Sketch and Patch: Efficient 3D Gaussian Representation for Man-Made Scenes

Yuang Shi, Simone Gasparini, Géraldine Morin, Chenggang Yang, Wei Tsang Ooi

Sketch and Patch: Efficient 3D Gaussian Representation for Man-Made Scenes

Yuang Shi[1,3], Simone Gasparini[2,3], Géraldine Morin[2,3], Chenggang Yang[1], Wei Tsang Ooi[1,3]
[1]National University of Singapore, Singapore
[2]IRIT - University of Toulouse, France
[3]IPAL, IRL2955, Singapore

ABSTRACT

3D Gaussian Splatting (3DGS) has emerged as a promising representation for photorealistic rendering of 3D scenes. However, its high storage requirements pose significant challenges for practical applications. We observe that Gaussians exhibit distinct roles and characteristics: some capture high-frequency features like edges and contours, while others represent broader, smoother regions. Based on this observation, we propose a novel hybrid representation that categorizes Gaussians into (i) *Sketch Gaussians*, which define scene boundaries, and (ii) *Patch Gaussians*, which cover smooth regions. Sketch Gaussians are efficiently encoded using parametric models, leveraging their geometric coherence, while Patch Gaussians undergo optimized pruning and retraining to maintain volumetric consistency and storage efficiency. Our comprehensive evaluation across diverse indoor and outdoor scenes demonstrates that this structure-aware approach achieves up to 11.52 % improvement in PSNR, 1.83 % in SSIM, and 19.20 % in LPIPS at equivalent model sizes, and correspondingly, for an indoor scene, our model maintains the visual quality with 7.9 % of the original model size.

CCS CONCEPTS

• **Information systems** → **Multimedia streaming**; • **Computing methodologies** → **Animation**.

KEYWORDS

3D Gaussian Splatting; Efficient 3D Representation; Hybrid Representation; Line Segment Prior; Geometric Encoding

1 INTRODUCTION

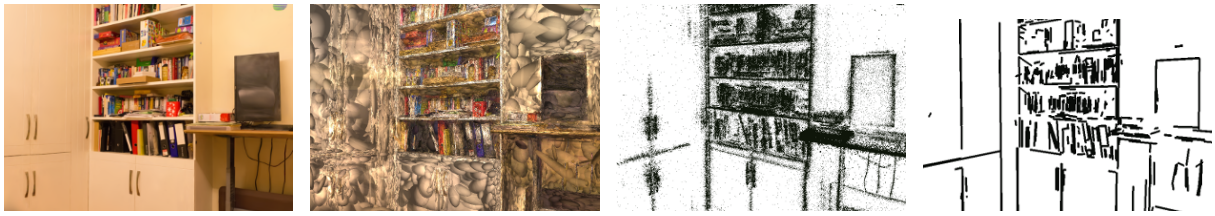
Recently, 3D Gaussian Splatting (3DGS) [9] has emerged as a leading approach for learning more explicit and more efficient point-based 3D representations while keeping the mix between geometry and appearance, leading to a very satisfying visual quality. Compared to point clouds, 3DGS represents 3D scenes with a collection of 3D ellipsoidic Gaussian splats (or just Gaussians for short), which can mitigate the presence of holes with much sparser distribution. By leveraging the properties of Gaussians and taking advantages of GPU, 3DGS has demonstrated its ability to achieve photorealistic and real-time rendering, showing great promise as a foundational representation for immersive multimedia applications. While 3DGS achieves impressive visual fidelity, it still generates vast numbers of Gaussians to capture fine geometric and appearance details.

Common point-based streaming strategies, such as 3D downsampling or tiling, are designed to address storage issues in 3D point clouds by generating scalable and multi-resolution representations [5, 20, 22, 25]. However, without tailored adaptation, these methods are unsuitable for 3DGS due to the unique properties of its Gaussian-based representation [23]. Unlike conventional point clouds, 3DGS employs heterogeneous Gaussians parameterized by ellipsoid shape and view-dependent color. These attributes are optimized iteratively to minimize discrepancies between rendered and ground-truth images, leading to a nonuniform and highly interdependent distribution of Gaussians, as shown in Figure 1(b). This tightly coupled representation optimizes a set of Gaussians, thus significantly limiting the effectiveness of 3D downsampling or partitioning techniques, which often result in substantial degradation of the visual quality [24].

Additionally, 3DGS's adaptive density control mechanism introduces storage inefficiencies. The adaptive density control aims to enhance fidelity by cloning Gaussians in under-reconstructed regions and splitting those in over-reconstructed areas, based on 2D positional gradients. The reliance on 2D gradients often overestimates the need for densification in high-frequency and boundary-defining areas, such as edges and contours. For instance, Figure 1(c) depicts the distribution of ellipsoid centers in the scene. As illustrated, this process results in dense Gaussian clusters along structural boundaries, like architectural edges, furniture boundaries, and other manufactured object contours, leading to redundant representation and increased computational overhead [12, 14].

Based on these characteristics of 3DGS, we propose to categorize Gaussians into two distinct roles: *Sketch Gaussians* and *Patch Gaussians*, drawing inspiration from traditional artistic techniques. Like how artists first sketch outlines before filling in broader areas with color, Sketch Gaussians capture boundary-defining features such as edges and contours, serving as the semantic scaffolding of the scene. These Sketch Gaussians exhibit strong coherence along linear structures, particularly in man-made scenes where edges and contours often follow consistent geometric patterns. In contrast, Patch Gaussians, analogous to broader brush strokes that add volume and depth to a painting, provide volumetric coverage for smoother and broader regions.

We further propose leveraging the unique structural properties of edges and contours, which typically represent high-frequency and boundary-defining features that are inherently linear or curvilinear. Specifically, instead of densely populating these areas with Sketch Gaussians explicitly, we propose encoding Sketch Gaussians with the prior of 3D line segments. By abstracting the coherent Sketch Gaussians into compact parametric models, such as polynomials or splines, we preserve both the visual quality and semantic information with data of significantly smaller size. For smoother and low-frequency regions, the Patch Gaussians are stored with their Gaussian parameters, given the naturally sparse distribution of Gaussians in these areas. This dual-role categorization forms the



(a) 3DGS Model. (b) Ellipsoids of 3DGS. (c) Ellipsoid centers. (d) 3D lines [8].

Figure 1: Illustration of the characteristics of 3DGS. Note the correlation between the density of 3DGS and the 3D lines.

basis of our hybrid Gaussian representation, designed to enhance scalability, structure awareness, and storage efficiency for 3DGS.

While our approach works for any 3DGS scenes, it is especially effective in man-made (as opposed to natural) scenes and objects, with well-defined straight edges and planar surfaces that naturally align with our Sketch and Patch categorization. The prevalence of regular geometric primitives in architectural elements, furniture, and manufactured objects enables the effective separation of boundary-defining features from broader regions, allowing for efficient encoding through our hybrid representation strategy.

Our contributions can be summarized as follows: (i) A novel dual-role categorization of 3DGS into Sketch and Patch Gaussians, reflecting their distinct functions in scene representation; (ii) A hybrid Gaussian representation that encodes high-frequency and boundary-defining features with structural 3D line prior and preserves volumetric detail for low-frequency regions using 3D Gaussians; (iii) Relying on the intrinsic structure of man-made scenes, our approach achieves consistent improvements in visual quality by up to 11.52 % in PSNR, 1.83 % in SSIM, and 19.20 % in LPIPS at the equivalent model size and correspondingly, a proposed hybrid model of size around 7.9 % of the original model can maintain the visual quality of an indoor scene. The proposed approach achieves substantial reduction in storage requirements while maintains high fidelity rendering, enabling deployment of complex 3D environments on heterogeneous mixed or virtual reality devices.

2 BACKGROUND AND RELATED WORK

2.1 3D Gaussian Splatting

The 3DGS represents a scene using a collection of Gaussians optimized to fit a set of input images. Each Gaussian is parameterized by a center position, a 3D covariance matrix, an opacity, and a color which is described by a set of Spherical Harmonic (SH) coefficients. To construct a collection of 3D Gaussians that accurately represent the scenes, 3DGS introduces an optimization method using differentiable rendering to estimate the parameters of the 3D Gaussians, fitting a set of calibrated input images of the given scene. Density of Gaussians is controlled by a positional gradient threshold, balancing between the number of parameters and the fitting accuracy. Starting from a sparse point set generated by Structure from Motion (SfM), the model iteratively removes the Gaussians whose opacities are below a threshold, and densifies the Gaussians in regions with high positional gradients. In areas lacking detail, low-density Gaussians are duplicated and shifted along the gradient to create new geometry. In areas of excessive overlap, large Gaussians are split into smaller ones to achieve finer granularity. This adaptive density control refines the scene representation while managing the number of Gaussians to balance model complexity and quality.

2.2 Compact 3DGS Representation

The model size in 3DGS is primarily affected by two factors. First, complex, high-frequency areas require sophisticated parameters to represent. Second, as discussed in Section 2.1, the model generation process introduces additional Gaussian splats based on density thresholds during scene fitting. The combination of these factors — the parameter complexity per Gaussian and the total number of Gaussian splats — leads to significant storage overhead.

Recent research has approached these challenges from multiple angles. To address parameter complexity, several methods have been proposed: region-based vector quantization [16, 17], K-means codebooks [16], and view-direction exclusion [2]. Parallel efforts have focused on managing Gaussian density through various Level-of-Detail (LOD) techniques [11, 15, 18, 21]. These methods, however, primarily focus on either parameter compression or LOD structuring without considering the inherent roles of different Gaussians in scene representation. Our approach fundamentally differs by recognizing and leveraging the distinct characteristics of Gaussians in man-made scenes, categorizing them into Sketch and Patch components based on their geometric significance. This structure-aware strategy enables more efficient representation by applying appropriate encoding techniques to each category. Meanwhile, our Sketch and Patch categorization is complementary to these existing compression and LOD techniques. This compatibility ensures that our method can serve as a foundation for further optimizations while providing its unique benefits in storage efficiency.

3 PROPOSED METHODOLOGY

The proposed method consists of the following steps (see Figure 2). First, we extract 3D line segments that abstract edges and contours to identify Sketch Gaussians. Based on the prior of line segments, we identify the Sketch Gaussians (§3.1). The Sketch Gaussians are then encoded using parametric models on a given 3D line to model their attributes compactly, significantly reducing storage requirements. For smoother regions, represented by Patch Gaussians, we apply pruning and retraining to sparsify their distribution while maintaining visual fidelity (§3.2). Finally, the Sketch and Patch Gaussians are integrated into a unified hybrid representation, enabling efficient storage, transmission, and rendering of 3D scenes.

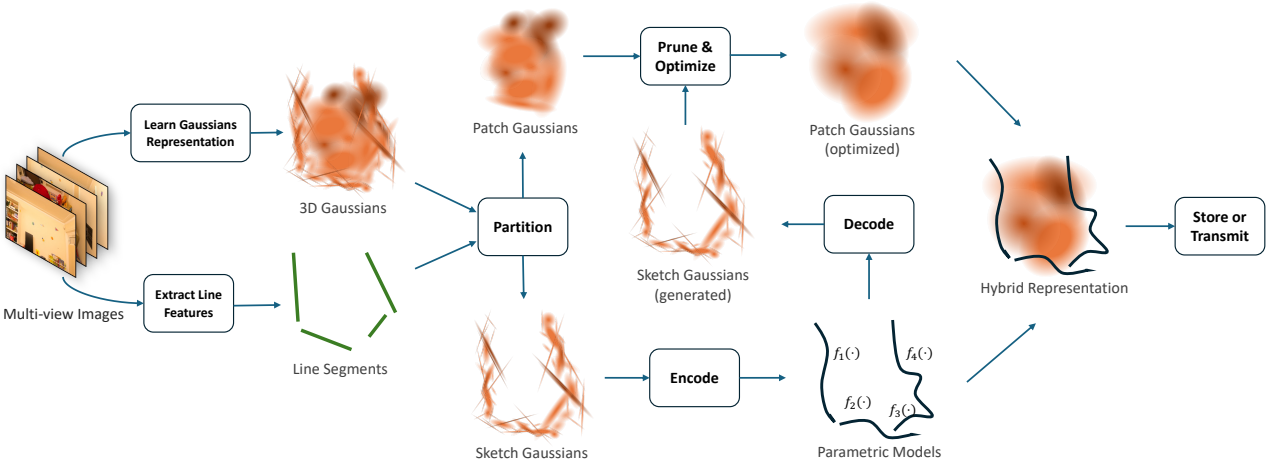


Figure 2: The framework of constructing our compact hybrid representation.

3.1 Partitioning Sketch and Patch Gaussians

Extraction of 3D Line Segments. Geometry-guided line reconstruction methods extract edges, contours, or high-frequency regions corresponding to high Gaussians density (see Figure 1 (c) and (d)). Here, we choose the line reconstruction method Line3D++ [7]. It takes as input the images and camera poses and provides as output the 3D lines by abstracting object boundaries into discrete 2D line segments [3, 26] and back-projecting them into 3D space.

Identification and modeling of Sketch Gaussians. Identifying Sketch Gaussians is a critical step in constructing a structure-aware representation. These Gaussian splats, concentrated along edges and contours, play an important role in preserving the semantic and visual essence of the scene. However, their identification is nontrivial due to the densely coupled and anisotropic nature of Gaussian splats in 3DGS. To address this, we propose a two-step robust process for extracting Sketch Gaussians based on structural priors provided by 2D line segments: (i) radius search and (ii) random sample consensus (RANSAC) filtering [4].

Sketch Gaussians along a 3D line. In the first step, we identify a rough set of Gaussian splats associated with a line segment extracted. Let $\mathbf{L}_i(t)$ denote a 3D line segment parameterized as: $\mathbf{L}_i(t) = (1-t)\mathbf{p}_{\text{start}} + t\mathbf{p}_{\text{end}}, t \in [0,1]$, where $\mathbf{p}_{\text{start}}$ and $\mathbf{p}_{\text{end}}$ are the endpoints of the segment. We include a Gaussian splat $\mathcal{G}_j$ for line $\mathbf{L}_i$ in the candidate set $\mathcal{S}_i$, if its center $\mu_j \in \mathbb{R}^3$ lies within a radius r from the line: $d(\mu_j, \mathbf{L}_i) = \min_{t\in[0,1]} \|\mu_j - \mathbf{L}_i(t)\| \leq r$. The three remaining steps further ensure that the Gaussian splats selected on a line also have coherent attributes.

Robust selection of Sketch Gaussians relative to the 3D line. After the initial radius search identifies a rough set of candidate Gaussians, the next step is to refine this set using RANSAC, to ensure coherent attribute values among the Sketch Gaussians on a given line. For edges and contours, which are inherently linear or curvilinear features in 3D scenes, RANSAC enables the extraction of Gaussians that align well with the line segments while discarding outliers due to noise and artifacts.

Robust selection of Sketch Gaussians relative to attributes. Given a set of Sketch Gaussians associated with a 3D line, to account for the heterogeneous nature of Gaussian attributes, we perform RANSAC separately for each attribute (opacity, color, scaling, and rotation), thereby leveraging the unique characteristics of each. The independent RANSAC results for each attribute are combined by intersecting the sets of inliers across all attributes. This ensures that only Gaussians consistent across multiple attributes are retained, improving the robustness of the model to be estimated.

Robust selection of Sketch Gaussians relative to the model. For each attribute, we fit a polynomial regression model $f(\mu)$ to the robustly selected Sketch Gaussians along a given line (see Section 3.2). We then evaluate the residual error for each Gaussian as the squared deviation between its attribute value and the model's prediction. Gaussians are classified as inliers if their residual error falls below a dynamically determined threshold $\epsilon = \eta \cdot \text{MAD}$, where MAD denotes the Median Absolute Deviation, and η is a hyperparameter that controls the sensitivity of RANSAC to deviations. Through this process, we identify a refined, robust and coherent set of Sketch Gaussians for each line. This set of Gaussians is represented efficiently through a compact polynomial model of the attributes.

Generated Sketch Gaussians filtering. Even though the two previous steps filter outliers within Sketch Gaussians along 3D lines, we observe that the scaling model tends to introduce bias, which can result in poorly modeled long and thin Gaussians. We introduce a further post-processing phase to address such critical issue. Specifically, we decode the Sketch Gaussians and use the Interquartile Range method to identify outlier Gaussians with extreme scaling values that deviate from the expected distribution. We then filter out the Gaussians among them that are not aligned with the direction of the line segment. This post-processing ensures that only visually consistent Gaussians are retained for the later encoding, improving the accuracy of the Sketch Gaussian representation and ensuring better alignment with the underlying scene structure.

Gaussians that are filtered out during this process are not discarded but instead reclassified as Patch Gaussians, ensuring that no

geometric information is lost while maintaining the most appropriate representation for each Gaussian based on its characteristics.

3.2 Gaussian Encoding and Optimization

Sketch Gaussian Encoding. After identifying the Sketch Gaussians, we aim to encode their attributes efficiently while preserving visual quality. Thanks to the coherence of the set of Sketch Gaussians along a 3D line, strengthened by the different robust steps detailed in the previous section, we can compactly code the Sketch Gaussians. We choose polynomial regression (PR) to model the Gaussians' attributes as it is a straightforward yet effective method for encoding complex features in a compact form. While more sophisticated methods could be used for encoding, PR combines simplicity with efficiency, which is crucial for real-time performance and memory consumption.

For each line corresponding to a set of Sketch Gaussians, we train four PR models to encode their key attributes: opacity, color, scaling, and rotation. This separation of attributes allows for more accurate and flexible decoding and each model captures the relationships between the attribute and the line segments. The degree is chosen by grid search in the range of $[1, 10]$. Once the models are trained, the attributes of the Sketch Gaussians are encoded as a parametric polynomial, significantly reducing the storage requirements.

By encoding the Sketch Gaussians with PR models, we efficiently represent the boundary-defining features while maintaining visual fidelity. Our experiments (Section 4), and in particular Table 1, will stress out the efficiency of the Sketch Gaussians encoding.

Patch Gaussian Pruning and Optimization. The core idea behind Patch Gaussian optimization is to reduce the number of Gaussians in smoother regions of the scene, where low-frequency variations are typically captured. We leverage retraining, which allows for pruning of Patch Gaussians while maintaining visual quality.

The process starts with the encoding and decoding of Sketch Gaussians. Once the Sketch Gaussians are (compactly) encoded, we decode them and fix the decoded version for subsequent operations. The set of decoded Sketch Gaussians remains unchanged during the pruning process. Note that the decoded Sketch Gaussians are used during retraining, rather than the original Sketch Gaussians, allowing the Patch Gaussians to compensate for errors introduced by the encoding-decoding process of the Sketch Gaussians. The retraining focuses on optimizing the Patch Gaussians, which are first pruned, by randomly and uniformly removing some of them to improve the compactness of the model, and then retrained to improve the visual quality of the results. This retraining process ensures that the Patch Gaussians are optimized to align with the visual structure of the scene w.r.t. the given set of Sketch Gaussians.

Through pruning and retraining, we achieve a more compact 3DGS representation where each Gaussian type serves its designated role: Sketch Gaussians efficiently capture linear features and boundaries, while optimized Patch Gaussians provide comprehensive coverage of smoother regions. The result is a hybrid representation that efficiently allocates storage resources according to the geometric characteristics of different scene regions, *i.e.* compact parametric models for boundary features and optimized sparse distribution for volumetric regions.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

Dataset and Metrics. We evaluated our method across four representative scenes from three distinct datasets: *Playroom* and *Drjohnson* from the Deep Blending dataset [6], *Room* from the Mip-NeRF360 dataset [1], and *Truck* from the Tanks&Temples dataset [13]. These scenes were specially selected to encompass a diverse range of geometric complexities, including bounded indoor environments and expansive unbounded outdoor scenes. This selection allows for an assessment of our method's adaptability across varying scene morphologies characteristic of man-made environments. We quantitatively assessed the visual fidelity of our representations using three complementary metrics: PSNR, SSIM [27], and LPIPS [29].

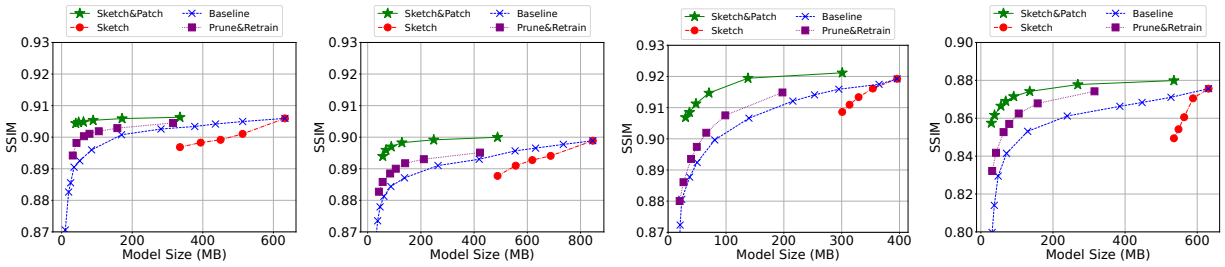
Implementation. Our implementation is based on the official 3D Gaussian Splatting codebase [10]. We initially trained 3DGS models for each scene using the default hyperparameters on an NVIDIA H100 GPU. Subsequently, we employed Line3D++ using the official release code [8] to extract 3D line segments. After distinguishing the Sketch and Patch Gaussians with the line prior, the polynomial regression (PR) is utilized to model Sketch Gaussians while Patch Gaussians are pruned and retrained for further compact representation. The PR model's degree was determined through a comprehensive grid search. While our method introduces additional computational overhead primarily from the PR modeling of Sketch Gaussians, this phase is inherently parallelizable across line segments on either CPU or GPU. We maintained consistent hyperparameters across all training stages.

Comparison Methods. We implemented three approaches: (i) ***Baseline***. We train 3DGS for each scene with different densification thresholds, producing multiple 3DGS models with different sizes. Standard 3DGS relies primarily on 2D positional gradients to guide the densification of Gaussians, which frequently results in significant redundancy, particularly in boundary-defining regions. Serving as the baseline, this approach globally adapts the densification gradient by changing the threshold value. (ii) ***Sketch***. The method focuses exclusively on the encoding of Sketch Gaussians, deliberately omitting the pruning and retraining of Patch Gaussians. To quantify the storage efficiency and visual quality trade-offs inherent in compact boundary feature representation, we progressively reduce the number of line segments used for Sketch Gaussian identification and modeling and measure the visual quality and model size. Note that this method is the only one not to retrain the model. (iii) ***Prune&Retrain***. This method applies a uniform pruning and retraining strategy across all Gaussians, without distinguishing between Sketch and Patch categories. This method serves as an ablation study to demonstrate the importance of recognizing and processing Gaussians differently based on their structural roles.

4.2 Results Analysis

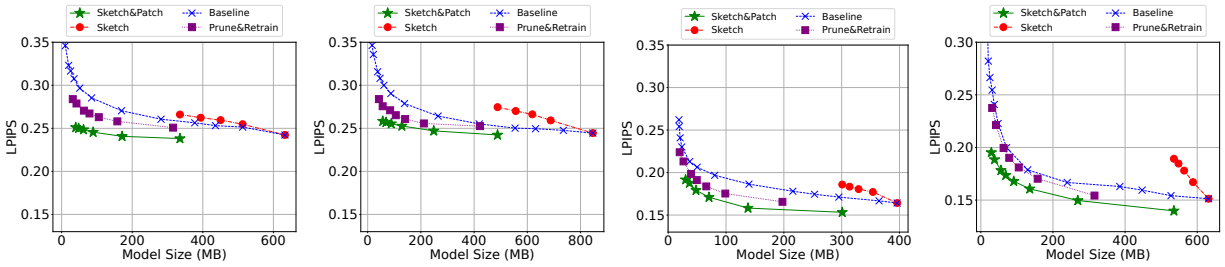
Figures 3 and 4 plot the visual quality of the four scenes versus the model size for four methods [1]. To generate rate-distortion (R-D) curves, we varied parameters for each method. For the *Baseline*, we adjusted the densification gradient threshold from 0.002 (default) to 0.2 with an interval of 0.0005. For the *Prune&Retrain*,

[1] Due to page limit, figures on PSNR are omitted.



(a) Playroom. (b) Drjohnson. (c) Room. (d) Truck.

Figure 3: R-D curves for SSIM: (a) Playroom, (b) Drjohnson, (c) Room, and (d) Truck.



(a) Playroom. (b) Drjohnson. (c) Room. (d) Truck.

Figure 4: R-D curves for LPIPS: (a) Playroom, (b) Drjohnson, (c) Room, and (d) Truck.

we created multiple models by reducing the number of Gaussian splats by factors of $\{2, 4, 6, 8, 10, 15, 20\}$ through uniform pruning, followed by retraining. For the *Sketch* method, we varied line segments used for Sketch Gaussian identification by selecting the top $\{25\,\%, 50\,\%, 75\,\%, 100\,\%\}$ longest lines. As for our proposed method (denoted as *Sketch&Patch*), we first utilized all line segments for Sketch Gaussian identification and modeling, then reduced Patch Gaussians by factors of $\{2, 4, 6, 8, 10, 15, 20\}$ through uniform pruning and retraining. We can make the following observations.

Visual Quality. First, our proposed method *Sketch&Patch* demonstrates superior performance across all evaluated metrics compared to alternative approaches at the same model size. For instance, in the Playroom scene, we observe enhancements by up to 11.52 % in PSNR, 1.83 % in SSIM, and 19.20 % in LPIPS. Similar patterns of improvement are evident in other scenes, with Drjohnson showing gains of 9.39 % in PSNR, 1.83 % in SSIM, and 16.15 % in LPIPS at most. Our method on Room scene achieves improvements by up to 11.50 % in PSNR, 2.99 % in SSIM, and 16.74 % in LPIPS, while the Truck scene demonstrates enhancements of 18.26 % in PSNR, 8.93 % in SSIM, and 26.71 % in LPIPS. These substantial improvements can be attributed to our method's strategic approach to Gaussian representation. By differentiating between Sketch and Patch Gaussians, our method preserves critical geometric features while efficiently representing broader scene regions. The particularly high performance on perceptual metrics (LPIPS) suggests that our approach effectively maintains the visually significant features contributing to human perception of scene quality.

Storage Efficiency. Second, a key advantage of our method lies in its significant reduction in storage requirements while maintaining visual fidelity. When comparing model sizes at equivalent SSIM values, for instance, our method achieves remarkable efficiency: requiring only 7.99 % of the original model size for Playroom, 10.81 % for Drjohnson, 19.34 % for Room, and 11.12 % for Truck, while maintaining the same visual quality. The achieved compression rates can be attributed to two key factors: the efficient encoding of high-frequency features through parametric models and the optimized representation of smooth regions through carefully pruned Patch Gaussians. This dual approach enables our method to maintain high visual quality while significantly reducing storage.

We further present Table 1, which provides a detailed breakdown of our hybrid representation's storage requirements, comparing Sketch and Patch Gaussians components with the baseline Vanilla 3DGS at an equivalent visual quality level in terms of SSIM (0.88 on Truck, 0.92 on Room, 0.90 on Drjohnson, and 0.91 on Playroom). The results reveal that our parametric encoding of Sketch Gaussians achieves remarkable efficiency, requiring no more than 9.62 MB of storage across all scenes. The compact representation of Sketch Gaussians is noteworthy, as these elements typically require dense sampling to maintain edge and contour fidelity. Our method's ability to encode these features efficiently while preserving their visual importance demonstrates the effectiveness of leveraging geometric coherence through parametric models, validating the effectiveness of our line segment-based parametric approach.

Qualitative Comparison. To illustrate the performance of the proposed method, we also present the visual comparison with the

Figure 5: Rendered images of 3DGS on *Playroom* generated from four methods.

Table 1: Model size (MB) at equivalent visual quality level.

Model	Playroom	Drjohnson	Room	Truck
Vanilla 3DGS	631.44	844.48	395.16	630.23
Sketch-Patch GS	54.25	136.98	73.50	138.05
Patch Gaussians	328.35	481.35	268.81	535.70
+ Pruned and Retrained	46.90	127.36	69.93	135.28
Sketch Gaussians	303.09	363.13	126.45	94.53
+ Encoded with PR	7.35	9.62	3.57	2.77

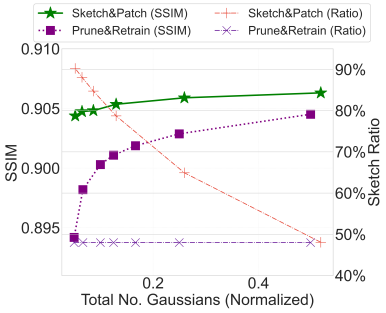


Figure 6: SSIM vs. ratio of Sketch Gaussians of *Playroom*.

other methods in Figure 5 for qualitative analysis. For fair comparison, we selected models representing each method's maximum achievable compression while maintaining acceptable visual quality, corresponding to the leftmost points in the R-D curves.

It can be noted that our method excels particularly in preserving geometric details and structural integrity. Compared to *Baseline* and *Prune&Retrain* method, the Sketch Gaussian encoding effectively preserves sharp edges and boundary features, especially in challenging areas such as bookshelves with stacked books and clearly readable book titles. This preservation of fine text and sharp edges is particularly impressive given the substantial reduction in storage requirements. Simultaneously, the optimized Patch Gaussians successfully preserve surface continuity in broader regions, eliminating redundant information while maintaining visual consistency.

Ablation Effect of Sketch and Patch. To better illustrate the importance of differential treatment of Gaussian categories, we show the visualization of the rendering performance vs. the ratio of Sketch Gaussians of the scene *Playroom* constructed by method *Sketch&Patch* in Figure 6. The Sketch ratio, defined as the number of Sketch Gaussians over the total number of Gaussian splats, provides valuable insight into the effectiveness of our method. As can be observed, in our *Sketch&Patch* method, the Sketch ratio increases as the model size decreases, ranging from approximately 48 % at the largest model size to 90 % at the most compressed state. This trend occurs because we fix the Sketch Gaussians while progressively pruning and optimizing the Patch Gaussians, ensuring the preservation of critical boundary features even under extreme compression. In contrast, the *Prune&Retrain* method maintains a constant Sketch ratio of approximately 48 % across all model sizes, as it uniformly prunes and optimizes all Gaussians without distinguishing their roles. This uniform treatment leads to suboptimal resource allocation, particularly evident in their SSIM values. Even when our method has a high Sketch ratio at small model sizes, it achieves better visual quality compared to *Prune&Retrain*. This superior performance demonstrates that preserving Sketch Gaussians while selectively optimizing Patch Gaussians is more effective than uniform pruning, especially under stringent storage constraints.

5 CONCLUSION

This paper introduces a novel perspective on 3D Gaussian scene representation by recognizing and leveraging the distinct roles of different Gaussian types in man-made environments, significantly reducing storage requirements while maintaining visual quality. Our current implementation focuses on static scenes, and extending this approach to dynamic scenarios presents interesting challenges. Also, the effectiveness of our line segment-based encoding may vary in natural scenes where boundaries are less geometric, suggesting the exploration of alternative parametric models for organic shapes [28]. Third, current evaluation methods for 3DGS focus on projection-based quality assessment with 2D metrics, neglecting the crucial aspects of immersive experience [19]. Sophisticated quality evaluation techniques, *e.g.* subjective evaluation, are required to assess the visual quality perceived by humans.

REFERENCES

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5460–5469. IEEE, 2022.

[2] Yuanhao Cai, Yixun Liang, Jiahao Wang, Angtian Wang, Yulun Zhang, Xiaokang Yang, Zongwei Zhou, and Alan L. Yuille. Radiative Gaussian splatting for efficient X-ray novel view synthesis. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part I*, volume 15059 of *Lecture Notes in Computer Science*, pages 283–299. Springer, 2024.

[3] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[4] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[5] Bo Han, Yu Liu, and Feng Qian. ViVo: visibility-aware mobile volumetric video streaming. In *MobiCom '20: The 26th Annual International Conference on Mobile Computing and Networking, London, United Kingdom, September 21-25, 2020*, pages 11:1–11:13. ACM, 2020.

[6] 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):1–15, 2018.

[7] Manuel Hofer, Michael Maurer, and Horst Bischof. Efficient 3D scene abstraction using line segments. *Computer Vision and Image Understanding*, 157:167–178, 2017.

[8] Manuel Hofer, Michael Maurer, and Horst Bischof. Line3D++. https://github.com/manhofer/Line3Dpp, 2020.

[9] 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.

[10] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. https://github.com/graphdeco-inria/gaussian-splatting, 2023.

[11] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4):62:1–62:15, 2024.

[12] Sieun Kim, Kyungjin Lee, and Youngki Lee. Color-cued efficient densification method for 3D Gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*, pages 775–783. IEEE, 2024.

[13] 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.

[14] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. GeoGaussian: Geometry-aware Gaussian splatting for scene rendering. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXV*, volume 15093 of *Lecture Notes in Computer Science*, pages 441–457. Springer, 2024.

[15] 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/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 20654–20664. IEEE, 2024.

[16] K. L. Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compact3D: Compressing Gaussian splat radiance field models with vector quantization. *CoRR*, abs/2311.18159, 2023.

[17] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3D Gaussian splatting for accelerated novel view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 10349–10358. IEEE, 2024.

[18] 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. *CoRR*, abs/2403.17898, 2024.

[19] Yuang Shi. 3D Gaussian-based immersive media streaming in networked extended reality. In *Proceedings of the 16th Conference on ACM Multimedia Systems, MMSys 2025, March 31-April 4, 2025, Stellenbosch, South Africa*, 2025.

[20] Yuang Shi, Bennett Clement, and Wei Tsang Ooi. QV4: QoE-based viewpoint-aware V-PCC-encoded volumetric video streaming. In *Proceedings of the 15th ACM Multimedia Systems Conference, MMSys 2024, Bari, Italy, April 15-18, 2024*, pages 144–154. ACM, 2024.

[21] Yuang Shi, Géraldine Morin, Simone Gasparini, and Wei Tsang Ooi. LapisGS: Layered Progressive 3D Gaussian Splatting for Adaptive Streaming. In *The 12th International Conference on 3D Vision, 3DV 2025, Singapore, March 25-28, 2025*. IEEE, 2025.

[22] Yuang Shi, Pranav Venkatram, Yifan Ding, and Wei Tsang Ooi. Enabling low bit-rate MPEG V-PCC-encoded volumetric video streaming with 3D sub-sampling. In *Proceedings of the 14th Conference on ACM Multimedia Systems, MMSys 2023, Vancouver, BC, Canada, June 7-10, 2023*, pages 108–118. ACM, 2023.

[23] Yuan-Chun Sun, Yuang Shi, Cheng-Tse Lee, Mufeng Zhu, Wei Tsang Ooi, Yao Liu, Chun-Ying Huang, and Cheng-Hsin Hsu. LTS: A DASH streaming system for dynamic multi-layer 3D Gaussian splatting scenes. In *Proceedings of the 16th ACM Multimedia Systems Conference, MMSys 2025, Stellenbosch, South Africa, March 31-April 4, 2025*. ACM, 2025.

[24] Yuan-Chun Sun, Yuang Shi, Wei Tsang Ooi, Chun-Ying Huang, and Cheng-Hsin Hsu. Multi-frame bitrate allocation of dynamic 3D Gaussian splatting streaming over dynamic networks. In *Proceedings of the 2024 SIGCOMM Workshop on Emerging Multimedia Systems, EMS 2024, Sydney, NSW, Australia, August 4-8, 2024*, pages 1–7. ACM, 2024.

[25] Irene Viola and Pablo César. Volumetric video streaming: Current approaches and implementations. *Immersive Video Technologies*, pages 425–443, 2023.

[26] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[27] 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.

[28] Chenggang Yang and Yuang Shi. LineGS: 3D line segment representation on 3D Gaussian splatting. *CoRR*, abs/2412.00477, 2024.

[29] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. IEEE, 2018.