3D SMoE Splatting for Edge-aware Realtime Radiance Field Rendering

YI-HSIN LI, Mid Sweden University, Sweden and Technische Universität Berlin, Germany
THOMAS SIKORA, Technische Universität Berlin, Germany
SEBASTIAN KNORR, Hochschule für Technik und Wirtschaft Berlin, Germany
MÅRTEN SJÖSTRÖM, Mid Sweden University, Sweden

Fig. 1. Conventional 3D Gaussian Splatting (3DGS) vs Our 3D SMoE Splatting (3DSMoES)

Steered Mixtures-of-Experts (SMoE) is an existing regression framework that has previously been applied for modeling and compression of 2D images and higher-dimensional imagery, including compression of light fields and light-field video. SMoE models are sparse, edge-aware representations that allow rendering of imagery with few Gaussians with excellent quality. In this paper a novel, edge-aware "3D SMoE Splatting" (3DSMoES) framework for 3D rendering is introduced, adopted to fit into the existing "3D Gaussian Splatting" (3DGS) CUDA optimization pipeline. Here, SMoE regression serves as a "plug-and-play" solution that replaces the established 3DGS regression as a novel workhorse. 3DSMoES achieves significant visual quality gains with drastically fewer Gaussian kernels compared to 3DGS. We observe up to approximately 4dB improvement in PSNR on individual scenes with kernel reductions between 20 to 50 percent. The sparse models are significantly faster to train and allow up to 30-50 percent improved rendering speeds.

Code for this paper is at https://github.com/yihsinli/3D-SMoE-Splatting.

CCS Concepts: • **Computing methodologies** → **Rasterization**; • **Mathematics of computing** → **Regression analysis**.

Additional Key Words and Phrases: Gaussian Splatting, Compression, Steered Mixture of Expert

ACM Reference Format:
Yi-Hsin Li, Thomas Sikora, Sebastian Knorr, and Mårten Sjöström. 2025. 3D SMoE Splatting for Edge-aware Realtime Radiance Field Rendering. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25), December 15–18, 2025, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3757377.3763899

Authors' Contact Information: Yi-Hsin Li, Computer and Electrical Engineering, Mid Sweden University, Sundsvall, Sweden and Technische Universität Berlin, Berlin, Germany, yi-hsin.li@miun.se; Thomas Sikora, Technische Universität Berlin, Berlin, Germany, thomas.sikora@tu-berlin.de; Sebastian Knorr, Hochschule für Technik und Wirtschaft Berlin, Berlin, Germany, Sebastian.Knorr@HTW-Berlin.de; Mårten Sjöström, Computer and Electrical Engineering, Mid Sweden University, Sundsvall, Sweden, Marten.Sjostrom@miun.se.

SA Conference Papers '25, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-2137-3/25/12
https://doi.org/10.1145/3757377.3763899

1 Introduction

Achieving realistic 3D scene representation remains a long-standing challenge in computer vision and computer graphics. Over the decades, the field has advanced through major milestones, from raster graphics in the 1960s to ray tracing in the 1980s and real-time GPU rendering in the 2000s. Despite these breakthroughs, capturing and rendering complex real-world scenes efficiently and with high fidelity continues to push the boundaries of research.

Neural Radiance Fields (NeRFs) [Mildenhall et al. 2021] introduced a paradigm shift in scene reconstruction by representing volumetric scenes with an implicit neural model. NeRFs set a new standard for photorealistic view synthesis but remain computationally expensive—requiring billions of evaluations per image—and hard to manipulate. Various extensions, such as Mip-NeRF [Barron et al. 2022] and Instant-NGP [Müller et al. 2022], improve efficiency, yet they suffer from scalability issues in large or sparse scenes.

3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] has emerged as a compelling explicit alternative, offering high rendering quality and direct scene manipulation. Instead of encoding scenes in neural networks, 3DGS represents them with Gaussian kernels that are projected and blended in screen space. As scene complexity grows, maintaining quality demands more Gaussians, increasing computation and memory costs. To address this, recent research has explored methods for reducing redundancy in 3DGS by refining densification strategies [Cheng et al. 2024; Fang and Wang 2024; Kim et al. 2024; Mallick et al. 2024; Rota Bulò et al. 2025; Zhang et al. 2024b,a]. These approaches optimize how new Gaussians are introduced and merged, significantly improving efficiency while preserving rendering quality. However, the "Weighted Sum of Gaussians" paradigm is

known to require many kernels to represent sharp edges [Fornberg et al. 2002], motivating sparsity-focused alternatives.

Sparsity of representation has been the prime focus of the well-established, edge-aware Steered Mixture of Experts (SMoE) approach. This regression strategy has been developed over the last 10 years for efficient representation and coding of 2D images and video, as well as for higher-dimensional images [Avramelos et al. 2018; Jongebloed et al. 2023; Lange et al. 2016; Verhack et al. 2018, 2017, 2016, 2020]. Alongside these developments, recent work introduces kernel-based alternatives to Gaussians—adaptive shapes and normalization schemes [Celarek et al. 2025; Diolatzis et al. 2024; Hou et al. 2025; Liu et al. 2025b; Zwicker et al. 2001b]—which also aim to improve regression quality and efficiency in 3D scene representation.

The prime purpose of this paper is to investigate the gains achievable with edge-aware 3D SMoE Splatting (3DSMoES) as an alternative regression strategy within the 3DGS framework. We prioritize improving regression accuracy and runtime efficiency over compression. To this end, we develop and integrate a novel, specifically tailored SMoE gating regression strategy into the existing 3DGS optimization pipeline - where it replaces the "Weighted Sum of Gaussians" concept.

This allows a direct comparison of the benefits and drawbacks of either method, as well as with other recent splatting approaches, on equal grounds. More specifically, SMoE is adopted to the 3DGS 2D projection space, leveraging 3DGS's rasterization pipeline to enable CUDA-accelerated training while retaining SMoE's sparsity and edge-preserving benefits. This enables seamless integration into existing 3DGS pipelines, complementing rather than replacing densification-based optimizations.

Figure 1 provides a snapshot of impressive gains achievable with SMoE regression. Besides drastic improvement of visual quality, 3D SMoE Splatting also improves the memory footprint as well as training and rendering speed significantly.

Applying SMoE in projected space as with 3DGS introduces occlusion challenges. Direct SMoE softmax normalization in projected space produces ghosting artifacts. We address this with a view-dependent decay term, inspired by NeRF and 3DGS, modulating Gaussian contributions based on depth and viewing direction, ensuring correct occlusion handling and depth consistency.

In this paper, our work focuses on optimizing the SMoE regression mechanism, making it complementary to existing Gaussian Splatting approaches. Our main contributions are the following:

- Introduction of an edge-aware 3D SMoE Splatting approach as a novel sparse work-horse to 3DGS, using "Weighted Sums of Soft-Gates". We extend prior SMoE gating networks with opacity-driven 3D soft-max gating functions, built on splatted Gaussians.
- Adaptation of SMoE strategies into the 3DGS rasterization pipeline in a 2D projected space. This results in a novel optimization concept compared to prior SMoE works.
- Introduction of a novel view-dependent decay term for SMoE regression, addressing occlusion in 2D projection space by adapting techniques from NeRF and Gaussian Splatting.
- Demonstration of drastic quality and computational gains with the sparse 3D SMoE Splatting approach, evaluated on equal grounds with 3DGS and related methods, within the prior established 3DGS benchmark framework.

2 Related work

2.1 3D Gaussian Splatting (3DGS)

3D Gaussian Splatting [Kerbl et al. 2023] has emerged as a powerful rendering technique, representing scenes by projecting 3D Gaussians onto a 2D image plane. By blending multiple Gaussians, 3DGS efficiently reconstructs complex visual details. Recent works have focused on improving 3DGS in terms of quality, real-time performance, and robustness. However, a fundamental limitation of 3DGS is its reliance on accumulated blending for color regression, which, while effective in dense representations, becomes inefficient in sparse settings where only a few Gaussians are available.

Research on 3DGS largely falls into adaptive densification and compression. The original densification process struggles with compactness—yielding too many Gaussians (overfitting) or too few (underfitting). Recent methods like Taming 3DGS [Mallick et al. 2024], Mini-Splatting [Fang and Wang 2024], GaussianPro [Cheng et al. 2024], Pixel-GS [Zhang et al. 2024a], and GS-Octree [Li et al. 2024] address this by refining Gaussian placement via geometric priors, multi-view cues, and hierarchies—substantially improving reconstruction. However, densification only adjusts spatial distribution; it leaves the regression mechanism untouched. Our 3D SMoE Splatting is orthogonal: rather than modifying how Gaussians are placed, we revise how they are blended, enabling seamless integration with existing densification strategies.

Compression methods aim to reduce both the number of Gaussians and their memory footprint. 3DGS-C [Niedermayr et al. 2024] applies quantization and pruning, followed by quantization-aware fine-tuning to maintain quality. 2DGS [Huang et al. 2024] collapses 3D into 2D, cutting memory without quantization or pruning. Unlike densification, compression methods fit naturally into our framework. Notably, 3DGS-C's fine-tuning transfers directly to SMoE. While HAC++ [Chen et al. 2025] and HEMGS [Liu et al. 2025a] push aggressive compression, our method prioritizes regression quality and runtime efficiency in compact regimes.

Beyond these, some works alter the rendering pipeline or kernel. Sort-free GS [Hou et al. 2025] skips depth sorting via a two-pass heuristic—losing physical transmittance. We retain sorting with a one-pass joint normalization that supports edge-aware regression. Surface Splatting [Zwicker et al. 2001b] normalizes only forward; we normalize both forward and backward, aligning gates during training. Other kernel variants include: N-D Gaussians [Diolatzis et al. 2024] for flexible shapes; Celarek et al. [2025], who normalize locally per Gaussian; and Deformable Beta Splatting [Liu et al. 2025b], which replaces Gaussians but retains weighted sums—without edge-aware gating.

2.2 Steered Mixture of Experts (SMoE)

The Steered Mixture of Experts (SMoE) approach shares strong conceptual ties with 3DGS. Originally developed for 2D image regression in 2016 [Verhack et al. 2016], early work on SMoE rooted in a statistical approach. 2D image signals were modeled with 3D Gaussian Mixture Models. As such, SMoE regression has conceptual

similarity with groundbreaking work on Mixture-of-Experts developed in the neural network domain. Here the gating functionality, later adopted in SMoE, was first introduced [Jacobs et al. 1991]. In SMoE, the so-called "experts" are multiple regression functions which are "gated" using soft-gating functions. The reader is referred to [Tok et al. 2018] for more insight into the subject. Optimizing the SMoE model parameters in early works involved Expectation-Maximization (EM) algorithms to minimize likelihood criteria. This resulted in a fast parameter optimization and sparse representation of images. Image quality, especially reconstruction of sharp edges, improved drastically with the introduction of mean-squared error (MSE) optimization using gradient descent [Bochinski et al. 2018; Tok et al. 2018], albeit to the expense of drastically larger training times. On the other hand, in contrast to early SMoE EM optimization, the gradient descent strategy allowed to design and optimize novel, very flexible SMoE models based on Gaussian kernels. In addition, the use of advanced objective criteria beyond MSE (i.e. SSIM) including regularization for sparsification was made possible [Jongebloed et al. 2019]. This includes concepts for quantization and coding of Gaussians for SMoE compression of images and video. Beyond 2D, SMoE has been extended to higher-dimensional data, including 3D video [Jongebloed et al. 2023], 4D light fields and 5D light field videos [Avramelos et al. 2018; Verhack et al. 2018, 2017, 2020]. These extensions model complex spatio-temporal variations using high-dimensional Gaussian kernels. However, they rely on the optimization in high-dimensional space, leading to prohibitively long training times. This computational bottleneck has limited the practical adoption of high-dimensional SMoE models. For small image blocks with limited amounts of steered kernels, real-time optimization using Autoencoder networks [Fleig et al. 2023b] were developed to speed up optimization. Extensions of the basic SMoE concept with Epanechikov kernels for 2D as well as light field imagery and video with excellent results appeared in [Liu et al. 2024, 2019]. Work on optimized initialization of gradient descent for SMoE and RBF networks in 2D imagery was recently published in [Li et al. 2025]. Here, also the gains achievable using SMoE compared to “weighted sum of Gaussians” (the work-horse of 3DGS, 2DGS and related methods) were shown, however on 2D images only.

Historically, SMoE has been applied across various domains including image denoising and super-resolution [Fleig et al. 2023a,b; Özkan et al. 2024; Özkan et al. 2023], yet the connection between Gaussian Splatting and SMoE has been largely overlooked.

3 Methodology

To allow comparison of 3DGS and related methods with SMoE regression on equal grounds, SMoE is integrated into the 3DGS framework. Core steps of 3DGS, including 3D-to-2D projection and CUDA rasterization, are thus employed with the novel 3DSMoES approach introduced in this paper. A SMoE "Sum of Weighted Soft-Gates" regression, specifically adapted and optimized for seamless integration, replaces the 3DGS "Weighted Sum of Gaussians" strategy.

3.1 Foundations of 3DGS and SMoE Regression

3.1.1 3D Gaussian Splatting. Kerbl et al. [Kerbl et al. 2023] introduced 3D Gaussian Splatting (3DGS), a differentiable rendering technique that represents scenes using 3D Gaussian primitives. Each Gaussian is parameterized by its position p_i and a 3D covariance matrix Σ, defined as:

$$G(p) = \exp\left(-\frac{1}{2}(p-p_i)^T\Sigma^{-1}(p-p_i)\right), \tag{1}$$

where $\Sigma = RSS^TR^T$ is factorized into a scaling matrix S and a rotation matrix R. To render an image efficiently, 3D Gaussians are first transformed into 2D to leverage CUDA rasterization, enabling highly parallelized computation. By projecting 3D Gaussians onto the image plane, all subsequent calculations operate in 2D space, allowing per-pixel parallel processing and maximizing GPU efficiency.

Each 3D Gaussian is transformed into camera coordinates using the world-to-camera transformation matrix W and then projected onto the image plane via a local affine transformation J, yielding a 2D Gaussian kernel G_{2D} with covariance matrix Σ_{2D}, obtained by discarding the third row and column of:

$$\Sigma' = JW\Sigma W^TJ^T. \tag{2}$$

For a detailed derivation of Σ_{2D}, we refer to [Yifan et al. 2019; Zwicker et al. 2001a]. Rendering is performed as a "Weighted Sum of Gaussians", where the final pixel color $C_{GS}(x)$ at 2D image coordinate x is computed as:

$$C_{GS}(x) = \sum_{i=1}^{K} c_i o_i G^i_{2D}(x) T_i(x), \tag{3}$$

where K is the number of Gaussians, c_i represents the view-dependent appearance (weight coefficients), derived from the spherical harmonics coefficients c_{sh} of the i-th Gaussian given the view direction d, and o_i (opacity) controls its contribution. The term T_i represents the accumulated transmittance, which governs view-dependent decay:

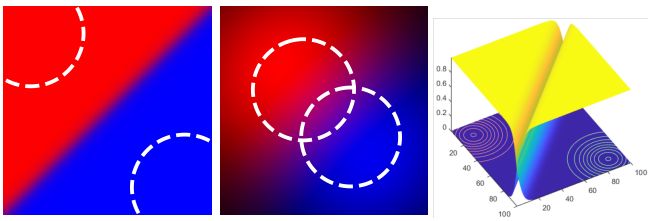
$$T_i(x) = \prod_{j=1}^{i-1}\left(1 - o_j G^j_{2D}(x)\right). \tag{4}$$

This ensures that Gaussians contribute less along the view direction d when occluded by nearer Gaussians, as their opacity dominates. However, if a distant Gaussian is not occluded, its weight can remain significant, as its visibility is unaffected by the nearer Gaussians. By integrating these visibility-based weighting mechanisms, CUDA rasterization enables efficient, real-time reconstruction of complex 3D scenes.

3.1.2 Steered Mixture of Experts. In contrast to Gaussian Splatting, Steered Mixture of Experts in its standard form (SMoE_{std}) employs a gating mechanism that normalizes all Gaussians, improving efficiency in sparse rendering. The equation for SMoE_{std} color regression is the weighted sum of soft-gates:

$$C_{\text{SMoE}_{\text{std}}}(x) = \sum_{i=1}^{K} c_i w_i(x), \tag{5}$$

where c_i is the weight coefficient (the SMoE expert) and $w_i(x)$ is the above-mentioned 2D soft-gating function. In prior SMoE works no opacity term was used to calculate the soft-gating function. To align with 3DGS, we integrate the opacity term into the calculation of the soft-gates $w_i(x)$, calculated based on the i-th Gaussian as:



(a) SMoE (b) GS (c) SMoE Gates

Fig. 2. Sparsity of SMoE vs GS. a) The sparse, edge-aware SMoE regression strategy is able to reconstruct the sharp edge between red and blue regions with only 2 Gaussians (white equilines) to almost perfect fidelity after MSE optimization. b) GS's "weighted sum of Gaussians" strategy with only 2 kernels is by far not sufficient to reconstruct any sharp transition. c) SMoE exhibits normalization-driven edge alignment: gates conform to signal boundaries, kernel overlapping remains minimal (red and green equilines), and soft-gates modulate their support to respect the edge structure.

$$w_i(x) = \frac{o_i G^i_{2D}(x)}{\sum_{j=1}^{K} o_j G^j_{2D}(x)}. \tag{6}$$

This formulation normalizes 2D Gaussian kernels per pixel and iteration—not outputs c_i. Like a spatial softmax, the normalization amplifies small response differences, enforcing competition. Pixels are softly assigned to dominant kernels, yielding implicit segmentation. In overlaps, normalization sharpens transitions so boundaries emerge. MSE optimization then aligns gates (Eq. 6) to edges, producing crisp, unsupervised segmentation through weighted sums of adjacent gates (Eq. 5).

Each Gaussian generates a 2D gating function (segment with soft-border transitions into neighboring segments). The gating mechanism, essentially a form of softmax-based normalization, acts as a soft decision segmentation. Segments can have sharp transitions for edges or smooth transitions to model smooth pixel variations.

3.1.3 Sparsity of SMoE vs Gaussian Splatting. Figure 2 illustrates the different regression functionality and abilities of Gaussian Splatting (GS) and SMoE on a synthetic test image with sharp edge after MSE optimization. The image presents a sharp diagonal transition between a red and a blue region. For both methods two "round" kernels were "splatted" into the image for reconstruction. With SMoE the two round optimized Gaussian kernels with same bandwidth completely suffice to reconstruct the sharp diagonal edge with almost perfect quality (reconstructed with Eq. 5). Based on the two Gaussians, two 2D gating functions (segments) are derived (Eq. 6), which support the red and blue regions. Each soft-gate in this case appears to have sharp transitions into the other gate, exactly at the location of the edge. Since the "color-weighted sum of the two soft-gates" reconstructs the image, the sharp edge is perfectly rendered. The reader is referred to [Li et al. 2025] for more information on gating functionality in SMoE.

Optimized GS with two round kernels on the other hand derives the "color-weighted sum of the two Gaussian kernels" (Eq. 3) for reconstruction. No gating mechanism is involved. With only two Gaussian kernels it is impossible to reconstruct the sharp edge. Even though weighted sum of Gaussians are universal approximators [Park and Sandberg 1991], many Gaussians are required for perfect reconstruction. For the test sample 4-6 kernels with different bandwidths would be required to render the edge with sufficient quality (see Figure 3).

Figure 2 also illustrates the global nature and sparsity of representation of SMoE regression. The two derived soft-gates, now shown after optimization, in fact extend to infinity. Thus, SMoE reconstruction with the two kernels (as weighted sum of the two soft-gates) extends the red and blue pixels infinitely into regions beyond the image in Figure 2. Two round kernels would suffice to reconstruct an image with infinite number of pixels containing an infinitely long sharp edge line or smooth transition between regions. Such image with infinite number of pixels is coded with neglectable number of bits.

In contrast, GS regression operates locally: its output directly depends on the raw Gaussian values, which decay rapidly as the distance from the mean increases. As a result, regions far from any Gaussian center receive negligible influence, leading to low-intensity or black areas. While both approaches use clipped Gaussians, only SMoE maintains a coherent structure and intensity across the domain—capturing global context through gating, rather than relying on local density alone.

Figure 10 in Appendix C depicts a natural image divided into thousands of soft-gates after SMoE Splatting and optimization. The arbitrarily-shaped soft segments capture the edges and edge-lines in the image and provide excellent edge-reconstruction fidelity.

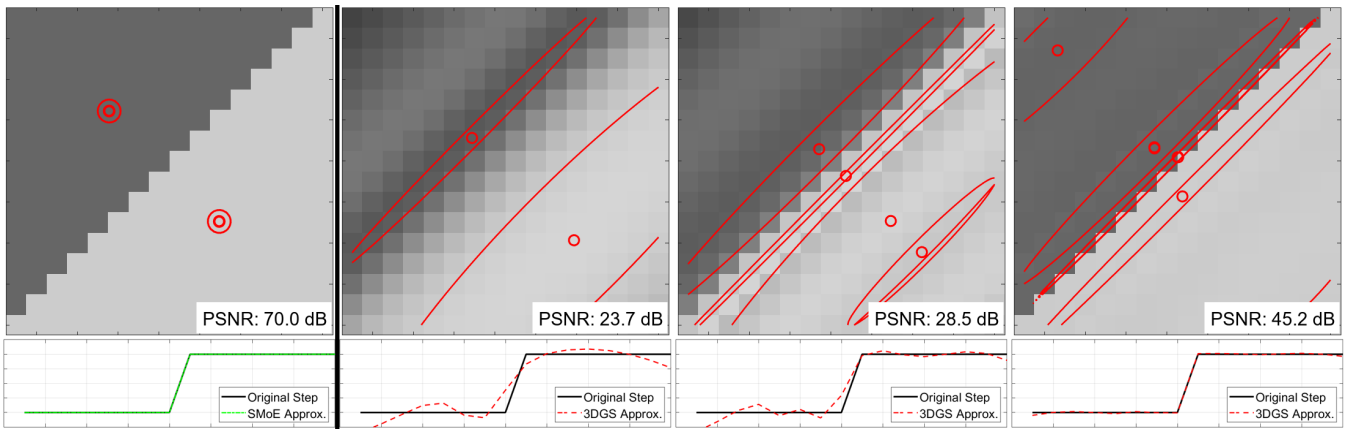
3.2 3D SMoE Splatting (3DSMoES)

In the 3DGS method, the *accumulated transmittance* term T has a crucial role for managing depth consistency and occlusion effects. This ensures that distant Gaussians are occluded appropriately, mitigating the risk of ghosting artifacts. Standard SMoE's normalization leads to compact and sharp representations, but does not incorporate the depth-sensitive occlusion handling present in 3DGS. A subject of investigation in this paper is to understand if and how an accumulated transmittance term T could enhance SMoE ability to handle depth-related effects in 3D rendering, without sacrificing sparsity of representation.

To this end, our final formulation of 3D SMoE Splatting integrates the accumulated transmittance term T directly into the gating function, resulting in a depth-aware extension of the original SMoE regression:

$$C_{SMoE}(x) = \sum_{i=1}^{K} c_i w^i_T(x) = \sum_{i=1}^{K} c_i \frac{o_i G^i_{2D}(x) T_i(x)}{\sum_{j=1}^{K} o_j G^j_{2D}(x) T_j(x)}. \tag{7}$$

This modification addresses two key aspects. First, the transmittance term T_i governs the visibility of each Gaussian, ensuring that distant Gaussians are properly occluded when necessary. This prevents the ghosting artifacts commonly observed in traditional SMoE rendering. Second, incorporating T in the denominator ensures that the blending weights are dynamically scaled relative to each Gaussian's visibility. This adjustment prevents distant Gaussians from exerting undue influence on the final color, reinforcing a depth-aware



(a) SMoE - 2 Gaussians (b) GS - 2 Gaussians (c) GS - 4 Gaussians (d) GS - 6 Gaussians

Fig. 3. Sparsity of representation with SMoE (ours) vs "weighted sum of Gaussians" (workhorse in 3DGS). (a) SMoE reconstructs a sharp edge with only 2 round Gaussians with same bandwidth, while (b–d) "weighted sum of Gaussians" requires 4-6 steered Gaussians with individual bandwidth to gain sufficient quality. The bottom 1D plots depict a line of pixels orthogonal to the edge (from upper left to bottom right). Reconstruction with "weighted sum of Gaussians" with few Gaussians results in overshoot artifacts and smeared edges.

weighting scheme. This visibility-aware normalization prevents inconsistencies in accumulated color blending, leading to more accurate and realistic depth perception as shown in Section 5.3. We apply a transparency cutoff on Gaussians with negligible visibility (similar to the original 3DGS) to improve performance. In practice, we discard Gaussian contributions when $G^i_{2D}(x) < 10^{-5}$.

4 Experimental setting

The evaluation focused on rendering quality, computational efficiency, and sparsity in the context of 3D scene representation.

4.1 Benchmarks

We compared our method with three baselines:

- **Original 3DGS (3DGS)** [Kerbl et al. 2023]: Standard 3D Gaussian Splatting using accumulated blending for color regression.
- **2D Gaussian Splatting (2DGS)** [Huang et al. 2024]: A 2D projection of 3DGS, reducing memory usage but lacking explicit quantization.
- **Deformable Beta Splatting (DBS)** [Liu et al. 2025b]: A state-of-the-art method replacing Gaussian kernels with learned Beta splats.

To test plug-and-play compatibility, we integrate 3DSMoES into four 3DGS variants: 3DGS-C [Niedermayr et al. 2024] which uses quantization and pruning, Taming 3DGS [Mallick et al. 2024] with score-based control, and the state-of-the-art HAC [Chen et al. 2024] and HAC++ [Chen et al. 2025] employing structure pruning and content-aware densification. The resulting models—3DSMoES-C, -Taming, -HAC, and -HAC++—show that our method integrates seamlessly into diverse compression and densification regimes.

4.2 Controlling the Number of Gaussians

Rate-distortion analysis evaluated the trade-off between compression and rendering quality, with Gaussian count (number of Gaussians, #G) being a critical factor affecting memory and performance.

3DGS increased Gaussian count via densification, driven by gradient magnitude and Gaussian scale—larger gradients triggered refinement, and larger Gaussians were split for local accuracy. Our 3DSMoES adopted the same densification strategy, adjusting the gradient threshold to ensure a fair comparison.

While we did not compare with specialized densification approaches, we retained 3DGS's original mechanism for consistency. This maintained the focus on our core contribution — replacing 3DGS's accumulated blending with 3DSMoES's gated Gaussian regression while keeping the densification techniques the same.

4.3 Datasets

We evaluated our method on three widely used datasets to ensure consistency with 3DGS benchmarks. From Tanks and Temples (TnT) [Knapitsch et al. 2017], we selected the "truck" and "train" scenes, both unbounded outdoor environments. From Mip-NeRF 360 [Barron et al. 2022], we used five indoor and four outdoor scenes matching 3DGS selections. From Deep Blending (DB) [Hedman et al. 2018], we included the "Dr Johnson" and "Playroom" indoor scenes. This selection aligned with prior works, ensuring comparability.

4.4 Evaluation Metrics

The rendering quality was measured using PSNR (pixel-wise fidelity), SSIM (structural coherence), and LPIPS (perceptual similarity). Sparsity was measured by the number of Gaussians required to meet a quality threshold, while compression efficiency was assessed via file size. Training and inference times were recorded to quantify computational cost.

4.5 Implementation

To ensure a fair comparison, we adopted the same train-test split as in the original 3DGS paper, selecting every 8th image for testing. Training followed 3DGS defaults, and evaluations used the test set

Table 1. Gains with 3DSMoES (Ours) compared to benchmarks. We provide objective quality measures and number of Gaussians #G in millions (M) required to evaluate the sparsity of the models. The resulting file size in megabytes (MB) provides information on Rate-Distortion efficiency and memory requirements. Comparisons target sparse (#G < 0.3M) and dense (#G > 1M) regimes, corresponding to thresholds 0.005 and 0.0002, the first and fifth of six tested in Fig. 3.

		Deep Blending (DB)					Tanks and Temples (TnT)					Mip-Nerf-360				
	Method	PSNR [dB]	SSIM	LPIPS	#G	Size	PSNR [dB]	SSIM	LPIPS	#G	Size	PSNR [dB]	SSIM	LPIPS	#G	Size
#G < 0.3 M	3DGS	26.98	0.87	0.33	0.059	13.9	20.21	0.75	0.30	0.106	25.3	23.32	0.68	**0.39**	0.088	20.9
	2DGS	24.94	0.85	0.37	**0.038**	**8.8**	18.65	0.71	0.36	**0.053**	**12.3**	21.59	0.65	0.43	**0.061**	**14.2**
	DBS	27.11	0.87	**0.32**	0.300	36.0	21.17	**0.78**	**0.27**	0.300	36.0	22.43	0.68	**0.39**	0.150	18.0
	Ours	**28.63**	**0.89**	**0.32**	0.039	9.4	**21.21**	0.77	0.30	0.062	14.9	**24.43**	**0.69**	**0.39**	0.064	15.4
#G > 1 M	3DGS	29.65	0.90	0.25	3.019	715.5	23.91	0.85	0.17	2.018	474.5	27.59	0.82	0.22	3.282	774.6
	2DGS	29.68	0.90	0.26	**1.506**	**350.4**	23.16	0.83	0.21	**0.875**	**203.6**	26.88	0.80	0.26	**1.904**	443.1
	DBS	29.63	**0.91**	**0.24**	3.000	360.0	**24.78**	**0.87**	**0.14**	1.75	210.0	**28.42**	**0.83**	**0.20**	3.110	**373.0**
	Ours	**30.08**	**0.91**	0.25	2.328	555.0	23.81	0.85	0.18	1.580	374.3	27.49	0.81	0.23	2.432	576.9

with metrics from Section 4.4. All experiments were run on a single NVIDIA A100 GPU. To analyze Gaussian count reduction, we varied the gradient threshold across six values. For reproducibility, all variants use Eq. 7 for forward rendering with normalized Gaussian weights, and the same form for gradients in backpropagation. Derivations are in the supplemental material and the released code.

The view-dependent transmittance term T remains in the numerator in our method, preserving correct occlusion and preventing ghosting. To verify that our normalization approach does not alter view-dependent effects, we conducted two ablation studies on the view-dependent transmittance term T: one assessing its inclusion and another isolating its impact in the denominator. We trained two additional models—one without T entirely, corresponding to Eq. 6, and another with T only in the numerator, using the following regression:

$$C_{SMoE_{T_n}}(x) = \sum_{i=1}^{K} c_i w^i_{T_n}(x) = \sum_{i=1}^{K} c_i \frac{o_i G^i_{2D}(x) T_i(x)}{\sum_{i=1}^{K} o_i G^i_{2D}(x)}. \quad (8)$$

5 Results and analysis

This section presents the strengths, efficiency, and design rationale of the proposed 3DSMoES. It begins with quantitative comparisons to establish core improvements, followed by analyses of efficiency. Qualitative evaluations further demonstrate the advantages of our method, and an ablation study concludes by isolating the contributions of key components.

5.1 Quantitative Performance Gains

Our method outperforms benchmarks in the sparsity–distortion trade-off. Figure 4 shows representative Rate-Distortion (RD) curves comparing 3DSMoES(-C) with key baselines—3DGS, 2DGS, and the compressed 3DGS-C. While not exhaustive, these plots highlight general trends across low- and high-rate regimes. More results are in Table 1, plug-and-play variants and memory in Table 2. These findings demonstrate our method's effectiveness in achieving high reconstruction quality with fewer primitives. The following subsections analyze the low-rate and high-rate regimes in more detail.

Table 2. Plug-and-play SMoE integration across 3DGS pipelines. PSNR is reported in dB; Train indicates training time in seconds. Results averaged across three datasets.

	Method	PSNR	SSIM	LPIPS	#G	Size	Train	FPS
#G ≲ 0.1 M	3DGS-C	23.40	0.72	0.37	0.086	2.9	1112	257
	3DSMoES-C	**24.47**	**0.73**	**0.36**	**0.059**	**2.4**	**796**	**329**
	3DGS-taming	25.67	0.75	0.34	0.113	3.8	323	151
	3DSMoES-taming	**25.71**	**0.76**	**0.33**	0.113	3.8	**280**	**194**
	3DGS-HAC	**26.36**	0.78	0.30	0.102	3.01	1817	162
	3DSMoES-HAC	26.21	0.78	0.30	**0.099**	**2.81**	**1274**	**186**
	3DGS-HAC++	**26.13**	0.76	0.33	0.103	2.1	2142	169
	3DSMoES-HAC++	25.98	0.76	0.33	**0.101**	**2.0**	**1606**	**189**
#G ≳ 0.4 M	3DGS-C	27.03	0.82	0.23	2.808	27.6	2074	259
	3DSMoES-C	27.03	0.82	0.23	**2.153**	**22.0**	**1594**	**284**
	3DGS-taming	**27.45**	0.82	0.24	0.810	7.9	617	109
	3DSMoES-taming	27.23	0.82	0.24	**0.782**	7.9	**570**	**137**
	3DGS-HAC	**27.46**	0.83	**0.23**	**0.467**	12.6	2378	97
	3DSMoES-HAC	27.28	0.83	0.24	0.476	**11.8**	**1723**	**110**
	3DGS-HAC++	**27.51**	**0.83**	**0.24**	**0.467**	7.1	3038	109
	3DSMoES-HAC++	27.34	0.82	0.25	0.472	**6.6**	**2343**	**123**

5.1.1 Quality for Small Numbers of Gaussians (<0.3 Million (M)). 3DSMoES reveals clear advantages in the low-rate regime. At the lowest rate, it surpassed 3DGS by an average of 1.7 dB across test sets. To reach a fixed PSNR of 24–25 dB, it needed just 60K Gaussians—nearly half the 115K required by 3DGS. The compressed variant (3DSMoES-C) mirrored these gains. While 2DGS sometimes used fewer Gaussians (see Table 1), its visual quality remained markedly inferior. DBS achieved slightly less distortion than 3DSMoES on Tanks and Temples but worse on other datasets. They did so using nearly the 0.3M Gaussian budget, indicating limited efficiency at low rates. These results affirm the strength of our gated regression, which maintains fidelity with far fewer primitives. In contrast, 2DGS and DBS reduce either error or kernels—but not both.

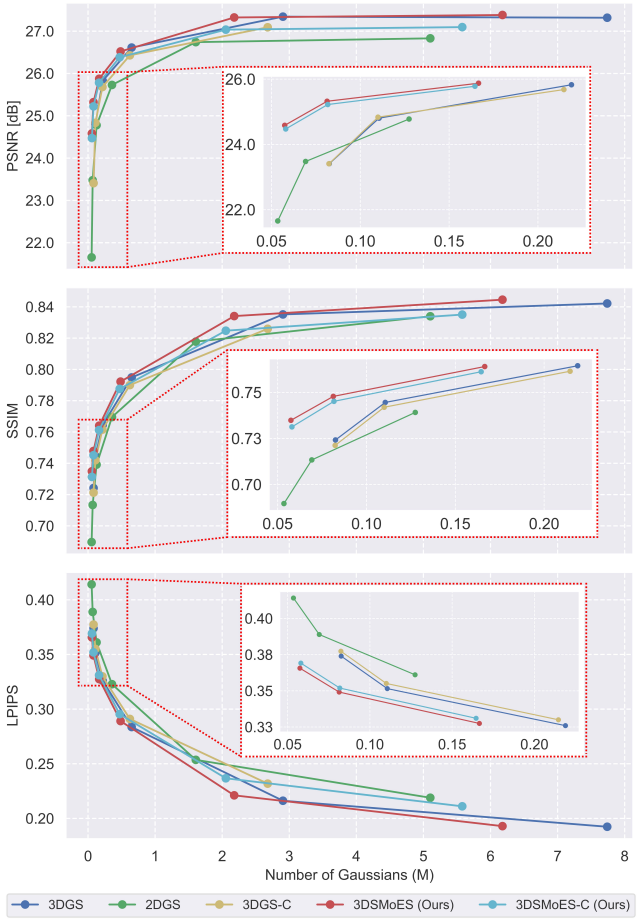


Fig. 4. Distortion vs number of Gaussians (in Millions M) required for the regression models. Our methods outperforms others especially for smaller numbers of Gaussians.

5.1.2 Quality for Large Numbers of Gaussian (1-8 Million (M)). Visual quality converged across methods, with similar metric scores as the rate increased. Nevertheless, 3DSMoES reached this level using 25–35% fewer Gaussians than 3DGS across all datasets. The compressed setting exhibited the same behavior. Though 2DGS remained the sparsest overall, its quality lagged behind. DBS tops PSNR and perceptual scores on Tanks & Temples and Mip-NeRF-360 but always uses more primitives than our method. 3DSMoES may trade quality for efficiency.

5.1.3 Run-Time Gain. Figure 5 presents runtime comparisons across the datasets from Table 1, showing that 3DSMoES outperforms most benchmarks.

At low rates, our method achieved a 1.1 dB PSNR gain with 31% fewer Gaussians, resulting in 30% faster training and 30% higher rendering throughput (in FPS) compared to 3DGS. These advantages extended to high-rate and compressed settings. In the low-bit regime, we render faster but train slower than DBS. At high-bit rates, we outperform DBS in both training and rendering speed.

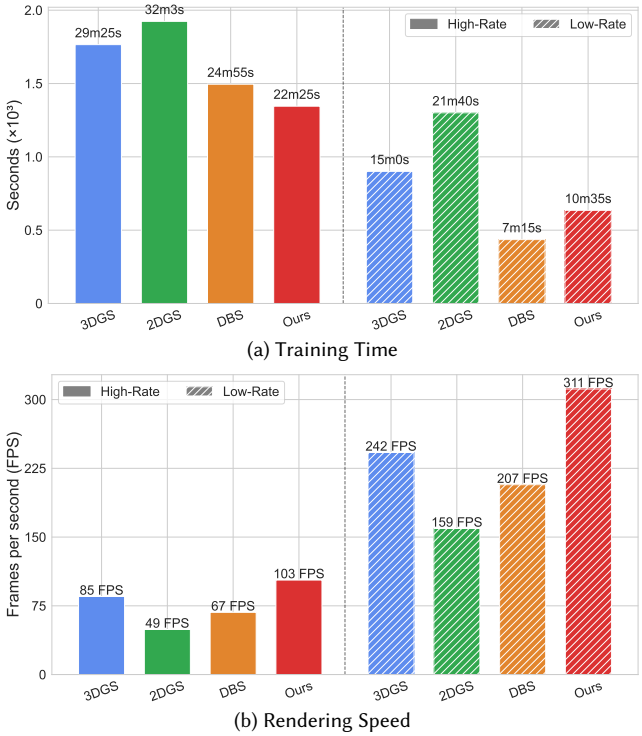


Fig. 5. Training time and rendering speed for methods under investigation. 3DSMoES (ours) achieves superior rendering speed (FPS).

2DGS is less efficient due to the transformation to 2D Gaussians. Overall, these results highlight the speed improvements of our approach, with potential for further efficiency gains by integrating our gated regression into 2DGS.

5.1.4 Plug-and-Play Integration. Table 2 evaluates our SMoE gating as a plug-and-play module across four 3DGS variants: compressed baseline (3DGS-C), Taming3DGS, HAC, and HAC++. We report quality, training and rendering speed, alongside memory for sparse and dense settings.

In the sparse regime, 3DSMoES-C boosts PSNR by 1.07 dB over 3DGS-C while cutting Gaussians by 31%, training time by 28%, and memory by 17%. Rendering speed rises 28%, showing gating's efficiency with fewer, sharper primitives. Integration into Taming3DGS yields consistent quality, 15% faster training, and higher FPS. In HAC and HAC++, SMoE preserves fidelity while speeding training and rendering. In the dense regime, 3DSMoES-C maintains PSNR and SSIM, reduces model size by 20%, and accelerates the pipeline. Other variants sustain high-quality reconstruction with modest efficiency gains. These results demonstrate that 3DSMoES gating generalizes well—improving efficiency and sparsity without quality loss. Its plug-and-play nature validates our claim: SMoE enhances splatting pipelines with minimal change.

5.2 Qualitative Results

Figure 6 illustrates the impressive visual quality gains achieved by 3DSMoES(-C) in the rendered images at low bit rates. Foreground details are comparable across methods. The gain is visible mainly

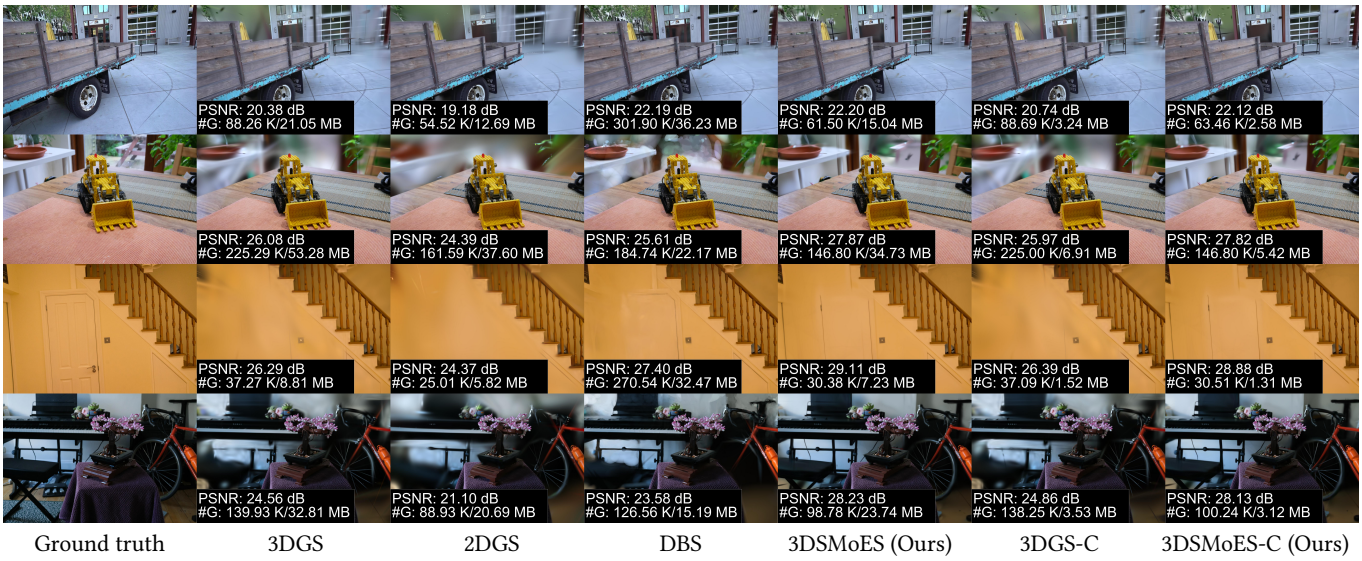


Fig. 6. Our 3DSMoES(-C) compared to 3DGS, 2DGS, DBS, and 3DGS-C in the low-rate domain.

in the background, where our method recovered drastically more structure and content. Contents not seen in 3DGS, 2DGS, and DBS are rendered well by 3DSMoES, while in other parts, edges and other structures are significantly sharpened. Note that this was achieved with significantly reduced numbers of Gaussian (converted to file sizes). We claim that, on average, the low-rate rendered images improve in quality from "useless" to "satisfactory" or even "good" or "very good". Even with a smaller number of Gaussians, 3DSMoES(-C) provides excellent quality in the foreground compared to benchmarks, allowing the optimization algorithm to allocate redundant Gaussians to the background and render these regions with improved quality.

For configurations exceeding 2 million Gaussians, the visual results are shown in Figure 7. At this scale, the visual differences between our method and the state-of-the-art became less pronounced. To emphasize compression capabilities, we highlight specific excerpts where our method still outperforms the others. We observe floaters in 3DGS at high rates eliminated by 3DSMoES/-C), not reflected in metrics or Figure 7. 3DSMoES(-C) excels in both training and rendering efficiency while achieving similar visual quality. Notably, rendering speed is significantly improved. This demonstrates that our method delivers comparable results with a substantially reduced computational cost.

5.3 Ablation Study

We evaluated three variants of our method to analyze the impact of the transmittance term T: the full model 3DSMoES as defined in Eq. 7, a variant without T in the denominator (3DSMoES-T_d, Eq. 8), and one without T in both numerator and denominator (3DSMoES-$T_{n,d}$, Eq. 6). Removing T led to visible ghosting in occluded regions, confirming that the view-dependent transmittance term is critical for correct occlusion handling. The full model effectively eliminated these artifacts through its view-dependent formulation (Figure 8).

Table 3. Comparison of three variants of our method: (1) the full model (3DSMoES), (2) without transmittance term T in the denominator (3DSMoES-T_d), and (3) without T in both numerator and denominator (3DSMoES-$T_{n,d}$). #G is measured in millions.

Method	PSNR [dB]	SSIM	LPIPS	Train. Time [s]	#G
Full model	23.81	0.85	0.17	841	1.580
- Td	22.71	0.82	0.21	1093	1.099
- Tn, d	22.328	0.81	0.22	971	1.195

This confirms that normalization does not degrade view-dependent quality; the correct handling of occlusions and specular effects is ensured by the transmittance term in the numerator. Table 3 reports that 3DSMoES not only achieves superior visual quality but also converges faster, despite requiring more Gaussians.

6 Conclusion

SMoE "Sum of Weighted Soft-Gates" regression is a "plug-and-play" solution that can readily replace the established "Sum of Weighted Gaussians" regression workhorse in 3DGS and related methods. 3DSMoES integrates seamlessly—demanding minimal architectural changes while delivering significant efficiency improvements across diverse pipelines.

In the low-rate regime, 3DSMoES improves quality drastically and significantly reduces Gaussian counts and training times, and boosts rendering efficiencies under standard setups. At high rates, it retains efficiencies—training faster and rendering smoother—even as Gaussian counts remain similar.

3DSMoES regression is expected to provide similar improvements for the plethora of existing 3DGS-related methods, including those using 2D, 3D, and higher dimensional Gaussians.

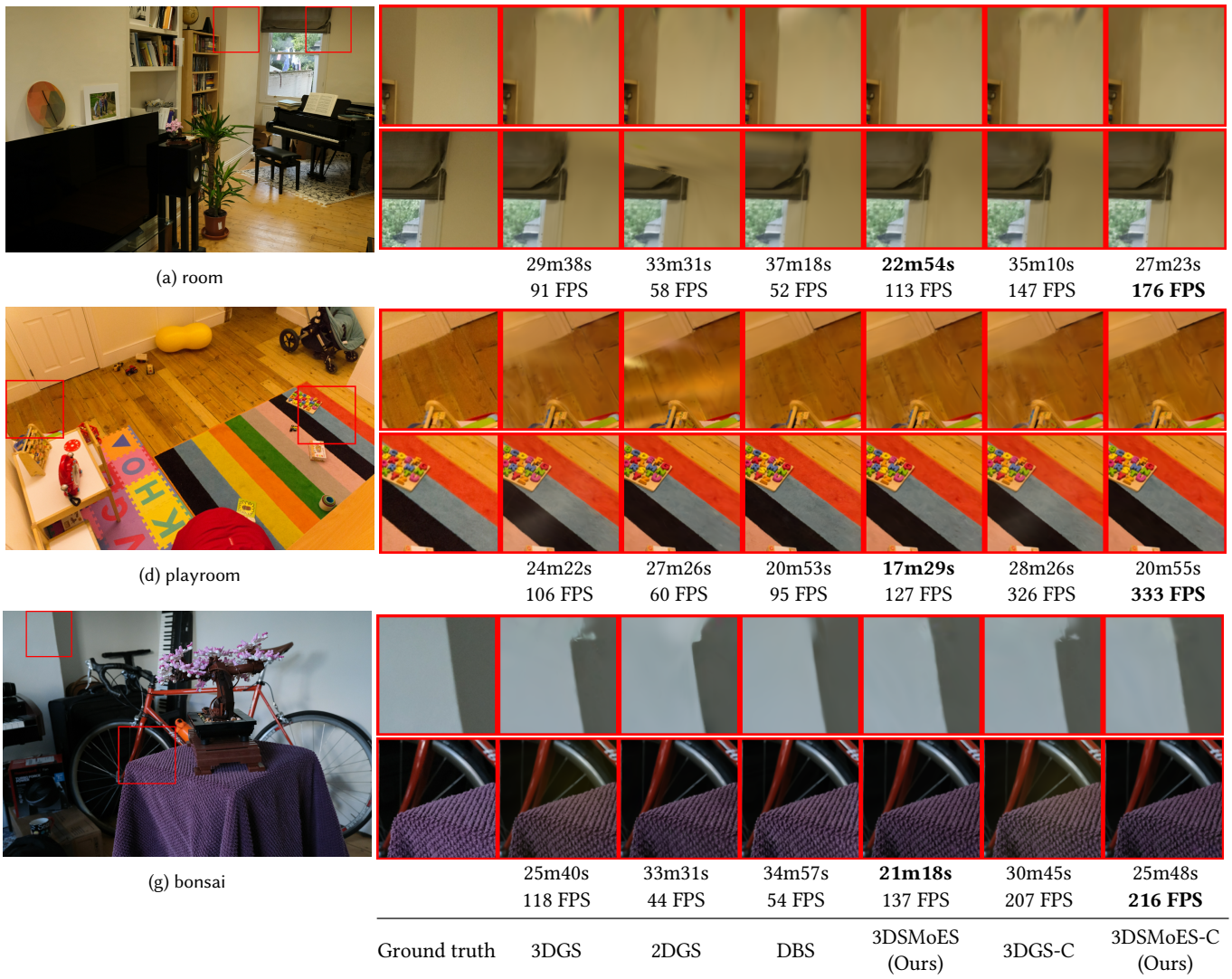


Fig. 7. Gains with our edge-aware 3DSMoES(-C) on Room, Playroom, and Bonsai with large numbers of Gaussians. While objective quality measures appear similar, our method provides visual benefits for the reconstruction of edges. This is achieved with a reduced number of Gaussians - which translates into reduced training times (in seconds, s) and significantly increased rendering speed (FPS).

(a) Ours-without T (b) Ours-without T in the denominator (c) Ours suggested (Full model)

Fig. 8. Gains using the transmittance term T to eliminate "ghosting" effects with our 3DSMoES regression. We compare gating functions (a) without T (3DSMoES-$T_{n,d}$), (b) with T in denominator (3DSMoES-T_n), against (c) T in both nominator and denominator (full model, our suggested variant). The suggested solution effectively eliminates ghosting.

Acknowledgments

This work received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956770. The authors thank the anonymous reviewers for their insightful comments and constructive feedback.

References

Vasileios Avramelos, Ignace Saenen, Ruben Verhack, Glenn Van Wallendael, Peter Lambert, and Thomas Sikora. 2018. Steered mixture-of-experts for light field video coding. In *Applications of Digital Image Processing XLI*, Andrew G. Tescher (Ed.), Vol. 10752. International Society for Optics and Photonics, SPIE, 107520B. doi:10.1117/12.2320563

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5460–5469. doi:10.1109/CVPR52688.2022.00539

Erik Bochinski, Rolf Jongebloed, Michael Tok, and Thomas Sikora. 2018. Regularized Gradient Descent Training of Steered Mixture of Experts for Sparse Image Representation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. 3873–3877. doi:10.1109/ICIP.2018.8451823

Adam Celarek, George Kopanas, George Drettakis, Michael Wimmer, and Bernhard Kerbl. 2025. Does 3D Gaussian Splatting Need Accurate Volumetric Rendering? arXiv:2502.19318 [cs.GR] https://arxiv.org/abs/2502.19318

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. 2024. HAC: Hash-Grid Assisted Context for 3D Gaussian Splatting Compression. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VII* (Milan, Italy). Springer-Verlag, Berlin, Heidelberg, 422–438. doi:10.1007/978-3-031-72667-5_24

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. 2025. HAC++: Towards 100X Compression of 3D Gaussian Splatting. arXiv:2501.12255 [cs.CV] https://arxiv.org/abs/2501.12255

Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. 2024. GaussianPro: 3D Gaussian splatting with progressive propagation. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org, Article 320, 18 pages.

Stavros Diolatzis, Tobias Zirr, Alexander Kuznetsov, Georgios Kopanas, and Anton Kaplanyan. 2024. N-Dimensional Gaussians for Fitting of High Dimensional Functions. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 126, 11 pages. doi:10.1145/3641519.3657502

Guangchi Fang and Bing Wang. 2024. Mini-Splatting: Representing Scenes with a Constrained Number of Gaussians. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXVII* (Milan, Italy). Springer-Verlag, Berlin, Heidelberg, 165–181. doi:10.1007/978-3-031-72980-5_10

Elvira Fleig, Erik Bochinski, and Thomas Sikora. 2023a. Steered mixture-of-experts autoencoder design for real-time image modelling and denoising. In *Real-time Processing of Image, Depth and Video Information 2023*, Matthias F. Carlsohn (Ed.), Vol. 12571. International Society for Optics and Photonics, SPIE, 125710K. doi:10.1117/12.2667082

Elvira Fleig, Jonas Geistert, Erik Bochinski, Rolf Jongebloed, and Thomas Sikora. 2023b. Edge-Aware Autoencoder Design for Real-Time Mixture-of-Experts Image Compression. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5. doi:10.1109/ISCAS46773.2023.10181937

Bengt Fornberg, Tobin A Driscoll, Grady Wright, and Richard Charles. 2002. Observations on the behavior of radial basis function approximations near boundaries. *Computers & Mathematics with Applications* 43, 3-5 (2002), 473–490.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.* 37, 6, Article 257 (Dec. 2018), 15 pages. doi:10.1145/3272127.3275084

Qiqi Hou, Randall Rauwendaal, Zifeng Li, Hoang Le, Farzad Farhadzadeh, Fatih Porikli, Alexei Bourd, and Amir Said. 2025. Sort-free Gaussian Splatting via Weighted Sum Rendering. arXiv:2410.18931 [cs.CV] https://arxiv.org/abs/2410.18931

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery. doi:10.1145/3641519.3657428

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87.

Rolf Jongebloed, Erik Bochinski, Lieven Lange, and Thomas Sikora. 2019. Quantized and Regularized Optimization for Coding Images Using Steered Mixtures-of-Experts. In *2019 Data Compression Conference (DCC)*. 359–368. doi:10.1109/DCC.2019.00044

Rolf Jongebloed, Erik Bochinski, and Thomas Sikora. 2023. Sparse video representation using steered mixture-of-experts with global motion compensation. In *Real-time Processing of Image, Depth and Video Information 2023*, Matthias F. Carlsohn (Ed.), Vol. 12571. International Society for Optics and Photonics, SPIE, 125710J. doi:10.1117/12.2665600

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Sieun Kim, Kyungjin Lee, and Youngki Lee. 2024. Color-cued Efficient Densification Method for 3D Gaussian Splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 775–783. doi:10.1109/CVPRW63382.2024.00082

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.* 36, 4, Article 78 (July 2017), 13 pages. doi:10.1145/3072959.3073599

Lieven Lange, Ruben Verhack, and Thomas Sikora. 2016. Video representation and coding using a sparse steered mixture-of-experts network. In *2016 Picture Coding Symposium (PCS)*. 1–5. doi:10.1109/PCS.2016.7906369

Jiaze Li, Zhengyu Wen, Luo Zhang, Jiangbei Hu, Fei Hou, Zhebin Zhang, and Ying He. 2024. GS-Octree: Octree-based 3D Gaussian Splatting for Robust Object-level 3D Reconstruction Under Strong Lighting. *Computer Graphics Forum* 43, 7 (2024), e15206. doi:10.1111/cgf.15206 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.15206

Yi-Hsin Li, Sebastian Knorr, Mårten Sjöström, and Thomas Sikora. 2025. Adaptive Segmentation-Based Initialization for Steered-Mixture-of-Experts Image Regression. *IEEE Transaction on Multimedia* (2025).

Boning Liu, Yan Zhao, Xiaomeng Jiang, Xingguang Ji, Shigang Wang, Yebin Liu, and Jian Wei. 2024. 5-D Epanechnikov Mixture-of-Experts in Light Field Image Compression. *IEEE Transactions on Image Processing* (2024).

Boning Liu, Yan Zhao, Xiaomeng Jiang, and Shigang Wang. 2019. An image coding approach based on mixture-of-experts regression using epanechnikov kernel. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1807–1811.

Lei Liu, Zhenghao Chen, Wei Jiang, Wei Wang, and Dong Xu. 2025a. HEMGS: A Hybrid Entropy Model for 3D Gaussian Splatting Data Compression. arXiv:2411.18473 [cs.CV] https://arxiv.org/abs/2411.18473

Rong Liu, Dylan Sun, Meida Chen, Yue Wang, and Andrew Feng. 2025b. Deformable Beta Splatting. arXiv:2501.18630 [cs.CV] https://arxiv.org/abs/2501.18630

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. 2024. Taming 3DGS: High-Quality Radiance Fields with Limited Resources. In *SIGGRAPH Asia 2024 Conference Papers (SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 2, 11 pages. doi:10.1145/3680528.3687694

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (Dec. 2021), 99–106. doi:10.1145/3503250

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

Simon Niedermayr, Josef Stumpfegger, and Rudiger Westermann. 2024. Compressed 3D Gaussian Splatting for Accelerated Novel View Synthesis . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 10349–10358. doi:10.1109/CVPR52733.2024.00985

Aytaç Özkan, Elena Stoykova, Thomas Sikora, and Violeta Madjarova. 2024. Denoising OCT images using steered mixture of experts with multi-model inference. In *Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XXVIII*, Rainer A. Leitgeb and Yoshiaki Yasuno (Eds.), Vol. 12830. International Society for Optics and Photonics, SPIE, 1283009. doi:10.1117/12.3000625

Jooyoung Park and Irwin W Sandberg. 1991. Universal approximation using radial-basis-function networks. *Neural computation* 3, 2 (1991), 246–257.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. 2025. Revising Densification in Gaussian Splatting. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland, Cham, 347–362.

Michael Tok, Rolf Jongebloed, Lieven Lange, Erik Bochinski, and Thomas Sikora. 2018. An mse approach for training and coding steered mixtures of experts. In *2018 Picture Coding Symposium (PCS)*. IEEE, 273–277.

Ruben Verhack, Nilesh Madhu, Glenn Van Wallendael, Peter Lambert, and Thomas Sikora. 2018. Steered Mixture-of-Experts Approximation of Spherical Image Data. In *2018 26th European Signal Processing Conference (EUSIPCO)*. 256–260. doi:10.23919/EUSIPCO.2018.8553065

Ruben Verhack, Thomas Sikora, Lieven Lange, Rolf Jongebloed, Glenn Van Wallendael, and Peter Lambert. 2017. Steered mixture-of-experts for light field coding, depth estimation, and processing. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*. 1183–1188. doi:10.1109/ICME.2017.8019442

Ruben Verhack, Thomas Sikora, Lieven Lange, Glenn Van Wallendael, and Peter Lambert. 2016. A universal image coding approach using sparse steered Mixture-of-Experts regression. In *2016 IEEE International Conference on Image Processing (ICIP)*. 2142–2146. doi:10.1109/ICIP.2016.7532737

Ruben Verhack, Thomas Sikora, Glenn Van Wallendael, and Peter Lambert. 2020. Steered Mixture-of-Experts for light field images and video: Representation and coding. *IEEE Transactions on Multimedia* 22, 3 (March 2020), 579–593. doi:10.1109/TMM.2019.2932614

Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. 2019. Differentiable surface splatting for point-based geometry processing. *ACM Trans. Graph.* 38, 6, Article 230 (Nov. 2019), 14 pages. doi:10.1145/3355089.3356513

Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. 2024b. FreGS: 3D Gaussian Splatting with Progressive Frequency Regularization . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 21424–21433. doi:10.1109/CVPR52733.2024.02024

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. 2024a. Pixel-GS: Density Control with Pixel-Aware Gradient for 3D Gaussian Splatting. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XIX* (Milan, Italy). Springer-Verlag, Berlin, Heidelberg, 326–342. doi:10.1007/978-3-031-72655-2_19

Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. 2001a. EWA volume splatting. In *Proceedings of the Conference on Visualization '01* (San Diego, California) *(VIS '01)*. IEEE Computer Society, USA, 29–36.

Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. 2001b. Surface splatting. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 371–378. doi:10.1145/383259.383300

Aytac Özkan, Yi-Hsin Li, and Thomas Sikora. 2023. Steered-Mixture-of-Experts Regression for Image Denoising with Multi-Model Inference. In *2023 31st European Signal Processing Conference (EUSIPCO)*. 546–550. doi:10.23919/EUSIPCO58844.2023.10289994