selection in Level 2 and center acceptance in Level 3, and hence the fallback estimate. Both levels estimate ρ_i as the median of f_i/z_{ip} over the selected points $\mathcal{V}_i$.

Center Fallback. When neither point set provides sufficient support, Level 3 estimates projection density using a robust center $\mathbf{c}$ of the training-associated point cloud $\mathcal{P}_{\text{tr}}$ in world coordinates. We obtain $\mathbf{c}$ by radially trimming distant points around the initial geometric median $\mathbf{c}_0$ and recomputing the geometric median of the retained points. The geometric median minimizes the sum of Euclidean distances to the points. For camera i, $z_{i\mathbf{c}}$ is the depth of $\mathbf{c}$ after applying the world-to-camera transform $\mathbf{W}_i$, and the fallback density is $\rho_i = f_i/z_{i\mathbf{c}}$. The center must satisfy the minimum-depth

Algorithm 1 Unified scene-scale estimation with three levels of point support

Require: Nonempty training and test camera sets with valid calibration and matching COLMAP records $\mathcal{P}_i$.
Require: Q_q: empirical q-quantile, ϵ_{mach}: double-precision machine epsilon.
Ensure: Finite positive ρ_i, ρ_{ref}, and λ_i for all cameras, or scene-level failure.

$\mathcal{P}_{\text{tr}} \leftarrow$ finite training-associated COLMAP points, deduplicated by point ID
if $|\mathcal{P}_{\text{tr}}| < 128$ **then**
  return scene-level failure
end if
$\mathcal{A}_{\text{tr}} \leftarrow \varnothing$
for each training or test camera i **do**
  // Treat both image axes symmetrically
  $f_i \leftarrow \sqrt{f_{x,i} f_{y,i}}$
  // Level 1
  $\mathcal{V}_i \leftarrow \{p \in \mathcal{P}_i : \text{camera-space coordinates are finite},\ z_{ip} > 0\}$
  if $|\mathcal{V}_i| \geq 32$ **then**
    Add i to $\mathcal{A}_{\text{tr}}$ if i is a training camera
  else
    // Level 2
    $\mathcal{V}_i \leftarrow \{p \in \mathcal{P}_{\text{tr}} : \text{camera-space coordinates are finite},\ z_{ip} > 0\}$
    Retain only points in $\mathcal{V}_i$ that project inside the native image
    // Level 3: training-point center
    if $|\mathcal{V}_i| < 32$ **then**
      if this is the first camera requiring Level 3 **then**
        // Compute the initial geometric-median center
        $\mathbf{c}_0 \leftarrow \text{GeoMedian}(\mathcal{P}_{\text{tr}})$
        // Recompute the center within the 95th radial percentile
        $\mathbf{c} \leftarrow \text{GeoMedian}(\{p \in \mathcal{P}_{\text{tr}} : \|p - \mathbf{c}_0\| \leq Q_{0.95}(\{\|p - \mathbf{c}_0\| : p \in \mathcal{P}_{\text{tr}}\})\})$
        // Guard near-zero depth
        $\epsilon_{\text{depth}} \leftarrow \max(10^{-8} \text{median}_{p \in \mathcal{P}_{\text{tr}}} \|p - \mathbf{c}\|, \epsilon_{\text{mach}})$
      end if
      if $z_{i\mathbf{c}} > \epsilon_{\text{depth}}$ and $\mathbf{c}$ projects inside $[-0.25W_i, 1.25W_i) \times [-0.25H_i, 1.25H_i)$ **then**
        $\rho_i \leftarrow f_i / z_{i\mathbf{c}}$
        // Center-based density is already assigned
        continue to the next camera
      else
        return scene-level failure
      end if
    end if
  end if
  // Robust projected sampling density
  $\rho_i \leftarrow \text{median}_{p \in \mathcal{V}_i}(f_i / z_{ip})$
end for
if $|\mathcal{A}_{\text{tr}}| < 3$ **then**
  return scene-level failure
end if
// Level 1 training views only
$\rho_{\text{ref}} \leftarrow \text{median}_{i \in \mathcal{A}_{\text{tr}}} \rho_i$
$\lambda_i \leftarrow \rho_i / \rho_{\text{ref}}$ for every training and test camera i
if any ρ_i, ρ_{ref}, or λ_i is nonfinite or nonpositive **then**
  return scene-level failure
end if
return $\{\rho_i, \lambda_i\}_i$ and ρ_{ref}

and extended-image-bound conditions in Alg. 1. The center and depth threshold are computed from training-associated points and shared across cameras.

Density and Normalization. Only Level 1 training cameras determine ρ_{ref}, so fallback estimates and test cameras do not affect the reference density. All training and test cameras are normalized by this same reference to obtain λ_i. Scale construction fails for the entire scene if any camera lacks a valid estimate or the training reference has insufficient support, as specified in Alg. 1. No scale-construction failures occurred in any of the scenes evaluated in this work.

Potential Fallback Failures. The global center may nevertheless provide no valid fallback for cameras with insufficient local point support. Its projection may lie outside the extended image bounds or have nonpositive depth, causing scene-level termination under the current procedure. The fallback hierarchy could be extended with local interpolation of projection densities from nearby cameras with reliable geometric support. Such estimates would retain the same training reference for normalization. If reliable estimates remain unavailable, the same affected test cameras could be excluded from evaluation across all methods, with the retained coverage explicitly reported.

A.2 A Functional-Data View of the Evaluation Protocol

A.2.1 Shared Curves from Sparse and Irregular Observations

For a metric M, let $F_i^M(\ell)$ denote the underlying performance curve of test view i over log scene scale $\ell = \log_2 \tau$. The evaluation observes this trajectory only at the realized nodes ℓ_{ij}:

$$Y_{ij}^M = F_i^M(\ell_{ij}), \tag{12}$$

where integer output sizes produce irregular nodes. The native scales λ_i also give the views different support intervals. This observation structure resembles sparse functional data (Wang et al., 2016), in which each subject contributes a small set of irregularly spaced measurements from an underlying curve. The analogy concerns the observation structure. Our metric values are deterministic outputs of a fixed rendering protocol. We do not introduce a stochastic observation model or estimate covariance across a population of cameras.

The scalar Y_{ij}^M is the measured metric value at node ℓ_{ij}. We construct $\widehat{y}_i^M(\ell)$ from these measurements to integrate performance along the scale axis. After merging duplicate realized nodes, we order the nodes of each view by increasing ℓ. For an admissible pair of adjacent nodes, we define the interpolant on $\ell \in [\ell_{ij}, \ell_{i,j+1}]$ as:

$$\widehat{y}_i^M(\ell) = \frac{\ell_{i,j+1} - \ell}{\ell_{i,j+1} - \ell_{ij}} Y_{ij}^M + \frac{\ell - \ell_{ij}}{\ell_{i,j+1} - \ell_{ij}} Y_{i,j+1}^M. \tag{13}$$

The weights in Eq. (13) are nonnegative and sum to one. The interpolant reproduces the endpoint measurements, satisfying $\widehat{y}_i^M(\ell_{ij}) = Y_{ij}^M$ and $\widehat{y}_i^M(\ell_{i,j+1}) = Y_{i,j+1}^M$. Between these endpoints, the interpolated value approximates $F_i^M(\ell)$ under a local linearity assumption. We organize the evaluation in a shared manifest, a record of each view's realized sampling nodes and admissible interpolation intervals that we fix before comparing methods. Let $\mathcal{J}_i$ index all consecutive pairs $(j, j+1)$ of the ordered realized nodes. All 17 requested targets must be valid, with none discarded. Duplicate realized image sizes are then merged before constructing the interpolant. The interpolation domain is $\mathcal{D}_i = \bigcup_{j \in \mathcal{J}_i} [\ell_{ij}, \ell_{i,j+1}]$. We leave $\widehat{y}_i^M$ undefined outside $\mathcal{D}_i$ to exclude extrapolation, with unsupported gaps lying outside $\bigcup_i \mathcal{D}_i$.

The union of the admissible interval boundaries defines positive-width atomic cells $B_r = [\ell_r, \ell_{r+1}]$. A view contributes only when its interpolation domain covers the entire cell, giving $\mathcal{I}_r = \{i | B_r \subseteq \mathcal{D}_i\}$ and $N_r = |\mathcal{I}_r|$. This partition splits partially overlapping view intervals at every endpoint and internal sampling node, so the contributing views remain fixed within each cell. Where several views overlap, we evaluate their interpolants at the same scale coordinate and average them with equal weight $1/N_r$ before integrating along the scale axis. For example, suppose views 1 and 2 have admissible intervals $[a, c]$ and $[b, d]$, respectively, with $a < b < c < d$. Their endpoints give $(\ell_k, \ell_{k+1}, \ell_{k+2}, \ell_{k+3}) = (a, b, c, d)$ and induce the cells $B_k = [a, b]$, $B_{k+1} = [b, c]$, and $B_{k+2} = [c, d]$. The corresponding support sets are $\mathcal{I}_k = \{1\}$, $\mathcal{I}_{k+1} = \{1, 2\}$, and $\mathcal{I}_{k+2} = \{2\}$. Thus, only the middle cell averages two interpolants, with $N_{k+1} = 2$. For a generic nonempty cell B_r, let $\Delta\ell_r = \ell_{r+1} - \ell_r$. We average the interpolated metric over the cell and its supporting views:

$$V_r^M = \frac{1}{N_r} \sum_{i \in \mathcal{I}_r} \frac{1}{\Delta\ell_r} \int_{\ell_r}^{\ell_{r+1}} \widehat{y}_i^M(\ell)\, d\ell. \tag{14}$$

By linearity of integration, this expression equals the normalized integral of the cross-view mean curve within the cell. The cell integral is $\Delta\ell_r V_r^M$, and its covered scale length is $\Delta\ell_r$. Overlapping support is therefore counted once regardless of N_r. Each contributing interpolant is linear within an

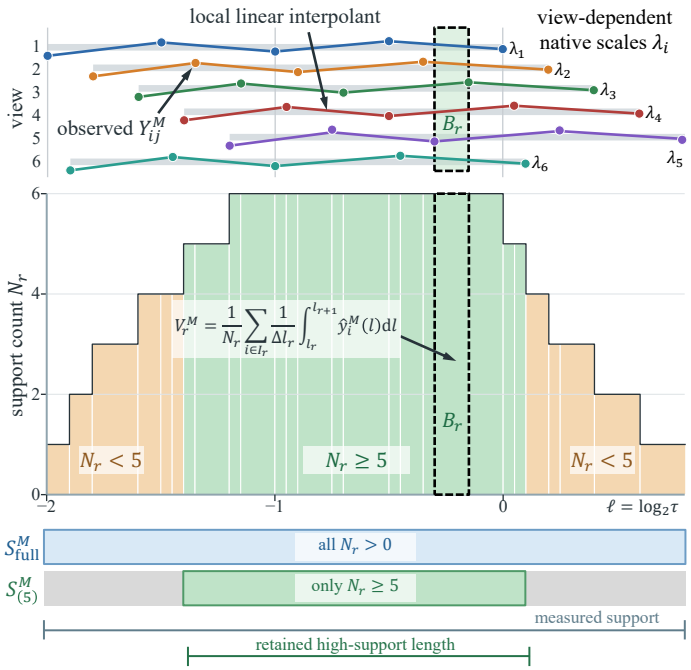


Figure 6: Atomic cells and view support. Only five nodes per view are shown for illustration. Evaluation requests 17 target scale nodes per view.

atomic cell, so the normalized integral equals the mean of its endpoint values:

$$V_r^M = \frac{1}{2N_r} \sum_{i \in \mathcal{I}_r} \left[\widehat{y}_i^M(\ell_r) + \widehat{y}_i^M(\ell_{r+1}) \right]. \tag{15}$$

An atomic boundary may lie between two measured nodes of a contributing view. We obtain the value at that boundary from Eq. (13). Cells with $N_r = 0$ contribute neither metric values nor covered length to the aggregate scores.

Fig. 6 illustrates how view-specific sampling nodes and overlapping support intervals induce atomic cells on the shared scale axis. Within each cell, the interpolants of the covering views determine V_r^M, while N_r records the number of contributing views. This construction provides the cell values and support counts used in the aggregation described below.

A.2.2 Global Atomic-Cell Construction

Each view supplies a local linear interpolant only between adjacent measured nodes. The union of all node boundaries defines a common partition. Within each cell, the supporting views remain fixed and each contributing interpolant remains linear. The trapezoidal expression in Eq. (15) exactly integrates the constructed piecewise-linear interpolant. The approximation error relative to F_i^M depends on the unobserved behavior between measured nodes. This construction requires no fitted smoothing bandwidth and keeps unsupported gaps separate.

We sample 17 target nodes uniformly in log scale over each view's two-octave interval, giving a nominal spacing of 0.125 octave. Rounding output dimensions to integers can merge nodes or change the realized spacing, so we construct the integration cells from the realized nodes. Before comparing methods, we freeze the realized nodes and admissible intervals in a shared manifest. Following § 3.3, we record $\Delta_{\max}$ as the largest gap between adjacent realized nodes within a view, providing a diagnostic of the interpolation intervals. Across the evaluation manifests for all 21 scenes, the maximum realized adjacent-node gap is 0.1289 octaves, close to the nominal spacing of 0.125 octaves. This design limits the distance over which we interpolate, although metric curves may still bend between samples. We evaluate all methods on this fixed grid to ensure that their observed curve shapes do not influence the sampling locations.

Table 3: Native camera-scale statistics and shared supported intervals in octaves, together with retained high-support coverage Q_5.

Dataset	Scenes	Views	Median $\log_2 \lambda_i$	Q10–Q90	Central 80% span	Q_5	Supported $\log_2 \tau$ 50% / 80% of views
Tanks and Temples	2	70	-0.001	$[-0.926, 0.719]$	1.644	66.3%	$[-1.964, -0.014]$ / $[-1.414, -0.614]$
Deep Blending	2	62	0.013	$[-0.616, 0.668]$	1.284	74.5%	$[-1.947, 0.003]$ / $[-1.547, -0.397]$
Mip-NeRF 360	9	246	0.051	$[-0.833, 0.972]$	1.805	77.2%	$[-1.911, 0.039]$ / $[-1.611, -0.411]$
BungeeNeRF	8	155	0.009	$[-1.364, 1.799]$	3.164	65.7%	$[-1.977, -0.077]$ / N/A

Table 4: Efficiency for the component configurations on Mip-NeRF 360, normalized to native 3DGS.

Configuration	Gaussian Count	Extra Scalars	Train Time	Peak Memory
3DGS	1.000×	0	1.000×	1.000×
+Target-first Training	0.534×	0	1.284×	0.548×
+Joint Residuals	0.545×	12	1.341×	0.634×
+Evidence Calibration	0.544×	12	1.447×	0.684×
+Detail Allocation (Ours)	0.601×	12	1.463×	0.743×

A.2.3 Scale-Length Aggregation and View Support

Following Eq. (11), we first average the supporting views within each cell to obtain V_r^M, then aggregate these values by the log-scale length $\Delta\ell_r$. Covered intervals of equal log-scale length therefore have equal influence on the score. We report S^M_{Full} over all measured support and $S^M_{(5)}$ over cells supported by at least five views, using five as an empirically chosen support threshold. The fraction Q_5 measures the scale length retained by this restriction. Fig. 6 illustrates these full-range and high-support summaries on the same atomic-cell partition. These summaries describe performance across the covered scale range and within regions supported by more views, with Q_5 indicating how much of the range those regions represent.

A.3 Additional Experiments

A.3.1 Dataset Scale Distributions

The evaluation manifests contain 70 test views for Tanks and Temples, 62 for Deep Blending, 246 for Mip-NeRF 360, and 155 for BungeeNeRF. For $s_{\min} = 0.25$, the nominal support of camera i on the log-scale axis is $[\log_2 s_{\min}\lambda_i, \log_2 \lambda_i] = [\log_2 \lambda_i - 2, \log_2 \lambda_i]$, spanning two octaves.

Per-scene normalization leaves substantial variation in native camera scales, as shown in Tab. 3. The central spans range from 1.284 to 3.164 octaves despite pooled medians near zero. BungeeNeRF has the widest distribution, with a central span exceeding the two-octave support of a single view and no scale reaching 80% view coverage. The frozen scale manifest determines Q_5 independently of the rendering method. We therefore report Q_5 once in Tab. 3. Mip-NeRF 360 has a wider central span than Tanks and Temples or Deep Blending but the largest Q_5. The central span measures the dispersion of native camera scales, whereas Q_5 measures the retained scale length after imposing the support threshold. A wide span can therefore coexist with high retained support when the view-specific intervals overlap across that span.

A.3.2 Representation Cost and Efficiency

Tab. 4 reports Gaussian count, training time, and peak memory on Mip-NeRF 360 relative to native 3DGS, together with the number of additional learned scalars per Gaussian. Target-first multi-scale training reduces Gaussian count and peak memory while increasing training time. Joint residuals and evidence calibration further increase training time and peak memory with little change in Gaussian count. Detail allocation increases Gaussian count and peak memory with a modest additional training cost. Coarser-scale supervision may reduce the need for gradient-driven densification, accounting for the more compact representation. However, online image resizing adds computation at each training iteration. Residual evaluation and regularization also operate over all Gaussians. Evidence calibration adds view-support statistics, indexed updates, and synchronization between the host and device. These operations offset the computational savings from fewer Gaussians. Overall,

Table 5: Mip-Splatting training-strategy comparison on Mip-NeRF 360. The original-strategy results are taken from Tab. 1.

Training strategy	PSNR↑	SSIM↑	1-LPIPS↑	$S^{\text{PSNR}}_{\text{Full}}$ ↑	$S^{\text{SSIM}}_{\text{Full}}$ ↑	$S^{\text{1-LPIPS}}_{\text{Full}}$ ↑	$S^{\text{PSNR}}_{(5)}$ ↑	$S^{\text{SSIM}}_{(5)}$ ↑	$S^{\text{1-LPIPS}}_{(5)}$ ↑
Original multi-scale	27.770	0.829	0.847	27.192	0.849	0.894	27.996	0.864	0.908
→ our target-first	27.683	0.819	0.833	27.293	0.858	0.902	28.090	0.869	0.913

the full model uses fewer Gaussians and less peak memory than native 3DGS, at the cost of longer training. This compactness is an additional benefit alongside the improvements in rendering quality.

A.3.3 Extending Target-First Training to Mip-Splatting

Motivated by the strong performance of Mip-Splatting (Yu et al., 2024) in Tab. 1, its simple pipeline, and the ease of adapting it to our training scheme, we introduce target-first training into Mip-Splatting and evaluate it on Mip-NeRF 360, with results reported in Tab. 5. Replacing its original multi-scale sampling strategy yields modest improvements across all six continuous-scale metrics, accompanied by a decline in native-resolution quality. Redistributing supervision more evenly across feasible scene scales may reduce the relative emphasis on native-resolution detail, introducing a trade-off within the shared representation. The modest gains may also reflect the strong baseline established by Mip-Splatting's algorithmic design and multi-scale training, leaving limited room for improvement through sampling alone. These results suggest that target-first sampling can complement an established multi-scale pipeline.

A.3.4 Additional Qualitative Comparisons

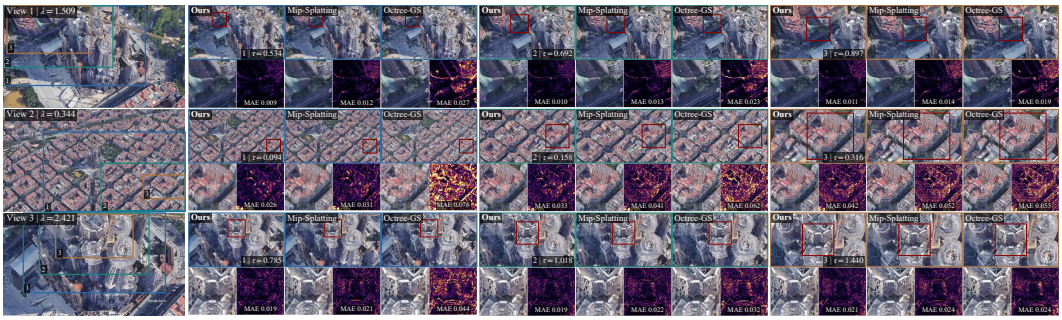
Figs. 7 to 9 extend Fig. 5 to six additional scenes, with three views and three selected unified scales per view. Each row places the native GT view on the left and three sampled scale groups on the right. Numbered frames on the native image identify progressively smaller context regions as the sampled scale increases. Each scale group compares ContinuousGS, Mip-Splatting, and Octree-GS within the same context window. Within these context windows, all methods use the same error-analysis region, whose location and extent remain fixed in normalized coordinates of the native image across scales. Below each prediction, a local enlargement and an absolute-error map show this region. We compute local RGB MAE against the scale-matched GT before enlarging the crops for display.

A.4 Limitations and Future Directions

Scale Estimation and Calibration. We represent the sampling scale of each camera with a single statistic derived from calibrated intrinsics and sparse COLMAP geometry, which provides a common coordinate across views. However, a single view-level statistic cannot capture spatial variations in sampling density caused by depth changes within an image. The scale estimate also depends on the availability of reliable geometric observations. When too few valid COLMAP points are directly associated with a view, the fallback procedure estimates projection density using either projected training points or the center of the training point cloud. This estimate may not be representative of the surfaces visible in that view. Possible strategies for addressing these limitations include spatially varying scale fields, anisotropic footprint models, and visibility-aware point selection, which warrant further investigation.

Training Efficiency. Our full model requires longer training than native 3DGS despite using fewer Gaussians and less peak memory (Tab. 4). Future work could combine our framework with more efficient rasterization and optimization schedules to reduce training time while preserving rendering quality across scales.

Evaluation and Generalization. We use 17 log-uniform target nodes per view and piecewise-linear integration to balance scale coverage, evaluation cost, and reproducible comparison. Atomic-cell integration is exact for the constructed interpolants, while approximation of continuous performance depends on local curve variation and the realized node spacing. Further exploration of shared adaptive sampling and efficient continuous performance estimators is needed to better capture sharp scale-dependent changes under common evaluation conditions. Computational constraints also limit

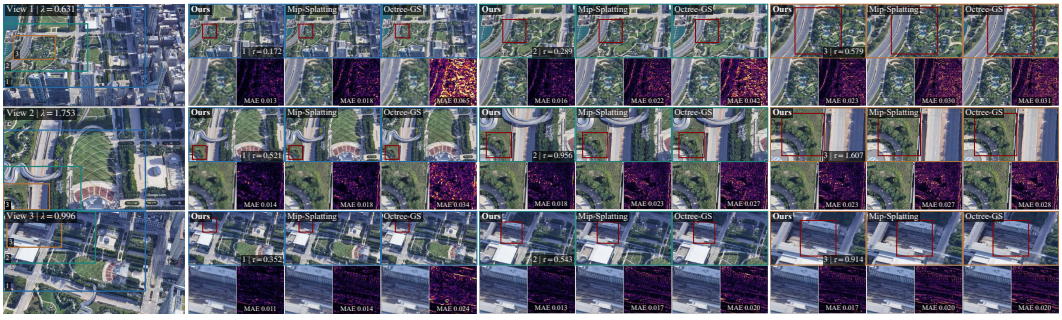


(a) Scene *Barcelona* from BungeeNeRF (Xiangli et al., 2022).

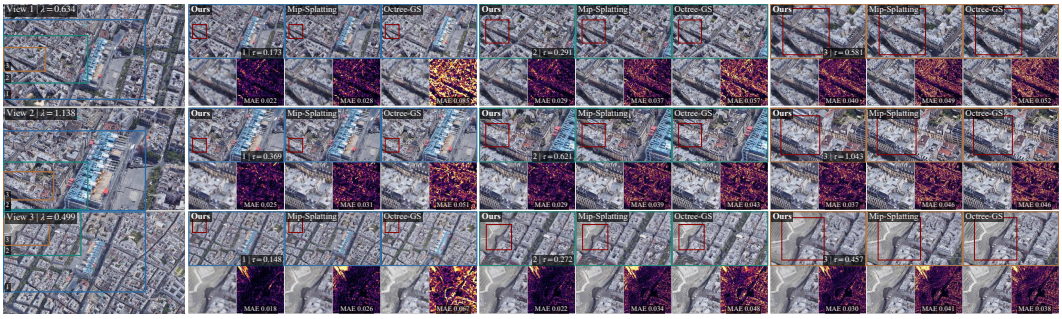


(b) Scene *Bicycle* from Mip-NeRF 360 (Barron et al., 2022).

Figure 7: Qualitative comparisons across three views and three sampled unified scales per view.



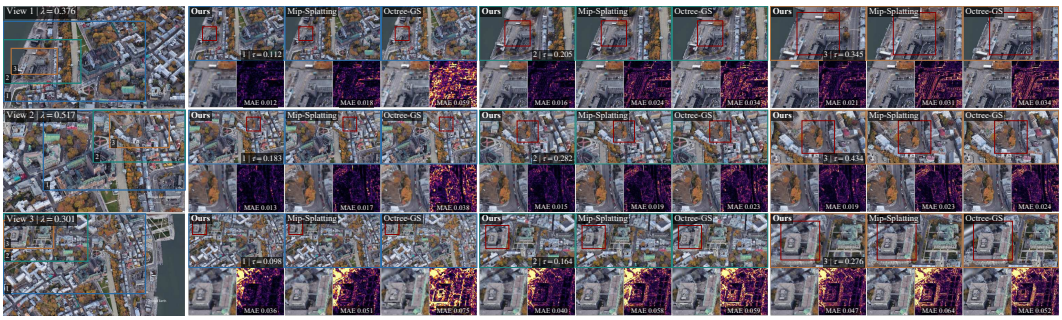
(a) Scene *Chicago* from BungeeNeRF (Xiangli et al., 2022).



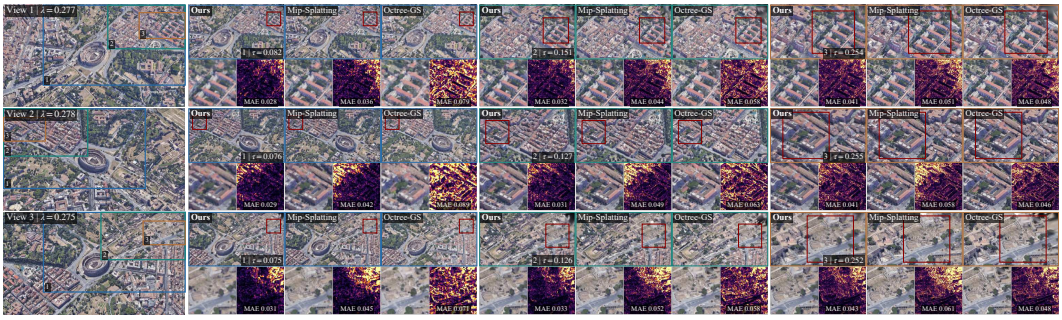
(b) Scene *Pompidou* from BungeeNeRF (Xiangli et al., 2022).

Figure 8: Qualitative comparisons across three views and three sampled unified scales per view.

our evaluation to 21 static scenes. Future assessment could extend to sparse capture, optical zoom, and scene motion, and examine performance under explicit memory and latency budgets.



(a) Scene *Quebec* from BungeeNeRF (Xiangli et al., 2022).



(b) Scene *Rome* from BungeeNeRF (Xiangli et al., 2022).

Figure 9: Qualitative comparisons across three views and three sampled unified scales per view.