CSGaussian: Progressive Rate-Distortion Compression and Segmentation for 3D Gaussian Splatting

Yu-Jen Tseng[1], Chia-Hao Kao[2], Jing-Zhong Chen[1], Alessandro Gnutti[2],
Shao-Yuan Lo[3], Yen-Yu Lin[1], Wen-Hsiao Peng[1]
[1]National Yang Ming Chiao Tung University, Taiwan
[2]University of Brescia, Italy [3]National Taiwan University, Taiwan
alan.cs12@nycu.edu.tw, wpeng@cs.nycu.edu.tw

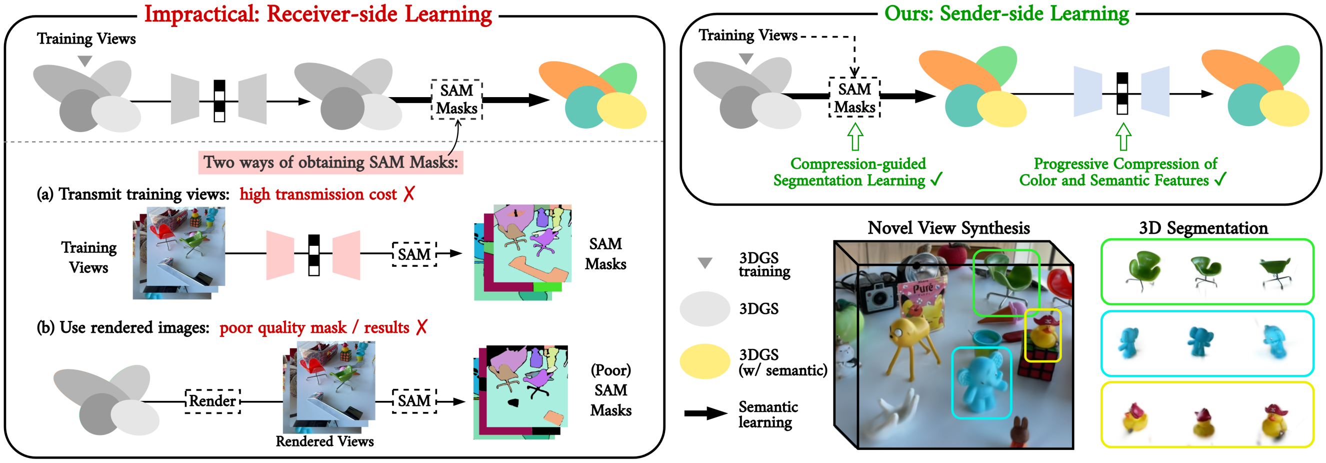

Impractical: Receiver-side Learning
Training Views
SAM Masks
Two ways of obtaining SAM Masks:
(a) Transmit training views: high transmission cost ✗
Training Views
SAM
SAM Masks
(b) Use rendered images: poor quality mask / results ✗
Render
SAM
(Poor) SAM Masks
Rendered Views
Ours: Sender-side Learning
Training Views
SAM Masks
Compression-guided Segmentation Learning ✓
Progressive Compression of Color and Semantic Features ✓
3DGS training
3DGS
3DGS (w/ semantic)
Semantic learning
Novel View Synthesis
3D Segmentation

Figure 1. Comparison of approaches for efficient 3DGS segmentation at the receiver. Left: The naive solution of learning semantics at the receiver is impractical, as it either (a) requires transmitting all training-view images, or (b) relies on rendered views that yield poor-quality SAM masks and suboptimal segmentation. Right: Our proposed method introduces sender-side compression-guided segmentation learning and transmits the semantically enriched 3DGS with RD-optimized compression, enabling efficient 3D segmentation at the receiver.

Abstract

We present the first unified framework for rate-distortion-optimized compression and segmentation of 3D Gaussian Splatting (3DGS). While 3DGS has proven effective for both real-time rendering and semantic scene understanding, prior works have largely treated these tasks independently, leaving their joint consideration unexplored. Inspired by recent advances in rate-distortion-optimized 3DGS compression, this work integrates semantic learning into the compression pipeline to support decoder-side applications–such as scene editing and manipulation–that extend beyond traditional scene reconstruction and view synthesis. Our scheme features a lightweight implicit neural representation-based hyperprior, enabling efficient entropy coding of both color and semantic attributes while avoiding costly grid-based hyperprior as seen in many prior works. To facilitate compression and segmentation, we further develop compression-guided segmentation learning, consisting of quantization-aware training to enhance feature separability and a quality-aware weighting mechanism to suppress unreliable Gaussian primitives. Extensive experiments on the LERF and 3D-OVS datasets demonstrate that our approach significantly reduces transmission cost while preserving high rendering quality and strong segmentation performance.

1. Introduction

Recent advances in 3D vision have made 3D Gaussian Splatting (3DGS) [15] a promising representation for 3D scenes due to its excellent reconstruction quality and real-time rendering capabilities. However, 3DGS typically involves a large number of Gaussian primitives, which leads to high storage cost and memory footprint. This limitation has spurred research into efficient 3DGS compression.

Aimed at minimizing parameter count, memory footprint, and file size, early research focuses primarily on developing more compact 3DGS representations [11, 18, 25]. More recently, rate-distortion (RD)-optimized compression of 3DGS [7, 36, 37, 43] emerged as a new school of thought, targeting transmission and storage efficiency. Departing from earlier approaches that merely reduce parameter count without entropy coding, these methods perform quantization and entropy encoding of Gaussian attributes to achieve a balanced trade-off between compressed file size (*rate*) and rendering quality (*distortion*) via end-to-end, per-scene optimization.

Meanwhile, the rising demand for 3D scene understanding in applications such as robotics and autonomous driving has catalyzed rapid advancements in 3DGS segmentation [5, 20, 26, 30, 33, 39, 41]. Recent methods learn 3D semantic features by leveraging multi-view images and their associated camera poses. A common approach to learning semantic features involves self-supervised contrastive learning guided by foundation models, such as Segment Anything Model (SAM) [17] or CLIP [31]. Conceptually, the semantic features of individual Gaussian primitives are acquired by first rendering them onto 2D training views and then aligning them with 2D segmentation masks produced by SAM or other foundation models. These semantic features empower the resulting 3DGS representation to support downstream tasks, including open-vocabulary segmentation [16, 22, 30] and 3D scene manipulation [6, 14, 38, 40].

This work explores a novel application scenario in which 3DGS optimization is performed on a server, while rendering, scene editing, manipulation, and understanding are carried out on a remote end device—such as an Augmented Reality (AR) headset. A naive approach would transmit color-only Gaussian primitives from the server (sender) and attempt to learn semantic features locally on the headset (receiver). However, this solution is infeasible as it either requires additionally sending all training-view images, which is prohibitively expensive (Figure 1a), or learning semantics from rendered views of compressed 3DGS, which leads to severe performance degradation (Figure 1b). In the latter case, the compromised visual quality also undermines the effectiveness of pre-trained vision models, such as SAM, leading to poor mask generation and suboptimal segmentation outcomes. A more practical solution is to learn semantic features on the server and transmit them alongside other color-related Gaussian attributes, as illustrated on the right of Figure 1. This approach calls for an efficient compression system that accounts for both color and semantic features.

So far, only few studies [10, 20] have explored 3DGS compression and segmentation simultaneously. InstanceGS [20] utilizes a hierarchical structure [25] to efficiently represent both color and semantic features, while DF-3DGS [10] decouples the semantic field from color information. Both approaches work under the assumption that semantic features require less granularity than color and are amenable to compression. Nevertheless, they do not consider rate-distortion (RD)-optimized compression of color and segmentation features.

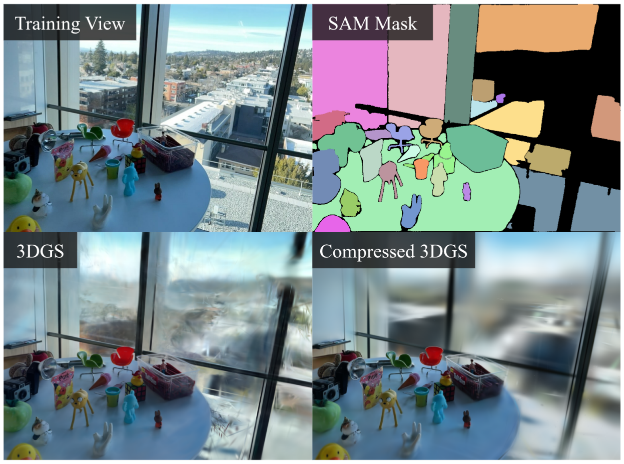


Figure 2. Visualization of training view, SAM mask, and 3DGS renderings with and without compression. Background regions of 3DGS, especially when compressed, contain low-quality primitives that fail to represent objects meaningfully. This discrepancy with SAM masks hinders segmentation learning when all primitives are treated equally.

In this paper, we present the first 3DGS framework to perform progressive RD-optimized compression and segmentation, enabling efficient yet accurate representation and understanding of 3D scenes. Our proposed scheme, CS-Gaussian, employs a hierarchical anchor-based representation [25] to encode both color and semantic information. To facilitate their entropy coding, we incorporate an Implicit Neural Representation (INR)-based hyperprior to model their distributions. To support compression and segmentation, we propose a compression-guided segmentation learning strategy that tightly couples the two tasks via (1) quantization-aware training and (2) quality-aware weighting. The former is applied to semantic features; it improves both compression and segmentation performance. The latter stems from the observation that Gaussian primitives vary in quality, as shown in Figure 2. Low-quality and background primitives often misalign with semantic masks, thereby hindering contrastive learning of semantic features when they are treated equally. Compared to RD-optimized 3DGS frameworks that compress only color [7, 37, 43], our method progressively encodes color and semantic information into a single bitstream, supporting both high-fidelity reconstruction and accurate segmentation.

In summary, this work makes three primary contributions:

- We pioneer a novel RD-optimized framework for simultaneous compression and segmentation on 3DGS.
- Our framework introduces a simple yet effective INR-based hyperprior for Gaussian attributes along with semantic information, leading to a significant bitrate reduction.
- We develop a quantization-aware training strategy and a quality-aware weighting mechanism to notably improve 3D scene understanding.

Extensive experiments demonstrate the effectiveness of our method, which substantially reduces transmission cost while performing favorably against prior methods.

2. Related Work

2.1. 3DGS Compression

3DGS [15] represents a 3D scene using numerous learnable Gaussian primitives. High-quality, real-time rendering requires substantial storage and bandwidth, highlighting the need for efficient compression.

Compact 3DGS Representation. Early efforts on 3DGS compression primarily focus on making the representation more compact in terms of storage size and memory footprint. One representative paradigm is to directly reduce the number of Gaussian primitives by pruning those that contribute minimally to rendered image quality [1, 11, 12, 18]. Another strategy employs vector quantization to quantize Gaussian attributes into fixed-length codewords in a learned codebook [11, 18, 27, 28]. Several existing methods explore hierarchical representations of Gaussian primitives [25, 32, 34]. For example, Scaffold-GS [25] employs an anchor-based design, where a group of Gaussian primitives is represented by one anchor, significantly reducing the parameter count.

Rate-distortion-optimized 3DGS Representation. Building on recent advances in learned image and video compression [2, 3, 13], several recent works have started integrating entropy coding into 3DGS frameworks to enable end-to-end RD-optimization [7, 8, 23, 24, 37, 43, 44]. These methods typically apply scalar quantization to Gaussian attributes and model their distributions with a hyperprior, enabling efficient arithmetic coding. For instance, HAC [7], built on Scaffold-GS, presents one of the earliest RD-optimized 3DGS compression systems by leveraging a hash grid to capture spatial relationships. CAT-3DGS [43] further improves coding efficiency through a triplane-based hyperprior and autoregressive modeling in both spatial and channel dimensions. Similarly, ContextGS [37] exploits local correlations between Gaussian primitives to construct an autoregressive model for entropy coding their attributes. Although these methods achieve a good trade-off between transmission bitrate and rendering quality, they do not incorporate semantic information or support 3D scene understanding.

2.2. 3DGS Segmentation

Integrating 3DGS with vision foundation models [4, 17, 19, 31], has led to significant progress in 3DGS segmentation [5, 9, 20, 29, 30, 39, 45]. The core principle involves attaching semantic features to Gaussian primitives, which allows 3D scenes to be represented by both color and semantic information. Based on the way of feature acquisition, existing methods fall into two categories: feature distillation and mask lifting.

Feature Distillation. These approaches distill features from foundation models and embed them into 3DGS. For instance, LangSplat [30] employs an autoencoder to project CLIP features into a 3D latent space, while LEGaussians [33] applies codebook quantization to distill CLIP and DINO [4] features. On the other hand, Feature3DGS [46] extracts features from the SAM encoder and employs the SAM decoder for segmentation. However, this early line of work has a key drawback: it requires a dedicated decoder to recover features during inference, leading to high computational overhead and inefficient rendering.

Mask Lifting. Another line of research explored learning 3D semantic features by leveraging segmentation masks produced by vision foundation models through self-supervised learning objectives. For example, SAGA [5] adopts a classic contrastive loss with SAM masks to guide the learning process. ClickGaussian [9] introduces global feature-guided learning to address cross-view inconsistency. More recently, OpenGaussian [39] combines SAM masks and CLIP embeddings and introduces a 3D-to-2D feature association for open-vocabulary segmentation. Building on this, InstanceGS [20] reformulates the representation within the Scaffold-GS framework, enabling Gaussian primitives associated with the same anchor to share semantic features. Compared to feature distillation, mask lifting offers a more scalable and computationally efficient solution by circumventing the need to decode dense feature embeddings.

However, most existing methods for 3DGS segmentation overlook the storage and transmission costs associated with 3DGS representations and their learned semantic features, which poses a challenge in practical applications.

3. Preliminary

3.1. Scaffold-GS: Anchor-based Representation

Scaffold-GS [25] introduces a storage-efficient, anchor-based representation for organizing Gaussian primitives. Specifically, anchor points are initialized on a predefined voxel grid, and each anchor carries information about a fixed number of K (e.g. 10) Gaussian primitives. For example, the positions of these Gaussian primitives, denoted as $\{\boldsymbol{\mu}_i \in \mathbb{R}^3\}_{i=1}^K$, are computed from the anchor position $\boldsymbol{x}$, learnable offsets $\{\boldsymbol{O}_i \in \mathbb{R}^3\}_{i=1}^K$ and a scaling factor $\boldsymbol{l} \in \mathbb{R}^6$, following the formulation: $\{\boldsymbol{\mu}_i\}_{i=1}^K = \boldsymbol{x} + \{\boldsymbol{O}_i\}_{i=1}^K \cdot \boldsymbol{l}$. Additional structural and color attributes of the Gaussian primitives, including the color $\{\boldsymbol{c}_i \in \mathbb{R}^3\}_{i=1}^K$, opacity $\{\boldsymbol{\alpha}_i \in \mathbb{R}^1\}_{i=1}^K$, scale $\{\boldsymbol{S}_i \in \mathbb{R}^3\}_{i=1}^K$, and rotation $\{\boldsymbol{R}_i \in \mathbb{R}^4\}_{i=1}^K$, are decoded from the anchor feature $\boldsymbol{f} \in \mathbb{R}^{50}$ via an MLP decoder.

3.2. Contrastive Semantic Feature Learning

In mask lifting-based 3DGS segmentation frameworks–such as OpenGaussian [39] and InstanceGS [20]–a semantic feature $\boldsymbol{s} \in \mathbb{R}^6$ is learned for each Gaussian primitive to encode its semantic meaning. Similar to color attributes, these features can be rendered onto a 2D feature map $\boldsymbol{F}$ using α-blending [15]. To guide the self-supervised learning of semantic features, 2D segmentation masks $\boldsymbol{M}$ are first generated by SAM for each training image. The semantic learning process is then supervised using two complementary losses: (1) the intra-mask smoothing loss $\mathcal{L}_s$, which encourages semantic features within the same 2D mask to converge toward their mean, and (2) the inter-mask contrastive loss $\mathcal{L}_c$, which promotes the separation between semantic features correspond to different instance masks. In symbols, we have:

$$\mathcal{L}_s = \sum_{i=1}^{m} \sum_{h=1}^{H} \sum_{w=1}^{W} \boldsymbol{M}_{i,h,w} \cdot \|\boldsymbol{F}_{:,h,w} - \bar{\boldsymbol{F}}_i\|^2, \tag{1}$$

$$\mathcal{L}_c = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} \frac{1}{\|\bar{\boldsymbol{F}}_i - \bar{\boldsymbol{F}}_j\|^2}, \tag{2}$$

where H and W denote the height and width of the image, respectively, m is the number of SAM masks in the current view, and $\bar{\boldsymbol{F}}_i$, $\bar{\boldsymbol{F}}_j$ are the mean features of two distinct instance masks.

4. Method

4.1. System Overview

In this work, we propose a novel RD-optimized 3DGS compression system that enables efficient transmission of color and semantic information of a 3D scene within a single bitstream (Figure 3). To achieve this, we adopt Scaffold-GS [25] as our base 3DGS representation and structure the encoding process into three main stages, as outlined below.

Stage 1: Color-only 3DGS Optimization. The first stage focuses exclusively on learning color (appearance) attributes in an end-to-end, RD-optimized manner, targeting both rendering quality and transmission bitrate. In a way similar to prior RD-optimized 3DGS compression methods [7, 8, 43], we leverage entropy coding to efficiently encode quantized anchor attributes, including the offsets $\{\boldsymbol{O}_i\}_{i=1}^K$, scaling $\boldsymbol{l}$, and anchor feature $\boldsymbol{f}$. Unlike existing approaches that rely on heavy spatial priors such as hash grids or triplanes, we propose a lightweight hyperprior based on an implicit neural representation (INR). Taking the 3D position of each anchor as input, the INR hyperprior learns a function to model the probability distributions of anchor attributes scatted throughout the 3D space without requiring hash grids or triplanes.

Stage 2: Compression-guided Semantic Feature Learning. This step aims to learn semantic features $\boldsymbol{s}$ associated with every anchor, to support 3D segmentation. Since semantic information typically require less granularity than color information [10, 20], all Gaussian primitives associated with an anchor share the same semantic feature vector. Our feature learning builds on the self-supervised framework of [20, 39], with two key enhancements motivated by the joint demands of RD-optimized compression and segmentation: (i) quantization-aware training for semantic features, and (ii) quality-aware weighting (Sec. 4.3).

Stage 3: Semantic Feature Compression. Lastly, semantic learning and semantic feature compression are jointly optimized within an end-to-end, RD-optimized framework. To entropy encode (or decode) semantic features, their probability distributions are estimated by a dedicated INR-based hyperprior model. Both the semantic features and the hyperprior are updated simultaneously to balance segmentation accuracy against compressed bitrate.

4.2. INR-based Hyperprior

To ensure efficient transmission of 3DGS, we propose a lightweight INR-based hyperprior. It comprises two distinct models: one for Stage 1 (color compression) and another for Stage 3 (semantic feature compression). As depicted in Figure 3a, our INR-based hyperprior models the distributions of 3DGS anchor attributes for entropy coding. Specifically, it consists of two compact multilayer perceptrons. The first predicts the distributions of anchor features $\boldsymbol{f}$, offsets $\{\boldsymbol{O}_i\}_{i=1}^K$, and scaling factors $\boldsymbol{l}$; the second models the distributions of semantic features $\boldsymbol{s}$. Given the positional embedding [35] of an anchor location $\boldsymbol{x}$, the model outputs the corresponding Gaussian distribution parameters–means $\boldsymbol{\mu}$ and variances $\boldsymbol{\sigma}$–along with the quantization step size q for each attribute type. In particular, each attribute type is assigned a distinct quantization step size, which is shared across its components. The probability of a quantized attribute $\hat{\boldsymbol{a}} = q \cdot \text{round}(\boldsymbol{a}/q)$ is then estimated as the probability mass under the learned Gaussian prior:

$$p(\hat{\boldsymbol{a}} \mid \boldsymbol{x}) = \int_{\hat{\boldsymbol{a}}-\frac{q}{2}}^{\hat{\boldsymbol{a}}+\frac{q}{2}} \mathcal{N}\big(\boldsymbol{\mu}, \boldsymbol{\sigma}\big)\, d\boldsymbol{a}.$$

Our grid-free formulation enables faster training and rendering compared to triplane-based hyperpriors, which require additional compression of grid points on each triplane typically through time-consuming autoregressive modeling.

To further enhance the entropy modeling for color attributes $\boldsymbol{f}$, we integrate a Channel-wise AutoRegressive Model (CARM) following [43, 44]. This model sequentially

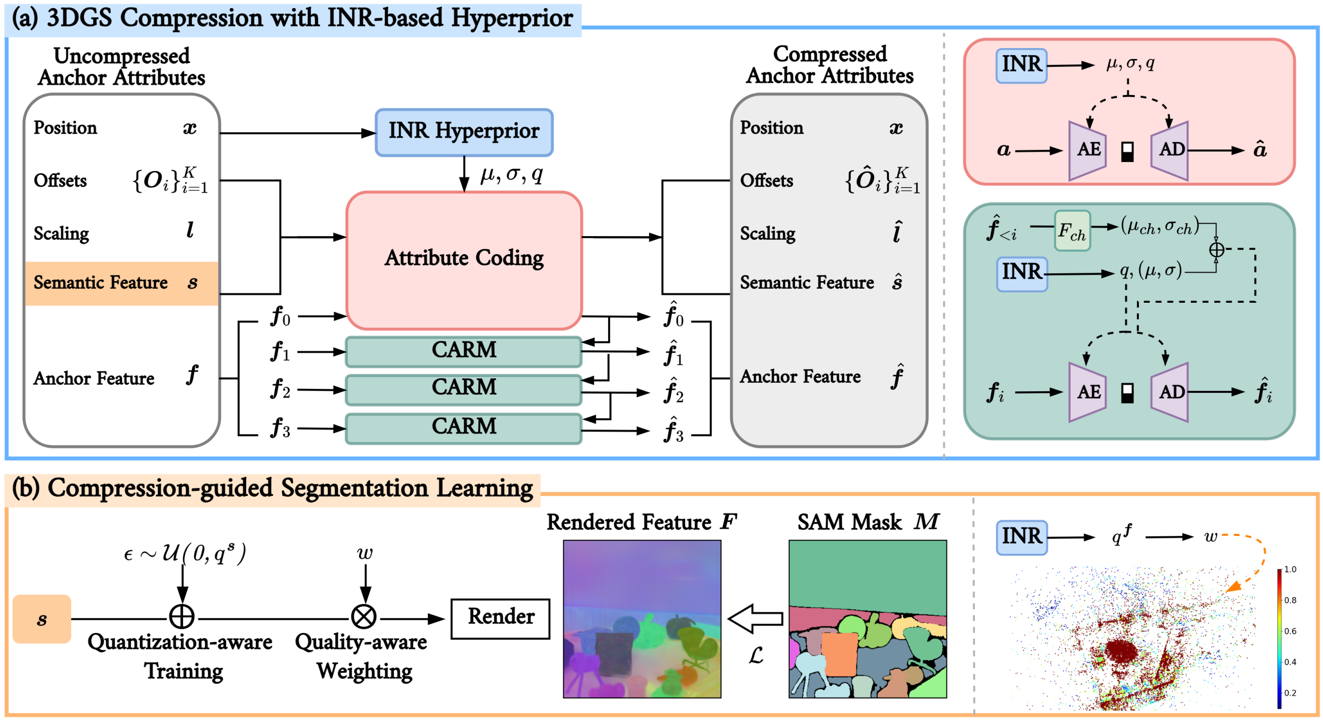

(a) 3DGS Compression with INR-based Hyperprior
Uncompressed Anchor Attributes
Position $\boldsymbol{x}$
Offsets $\{\boldsymbol{O}_i\}_{i=1}^{K}$
Scaling $\boldsymbol{l}$
Semantic Feature $\boldsymbol{s}$
Anchor Feature $\boldsymbol{f}$
INR Hyperprior
μ, σ, q
Attribute Coding
$\boldsymbol{f}_0$
$\boldsymbol{f}_1$
$\boldsymbol{f}_2$
$\boldsymbol{f}_3$
CARM
CARM
CARM
$\hat{\boldsymbol{f}}_0$
$\hat{\boldsymbol{f}}_1$
$\hat{\boldsymbol{f}}_2$
$\hat{\boldsymbol{f}}_3$
Compressed Anchor Attributes
Position $\boldsymbol{x}$
Offsets $\{\hat{\boldsymbol{O}}_i\}_{i=1}^{K}$
Scaling $\hat{\boldsymbol{l}}$
Semantic Feature $\hat{\boldsymbol{s}}$
Anchor Feature $\hat{\boldsymbol{f}}$
INR
μ, σ, q
$\boldsymbol{a}$
AE
AD
$\hat{\boldsymbol{a}}$
$\hat{\boldsymbol{f}}_{<i}$
F_{ch}
(μ_{ch}, σ_{ch})
INR
$q, (\mu, \sigma)$
$\boldsymbol{f}_i$
AE
AD
$\hat{\boldsymbol{f}}_i$
(b) Compression-guided Segmentation Learning
$\boldsymbol{s}$
$\epsilon \sim \mathcal{U}(0, q^s)$
w
Quantization-aware Training
Quality-aware Weighting
Render
Rendered Feature $\boldsymbol{F}$
SAM Mask $\boldsymbol{M}$
$\mathcal{L}$
INR
q^f
w
1.0
0.8
0.6
0.4
0.2

Figure 3. Overview of our proposed framework. Top (a): The left illustrates the overall architecture of our compression system with an INR-based hyperprior, and the right details attribute coding for attributes $\boldsymbol{a}$ and channel-wise autoregressive modeling (CARM). For better visualization, only one INR module is depicted, though two models are employed to compress color and semantic features. Bottom (b): Pipeline of compression-guide segmentation learning, including quantization-aware training and quality-aware weighting. The weighting visualization (bottom right) shows foreground anchors are emphasized with larger weightings while background ones are down-weighted.

predicts the distribution of feature channels conditioned on previously decoded ones, thereby exploiting intra-feature dependencies. In practice, the predicted means $\boldsymbol{\mu}_{ch}$ and variances $\boldsymbol{\sigma}_{ch}$ are added to the initial prediction output by the proposed INR-based hyperprior. We also adopt a view-frequency-aware masking mechanism [7, 43] to prune primitives with little contribution to rendering quality.

4.3. Compression-guided Semantic Learning

We enable contrastive semantic feature learning (Sec. 3.2) in Stage 2. The joint pursuit of compression and segmentation motivates enhancements to the feature learning process. As illustrated in Figure 3b, we introduce two key techniques: quantization-aware training and quality-aware weighting.

Quantization-Aware Training (QAT). Quantization-aware training, a widely adopted technique in learned image and video compression [2, 3, 13], aims to approximate quantization effects during training and enable end-to-end optimization under a rate-distortion objective. In a similar vein, we simulate scalar quantization with step size q^s by injecting uniform noise $\epsilon \sim \mathcal{U}(0, q^s)$ [2] to the learned semantic features $\boldsymbol{s}$ prior to rendering them as a 2D feature map for self-supervised learning. In Stage 2, q^s is set to a fixed value $\mathcal{Q}$ since RD-optimized compression is activated only in Stage 3. In Stage 3, q^s is subsequently refined by a modulation factor predicted by the INR-based hyperprior.

In passing, QAT serves a dual purpose in our framework. Beyond simulating quantization effects for end-to-end optimization, the additive noise has a regularization effect on semantic learning itself. Since the contrastive loss aims to enforce separation between semantic features corresponding to different SAM masks, additive noise implicitly encourages semantic features to be pushed further apart. The resulting features are more robust and remain distinguishable under perturbations or noise (See Sec. 5.3 and supplementary material for an in-depth analysis of this aspect). This result is consistent with the observation made in [21, 42].

Quality-aware Weighting Mechanism (QWM). Existing 3DGS segmentation methods typically optimize semantic features by treating all Gaussian primitives uniformly. However, Gaussian primitives that contribute to the rendering of the background regions are often less well learned and tend to generate poor rendering quality in those regions. They can thus mislead and negatively impact the semantic learning process because SAM masks in these background regions can be inconsistent or even random. This, in turn, has a significant impact on the contrastive learning of foreground semantic features.

We address this issue through a quality-aware weighting mechanism that explicitly accounts for the reliability of each anchor during contrastive learning. Notably, the quantization step size $q^{\boldsymbol{f}}$, estimated by the INR-based hyperprior in Stage 1, serves as a strong proxy for anchor quality. Intuitively, anchors with smaller step sizes tend to be more reliable and are often situated in foreground-like regions, whereas those with larger step sizes are typically less accurate and of lower quality. This relationship establishes a natural link between compression and segmentation.

Formally, we specify a weight w for each anchor according to $q^{\boldsymbol{f}}$ as follows:

$$w = \begin{cases} 1, & \text{if } 1 - \|q^{\boldsymbol{f}}\| \geq \mathcal{T} \\ 1 - \|q^{\boldsymbol{f}}\|, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathcal{T}$ is an empirically chosen threshold and $\|\cdot\|$ denotes a normalization operator that maps its input to an interval [0, 1]. As a result, low-quality background anchors with large step sizes are down-weighted, effectively minimizing their influence and treating them as if they carried little semantic content (see the bottom-right of Figure 3b). With both QAT and QWM, the modulated semantic features rendered onto a 2D image plane for contrastive learning is $\tilde{\boldsymbol{s}} = (\boldsymbol{s} + \epsilon) \cdot w$.

4.4. Training Objectives

Our framework addresses color/appearance learning, semantic learning, and semantic feature compression in three progressive stages. Each stage has a specific training objective.

Stage 1 learns a 3DGS representation to capture color information of the 3D scene in an RD-optimized manner by a typical rate-distortion training objective [7, 43]:

$$\mathcal{L}_{\text{Stage 1}} = \mathcal{L}_{\text{distortion}} + \lambda_{\text{rate}} \left(\mathcal{L}_{\text{rate}} + \lambda_{\text{offset mask}} \mathcal{L}_{\text{offset mask}}\right),$$

where $\mathcal{L}_{\text{distortion}}$ measures rendering quality using a combination of L1 loss and SSIM, and $\mathcal{L}_{\text{rate}}$ denotes the average number of bits required to entropy encode an anchor. The term $\mathcal{L}_{\text{offset mask}}$ promotes spatial sparsity by penalizing redundant offsets. The trade-offs among these factors are governed by the weighting factors λ_{rate} and $\lambda_{\text{offset mask}}$. Notably, $\mathcal{L}_{\text{offset mask}}$ is multiplied further by λ_{rate} to enable more aggressive pruning of anchors at lower bitrates.

With the frozen anchor features, position, scaling, and offsets, Stage 2 introduces semantic features to each anchor and optimizes them for 3D segmentation. We adopt the self-supervised learning scheme from [20, 39], guided by SAM-generated masks. The training objective $\mathcal{L}_{\text{Stage 2}}$ includes both the inter-mask contrastive loss $\mathcal{L}_c$ and intra-mask smoothing loss $\mathcal{L}_s$ in Sec. 3.2. In this stage, no entropy coding is applied.

In the final stage, semantic features are entropy coded using a separate INR hyperprior model. Here, the semantic features and the hyperprior are jointly optimized with another rate-distortion objective that incorporates the estimated bitrate $\mathcal{L}_{\text{feature rate}}$ of semantic features in addition to the feature loss $\mathcal{L}_{\text{Stage 2}}$:

$$\mathcal{L}_{\text{Stage 3}} = \mathcal{L}_{\text{Stage 2}} + \lambda_{\text{feature rate}} \mathcal{L}_{\text{feature rate}},$$

where $\lambda_{\text{feature rate}}$ controls the trade-off between the two terms. Our progressive training optimizes semantic features to minimize transmission overhead while maximizing segmentation quality, without compromising rendering fidelity.

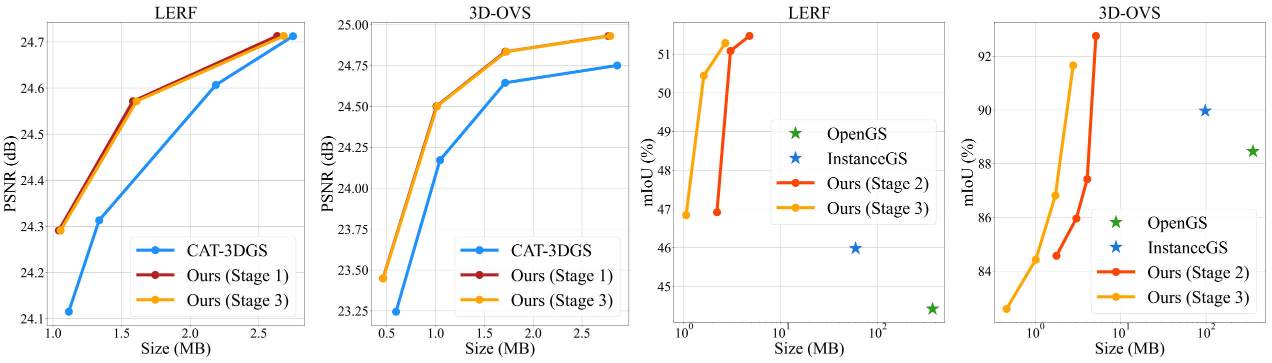
5. Experiments

5.1. Experimental Setup

Datasets and Metrics. We evaluate our proposed 3DGS compression and segmentation framework on two representative datasets: LERF [16] and 3D-OVS [22]. The LERF dataset, designed for 3D object localization, contains complex real-world indoor scenes captured with an iPhone and is annotated by [33]. In contrast, the 3D-OVS dataset focuses on forward-facing scenes targeting open-vocabulary 3D segmentation. Following the evaluation protocols adopted in [20, 30, 39], we evaluate open-vocabulary 3D segmentation with mean Intersection-over-Union (mIoU) and Peak Signal-to-Noise Ratio (PSNR) as metrics for segmentation accuracy and reconstruction quality, respectively. To assess coding efficiency, we also report the compressed file size (bitrate) in megabytes (MB).

Implementation Details. We implement our method based on [25, 43] using the PyTorch framework and train it on a single NVIDIA RTX 4090 GPU. During quantization-aware training in Stage 2, we set the quantization step size $\mathcal{Q}$ to 1 and the threshold $\mathcal{T}$ of quality-aware weighting to 0.65. To evaluate performance across different bitrates, we use a range of rate-control factors λ_{rate}. Specifically, we choose $\lambda_{\text{rate}} = \{0.004, 0.01, 0.02\}$ for the LERF dataset and $\lambda_{\text{rate}} = \{0.004, 0.01, 0.02, 0.05\}$ for the 3D-OVS dataset.

Baselines. Our comparison encompasses two key aspects: reconstruction quality and segmentation performance, both of which also consider transmission costs. For reconstruction, we assess the efficiency of our system with INR-based hyperprior against CAT-3DGS [43] with a triplane hyperprior. We directly quantify the overhead of enabling segmentation at the decoder side by evaluating 3DGS after both Stage 1 (color-only compression) and Stage 3 (with semantic features) of our framework. For segmentation, we compare our method against two recent state-of-the-art 3DGS segmentation methods, OpenGaussian [39] and InstanceGS [20], which do not incorporate RD-optimized compression. To further examine the impact of semantic feature compression, we include Stage 2 results, where semantic features are optimized but left uncompressed, alongside the final Stage 3 results of our method.



(a) Reconstruction Quality

(b) Segmentation Performance

Figure 4. Performance comparison, where the x axis represents bitrate, and y axis represents rendering quality or segmentation accuracy.

5.2. Performance Comparison

Reconstruction Quality. Figure 4a shows that our proposed framework achieves 3D segmentation with marginal bitrate overhead (2-4%) relative to the color-only 3DGS representation (Ours, Stage 1), without compromising reconstruction quality. Compared to CAT-3DGS [43], which relies on a parameter-heavy triplane, our INR-based hyperprior achieves superior RD performance without the need to learn and signal triplanes. As shown in Table 1, our method further reduces average training and decoding times of CAT-3DGS by 30% and 50%, respectively, while maintaining high rendering speed on the LERF dataset. This confirms the superiority of our proposed method in terms of both coding efficiency and computational cost.

Segmentation Performance. As shown in Figure 4b, our framework delivers substantial performance gains over state-of-the-art 3D Gaussian segmentation methods, which do not incorporate RD-optimized compression. Compared to OpenGaussian [39] and InstanceGS [20], our method achieves bitrate reductions of over 140× and 23×, respectively, through efficient entropy coding. Simultaneously, it improves mIoU by 2–5% via compression-guided segmentation learning.

Focusing on semantic feature compression (Stage 2 to Stage 3), our INR-based hyperprior consistently demonstrates high efficiency in compressing additional semantic features—significantly reducing transmission cost while preserving segmentation performance. The key distinction between Stage 2 and Stage 3 lies in their treatment of semantic features: Stage 2 learns uncompressed features, whereas Stage 3 introduces an INR-based hyperprior to encode them in a RD-optimized manner.

5.3. Ablation Studies

We conduct several ablation studies to validate the effectiveness of our proposed compression-guided learning and to justify key hyperparameter choices. All the experiments are conducted on the LERF [16] dataset. Except for the first one, all experiments adopt the same compressed color-only 3DGS (Stage 1) and keep the additional semantic features uncompressed to ensure fair comparisons.

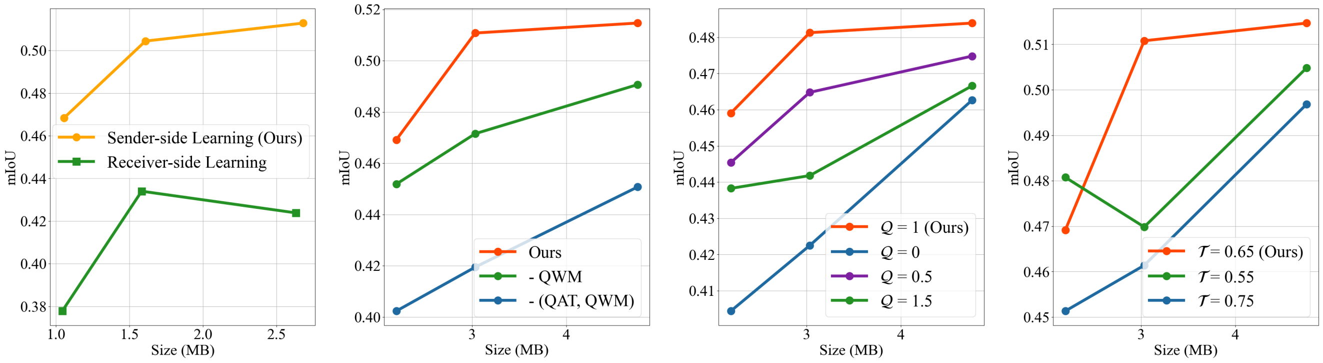
Table 1. Runtime comparison for Stage 1.

Method	Training Time (min.) ↓	Decoding Time (sec.) ↓	Rendering Speed (FPS) ↑
CAT-3DGS	45.8	23.1	198.9
Ours	29.5	11.2	201.5

Comparison with Receiver-side Learning Approach. We present the experimental results for the receiver-side learning scenario depicted in Figure 1b, where SAM masks are generated from the rendered image for segmentation learning. We compare this approach against our proposed sender-side learning framework. Figure 5a shows that the poor quality of rendered images leads to suboptimal SAM masks and significant failures in segmentation performance, highlighting the necessity of our proposed approach.

Compression-guided Semantic Learning. Quantization-Aware Training (QAT) and Quality-Aware Weighting (QWM) leverage compression information for semantic learning. To assess their contributions, we remove each component one by one and report rate-mIoU performance in Figure 5b. As shown, removing QWM (green curve), which serves to mitigate the negative impact of low-quality Gaussian primitives, leads to a 1-4% drop in mIoU across bitrates. Further removing QAT (blue curve) causes an additional 5% drop, confirming its benefits in distinguishing features under contrastive learning. We also provide additional analysis in the supplementary material.

Quantization Step Size. This experiment further examines the impact of applying different quantization step sizes during QAT. We conduct the experiment with QWM disabled to isolate the effects of the quantization alone. In Figure 5c, training without QAT ($\mathcal{Q} = 0$) yields the lowest performance among all variants. Furthermore, a smaller quantization step size ($\mathcal{Q} = 0.5$) limits the potential improvement, while a step



(a) Receiver-side Learning

(b) Compression-guided Learning

(c) Quantization Step Size

(d) Weighting Threshold

Figure 5. Rate-mIoU performance comparison for each ablation study.

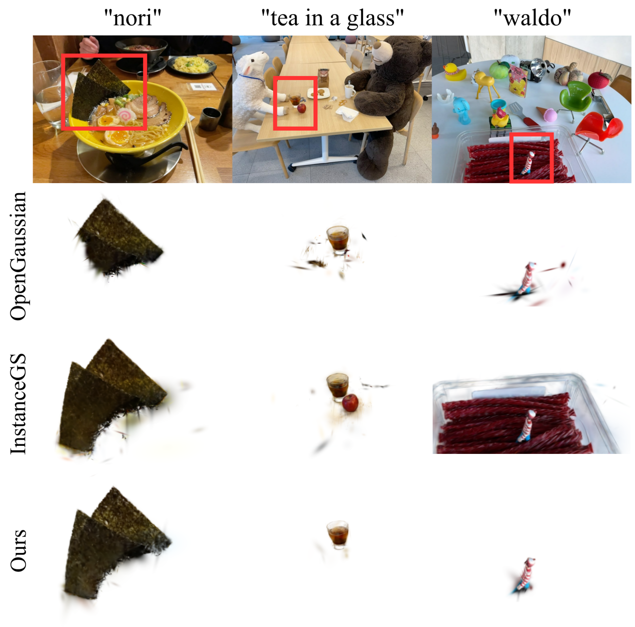


Figure 6. Qualitative comparison of 3D segmentation on LERF.

size too large ($\mathcal{Q} = 1.5$) may introduce excessive noise and harm performance. Our choice of $\mathcal{Q} = 1$ strikes a balance and results in superior segmentation accuracy.

Quality-aware Weighting Threshold. Finally, we evaluate the effect of the threshold $\mathcal{T}$ of QWM, which is used in Eq. 3 to distinguish the high- and low-quality anchors for applying QWM. Figure 5d reports the results across different values of $\mathcal{T}$. An overly high threshold ($\mathcal{T} = 0.75$) may exclude many valid high-quality anchors, while a threshold too low ($\mathcal{T} = 0.55$) insufficiently suppresses low-quality ones. Our chosen threshold of $\mathcal{T} = 0.65$ achieves the best overall mIoU performance, validating its effectiveness.

5.4. Visualization

A qualitative comparison on open vocabulary segmentation with OpenGaussian [39] and InstanceGS [20] is presented in Figure 6. Our method is able to more effectively separate objects with noticeably clearer boundaries. In contrast, InstanceGS tends to include neighboring objects within the same segment (see "waldo"), while OpenGaussian results in incomplete segmentation (see "nori"). In addition to open-vocabulary segmentation, we present the result for click-based object selection of our proposed framework, similarly to [5, 9, 39]. By clicking on an object of a 2D rendered image, we can successfully retrieve the corresponding 3D object, as shown Figure 7. Additional visualizations can be found in the supplementary material.

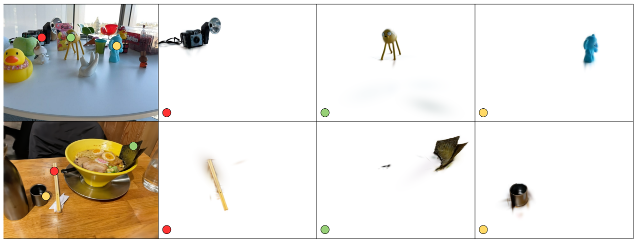


Figure 7. Click-based Object Selection.

6. Conclusion

This work introduces a novel framework for 3DGS that simultaneously addresses rate-distortion compression and segmentation. Our lightweight INR-based hyperprior significantly reduces the size of 3DGS while maintaining high rendering quality and efficient decoding. In addition, the proposed compression-guided segmentation learning further enhances segmentation performance. Together, these contributions enable an efficient and semantically meaningful representation of 3DGS, paving the way for practical downstream applications. We also discuss the limitations of our framework and potential future work in the supplementary material.

7. Acknowledgement

This work is supported by MediaTek Advanced Research Center and National Science and Technology Council (NSTC), Taiwan, under Grants 113-2634-F-A49-007-, 112-2221-E-A49-092-MY3, and 114-2221-E-A49-035-MY3. We thank to National Center for High-performance Computing (NCHC) for providing computational and storage resources.

References

[1] Muhammad Salman Ali, Maryam Qamar, Sung-Ho Bae, and Enzo Tartaglione. Trimming the fat: Efficient compression of 3d gaussian splats through pruning. In *British Machine Vision Conference*, 2024. 3

[2] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017. 3, 5

[3] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018. 3, 5

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021. 3

[5] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. In *the AAAI Conference on Artificial Intelligence*, 2025. 2, 3, 8

[6] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[7] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, 2024. 2, 3, 4, 5, 6

[8] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac++: Towards 100x compression of 3d gaussian splatting. *arXiv preprint arXiv:2501.12255*, 2025. 3, 4

[9] Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. Click-gaussian: Interactive segmentation to any 3d gaussians. In *European Conference on Computer Vision*, 2024. 3, 8

[10] Zhenqi Dai, Ting Liu, and Yanning Zhang. Efficient decoupled feature 3d gaussian splatting via hierarchical compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2, 4

[11] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, Zhangyang Wang, et al. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 2024. 2, 3

[12] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, 2024. 3

[13] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3, 5

[14] Sheng-Yu Huang, Zi-Ting Chou, and Yu-Chiang Frank Wang. 3d gaussian inpainting with depth-guided cross-view consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Conference*, 2025. 2

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 1, 3, 4

[16] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *IEEE/CVF International Conference on Computer Vision*, 2023. 2, 6, 7

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, 2023. 2, 3

[18] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3

[19] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 3

[20] Haijie Li, Yanmin Wu, Jiarui Meng, Qiankun Gao, Zhiyao Zhang, Ronggang Wang, and Jian Zhang. Instancegaussian: Appearance-semantic joint gaussian representation for 3d instance-level perception. In *IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2025. 2, 3, 4, 6, 7, 8

[21] Xuelong Li. Positive-incentive noise. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 5

[22] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. *Advances in Neural Information Processing Systems*, 2023. 2, 6

[23] Lei Liu, Zhenghao Chen, Wei Jiang, Wei Wang, and Dong Xu. Hemgs: A hybrid entropy model for 3d gaussian splatting data compression. *arXiv preprint arXiv:2411.18473*, 2024. 3

[24] Xiangrui Liu, Xinju Wu, Pingping Zhang, Shiqi Wang, Zhu Li, and Sam Kwong. Compgs: Efficient 3d scene representation via compressed gaussian splatting. In *ACM International Conference on Multimedia*, 2024. 3

[25] 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*, 2024. 2, 3, 4, 6

[26] Weijie Lyu, Xueting Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any gaussians via 3d-aware memory bank. *arXiv preprint arXiv:2404.07977*, 2024. 2

[27] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compgs: Smaller and faster gaussian splatting with vector quantization. In *European Conference on Computer Vision*, 2024. 3

[28] 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*, 2024. 3

[29] Qucheng Peng, Benjamin Planche, Zhongpai Gao, Meng Zheng, Anwesa Choudhuri, Terrence Chen, Chen Chen, and Ziyan Wu. 3d vision-language gaussian splatting. In *International Conference on Learning Representations*, 2025. 3

[30] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3, 6

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 2, 3

[32] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 3

[33] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3, 6

[34] Xiangyu Sun, Joo Chan Lee, Daniel Rho, Jong Hwan Ko, Usman Ali, and Eunbyung Park. F-3dgs: Factorized coordinates and representations for 3d gaussian splatting. In *ACM International Conference on Multimedia*, 2024. 3

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017. 4

[36] Henan Wang, Hanxin Zhu, Tianyu He, Runsen Feng, Jiajun Deng, Jiang Bian, and Zhibo Chen. End-to-end rate-distortion optimized 3d gaussian representation. In *European Conference on Computer Vision*, 2024. 2

[37] Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex Kot, and Bihan Wen. Contextgs: Compact 3d gaussian splatting with anchor level context model. *Advances in neural information processing systems*, 2024. 2, 3

[38] Chung-Ho Wu, Yang-Jung Chen, Ying-Huan Chen, Jie-Ying Lee, Bo-Hsu Ke, Chun-Wei Tuan Mu, Yi-Chuan Huang, Chin-Yang Lin, Min-Hung Chen, Yen-Yu Lin, et al. Aurafusion360: Augmented unseen region alignment for reference-based 360deg unbounded scene inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Conference*, 2025. 2

[39] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 2024. 2, 3, 4, 6, 7, 8

[40] Hanyuan Xiao, Yingshu Chen, Huajian Huang, Haolin Xiong, Jing Yang, Pratusha Prasad, and Yajie Zhao. Localized gaussian splatting editing with contextual awareness. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025. 2

[41] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*, 2024. 2

[42] Xiaowei Yu, Zhe Huang, Yao Xue, Lu Zhang, Li Wang, Tianming Liu, and Dajiang Zhu. Noisynn: Exploring the influence of information entropy change in learning systems. *arXiv preprint arXiv:2309.10625*, 2023. 5

[43] Yu-Ting Zhan, Cheng-Yuan Ho, Hebi Yang, Yi-Hsin Chen, Jui Chiu Chiang, Yu-Lun Liu, and Wen-Hsiao Peng. Cat-3dgs: A context-adaptive triplane approach to rate-distortion-optimized 3dgs compression. In *International Conference on Learning Representations*, 2025. 2, 3, 4, 5, 6, 7

[44] Yu-Ting Zhan, He-bi Yang, Cheng-Yuan Ho, Jui-Chiu Chiang, and Wen-Hsiao Peng. Cat-3dgs pro: A new benchmark for efficient 3dgs compression. *arXiv preprint arXiv:2503.12862*, 2025. 3, 4

[45] Can Zhang and Gim Hee Lee. econsg: Efficient and multi-view consistent open-vocabulary 3d semantic gaussians. In *International Conference on Learning Representations*, 2025. 3

[46] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3