Plug-and-Play Optimization for 3D Gaussian Splatting Compression: Distribution Regularization, Probabilistic Pruning and Detail Compensation

Tian Bai[1,2*†], **Zheng Qiu**[1,2†], **Haojie Chen**[1,2], **Ziyang Dai**[1,2]

[1]Suzhou Institute for Advanced Research, University of Science and Technology of China
[2]School of Software Engineering, University of Science and Technology of China
baitian@ustc.edu.cn, {qiuzheng, haojiechen, daiziyangcn}@mail.ustc.edu.cn

Abstract

Recent advancements in 3D Gaussian Splatting (3DGS) have demonstrated remarkable rendering quality, However, their substantial computational demands hinder practical deployment on resource-constrained devices. We propose a novel plug-and-play structured compression framework that significantly reduces computational overhead while maintaining rendering fidelity. We first discover that the statistical distribution of anchor vectors is decoupled from rendering quality. Based on this finding, we propose a distribution regularization method that enforces alignment to standard Gaussian distribution through KL divergence while optimizing Gaussian radius, significantly improving entropy coding efficiency. Second, we innovatively introduce an opacity-based probabilistic pruning mechanism that transforms pruning into an opacity optimization problem, achieving intelligent scene sparsification while allowing flexible adjustment according to hardware resources. Finally, we design a lightweight high-frequency compensation network that regards the high-frequency loss caused by over-compression as a residual and effectively recovers the high-frequency details lost during the compression process through residual learning. All modules are plug-and-play and can be seamlessly integrated into mainstream structured 3DGS frameworks. Extensive experiments on Synthetic-NeRF, Tanks&Temples, Mip-NeRF360 and DeepBlending datasets demonstrate that our method significantly reduces size by over 80x compared to vanilla 3DGS while simultaneously improving fidelity. Furthermore, it achieves a better size reduction and a 20% improvement in entropy encoding efficiency when compared to HAC, while meeting the requirements for real-time rendering.

Code — https://github.com/fl0a1e/GSComp-ppo.git

1 Introduction

The rapid advancements in computer vision and graphics have propelled 3D scene reconstruction to a pivotal role in enabling applications such as virtual reality and augmented reality (Kalkofen, Mendez, and Schmalstieg 2009; Patney et al. 2016). Within this domain, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has emerged as a prominent explicit rendering technique. Distinct from implicit methods like NeRF (Mildenhall et al. 2020), 3DGS offers not only real-time rendering capabilities but also superior scene editability. Nevertheless, its substantial memory and storage requirements present a formidable challenge for deployment in resource-constrained environments (Navaneet et al. 2024; Yan et al. 2024; Bagdasarian et al. 2025; Chen et al. 2023).

To mitigate storage constraints, current efforts largely concentrate on the compression of 3DGS representations, broadly classified into non-structured and structured approaches (Bagdasarian et al. 2025; Ali et al. 2025). Non-structured methods aim to reduce redundancy via techniques

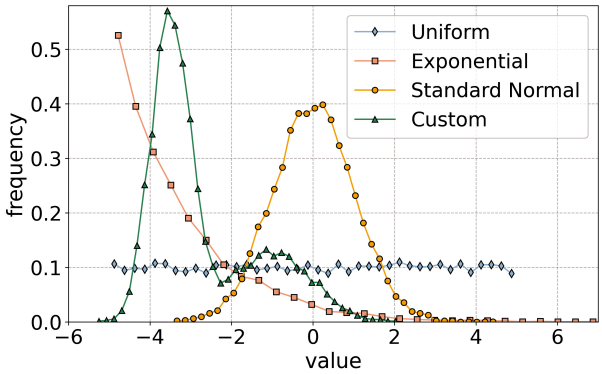

Figure 1: Different forms of feature distribution. We achieve different distribution patterns of Gaussian features by applying supervision on the data shape.

Distribution	PSNR↑	SSIM↑	LPIPS↓
Uniform	35.15	0.977	0.018
Exponential	35.19	0.979	0.019
Standard Normal	35.16	0.978	0.018
Custom	35.21	0.979	0.019

Table 1: Due to the powerful fitting capability of MLP, the influence of different feature distributions on the final outcome is small. We conducted tests on the lego scene of the Synthetic-NeRF dataset (Mildenhall et al. 2020).

*Corresponding author
†These authors contributed equally.

such as pruning, quantization, and entropy encoding, without fundamentally altering the Gaussian distribution's inherent structure. Conversely, structured techniques exploit the spatial and semantic relationships among Gaussians to attain superior compression ratios, a prime example being anchor-based structured compression (Lu et al. 2024; Chen et al. 2024). It is noteworthy that, among current 3DGS compression technologies, anchor-based structured methods exhibit the optimal balance between rendering quality and compression efficiency (Bagdasarian et al. 2025). By establishing spatial hierarchical relationships among Gaussian distributions, these methods are capable of achieving compression ratios an order of magnitude higher than their non-structured counterparts (Bagdasarian et al. 2025; Ali et al. 2025). Despite these notable advancements, existing methodologies continue to face challenges in maintaining a delicate balance between compression efficiency, rendering speed, and visual quality in complex and high-density scenes (Chen et al. 2022; Barron et al. 2022).

This paper proposes a plug-and-play optimization scheme for 3D Gaussian Splatting structured compression. First, we discover that different distributions of input vectors do not affect the final reconstruction performance, but they directly influence the entropy coding compression ratio. Therefore, optimal coding efficiency can be achieved by adjusting the input vector distribution. Second, to reduce runtime memory usage, we perform probabilistic Gaussian pruning by optimizing the opacity attributes of Gaussians, which significantly reduces the overall density of the Gaussian scene and, in turn, lowers memory requirements. Finally, we find that excessively pursuing higher compression ratios inevitably leads to the loss of high-frequency details; thus, it is necessary to perform some high-frequency compensation after compression.

Our key contributions are threefold:

- We introduce a method to optimize entropy coding efficiency by constraining input vector distributions towards a standard normal distribution with minimal variance, using probabilistic divergence measures. This plug-and-play design is fully compatible with existing structured 3DGS compression frameworks(Lu et al. 2024; Chen et al. 2024).
- We propose a novel probabilistic pruning mechanism that leverages Gaussian opacity to selectively discard low-impact Gaussians. This not only enhances compression but also ensures robustness across scenes via adjustable pruning rates.
- We design a residual learning network to compensate for high-frequency detail loss incurred during aggressive compression.

All presented modules are designed to be plug-and-play, ensuring seamless integration into mainstream structured 3DGS compression pipelines.

2 Related Work

2.1 Neural Radiance Fields and 3D Gaussian Splatting

Neural Radiance Fields (NeRF) (Mildenhall et al. 2020; Lin et al. 2025; Sheng et al. 2024; Pumarola et al. 2021) set a new standard for novel view synthesis using continuous volumetric representations parameterized by MLPs. Yet, its reliance on costly ray sampling led to multi-minute rendering times per frame (Chen et al. 2022; Garbin et al. 2021; Takikawa et al. 2021; Müller et al. 2022). To break this efficiency barrier, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) emerged. 3DGS models scenes explicitly as learnable Gaussian ellipsoids, achieving real-time performance via splatting. While offering comparable visual quality to NeRF variants, 3DGS significantly accelerates rendering. Critically, this explicit representation entails immense memory and storage demands for complex scenes, fueling extensive research into 3DGS compression.

2.2 3D Gaussian Splatting Compression

3DGS compression is broadly categorized into non-structured and structured methods.

Non-structured methods optimize raw Gaussian parameters via pruning (Lee et al. 2024; Fan et al. 2024; Sharath Girish 2024; Papantonakis et al. 2024; M. Salman Ali and Tartaglione 2024), quantization (Navaneet et al. 2024; Niedermayr, Stumpfegger, and Westermann 2024), and entropy coding (M. Salman Ali 2025; Sharath Girish 2024; Morgenstern et al. 2025). Examples include Lee et al.'s spatial importance pruning (Lee et al. 2024) (potentially sacrificing high-frequency details), Navaneet et al.'s hierarchical quantization for position, covariance, and color (Navaneet et al. 2024), and Girish et al.'s entropy coding utilizing spatial correlation (Sharath Girish 2024).

Structured methods achieve higher compression through representation paradigm shifts. Scaffold-GS (Lu et al. 2024) employs a hierarchical anchor system for shared parameters, significantly boosting efficiency. Yang et al. (Yang et al. 2024; Zhang et al. 2024; Liu et al. 2024) model Gaussians as graph structures, using spectral pruning to remove redundancy. Chen et al.'s methods (Chen et al. 2024, 2025b,a) optimizes with hash grid-assisted context and learnable quantization. Context-GS (Wang et al. 2024b) further innovates with an autoregressive context model, hierarchically encoding and using coarse-to-fine prediction with low-dimensional quantized features, surpassing Scaffold-GS.

Systematic evaluations (Bagdasarian et al. 2025; Ali et al. 2025) confirm structured methods superior average compression efficiency in complex scenes over non-structured ones. Despite minor decoding overhead, their compression-quality benefits make it acceptable. Recent work explores hybrid representations, like combining voxels and Gaussians (Zhang et al. 2024), to reduce storage while maintaining visual quality. Overall, structured methods offer greater compression potential in complex scenes, but real-time decoding remains a challenge (Bagdasarian et al. 2025).

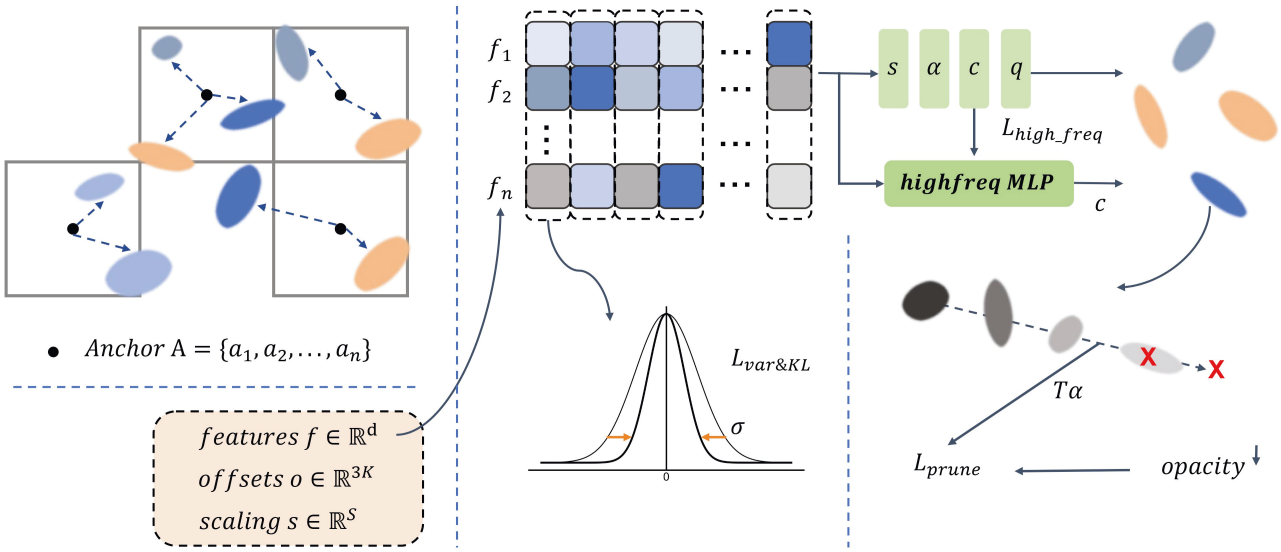

Figure 2: Overview of our method framework. Our method can base on any structured compression method (e.g. Scaffold-GS (Lu et al. 2024), HAC (Chen et al. 2024)). We compact the data distribution through variance-aware and aligned anchor feature vector f distribution patterns, enhancing data correlation while providing an ideal prior for entropy coding. Furthermore, we introduce pruning operations based on opacity α sorting, removing a certain proportion of anchor points that contribute less to the image, achieving fine-grained control over the anchor poins. It is worth noting that to minimize the effectiveness loss caused by pruning and quantization, we further incorporate high-frequency residual networks to compensate for the details of the scene.

3 Method

3.1 Preliminaries

3DGS (Kerbl et al. 2023) 3DGS is an explicit Gaussian-based real-time rendering method that fundamentally represents 3D scenes as a collection of numerous learnable Gaussian ellipsoids. Each Gaussian encapsulates both geometric properties (including position μ, covariance Σ, scale factor s, and opacity α) and appearance properties (represented by spherical harmonic coefficients f for view-dependent colors). During rendering, these 3D Gaussians are projected onto the 2D screen space via a GPU rasterization pipeline, depth-sorted, and then composited to yield the final pixel color.

$$\mathbf{C}(x) = \sum_{i \in N} \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) c_i = \sum_{i \in N} w_i c_i, \qquad (1)$$

where c_i denotes the decoded spherical harmonic feature, and w_i quantifies the contribution of the i-th Gaussian to the final rendered output. Compared to implicit neural representations, 3DGS offers the dual advantages of real-time rendering and explicit scene editing. Nevertheless, a typical scene often necessitates storing millions of Gaussian distributions, which presents a significant deployment challenge in resource-constrained environments. To address this limitation, we introduced innovative optimization method that employs a three-pronged strategy involving distribution regularization, probabilistic pruning, and detail compensation, as shown in Figure 2.

Scaffold-GS (Lu et al. 2024) Our research significantly leverages Scaffold-GS (Lu et al. 2024), a pioneering method that first proposed the concept of structured 3DGS compression. Scaffold-GS's core principle involves introducing "anchors" to organize local 3D Gaussians: shared base parameters are facilitated via anchors, and Gaussian attributes are dynamically predicted based on the viewing direction and distance from the view frustum. Moreover, its integrated anchor growth and pruning strategies effectively minimize redundant Gaussians while ensuring high-quality rendering. This design not only improves the model's adaptability to varying levels of detail and viewpoint-specific observations without compromising rendering speed, but also demonstrates superior performance and enhanced memory efficiency in managing complex scenes. Following this groundwork, subsequent studies like HAC (Chen et al. 2024) and Context-GS (Wang et al. 2024b) have further advanced the exploration in structured 3DGS compression. These cutting-edge contributions form a robust foundation for our current work.

3.2 Distribution Regularization for Anchor Vector Optimization

We propose an innovative distribution regularization method to boost compression efficiency in structured 3DGS without compromising rendering quality, achieved by systematically controlling anchor input vector distributions. Our motivation stems from a key insight: anchor vector distributions minimally affect final rendering quality but significantly impact

Figure 3: Qualitative results different scenes and datasets. We highlight the visual differences of various methods through box annotations for enhancing the visualization effect. In these scenarios, our approach demonstrates significant advantages in complex scenes, particularly in conditions with drastic lighting changes. At the same time, it demonstrates considerable advantages in scenes of varying scales and perspectives.

subsequent entropy coding efficiency.

Distribution vs. Rendering Quality Rigorous experiments confirmed this. Applying uniform, Gaussian and exponential constraints (Figure 1) to anchor vectors during training resulted in negligible differences in rendering quality metrics (Table 1). This demonstrates neural networks' strong distribution adaptability, allowing them to adjust parameters and maintain consistent visual quality despite varying input distributions.

Entropy Coding Distribution Optimization Nevertheless, when considering compression efficiency, the significance of data distribution properties becomes exceptionally salient. As a fundamental component of contemporary compression systems, entropy coding's efficiency is directly contingent upon the probability distribution characteristics of its input data, given its reliance on the probabilistic advantages offered by non-uniform data distributions (Shannon 1948; Witten, Neal, and Cleary 1987). The standard normal distribution, renowned for its excellent probability concentration, offers an ideal scenario for entropy coding. Guided by this insight, we formulated a dual optimization objective:

Distribution Alignment Loss We employ KL divergence to progressively align the distribution of anchor vectors with the standard normal distribution $N(0,1)$.

$$\mathcal{L}_{KL} = \lambda_{KL} \sum_{i=1}^{N_d} D_{KL}(P_i \,||\, N(0,1)), \tag{2}$$

where $N(0,1)$ denotes the standard normal distribution, P_i represents the frequency distribution of the i-th dimension (referencing the anchor parameterization method in Scaffold-GS (Lu et al. 2024)) and N_d signifies the total number of dimensions.

Variance Constraint To fully leverage the inherent probabilistic advantages within entropy coding, we introduce a variance constraint designed to ensure the compactness and tight concentration of the distribution.

$$\mathcal{L}_{var} = \lambda_{var} \sum_{i=1}^{N_d} \|\sigma_i - \sigma_0\|^2, \tag{3}$$

In this formula, $\sigma_i \in \mathbb{R}$ refers to the variance of the i-th dimension and $\lambda_{var} \in \mathbb{R}$ is a hyperparameter for modulating the supervision strength (a larger λ_{var} value leads to a more concentrated data distribution). N_d indicates the number of Gaussian feature dimensions in the scene, with i indexing the i-th dimension. This dual constraint mechanism not only guarantees the normalized morphology of the distribution but also, through precise variance regulation, further

Dataset	Synthetic-NeRF				Tanks&Temples				Deep Blending			
Mothods/Metrics	PSNR↑	SSIM↑	LPIPS↓	Size↓	PSNR↑	SSIM↑	LPIPS↓	Size↓	PSNR↑	SSIM↑	LPIPS↓	Size↓
3DGS-30K	33.31	-	-	-	23.14	0.841	0.183	411.0	29.41	0.903	**0.243**	676.0
Compact3DGS	33.33	**0.968**	**0.034**	5.8	23.32	0.831	0.202	20.9	29.73	0.900	0.258	23.8
LightGaussian	32.72	0.965	0.037	7.8	23.11	0.817	0.231	22.0	27.01	0.872	0.308	33.8
SOG	33.23	0.966	**0.034**	4.1	23.56	0.837	0.186	22.8	29.26	0.894	0.268	17.7
RDO-Gaussian	33.12	0.967	0.035	2.3	23.34	0.835	0.195	12.0	29.63	0.902	0.252	18.0
Scaffold-GS	33.40	0.966	0.035	15.7	24.12	0.851	0.175	77.9	30.13	0.905	0.257	52.7
HAC_lowrate	32.74	0.965	0.040	0.9	24.10	0.846	0.188	7.95	29.95	0.902	0.269	4.4
HAC_highrate	**33.72**	**0.968**	**0.034**	1.8	24.32	0.853	0.177	12.9	30.21	0.906	0.257	7.5
ContextGS_lowrate	32.79	0.965	0.040	1.1	24.12	0.849	0.186	9.9	30.09	0.907	0.265	3.7
ContextGS_highrate	33.51	**0.968**	0.035	1.6	24.29	**0.855**	0.176	11.8	**30.39**	**0.909**	0.258	6.6
Ours_lowrate	32.54	0.964	0.042	**0.8**	24.01	0.845	0.189	**5.8**	29.93	0.902	0.272	**3.0**
Ours_highrate	33.61	**0.968**	0.035	1.3	**24.36**	0.852	**0.166**	9.5	30.38	0.907	0.257	5.4

Table 2: Comparison of our method against other previous compression methods. The best and 2nd best results are in **bold** and underlined. The size is measured in MB.

Scene	Blender-Lego		DB-Drjohnson	
Model	PSNR↑	NUM	PSNR↑	NUM
None	35.51	55K	29.69	192K
w/ Var&KL	35.63	57K	29.73	200K
w/ Opa. Prune	35.59	**37K**	29.66	**141K**
w/ Detail Comp.	**35.78**	**40K**	**29.81**	**143K**

Table 3: Ablation studies. All ablation models are based on Scaffold-GS and seamlessly integrate the corresponding modules.

Dataset	HAC	Ours	Improvement Rate
Synthetic-NeRF	1.81s	1.49s	17.9%
Mip-NeRF360	21.71s	16.76s	22.8%
Tanks&Temples	15.05s	10.52s	30.1%
Deep Blending	7.94s	5.41s	31.9%

Table 4: Comparison of the average decoding efficiency of all scenes in the corresponding dataset. Our method makes the feature distribution more compact and reduces the number of anchors, improving the encoding and decoding efficiency of entropy coding.

elevates its probability concentration, thus furnishing superior input conditions for subsequent entropy coding.

Our method shows significant advantages in practical applications. Tests on multiple public datasets confirm that distribution optimization markedly enhances overall rendering quality and compression ratio. Notably, arithmetic coding (Witten, Neal, and Cleary 1987) efficiency improves by 25% over HAC (Chen et al. 2024), due to better alignment with entropy coding's theoretical assumptions. These gains make our approach ideal for resource-constrained scenarios.

Implementation-wise, our solution boasts excellent usability. It seamlessly integrates into existing structured 3DGS compression pipelines via a simple regularization term added to the loss function. This plug-and-play nature drastically cuts technical migration costs, granting our optimization broad applicability.

3.3 Opacity-Based Probabilistic Gaussian Pruning

Building upon our entropy coding optimization, we further introduce an opacity-based probabilistic pruning mechanism. This mechanism aims to achieve a highly efficient sparse representation of 3D scenes by systematically optimizing the opacity attribute of individual Gaussian distributions. Diverging from conventional hard threshold pruning, we reformulate the pruning process as an opacity optimization problem. The core tenet is that Gaussians contributing negligibly to the final rendered output should ideally have their opacity approach zero. This theoretical stance resonates with the findings in CompGS (Navaneet et al. 2024); however, we present significant enhancements by incorporating global constraints and an innovative probabilistic pruning strategy.

We have devised a constrained optimization framework featuring explicit boundary conditions. Central to this framework is the target total opacity T_α, a crucial control parameter that dynamically modulates the optimization intensity via a weighting coefficient λ. The objective function for this framework is designed to minimize overall opacity while concurrently preserving the fundamental structural integrity of the scene. Specifically, our opacity optimization objective function is defined as:

$$\mathcal{L}_{prune} = \lambda \left(\sum_{i=1}^{N} \alpha_i - T_\alpha \right)^2, T_\alpha = \beta N, \tag{4}$$

where, $\beta \in \mathbb{R}$ denotes the average opacity, N denotes the total number of Gaussians. $T_\alpha \in \mathbb{R}$ is the target total opacity, which governs the overall scene density; $\lambda \in \mathbb{R}$ is the weight coefficient; and $\alpha_i \in \mathbb{R}$ signifies the opacity of the i-th Gaussian. This function effectively drives the opacity

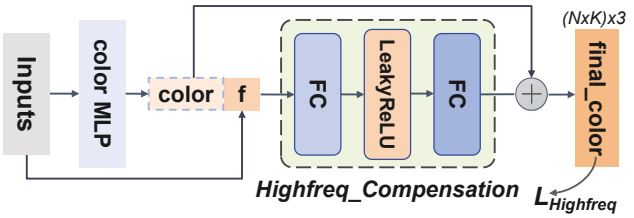

Figure 4: High frequency compensation module structure. For the color of all neural Gaussian points ($N \times K$), we input it into our lightweight MLP to obtain the refined final color, after obtaining the base color and concatenating it with the anchor point features f.

of redundant Gaussians towards zero while simultaneously ensuring that the aggregated scene density remains within a predefined range. This judicious design effectively mitigates the issue of rendering artifacts (holes) that can arise from aggressive over-pruning, thereby simultaneously guaranteeing robust compression efficiency.

To accommodate diverse hardware deployment environments, we have proposed a two-stage pruning strategy. The first, global optimization stage, leverages the aforementioned constrained optimization framework to automatically identify and attenuate non-essential Gaussian distributions. Subsequently, the probabilistic pruning stage employs a statistically-driven filtering mechanism: initially, the opacities of Gaussians associated with anchors are aggregated and sorted. Then, from those anchors whose sorted opacities fall within the bottom 40%, a random subset of 30% of their associated Gaussians are discarded. This stochastic pruning strategy offers marked advantages: it not only effectively prevents localized over-pruning but also helps to preserve scene detail diversity. Furthermore, it supports a configurable pruning rate ranging from 5% to 40%, providing flexible adaptation for various application scenarios.

Experimental evaluations consistently demonstrate the scheme's exceptional performance in resource-constrained settings. When subjected to 30% Gaussian pruning, our method achieves an 26.7% reduction in memory footprint compared to Scaffold-GS, while incurring only a 0.1-0.2dB decrease in PSNR. These characteristics render it particularly well-suited for resource-limited applications. Notably, the scheme also exhibits outstanding framework compatibility, allowing users to achieve a precise trade-off between rendering quality and computational efficiency merely by tuning parameters such as T_α and the pruning rate.

3.4 High-Frequency Detail Compensation

We observe that contemporary 3DGS compression methods, in their pursuit of high compression ratios, frequently lead to an excessive suppression of high-frequency signals within reconstructed scenes. This often manifests as visual artifacts such as blurred details and softened edges in the final rendered output. To effectively mitigate this loss of high-frequency information, we introduce a high-frequency compensation module, designed to augment high-frequency content in the output RGB images. The fundamental concept is inspired by the principle of high-frequency compensation inherent in sharpening filters within traditional image processing (Gonzalez and Woods 2018). At the core of this module lies a lightweight Multi-Layer Perceptron (MLP) network, denoted as $F(\cdot)$. Recognizing that the energy of high-frequency components is considerably lower than that of their low-frequency counterparts, we innovatively employ a residual learning architecture. Specifically, the module receives the compressed original RGB values $x \in \mathbb{R}^3$ as input, and through $F(x)$, it learns and predicts the high-frequency residual component. The final output is the compensated RGB $\mathcal{H} \in \mathbb{R}^3$.

$$\mathcal{H}_i = x_i + \Delta\mathcal{H}_i, \Delta\mathcal{H}_i = F(x_i \oplus f_i). \tag{5}$$

This residual learning design confers several notable benefits:

1) **Integrity of Base Information** The residual connection guarantees the complete preservation of the foundational color information, thus preventing extraneous data loss. 2) **Mitigated Learning Complexity** The MLP network's task is streamlined to solely learn the high-frequency residuals, rather than the entire signal. This substantially reduces the learning complexity and accelerates convergence. 3) **Adaptive Detail Restoration** By incorporating the Mean Squared Error (MSE) loss between the module's output $\mathcal{H}$ and the original high-frequency signal into the overall loss function for joint optimization, the network gains the ability to adaptively detect and restore high-frequency details that were compromised during compression. The high-frequency compensation loss function is formulated as follows:

$$\mathcal{L}_{HF} = \sum_{i=1}^{N} \|\mathcal{H}_i - x_i\|^2 . \tag{6}$$

Analogous to the previously detailed distribution regularization and probabilistic pruning modules, the high-frequency detail compensation module demonstrates exceptional compatibility, allowing for seamless integration into existing structured 3DGS compression frameworks.

4 Experiments

4.1 Setup

All experiments were executed on a computing platform configured with an NVIDIA Tesla V100 GPU (32GB VRAM). The implementations leveraged the PyTorch 1.12.1 framework alongside the CUDA 11.6 acceleration library. A thorough evaluation was conducted across four widely recognized benchmark datasets: 1) **Tanks&Temples (Knapitsch et al. 2017)** This dataset encompasses 2 large-scale, open-world scenes, primarily serving to assess method robustness within intricate geometric settings. 2) **Mip-NeRF360 (Barron et al. 2022)** Featuring 9 diverse indoor and outdoor environments, this dataset is crucial for validating rendering fidelity under challenging lighting conditions. 3) **DeepBlending (Hedman et al. 2018)** Comprising 2 scenes that include transparent objects, it is utilized to rigorously evaluate the material modeling capabilities of the approaches. 4) **Blender (Mildenhall et al. 2020)** With its

8 synthetic scenes, this dataset is specifically designed for quantifying the precision of reconstructed geometric details.

For performance evaluation, we adopted five complementary metrics: 1) **PSNR (Peak Signal-to-Noise Ratio (Damera-Venkata et al. 2000))** This metric primarily quantifies the pixel-wise fidelity between reconstructed and ground-truth images. 2) **SSIM (Structural Similarity Index Measure (Wang et al. 2004))** It assesses the preservation of image quality concerning brightness, contrast, and structural information. 3) **LPIPS (Learned Perceptual Image Patch Similarity(Zhang et al. 2018))** This metric measures high-level feature discrepancies from a human perceptual standpoint. 4) **SIZE (Model Storage Size)** Employed to evaluate the effectiveness of compression. 5) **NUM (Number of Anchors)** Utilized to gauge the memory footprint.

Furthermore, to comprehensively evaluate the practical utility of our proposed method, we meticulously recorded the entropy coding computational efficiency and rendering latency across different approaches.

4.2 Performance Evaluation

To comprehensively evaluate the performance of our proposed method, we conducted comparisons against 8 state-of-the-art approaches, specifically including: Scaffold-GS (Lu et al. 2024), HAC (Chen et al. 2024), Context-GS (Wang et al. 2024b), 3DGS-30k (Kerbl et al. 2023), LightGaussian (Fan et al. 2024), SOG(Morgenstern et al. 2025), Compact3DGS (Lee et al. 2024) and RDO-Gaussian (Wang et al. 2024a). To ensure the fairness and comparability of our experiments, we adopted HAC (Chen et al. 2024) as the foundational backbone. Into this framework, we seamlessly integrated the three distinct optimization modules proposed in this work.

With respect to hyperparameter configurations, the loss weights for our three modules were set as follows: the distribution regularization module's weight was 0.001, the probabilistic pruning module's weight was 1e-5, the average opacity T_α was 0.001, and the high-frequency detail compensation module's weight was 5e-4. During the initial distribution alignment process, the vector quantization resolution was configured to 7 bit. All other training parameters were maintained in strict consistency with those of Scaffold-GS and HAC, thereby ensuring the robust comparability of our experimental findings.

Table 1 meticulously detail the performance evaluation results of each method across the various datasets.

Table 2 illustrates that our method achieves superior model size reduction and higher compression ratios compared to state-of-the-art techniques. Relative to Vanilla 3DGS, our approach reduces model size by 100 times on average. We also demonstrate better storage efficiency than recent competitors like HAC and Context-GS. Furthermore, Table 4 confirms our method's significant boost to entropy coding efficiency, showing an 25% improvement over HAC under identical base settings.

Figure 3 shows visual comparisons, demonstrating that our method delivers better rendering quality with significantly reduced model size compared to most recent 3D compression models.

4.3 Ablation Studies

This subsection presents a comprehensive ablation study designed to systematically ascertain the individual contributions of each of the three proposed modules to the model's compression ratio and rendering quality. All evaluations were conducted under the full view configuration of the Blender and Deep Blending dataset.

Component Ablation We rigorously verified the efficacy of all plug-and-play modules comprising our method, with the quantitative results meticulously presented in Table 3. Further experiments and extended analyses can be found in the supplementary material.

The distribution regularization constraint demonstrably enhances the compression ratio during the entropy coding phase: the sole application of this module leads to a 9.6% reduction in model size, an 0.2dB improvement in PSNR, and concurrently an 5% increase in entropy coding efficiency.

Furthermore, the opacity-based probabilistic pruning module significantly diminishes the quantity of Gaussians. Under a 30% probabilistic pruning regime, the number of Gaussian anchors decreases by 30%, yet the PSNR experiences only a marginal drop of 0.1-0.3dB, thereby substantially curtailing memory demands. Additionally, direct pruning was found to introduce localized distortion, degrading the perceptual consistency of the reconstructed outputs. To further verify this effect, additional comparison experiments were conducted. The results indicate that random pruning yields more stable or slightly higher PSNR values, suggesting a superior balance between compression efficiency and reconstruction quality.

Concurrently, the high-frequency information loss, a potential consequence of aggressive compression, can be effectively recuperated by the high-frequency detail compensation network. Table 4 provide compelling evidence, showcasing the quantitative data comparisons, after processing with this network, thus robustly affirming its detail recovery capabilities.

Compatibility Analysis The three plug-and-play modules we propose are not only capable of seamless integration into existing structured 3DGS compression models (e.g., Scaffold-GS), but with minor adaptive modifications, they can also be deployed with other types of models. The experimental outcomes unequivocally demonstrate that the compression performance of all baselines was substantially improved through the incorporation of our modules, further underscoring the broad applicability of our proposed solution.

5 Conclusion

We present an innovative plug-and-play optimization for structured 3D Gaussian Splatting compression. Experiments confirm our method achieves substantial model size reduction with maintained real-time rendering, outperforming existing baselines. Its modular design ensures seamless integration into various structured compression pipelines, offering a practical solution for high-quality 3DGS in resource-constrained environments. Our full implementation is publicly available to accelerate future research.

References

Ali, M. S.; Zhang, C.; Cagnazzo, M.; Valenzise, G.; Tartaglione, E.; and Bae, S.-H. 2025. Compression in 3D Gaussian Splatting: A Survey of Methods, Trends, and Future Directions. arXiv:2502.19457.

Bagdasarian, M. T.; Knoll, P.; Li, Y.; Barthel, F.; Hilsmann, A.; Eisert, P.; and Morgenstern, W. 2025. 3DGS.zip: A Survey on 3D Gaussian Splatting Compression Methods. *Computer Graphics Forum*, e70078.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. TensoRF: Tensorial Radiance Fields. In *European Conference on Computer Vision (ECCV)*.

Chen, Y.; Li, M.; Wu, Q.; Lin, W.; Harandi, M.; and Cai, J. 2025a. PCGS: Progressive Compression of 3D Gaussian Splatting. *arXiv preprint arXiv:2503.08511*.

Chen, Y.; Wu, Q.; Li, M.; Lin, W.; Harandi, M.; and Cai, J. 2025b. Fast Feedforward 3D Gaussian Splatting Compression. In *International Conference on Learning Representations (ICLR)*.

Chen, Y.; Wu, Q.; Lin, W.; Harandi, M.; and Cai, J. 2024. HAC: Hash-grid Assisted Context for 3D Gaussian Splatting Compression. In *European Conference on Computer Vision (ECCV)*.

Chen, Z.; Funkhouser, T.; Hedman, P.; and Tagliasacchi, A. 2023. MobileNeRF: Exploiting the Polygon Rasterization Pipeline for Efficient Neural Field Rendering on Mobile Architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Damera-Venkata, N.; Kite, T.; Geisler, W.; Evans, B.; and Bovik, A. 2000. Image quality assessment based on a degradation model. *IEEE Transactions on Image Processing (TIP)*, 9(4): 636–650.

Fan, Z.; Wang, K.; Wen, K.; Zhu, Z.; Xu, D.; and Wang, Z. 2024. LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction and 200+ FPS. In *Neural Information Processing Systems (NeurIPS)*.

Garbin, S. J.; Kowalski, M.; Johnson, M.; Shotton, J.; and Valentin, J. 2021. FastNeRF: High-Fidelity Neural Rendering at 200FPS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Gonzalez, R. C.; and Woods, R. E. 2018. *Digital Image Processing*. Pearson, 4th edition.

Hedman, P.; Philip, J.; Price, T.; Frahm, J.-M.; Drettakis, G.; and Brostow, G. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.*, 37(6).

Kalkofen, D.; Mendez, E.; and Schmalstieg, D. 2009. Comprehensible Visualization for Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 15(2): 193–204.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics (TOG)*, 36(4).

Lee, J. C.; Rho, D.; Sun, X.; Ko, J. H.; and Park, E. 2024. Compact 3D Gaussian Representation for Radiance Field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21719–21728.

Lin, A.; Xiang, Y.; Li, J.; and Prasad, M. 2025. Dynamic Appearance Particle Neural Radiance Field. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 35(7): 6853–6866.

Liu, X.; Wu, X.; Zhang, P.; Wang, S.; Li, Z.; and Kwong, S. 2024. CompGS: Efficient 3D Scene Representation via Compressed Gaussian Splatting. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20654–20664.

M. Salman Ali, E. T., Sung-Ho Bae. 2025. ELMGS:Enhancing memory and computation scaLability through coMpression for 3D Gaussian Splatting. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

M. Salman Ali, S.-H. B., M. Qamar; and Tartaglione, E. 2024. Trimming the Fat: Efficient Compression of 3D Gaussian Splats through Pruning. In *BMVC*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*.

Morgenstern, W.; Barthel, F.; Hilsmann, A.; and Eisert, P. 2025. Compact 3D Scene Representation via Self-Organizing Gaussian Grids. In *European Conference on Computer Vision (ECCV)*, 18–34. Cham: Springer Nature Switzerland.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.*, 41(4): 102:1–102:15.

Navaneet, K.; Meibodi, K. P.; Koohpayegani, S. A.; and Pirsiavash, H. 2024. CompGS: Smaller and Faster Gaussian Splatting with Vector Quantization. *European Conference on Computer Vision (ECCV)*.

Niedermayr, S.; Stumpfegger, J.; and Westermann, R. 2024. Compressed 3D Gaussian Splatting for Accelerated Novel View Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10349–10358.

Papantonakis, P.; Kopanas, G.; Kerbl, B.; Lanvin, A.; and Drettakis, G. 2024. Reducing the Memory Footprint of 3D Gaussian Splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1).

Patney, A.; Salvi, M.; Kim, J.; Kaplanyan, A.; Wyman, C.; Benty, N.; Luebke, D.; and Lefohn, A. 2016. Towards foveated rendering for gaze-tracked virtual reality. *ACM Trans. Graph.*, 35(6).

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shannon, C. E. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3): 379–423.

Sharath Girish, A. S., Kamal Gupta. 2024. EAGLES: Efficient Accelerated 3D Gaussians with Lightweight EncodingS. In *European Conference on Computer Vision (ECCV)*.

Sheng, Z.; Liu, F.; Liu, M.; Zheng, F.; and Nie, L. 2024. Open-Set Synthesis for Free-Viewpoint Human Body Reenactment of Novel Poses. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 34(12): 12676–12691.

Takikawa, T.; Litalien, J.; Yin, K.; Kreis, K.; Loop, C.; Nowrouzezahrai, D.; Jacobson, A.; McGuire, M.; and Fidler, S. 2021. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, H.; Zhu, H.; He, T.; Feng, R.; Deng, J.; Bian, J.; and Chen, Z. 2024a. End-to-End Rate-Distortion Optimized 3D Gaussian Representation. In *European Conference on Computer Vision (ECCV)*.

Wang, Y.; Li, Z.; Guo, L.; Yang, W.; Kot, A.; and Wen, B. 2024b. ContextGS : Compact 3D Gaussian Splatting with Anchor Level Context Model. In *Neural Information Processing Systems (NeurIPS)*.

Wang, Z.; Bovik, A.; Sheikh, H.; and Simoncelli, E. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4): 600–612.

Witten, I. H.; Neal, R. M.; and Cleary, J. G. 1987. Arithmetic coding for data compression. *Commun. ACM*, 30(6): 520–540.

Yan, C.; Qu, D.; Xu, D.; Zhao, B.; Wang, Z.; Wang, D.; and Li, X. 2024. GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang, R.; Zhu, Z.; Jiang, Z.; Ye, B.; Chen, X.; Zhang, Y.; Chen, Y.; Zhao, J.; and Zhao, H. 2024. Spectrally Pruned Gaussian Fields with Neural Compensation. arXiv:2405.00676.

Zhang, F.; Luo, Y.; Zhang, T.; Zhang, L.; and Huang, Z. 2024. GaussianForest: Hierarchical-Hybrid 3D Gaussian Splatting for Compressed Scene Modeling. arXiv:2406.08759.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.