Binary-Gaussian: Compact and Progressive Representation for 3D Gaussian Segmentation

An Yang[1], Chenyu Liu[2], Jun Du[1*], Jianqing Gao[2], Jia Pan[2], Jinshui Hu[2], Baocai Yin[2], Bing Yin[2], Cong Liu[2]

[1]NERC-SLIP, University of Science and Technology of China
[2]iFLYTEK Research
yangan2002@mail.ustc.edu.cn, jundu@ustc.edu.cn, {cyliu7,jqgao,jiapan,jshu,bcyin,bingyin,congliu2}@iflytek.com

Abstract

3D Gaussian Splatting (3D-GS) has emerged as an efficient 3D representation and a promising foundation for semantic tasks like segmentation. However, existing 3D-GS-based segmentation methods typically rely on high-dimensional category features, which introduce substantial memory overhead. Moreover, fine-grained segmentation remains challenging due to label space congestion and the lack of stable multi-granularity control mechanisms. To address these limitations, we propose a coarse-to-fine binary encoding scheme for per-Gaussian category representation, which compresses each feature into a single integer via the binary-to-decimal mapping, drastically reducing memory usage. We further design a progressive training strategy that decomposes panoptic segmentation into a series of independent sub-tasks, reducing inter-class conflicts and thereby enhancing fine-grained segmentation capability. Additionally, we fine-tune opacity during segmentation training to address the incompatibility between photometric rendering and semantic segmentation, which often leads to foreground-background confusion. Extensive experiments on multiple benchmarks demonstrate that our method achieves state-of-the-art segmentation performance while significantly reducing memory consumption and accelerating inference.

1 Introduction

Understanding and manipulating 3D scenes—including representation, rendering, perception, and editing—is a long-standing goal in computer vision and graphics, supporting applications such as virtual reality, robotics, and digital twins. 3D Gaussian Splatting(3D-GS) (Kerbl et al. 2023a), with its discrete spatial structure and fast rasterization, is becoming a mainstream approach for 3D scene representation, offering significant advantages in real-time interaction (Choi et al. 2024; Zhao et al. 2025) and scene editing (Yan et al. 2024; Ye et al. 2024; Qu et al. 2025). For such applications, accurately and efficiently segmenting objects is important (Jiang et al. 2024). However, this remains a challenging problem, especially in complex environments where objects exhibit occlusion, varying scales, and weak boundaries.

Recently, various segmentation methods (Ye et al. 2024; Shen, Yang, and Wang 2024; Choi et al. 2024; Cen et al. 2025) based on 3D-GS have been proposed. To represent category information, each Gaussian is usually assigned an additional continuous-valued vector. These approaches learn per-Gaussian category features from multi-view 2D mask annotations using techniques such as global optimization or contrastive learning. During inference, segmentation is achieved either by taking the argmax over class probabilities, or by performing clustering in the feature space. However, the additional feature vector assigned to each Gaussian (typically 32 dimensions) accounts for over 50% of the original parameter size (62 dimensions), resulting in significant memory overhead. In addition, methods that support multi-granularity segmentation, such as SAGA (Cen et al. 2025), control the segmentation scale by simply multiplying a scale-controlling embedding with the category features. However, this approach is unstable and struggles to deliver accurate results for fine-grained segmentation.

To alleviate these issues, we first replace continuous category features with binary codes. By leveraging the one-to-one mapping between binary vectors and decimal integers, each Gaussian's category feature can be compactly stored as a single integer, significantly reducing the additional memory overhead introduced by segmentation. We further adopt a coarse-to-fine multi-granularity representation and progressive training strategy. Specifically, the binary code is divided into multiple partitions, each encoding feature information for a specific granularity level. The segmentation at a finer level depends not only on the current partition but also on all coarser-level partitions. In this hierarchical structure, coarse-level features determine broad category assignments, while finer-level features further subdivide those categories—akin to a multi-level indexing scheme. This design decomposes the complex segmentation task into a sequence of simpler, localized sub-problems, thereby reducing training complexity and significantly enhancing the model's ability to distinguish fine-grained categories. Moreover, existing models typically inherit opacity from photometric reconstruction and keep it fixed, which can cause low-opacity foreground objects to be misclassified as background. This issue is amplified under discrete representations like binary encoding, so we fine-tune opacity to better align rendering behavior with semantic segmentation objectives. Finally, our model achieves superior segmentation accuracy while significantly reducing storage overhead and accelerating infer-

*Corresponding author.

ence, outperforming existing state-of-the-art methods across multiple benchmarks. In summary, our main contributions are as follows:

- We introduce binary codes to represent category features, reducing the parameter overhead of segmentation to just 1.6% and achieving over 700 FPS in segmentation rendering.
- We propose a coarse-to-fine multi-granularity representation and progressive training strategy, which simplifies the segmentation task and improves overall accuracy, especially for fine-grained segmentation.
- We identify the incompatibility between opacity in photometric and semantic reconstruction, and address it by fine-tuning opacity during segmentation training, which improves foreground-background separation.
- Extensive qualitative and quantitative experiments across diverse scenes demonstrate that our method consistently outperforms existing state-of-the-art approaches in segmentation quality.

2 Related Work

3D Segmentation in Radiance Fields. The integration of semantic segmentation into radiance field representations has become an active area of research, driven by the increasing demand for 3D scene understanding and object-level interaction. Initial work (Kobayashi, Matsumoto, and Sitzmann 2022; Tschernezki et al. 2022; Goel et al. 2023; Kerr et al. 2023) on 3D segmentation in radiance fields often relied on lifting visual features from 2D foundation models into 3D. Methods such as DFFs (Kobayashi, Matsumoto, and Sitzmann 2022), N3F (Tschernezki et al. 2022), and ISRF (Goel et al. 2023) adopt feature distillation pipelines, transferring representations from self-supervised 2D models like DINO (Caron et al. 2021) into 3D fields. However, since these models are not designed for segmentation, their transferred features tend to lack spatial precision. NeRF-SOS (Fan et al. 2022) and ContrastiveLift (Bhalgat et al. 2023) distill 2D pixel or region level feature similarities into 3D representations, introducing contrastive learning as a new training paradigm. Furthermore, some work (Zhou et al. 2024; Chen et al. 2023) leverages the Segment Anything Model (Kirillov et al. 2023), distilling its encoder features into 3D Gaussians while using the decoder for segmentation inference. Despite improvements in segmentation quality, such methods often incur high computational costs.

Another line of work formulates 3D segmentation as a mask-lifting problem, where 2D masks are projected into 3D space across multiple views. Examples include SA3D (Cen et al. 2023), NVOS (Ren et al. 2022), MVSeg (Mirzaei et al. 2023), FlashSplat (Shen, Yang, and Wang 2024), and GaussianGrouping (Ye et al. 2024), which rely on user prompts or video tracking to align masks and guide segmentation. While effective in certain settings, these approaches are vulnerable to view inconsistency and often assume ideal camera trajectories or tracking conditions. More recent methods like OmniSeg3D (Ying et al. 2023) and GARField (Kim et al. 2024) mitigate these issues by introducing contrastive learning schemes and scale-conditioned features, but they typically build on NeRF-based architectures, which are computationally expensive and not well-suited to real-time applications. Later works (Choi et al. 2024; Cen et al. 2025) introduce contrastive learning into 3D Gaussians segmentation. Click-Gaussian (Choi et al. 2024) improves segmentation precision through coarse-to-fine global-guided learning, yet it is limited to dual-scale segmentation. SAGA (Cen et al. 2025) introduces scale-gated affinity, similar to GARField, for adaptive granularity control, but its simple gating design-based on a single fully connected layer-limits precision and stability. In contrast, our method employs a coarse-to-fine representation strategy to achieve stable multi-granularity segmentation.

Compression for 3D Gaussian Splatting. 3D-GS (Kerbl et al. 2023b) enables real-time rendering with high quality, but suffers from high memory usage, as each Gaussian stores multiple attributes, often resulting in gigabyte-scale scenes and reduced rendering efficiency (Fan et al. 2024). To address this, recent methods compress 3D Gaussian Splatting by discretizing Gaussian attributes via codebook-based vector quantization. CompGS (Navaneet et al. 2024) reduces redundancy via opacity regularization and K-means quantization. LightGaussian (Fan et al. 2024) integrates global significance-based pruning, SH distillation, and vector quantization into a unified compression framework. HAC (Chen et al. 2024) introduces a hash-grid-assisted context model for entropy-aware compression and spatially consistent pruning. Papantonakis et al. (2024) additionally prune resolution-insensitive Gaussians, adaptively reduce SH coefficients, and compress attributes via codebook quantization and half-precision representation. Other efforts adopt either sensitivity-aware clustering with quantization-aware training (Niedermayr, Stumpfegger, and Westermann 2024), or grid-based color fields as a replacement for spherical harmonics (Lee et al. 2024), to reduce memory usage. The above methods focus on compressing appearance-related parameters of 3D Gaussians. However, segmentation tasks require assigning additional attributes to each Gaussian to represent category information. Previous segmentation approaches typically rely on high-dimensional continuous features to encode such information, which significantly increases memory consumption. Unlike prior methods, our proposed binary feature encodes category semantics as compact binary codes, which can be stored using an integer (e.g., a 32-bit integer), substantially reducing the storage overhead for 3D-GS-based segmentation models.

3 Method

3.1 Preliminary: 3D Gaussian Splatting

3D Gaussian Splatting(3D-GS) represents a 3D scene with a collection of explicit 3D Gaussians and renders images through a differentiable rasterizer (Kerbl et al. 2023b). Given a training set of images $\mathcal{I} = \{I^v\}_{v=1}^V$ with camera poses, 3D-GS learns a set of 3D Gaussians $G = \{g_i\}_{i=1}^N$, where V is the number of views and N denotes the total number of Gaussians. Each Gaussian g_i is parameterized by its 3D position and covariance, opacity, and color represented using spherical harmonics coefficients. Given a specific cam-

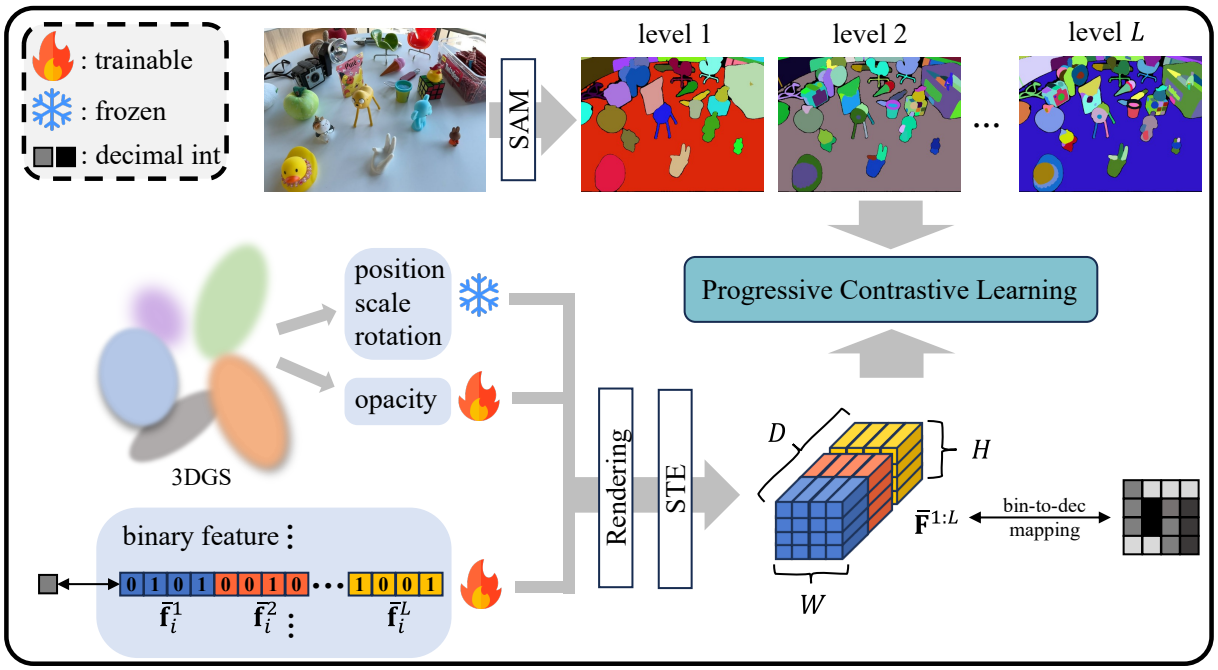


Figure 1: Overview of Binary-Gaussian. Our method builds upon pre-trained 3D Gaussians by augmenting each Gaussian with multi-granularity binary features. During training, we keep the opacity values learnable and optimize the multi-level binary features $\bar{\mathbf{F}}^{1:L}$ on the rendered 2D projections using a progressive contrastive learning strategy.

era pose, the color of pixel p is computed by projecting the Gaussians onto the 2D image plane and blending a depth-ordered subset $\mathcal{N}$ overlapping the pixel p:

$$C_p = \sum_{i \in \mathcal{N}} c_i \alpha_i T_i, \tag{1}$$

where c_i is the color of each Gaussian and α_i is given by evaluating a 2D Gaussian with covariance Σ multiplied with a learned per-Gaussian opacity (Kerbl et al. 2023b), and $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$ is the transmittance.

3.2 Coarse-to-Fine Binary Feature

Different from previous methods that adopt continuous values, we equip each 3D Gaussian with a binary code to represent its segmentation category. To support multi-granularity segmentation, we further introduce a coarse-to-fine representation scheme, where each Gaussian is associated with a set of hierarchical codes that capture segmentation labels at different levels of granularity.

Specifically, let L denote the maximum level of segmentation granularity. We assign L feature vectors $\{\bar{\mathbf{f}}_i^l \in \{0,1\}^{D_l} \mid l = 1, 2, \ldots, L\}$ to each Gaussian to represent features at different levels, which are then concatenated into a single feature vector $\bar{\mathbf{f}}_i \in \mathbb{R}^D$ for efficient rendering computation, where $D = \sum_{l=1}^{L} D_l$. Using the rasterizer, we can render the features of different granularity levels to the 2D pixel p:

$$\mathbf{F}_p = \sum_{i \in \mathcal{N}} \bar{\mathbf{f}}_i \alpha_i T_i, \tag{2}$$

Then, we apply a straight-through estimator (STE) to obtain binary features:

$$\bar{\mathbf{F}}_p = \text{stop\_grad}\left(\mathbb{I}(\mathbf{F}_p > 0.5) - \mathbf{F}_p\right) + \mathbf{F}_p \tag{3}$$

where $\mathbb{I}(\cdot)$ is the element-wise indicator function that outputs 1 if the input condition is true and 0 otherwise, and stop_grad$(\cdot)$ represents the stop-gradient operation that blocks gradients during backpropagation. $\bar{\mathbf{F}}_p \in \{0,1\}^D$ is the final binarized feature vector, and $\bar{\mathbf{F}}_p^l = \bar{\mathbf{F}}_p[\sum_{k=1}^{l-1} D_k : \sum_{k=1}^{l} D_k]$ denotes the binary features corresponding to the l-th level, with $l = 1, 2, \ldots, L$.

The coarse-to-fine design is realized by progressively aggregating features across granularity levels. Specifically, for each granularity level l, we concatenate the binary feature vectors from all levels less than or equal to l to form a unified feature for segmentation:

$$\bar{\mathbf{F}}_p^{1:l} = \text{Concat}(\bar{\mathbf{F}}_p^1, \bar{\mathbf{F}}_p^2, \ldots, \bar{\mathbf{F}}_p^l). \tag{4}$$

3.3 Progressive Contrastive Learning

We propose a novel progressive contrastive learning strategy tailored to multi-granularity binary features in a coarse-to-fine hierarchy. We next describe this strategy progressively. Let M_p^l denote the mask that pixel p belongs to at granularity level l.

Standard Contrastive Learning. We first adopt a standard contrastive learning approach for the level-1 features. For pixels p and q, if they belong to the same level-1 mask,

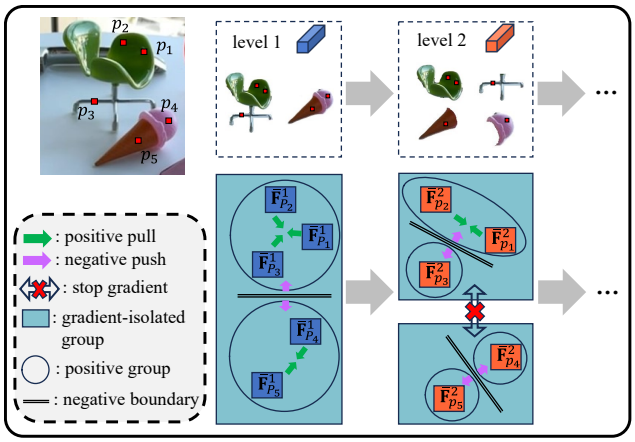


Figure 2: Illustration of Progressive Contrastive Learning. Fine-grained segmentation at each level is built on the coarser segmentation from the preceding level. Each positive group identified at the coarse level forms an independent, gradient-isolated sub-optimization task at the finer level.

we minimize the L2 distance between their level-1 features. Otherwise, we maximize the L2 distance:

$$\mathcal{L}_{\text{level-1}} = \begin{cases} \|\bar{\mathbf{F}}_p^1 - \bar{\mathbf{F}}_q^1\|_2, & \text{if } M_p^1 = M_q^1 \\ D_1 - \|\bar{\mathbf{F}}_p^1 - \bar{\mathbf{F}}_q^1\|_2, & \text{otherwise.} \end{cases} \quad (5)$$

Our objective is to ensure that $\bar{\mathbf{F}}_p^1 = \bar{\mathbf{F}}_q^1$ when $M_p^1 = M_q^1$, and that $\bar{\mathbf{F}}_p^1 \neq \bar{\mathbf{F}}_q^1$ when $M_p^1 \neq M_q^1$.

Progressive Contrastive Learning. Following the logic of mathematical induction, we assume that the level $l-1$ segmentation (with $l \geq 2$) has already been constrained. More specifically, the features $\bar{\mathbf{F}}^{1:l-1}$ are subject to constraints similar to Equation 5, ensuring that $\bar{\mathbf{F}}_p^{1:l-1} = \bar{\mathbf{F}}_q^{1:l-1}$ when $M_p^{l-1} = M_q^{l-1}$, and that $\bar{\mathbf{F}}_p^{1:l-1} \neq \bar{\mathbf{F}}_q^{1:l-1}$ when $M_p^{l-1} \neq M_q^{l-1}$.

For the segmentation at level l, we need to impose constraints to ensure that $\bar{\mathbf{F}}_p^{1:l} = \bar{\mathbf{F}}_q^{1:l}$ when $M_p^l = M_q^l$, and that $\bar{\mathbf{F}}_p^{1:l} \neq \bar{\mathbf{F}}_q^{1:l}$ when $M_p^l \neq M_q^l$. if $M_p^l = M_q^l$, then it necessarily follows that $M_p^{l-1} = M_q^{l-1}$. According to our inductive assumption, constraints have already been applied such that $\bar{\mathbf{F}}_p^{1:l-1} = \bar{\mathbf{F}}_q^{1:l-1}$. Therefore, we only need to minimize the L2 distance between their level l features $\bar{\mathbf{F}}_p^l$ and $\bar{\mathbf{F}}_q^l$, as shown by p_1 and p_2 at level 2 in Figure 2. if $M_p^l \neq M_q^l$, we further consider two cases:

$$(1) M_p^{l-1} \neq M_q^{l-1}; \qquad (2) M_p^{l-1} = M_q^{l-1}.$$

For case (1), according to our inductive assumption, constraints have already been applied such that $\bar{\mathbf{F}}_p^{1:l-1} \neq \bar{\mathbf{F}}_q^{1:l-1}$. Due to the inheritance property of the coarse-to-fine feature representation, it must hold that $\bar{\mathbf{F}}_p^{1:l} \neq \bar{\mathbf{F}}_q^{1:l}$. Therefore, we do not impose inconsistent constraints on level l features under this case, as shown by p_1 and p_4 at level 2 in Figure 2. For case (2), we have the constraint $\bar{\mathbf{F}}_p^{1:l-1} = \bar{\mathbf{F}}_q^{1:l-1}$. Hence, we maximize the L2 distance between their level l features $\bar{\mathbf{F}}_p^l$ and $\bar{\mathbf{F}}_q^l$ to induce separation, resulting in $\bar{\mathbf{F}}_p^{1:l} \neq \bar{\mathbf{F}}_q^{1:l}$, as shown by p_1 and p_3 at level 2 in Figure 2. Formally, we have:

$$\mathcal{L}_{\text{level-}l} = \begin{cases} \|\bar{\mathbf{F}}_p^l - \bar{\mathbf{F}}_q^l\|_2, & \text{if } M_p^l = M_q^l \\ D_l - \|\bar{\mathbf{F}}_p^l - \bar{\mathbf{F}}_q^l\|_2, & \text{else if } M_p^{l-1} = M_q^{l-1} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

This completes our inductive reasoning. It demonstrates that by applying the set of constraints $\sum_{l=1}^{L} \mathcal{L}_{\text{level-}l}$ derived at each granularity level, we can optimize the multi-scale segmentation task. Notably, the above derivations for different cases rely on ground-truth labels, and there are no dependencies between features at different levels during training. As a result, all levels can be optimized simultaneously. Our progressive contrastive learning decomposes the fine-grained segmentation task into a sequence of simpler subtasks, thereby reducing the complexity of the overall problem and significantly alleviating the overlap and confusion between features of different object categories.

3.4 Virtual Negative Guidance for Positive-Only Contrast

Contrastive learning depends on the equilibrium between positive attraction and negative repulsion. Disruptions to this balance have emerged as a key issue, motivating a wide range of research efforts to explore effective solutions (Grill et al. 2020; Chen and He 2021; Cha, Cho, and Moon 2023). In our progressive contrastive framework, this issue arises when a semantic mask becomes indivisible. At the next finer level, the resulting gradient-isolated group contains only positive pairs and receives no repulsive supervision. Counterintuitively, we observe that pull-only objectives in such cases often lead to feature drift or inconsistency, rather than feature collapse (as shown in Figure 4 of ablation study). To address this, we propose a simple yet effective solution: guiding feature learning using an all-one vector as the virtual negative. Specifically, we impose an L2 regularization constraint on each pixel feature $\bar{\mathbf{F}}_p^l$ within a positive-only group at level l, aiming to maximize its distance from the virtual negative. Formally, we have:

$$\mathcal{L}_{\text{VN-}l} = \|\bar{\mathbf{F}}_p^l\|_2. \quad (7)$$

Loss Function. In addition to the aforementioned contrastive loss, we introduce a regularization term on the pre-binarized features $\mathbf{F}_p$, encouraging them to approach binary values (0 or 1) for more semantically deterministic representations. The regularization term is defined as:

$$\mathcal{L}_{\text{reg}} = \|\mathbb{I}(\mathbf{F}_p > 0.5) - \mathbf{F}_p\|_2^2 \quad (8)$$

The final loss function is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{guiding}} \sum_{l}^{L} \mathcal{L}_{\text{VN-}l} + \lambda_{\text{con}} \sum_{l}^{L} \mathcal{L}_{\text{level-}l} \quad (9)$$

To balance the magnitudes of different loss terms, we set $\lambda_{\text{reg}} = 10$, $\lambda_{\text{guiding}} = 1$, and $\lambda_{\text{con}} = 1$.

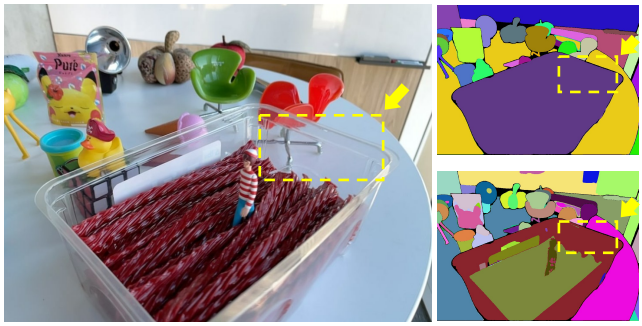


Figure 3: Opacity inconsistency between visual and segmentation rendering.

3.5 Additional Training Strategy

Counter-Visual Opacity Effects in Segmentation. Opacity, a fundamental attribute of each Gaussian, indicates its light-blocking strength and determines its contribution to the final pixel appearance through alpha blending. In prior 3D Gaussian segmentation methods, opacity values are inherited from color reconstruction and kept fixed during training. However, the opacity in color space does not always align with the semantics required for accurate segmentation. A typical failure case arises when handling semi-transparent objects, such as a plastic box, as shown in Figure 3. In the color space, the box is rendered with low opacity, allowing the background (e.g., a table) to be partially visible. However, when used in segmentation, low-opacity foreground Gaussians may contribute too little to dominate the pixel. This issue is further exacerbated under discrete representations such as binary encoding. Unlike continuous features—which can compensate for low-opacity Gaussians by amplifying their feature values—our discrete encoding is bounded in range. As a result, low-opacity Gaussians are easily suppressed during feature rendering, causing insufficient optimization and foreground-background confusion. Therefore, we propose to fine-tune the opacity values during category feature training.

Mask-Balanced Sampling. In previous methods, each training iteration randomly samples N_p pixels from an image to form $N_p \times N_p$ pixel pairs. This inevitably leads to data imbalance, where large masks occupying more pixels dominate the optimization, degrading segmentation performance for small masks (Cen et al. 2025). To address this issue, we randomly sample only a portion of pixels from the entire image. For the remaining pixels, we adopt a Mask-Balanced Pixel Sampling strategy: we randomly select a set of masks, and then sample a fixed number of pixels from each selected mask. This strategy increases the proportion of sampling pixels from small masks.

3.6 Inference

With the well-trained Gaussian binary features $\bar{\mathbf{f}}_i^{1:l}$, we can render the corresponding pixel-level binary feature map $\bar{\mathbf{F}}_p^{1:l}$ from arbitrary viewpoints. By applying binary-to-decimal mapping, we quickly obtain decimal integer labels for both $\bar{\mathbf{f}}_i^{1:l}$ and $\bar{\mathbf{F}}_p^{1:l}$:

$$\text{Class}_i^l = \text{Bin2Dec}(\bar{\mathbf{f}}_i^{1:l}), \text{Class}_p^l = \text{Bin2Dec}(\bar{\mathbf{F}}_p^{1:l}),$$

where $\text{Bin2Dec}(\cdot)$ denotes the mapping function from binary to decimal. Class_i^l and Class_p^l indicate the semantic category at granularity level l, used for 3D Gaussian segmentation and multi-view 2D segmentation, respectively. By associating these decimal values with each semantic object annotation, we can retrieve all Gaussians sharing the same specific value, thereby enabling object-specific 3D segmentation.

4 Experiments

4.1 Datasets

Following prior approaches (Choi et al. 2024; Cen et al. 2025), we employ a SAM-based automatic mask generation module to obtain segmentation annotations at multiple granularity levels within 3D scenes. To evaluate our model's 3D segmentation performance, we adopt two widely used public real-world datasets: the LERF-Mask dataset (Choi et al. 2024) and the SPIn-NeRF dataset (Mirzaei et al. 2023). The LERF-Mask dataset comprises three scenes (Kerr et al. 2023) with manually annotated ground truth masks for both coarse and fine-grained segmentation. The SPIn-NeRF dataset provides multi-view masks for single objects in the widely used NeRF datasets (Fridovich-Keil et al. 2022; Knapitsch et al. 2017; Mildenhall et al. 2019, 2021; Yen-Chen et al. 2022).

4.2 Implementation Details

We train the category features based on the 3D Gaussian parameters obtained from color-space reconstruction (Kerbl et al. 2023a). During this process, the learning rate for opacity is set to 0.001 (2% of the initial value) for fine-tuning, while the learning rate for category features is set to 0.005. In each iteration, 2k pixels are randomly sampled, and an additional 8k pixels are selected using mask-balanced sampling. More training details are provided in Appendix A.

4.3 Quantitative Evaluation

We evaluate our method on the LERF-Mask dataset under the setting of multi-object, multi-granularity segmentation, with the results from a multi-perspective evaluation summarized in Table 1. Firstly, leveraging the binary coding design, the category feature of each Gaussian occupies only 32 bits—merely 1.6% of the original parameter size—significantly lower than competing methods. Moreover, while previous approaches rely on clustering to construct category centers and compute distances between Gaussian features and all centers during inference, our model simply maps the binary code to a decimal index for direct category identification. This enables inference speeds over 10× faster than prior methods, dramatically accelerating the rendering process. Despite its lightweight and efficient design, our model also maintains high segmentation

Model	Extra Param / Gau		Multi-granularity Support	FPS	mIoU (%)	
	Size(bit)	Growth(%)			Coarse	Fine
OmniSeg3D (Ying et al. 2023)	-	-	✓✓✓	22.6	79.4	46.9
GARField (Kim et al. 2024)	-	-	✓✓✓	0.32	80.9	71.4
Gau-Group (Ye et al. 2024)	1024	51.6	✗	288	72.8	31.5
Feature3DGS (Zhou et al. 2024)	4096	206.5	✗	14.6	65.6	63.5
Click-Gaussian (Choi et al. 2024)	1024	51.6	✓	≈ 50–100	**89.1**	84.3
SAGA (Cen et al. 2025)	1024	51.6	✓✓✓	67.9	83.7	48.1
SAGA* (Cen et al. 2025)	32	1.6	✓✓✓	74.6	69.7	47.8
Ours	**32**	**1.6**	✓✓✓	**769**	87.1	**86.2**

Table 1: Comparison with baselines on LERF-Mask dataset in terms of parameter overhead, segmentation granularity, speed, and quality. The top two rows correspond to NeRF-based methods, and the bottom six rows to 3D-GS-based methods. SAGA* denotes the variant with parameter compression via codebook quantization. Symbol definitions: ✗ supports only a single segmentation granularity; ✓ supports two levels (coarse and fine); ✓✓✓ supports multi-granularity segmentation.

Method	mIoU (%)
MVSeg (Mirzaei et al. 2023)	90.9
SA3D (Cen et al. 2023)	92.4
SAGA (Cen et al. 2025)	88.0
Click-Gaussian (Choi et al. 2024)	94.0
Ours	**94.3**

Table 2: Quantitative comparison with baselines on the SPIn-NeRF dataset. We report the average mIoU across ten real-world scenes. Both MVSeg and SA3D are limited to segmenting a single object per training session.

	mIoU (%)	
	Coarse	Fine
Full	87.1	86.2
w/o Virtual Negative	85.3	77.9
w/o Opacity Fine-tuning	86.0	84.4
w/o MB Sampling	84.5	76.4

Table 3: Ablation results for multi-granularity segmentation

quality, achieving the best performance in fine-grained segmentation. The results on the SPIn-NeRF dataset across diverse real-world scenes, as shown in Table 2, further demonstrate the superiority of our method, which achieves state-of-the-art segmentation accuracy.

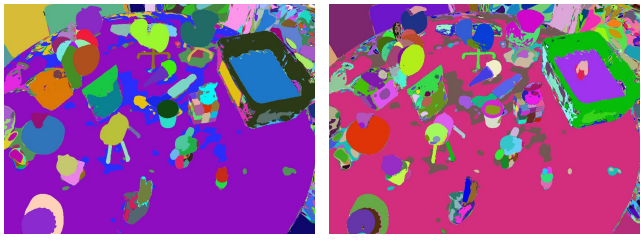
4.4 Qualitative Evaluation

In Figure 5, we qualitatively evaluate the segmentation performance of different models on novel views. Our model achieves high-quality segmentation, particularly under fine-grained settings—for example, it produces more precise contours for the egg yolk. For semi-transparent objects such as the glass cup in the first row and the plastic box in the bottom three rows, baseline models exhibit foreground-background confusion. In contrast, our model benefits from opacity fine-tuning and delivers superior segmentation quality in these challenging cases. Moreover, our model achieves more stable control over segmentation granularity. In the panoptic segmentation results, baseline models show poor sensitivity to granularity and even exhibit inconsistent or confused segmentations. By comparison, our model accurately controls segmentation granularity and exhibits smooth transitions across different levels. Finally, as shown in the finest-grained segmentation results in the last row, only our model successfully segments subtle object parts, such as the beak of bird, the duck eye, bunny ears, and the recessed area on the barrel—highlighting the significant advantage of Binary-Gaussian in fine-grained segmentation and validating the effectiveness of our progressive contrastive learning strategy. More visualization examples are provided in Appendix B.

4.5 Ablation Study

The ablation results for the virtual negative, opacity fine-tuning, and mask-balanced sampling are presented in Table 3.

Virtual Negative. Virtual negative is introduced to enhance consistency optimization for fine-grained segmentation, and removing this strategy leads to a significant drop in fine-grained segmentation performance. Figure 4 presents a qualitative comparison, where we observe that for masks with limited granularity (e.g., table), the absence of virtual negatives causes the features to fail to form consistent representations at fine-grained levels.



level 2 level 3

Figure 4: Segmentation results without the virtual negative across different granularities.

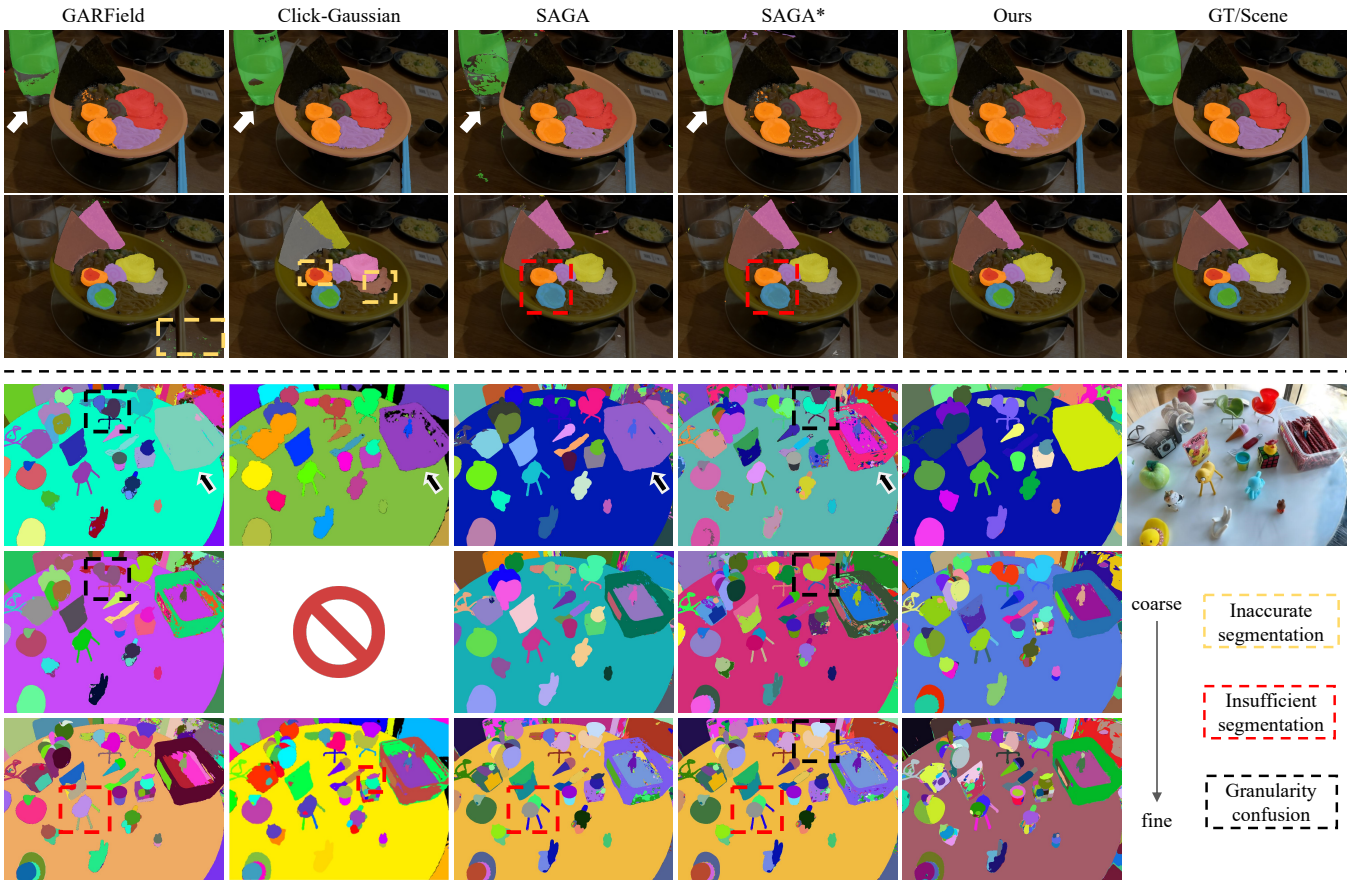


Figure 5: Comparison with baseline models on the LERF-Mask dataset. The top two rows illustrate object-specific segmentation results for coarse and fine-grained targets. The bottom three rows present panoptic segmentation results across different levels of granularity. The prohibition symbol indicates that Click-Gaussian lacks support for intermediate granularity.

Ours w/o opacity f-t

Figure 6: Ablation study on opacity fine-tuning for semi-transparent object segmentation.

Opacity Fine-tuning. Since widely used benchmark datasets rarely contain objects with high transparency, opacity fine-tuning does not lead to significant improvements in quantitative metrics. To better demonstrate its effect, we provide a visual comparison in Figure 6, where opacity fine-tuning noticeably improves the segmentation quality of semi-transparent objects.

Mask-Balanced Sampling. We adopt Mask-Balanced Sampling to address the under-sampling issue of small-sized masks. As shown in Table 3, this strategy brings limited improvements to coarse segmentation but leads to notable gains in fine-grained segmentation, where the masks are typically smaller in size.

5 Conclusion

We propose a coarse-to-fine binary feature encoding scheme for 3D-GS-based segmentation, along with a progressive contrastive learning strategy tailored to this task. To address various challenges in this domain, we further introduce three training techniques: virtual negative, opacity fine-tuning, and mask-balanced sampling. Finally, with minimal parameter overhead and high inference speed, our method achieves excellent segmentation performance, especially at fine-grained levels.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. U25A20409.

References

Bhalgat, Y.; Laina, I.; Henriques, J. F.; Zisserman, A.; and Vedaldi, A. 2023. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. *arXiv preprint arXiv:2306.04633*.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9650–9660.

Cen, J.; Fang, J.; Yang, C.; Xie, L.; Zhang, X.; Shen, W.; and Tian, Q. 2025. Segment any 3d gaussians. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 1971–1979.

Cen, J.; Zhou, Z.; Fang, J.; Shen, W.; Xie, L.; Jiang, D.; Zhang, X.; Tian, Q.; et al. 2023. Segment anything in 3d with nerfs. *Advances in Neural Information Processing Systems*, 36: 25971–25990.

Cha, S.; Cho, K.; and Moon, T. 2023. Regularizing with pseudo-negatives for continual self-supervised learning. *arXiv preprint arXiv:2306.05101*.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15750–15758.

Chen, X.; Tang, J.; Wan, D.; Wang, J.; and Zeng, G. 2023. Interactive segment anything nerf with feature imitation. *arXiv preprint arXiv:2305.16233*.

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*, 422–438. Springer.

Choi, S.; Song, H.; Kim, J.; Kim, T.; and Do, H. 2024. Click-gaussian: Interactive segmentation to any 3d gaussians. In *European Conference on Computer Vision*, 289–305. Springer.

Fan, Z.; Wang, K.; Wen, K.; Zhu, Z.; Xu, D.; Wang, Z.; et al. 2024. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37: 140138–140158.

Fan, Z.; Wang, P.; Jiang, Y.; Gong, X.; Xu, D.; and Wang, Z. 2022. Nerf-sos: Any-view self-supervised object segmentation on complex scenes. *arXiv preprint arXiv:2209.08776*.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5501–5510.

Goel, R.; Sirikonda, D.; Saini, S.; and Narayanan, P. 2023. Interactive segmentation of radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4201–4211.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284.

Jiang, Y.; Yu, C.; Xie, T.; Li, X.; Feng, Y.; Wang, H.; Li, M.; Lau, H.; Gao, F.; Yang, Y.; et al. 2024. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, 1–1.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023a. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

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

Kerr, J.; Kim, C. M.; Goldberg, K.; Kanazawa, A.; and Tancik, M. 2023. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19729–19739.

Kim, C. M.; Wu, M.; Kerr, J.; Goldberg, K.; Tancik, M.; and Kanazawa, A. 2024. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21530–21539.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026.

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): 1–13.

Kobayashi, S.; Matsumoto, E.; and Sitzmann, V. 2022. Decomposing nerf for editing via feature field distillation. *Advances in neural information processing systems*, 35: 23311–23330.

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*, 21719–21728.

Mildenhall, B.; Srinivasan, P. P.; Ortiz-Cayon, R.; Kalantari, N. K.; Ramamoorthi, R.; Ng, R.; and Kar, A. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4): 1–14.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Mirzaei, A.; Aumentado-Armstrong, T.; Derpanis, K. G.; Kelly, J.; Brubaker, M. A.; Gilitschenski, I.; and Levinshtein, A. 2023. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20669–20679.

Navaneet, K.; Meibodi, K. P.; Koohpayegani, S. A.; and Pirsiavash, H. 2024. CompGS: Smaller and Faster Gaussian Splatting with Vector Quantization. *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*, 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): 1–17.

Qu, Y.; Chen, D.; Li, X.; Li, X.; Zhang, S.; Cao, L.; and Ji, R. 2025. Drag Your Gaussian: Effective Drag-Based Editing with Score Distillation for 3D Gaussian Splatting. *arXiv preprint arXiv:2501.18672*.

Ren, Z.; Agarwala, A.; Russell, B.; Schwing, A. G.; and Wang, O. 2022. Neural volumetric object selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6133–6142.

Shen, Q.; Yang, X.; and Wang, X. 2024. Flashsplat: 2d to 3d gaussian splatting segmentation solved optimally. In *European Conference on Computer Vision*, 456–472. Springer.

Tschernezki, V.; Laina, I.; Larlus, D.; and Vedaldi, A. 2022. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 International Conference on 3D Vision (3DV)*, 443–453. IEEE.

Yan, Z.; Li, L.; Shao, Y.; Chen, S.; Wu, Z.; Hwang, J.-N.; Zhao, H.; and Remondino, F. 2024. 3dsceneeditor: Controllable 3d scene editing with gaussian splatting. *arXiv preprint arXiv:2412.01583*.

Ye, M.; Danelljan, M.; Yu, F.; and Ke, L. 2024. Gaussian Grouping: Segment and Edit Anything in 3D Scenes. In *ECCV*.

Yen-Chen, L.; Florence, P.; Barron, J. T.; Lin, T.-Y.; Rodriguez, A.; and Isola, P. 2022. Nerf-supervision: Learning dense object descriptors from neural radiance fields. In *2022 international conference on robotics and automation (ICRA)*, 6496–6503. IEEE.

Ying, H.; Yin, Y.; Zhang, J.; Wang, F.; Yu, T.; Huang, R.; and Fang, L. 2023. OmniSeg3D: Omniversal 3D Segmentation via Hierarchical Contrastive Learning. *arXiv preprint arXiv:2311.11666*.

Zhao, Y.; Xu, W.; Zheng, R.; Qiao, P.; Liu, C.; and Chen, J. 2025. iSegMan: Interactive Segment-and-Manipulate 3D Gaussians. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 661–670.

Zhou, S.; Chang, H.; Jiang, S.; Fan, Z.; Zhu, Z.; Xu, D.; Chari, P.; You, S.; Wang, Z.; and Kadambi, A. 2024. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21676–21685.