

CLIP-GS: CLIP-Informed Gaussian Splatting for View-Consistent 3D Indoor Semantic Understanding

Guibiao Liao^{1,2} Jiankun Li³ Zhenyu Bao^{1,2} Xiaoqing Ye³ Qing Li² Kanglin Liu^{2*}

¹Peking University, ²Pengcheng Laboratory, ³Baidu Inc.

Abstract

Exploiting 3D Gaussian Splatting (3DGS) with Contrastive Language-Image Pre-Training (CLIP) models for open-vocabulary 3D semantic understanding of indoor scenes has emerged as an attractive research focus. Existing methods typically attach high-dimensional CLIP semantic embeddings to 3D Gaussians and leverage view-inconsistent 2D CLIP semantics as Gaussian supervision, resulting in efficiency bottlenecks and deficient 3D semantic consistency. To address these challenges, we present CLIP-GS, efficiently achieving a coherent semantic understanding of 3D indoor scenes via the proposed Semantic Attribute Compactness (SAC) and 3D Coherent Regularization (3DCR). SAC approach exploits the naturally unified semantics within objects to learn compact, yet effective, semantic Gaussian representations, enabling highly efficient rendering (>100 FPS). 3DCR enforces semantic consistency in 2D and 3D domains: In 2D, 3DCR utilizes refined view-consistent semantic outcomes derived from 3DGS to establish cross-view coherence constraints; in 3D, 3DCR encourages features similar among 3D Gaussian primitives associated with the same object, leading to more precise and coherent segmentation results. Extensive experimental results demonstrate that our method remarkably suppresses existing state-of-the-art approaches, achieving mIoU improvements of 21.20% and 13.05% on ScanNet and Replica datasets, respectively, while maintaining real-time rendering speed. Furthermore, our approach exhibits superior performance even with sparse input data, substantiating its robustness.

1. Introduction

Neural Radiance Fields (NeRFs) [35] and 3D Gaussian Splatting (3DGS) [18] have emerged as promising methods for high-quality 3D scene modeling and novel view synthesis [12, 31, 32, 54]. Recent methods have made remarkable advancements in rendering novel views that en-

compass geometric and appearance details [2, 49]. However, achieving a comprehensive semantic understanding of 3D scenes [15, 25–27, 30, 53] remains a challenging task. To achieve it, one intuitive approach involves using manually annotated semantic labels to offer semantic supervision for existing 3D scene representations. Nevertheless, this resource-intensive manual annotation process impedes its practical application in real-world 3D semantic understanding. Compared to the traditional semantic labeling manner, the 2D vision-language model, Contrastive Language-Image Pre-Training (CLIP) is exploited to provide a new approach to semantic understanding without reliance on annotated image labels. The CLIP model, comprising an image encoder and a text encoder, is pre-trained on extensive image-text pairs collected from websites to establish vision-language associations. This paradigm enables CLIP to exhibit promising open-vocabulary semantic understanding capabilities, allowing it to segment objects based on textual queries [24, 28, 29, 46, 50]. Consequently, effectively harnessing image-text knowledge from CLIP for precise open-vocabulary 3D semantic understanding of indoor scenes is emerging as a valuable area.

Recently, LERF [19] pioneers a NeRF-based semantic field optimization with CLIP visual features, enabling text-driven 3D segmentation. Building upon this foundation, 3DOVS [33] further introduces a Relevancy-Distribution Alignment loss for segmentation accuracy improvement. However, NeRF’s ray-marching volume rendering technique significantly impedes rendering efficiency (as illustrated in the 2nd column of Fig. 1). To handle this computational limitation, several methods attempt to employ 3D Gaussian representation and tile-based rasterization for rendering acceleration. For example, Feature 3DGS [55] embeds high-dimensional semantic parameters into 3D Gaussians and optimizes them with CLIP semantic features. The state-of-the-art LangSplat [37] learns low-dimensional, compressed CLIP features at 3D Gaussians to accelerate rasterization, and then utilizes a pre-trained deep neural network for post-process feature upsampling to obtain semantic representation. Moreover, LangSplat utilizes region masks derived from the Segment Anything Model (SAM)

*Corresponding author

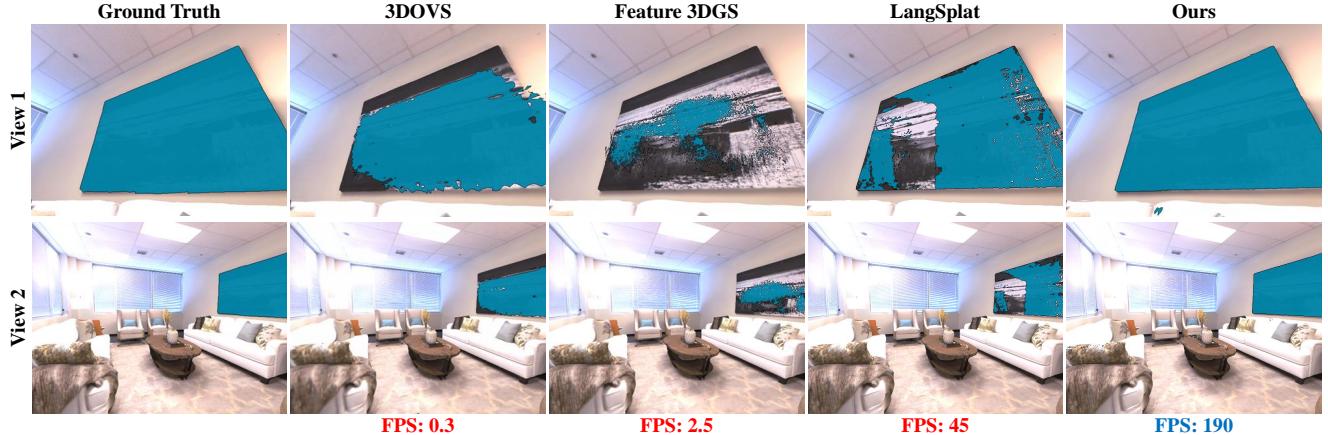


Figure 1. Visual comparisons between different CLIP-informed 3D semantic segmentation methods using the text query “Picture” across different views. The *NeRF*-based method, 3DOVS [33], and 3DGS-based methods, Feature 3DGS [55] and LangSplat [37] exhibit ambiguous semantics and efficiency bottlenecks. In contrast, our approach achieves more precise and consistent semantic segmentation results with a faster speed.

[20] to refine CLIP feature extraction, enhancing the precision of object boundaries for each training view.

Despite recent advances, certain limitations remain in achieving efficient and view-consistent 3D semantic understanding. **1) Efficiency Bottleneck.** While adopting efficient rasterization techniques, the high-dimensional feature rasterization in Feature 3DGS and the post-process feature upsampling in LangSplat make them fall short of achieving extremely efficient rendering. **2) Deficient 3D Semantic Consistency.** Additionally, although LangSplat employs SAM’s masks to refine object boundaries in each single training view, it struggles to achieve view-consistent semantic results as illustrated in Fig. 1. This limitation stems from the application of view-inconsistent 2D CLIP semantics [46] for 3D Gaussians optimization, without incorporating effective 3D-consistent constraints for cross-view consistency enhancement.

To tackle these limitations, we present **CLIP-GS**, an efficient method for achieving precise 3D semantic understanding using CLIP-informed 3D Gaussian representation. **To address the efficiency challenge**, we introduce the Semantic Attribute Compactness (SAC) approach. Motivated by the naturally unified semantics within objects, SAC extracts a single representative semantic feature for each object using SAM masks [20], and uses low-dimensional semantic indices to encapsulate them. In virtue of this low-dimensional representation, we attach low-dimensional semantic embeddings to 3D Gaussians for rasterization, extremely enhancing semantic rendering efficiency in 3D scenes (>100 FPS). **To tackle deficient 3D consistency**, we propose 3D Coherent Regularization (3DCR) with 2D and 3D-level consistency constraints. Specifically, in the 2D view space, 3DCR leverages enhanced inherently view-

consistent rendered outcomes from 3D models (i.e., trained 3D Gaussians) as coherent supervision signals, providing view-consistent semantic constraints. Additionally, in the 3D point space, 3DCR identifies 3D Gaussian primitives associated with the same object via ray-based intersection matching, and explicitly encourages semantic similarity among their semantic embeddings to enhance semantic consistency. By incorporating this semantic-consistent optimization, our method achieves view-consistent segmentation results, as shown in the last column of Fig. 1.

To evaluate the 3D semantic understanding performance of our approach, we conduct experiments on synthetic and real-world datasets: Replica [41], ScanNet [8], and 3DOVS [33]. Evaluations demonstrate superior performance of our method compared to existing state-of-the-art methods, especially achieving significant mIoU improvements of 21.20% and 13.05% on ScanNet and Replica, respectively. The key contributions of this work are as follows.

- We introduce a Semantic Attribute Compactness (SAC) approach that efficiently attaches compact and effective semantic information into 3D Gaussians, ensuring extremely efficient rendering.
- We propose a 3D Coherent Regularization (3DCR) approach that addresses the issue of deficient 3D consistency by imposing semantic-consistent constraints at 2D and 3D levels, leading to more coherent segmentation results across different viewpoints.
- Extensive experiments demonstrate that our approach outperforms state-of-the-art CLIP-informed 3D semantic understanding methods in both segmentation precision and rendering efficiency (>100 FPS). Moreover, our method shows superior performance in the sparse-view setting, validating its robustness.

2. Related Work

Gaussian Splatting for 3D Scene Representation. Representing 3D scenes via radiance fields has witnessed significant advancements in recent years. Neural Radiance Field (NeRF) [35] implicitly represents the appearance and geometry of 3D scenes using a coordinate-based neural network. Despite substantial efforts to improve optimization and rendering efficiency [3, 36], NeRF-based methods still face challenges of slow rendering speeds, primarily due to the neural network query process and volume rendering.

Recently, Kerbl et al. [18] proposed 3D Gaussian Splatting (3DGS), a novel approach to represent 3D scenes as collections of 3D Gaussians. By employing a fast tile-based rasterization technique, 3DGS enables real-time rendering at 1080p resolution while maintaining high-quality visual results. Building upon the efficiency demonstrated by 3DGS, numerous recent studies have extended its application to various tasks, such as head and human reconstruction [7, 42, 56], autonomous driving scene modeling [11, 47, 52], and 3D editing [16, 44, 45]. Besides, recent surveys [1, 10] provide a more comprehensive overview of the developments of 3DGS. Unlike these methods, our study focuses on harnessing 3DGS for CLIP-informed 3D scene semantic understanding.

CLIP-Informed 3D Semantic Fields. Early methods focused on integrating CLIP features [5, 38] into NeRFs to establish 3D semantic fields. DFF [21] explored the incorporation of features from CLIP-LSeg [24] to optimize the semantic feature field. LERF [19] extended this concept by introducing a scale-conditioned feature field supervised by multi-scale CLIP features from the CLIP visual encoder. Similarly, 3DOVS [33] optimized the semantic feature field using CLIP features and introduced a relevance-distribution alignment loss to enhance accuracy. Yet, these NeRF-based methods are constrained by the computational bottleneck of NeRF’s volume rendering, impeding efficient rendering.

Alternative to NeRFs, recent works explore embedding CLIP features into 3D Gaussians to construct a semantic field and employ tile-based rasterization for efficient semantic rendering. Feature 3DGS [55] attached high-dimensional CLIP embeddings to 3D Gaussians, optimizing them using CLIP semantic features. However, embedding high-dimensional parameters in millions of 3D Gaussians significantly hampers rendering efficiency. FMGS [57] integrated 3D Gaussians with multi-resolution hash encodings to form a combined feature field, supervised by multi-view CLIP features. Shi et. al. [40] embedded quantized CLIP features onto 3D Gaussians to reduce memory and storage requirements. LangSplat [37] learned low-dimensional semantic Gaussian features and utilized a pre-trained deep neural network to upsample rendered features, aligning them with high-dimensional CLIP features. However, pre-training the deep network and applying the post-

processing upsampling process inevitably introduce additional time overhead, thereby compromising the method’s overall efficiency. Moreover, while LangSplat used SAM to generate object masks and then utilized CLIP to encode these regions for single-view object boundary refinement, it still suffers from semantic ambiguity due to a lack of cross-view, semantic-consistent supervision, resulting in limited coherence in the segmentation results. Note that methods for 3D interactive segmentation, such as [6, 14, 39, 48], fall outside the scope of the CLIP-informed semantic fields. These approaches primarily generate region masks devoid of semantic meaning, thereby lacking the capability to directly segment objects through language queries. For instance, Gaussian Grouping [48] requires supplementary techniques like Grounding Dino [34] to annotate masks with semantic information. Moreover, it is constrained to single-query segmentation for each mask, thereby limiting the ability to efficiently generate a whole segmentation map that necessitates rendering times on the order of minutes. Guo et al. [13] projected 2D CLIP features onto 3D Gaussians via a spatial correspondence method that relied on rendered depth obtained from the depth rendering of 3D Gaussians. GSemSplat [43] tackled sparse-input 3D semantic understanding by utilizing ViT-based MAS3R [23] and an additional semantic MLP for feature prediction. Yet, its training pipeline involved $O(N^2)$ pairwise point map computations, leading to high computational cost and limited scalability with increasing views. Jiao et al. [17] proposed an image-text-3D Gaussian contrastive pretraining framework for 3D representation learning, but its performance depended on large-scale paired image-3D datasets.

Unlike previous approaches, our method addresses the inefficiency challenge by introducing a semantic attribute compactness strategy to compactly represent semantic Gaussian representations, facilitating fast training and inference. Additionally, we tackle the view-inconsistent issue through a novel 3D coherent regularization approach, enhancing the view consistency of semantics for coherent 3D semantic understanding.

3. Methodology

3.1. Preliminary and Overview

Preliminary. 3D Gaussian Splatting (3DGS) [18] represents a 3D scene using a suite of 3D Gaussians, each parameterized by a 3D position $\mathbf{p} = \{x, y, z\} \in \mathbb{R}^3$, a 3D size scaling factor $\mathbf{s} \in \mathbb{R}^3$, a rotation quaternion $\mathbf{q} \in \mathbb{R}^4$, a color $\mathbf{c} \in \mathbb{R}^3$, and an opacity value $o \in \mathbb{R}$. All parameters are learnable and can be collectively symbolized by $\Theta_i = \{\mathbf{p}_i, \mathbf{s}_i, \mathbf{q}_i, \mathbf{c}_i, o_i\}$, where i denotes the i -th Gaussian. To compute the pixel color C , 3DGS employs α -blending point-based rendering by blending N Gaussians

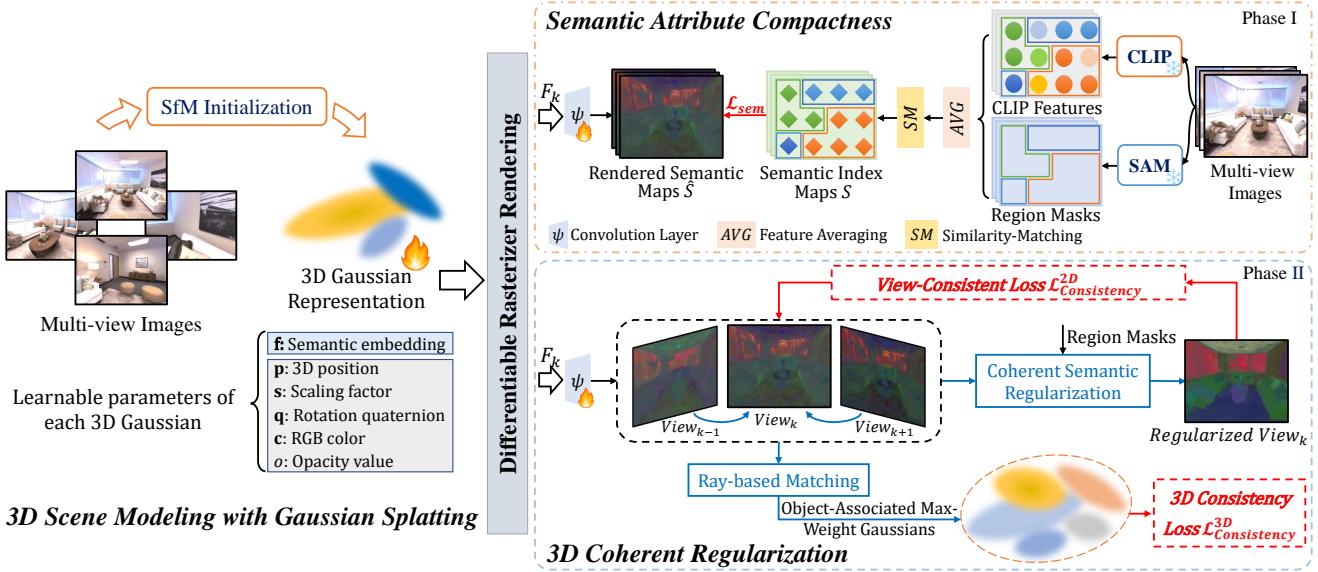


Figure 2. **Illustration of CLIP-GS optimization.** *Left:* CLIP-GS represents the 3D scene with a collection of 3D Gaussians [18] with learnable attributes, specifically adding a *semantic* attribute. *Right:* First, multi-view images undergo feature extraction using the frozen CLIP model [38] and region mask generation with SAM [20]. We then optimize CLIP-GS in an end-to-end manner through two phases. In Phase I, we introduce **Semantic Attribute Compactness** (SAC) to capture the *unified* semantics within each object, facilitating efficient optimization and rendering of semantic Gaussians. In Phase II, after training 3D Gaussians at certain iterations, we present **3D Coherent Regularization** (3DCR) to enhance 3D semantic consistency. 3DCR leverages self-predicted semantics derived from CLIP-GS, refined by cross-view coherent regularization, to provide view-consistent supervision signals for optimizing Gaussians. Additionally, 3DCR identifies 3D Gaussian primitives associated with the same object through ray-based intersection matching and encourages their semantics to be similar. The color optimization process follows 3DGs [18] and is omitted for brevity.

in the front-to-back depth order [22], formulated as:

$$C(x_p) = \sum_{i \in \mathcal{N}} T_i \alpha_i \mathbf{c}_i, \quad (1)$$

where \mathcal{N} is the set of Gaussian primitives that overlap with the given pixel x_p . α_i is calculated by $\alpha_i = o_i G_i^{2D}(x_p)$, where G_i^{2D} denotes the i -th Gaussian's 2D projection. The transmittance T_i is defined as $T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$.

To update the parameters of 3D Gaussians, Gaussian Splatting adopts a differentiable rendering technique that projects the Gaussians onto the 2D image plane [58] and optimizes them using color supervision. The reconstruction loss is defined as the discrepancy between the rendered image \hat{I} and the ground truth image I :

$$\mathcal{L}_{rgb} = (1 - \lambda) \mathcal{L}_1(\hat{I}, I) + \lambda \mathcal{L}_{D-SSIM}(\hat{I}, I), \quad (2)$$

where λ is set to 0.2 [18]. 3DGs has shown its effectiveness in 3D scene reconstruction. In this work, we propose **CLIP-GS**, a novel approach that extends 3DGs toward semantic understanding of complex 3D indoor scenes.

Overview. The overall framework of our CLIP-GS is illustrated in Fig. 2. Given a set of N posed images $I = \{I_i\}_{i=1}^N \in \mathbb{R}^{H \times W \times 3}$, CLIP-GS represents the 3D scene us-

ing 3D Gaussian primitives with learnable attributes, specifically adding a semantic attribute. To enable 3D semantic understanding, we optimize 3D Gaussians guided by multi-view CLIP features through two key components: **Semantic Attribute Compactness (SAC)** and **3D Coherent Regularization (3DCR)**. SAC constructs compact semantic Gaussian representations to support efficient semantic understanding (Section 3.2), while 3DCR introduces 2D and 3D-level semantic-consistent constraints to enhance coherent segmentation results (Section 3.3). We elaborate on each component in the following sections.

3.2. Semantic Attribute Compactness

To render novel views with semantic information via the Gaussian Splatting rendering, an intuitive method is attaching a learnable semantic parameter $\mathbf{f}_i \in \mathbb{R}^D$ to each Gaussian, and apply α -blending to compute the pixel-wise rendered feature F . This process can be formulated as:

$$F(x_p) = \sum_{i \in \mathcal{N}} T_i \alpha_i \mathbf{f}_i \in \mathbb{R}^D, \quad (3)$$

where D denotes the dimension of rendered features, typically set to a high value, such as 512, to align the CLIP semantic feature dimension for optimizing \mathbf{f}_i . However, this

direct embedding of high-dimensional parameters into millions of Gaussians for semantic modeling, such as Feature 3DGS [55], significantly decreases rasterization efficiency in Gaussian Splatting, leading to constrained rendering performance.

To tackle this efficiency challenge, we introduce **Semantic Attribute Compactness (SAC)**. The key insight of SAC is leveraging the inherently *unified* semantic meaning of the identical object for efficient representation. Concretely, SAC represents each object using a single representative CLIP feature, and further employs low-dimensional semantic indices to encapsulate the high-dimensional CLIP features. By leveraging low-dimensional indices as supervision, we thus attach low-dimensional semantic embeddings to 3D Gaussians for rasterization, resulting in a more efficient semantic rendering process.

Specifically, for k -th training view I_k , we transform the CLIP feature $\bar{F}_k \in \mathbb{R}^{D \times H \times W}$ into a representative version $\hat{F}_k \in \mathbb{R}^{D \times M}$, where M denotes the number of objects in I_k . To achieve this, we harness the powerful Segment Anything Model (SAM) [20] to yield region masks $R_k = \{R_k^q\}_{q=1}^M$ over the image I_k . For each region, we compute the weighted average of the CLIP feature in the spatial dimension, treating it as the unified feature to represent the semantics of this region uniformly. This process yields the representative CLIP feature $\hat{F}_k \in \mathbb{R}^{D \times M}$, where M typically ranges from a few dozen to a hundred, significantly smaller than the image size $H \times W$. Since the features within each region are still high-dimensional yet semantically consistent, we utilize a low-dimensional semantic index to efficiently represent the unified semantic feature of each region. To assign these indices for each region, we adopt a similarity-matching method that computes the cosine similarity between representative CLIP features \hat{F}_k and a set of text features T , producing the semantic index map S_k as:

$$S_k = \text{argmax}(\cos(\hat{F}_k, T)), \quad (4)$$

where T is obtained by encoding a set of text descriptions using the CLIP text encoder. In this way, each position within a region of the semantic index map $S_k \in \mathbb{R}^{1 \times H \times W}$ shares a consistent, low-dimensional semantic index. This method ensures a reliable and robust matching relationship in the CLIP feature space, generating effective semantic index maps for 3D Gaussian optimization and CLIP feature retrieval.

Leveraging SAC, we can transform semantic Gaussian representation learning into a low-dimensional space. Thus, we embed a low-dimensional, learnable semantic parameter $\mathbf{f}_i \in \mathbb{R}^d$ into each 3D Gaussian to efficiently construct the 3D semantic field. Specifically, we adopt the α -blending rendering pipeline to project 3D Gaussians onto the 2D image plane and obtain the rendered feature map

$F \in \mathbb{R}^{d \times H \times W}$. Each pixel-wise feature is computed as $F(x_p) = \sum_{i \in \mathcal{N}} T_i \alpha_i \mathbf{f}_i$, where \mathcal{N} denotes the set of Gaussians contributing to pixel location x_p . Then, we use a trainable, lightweight convolution layer ψ to produce the rendered semantic map \hat{S} , which can be formulated as:

$$\hat{S} = \psi(F). \quad (5)$$

The rendered semantic map \hat{S} is supervised using the semantic index map S , optimizing the learnable semantic embedding per Gaussian, and the semantic loss is defined as:

$$\mathcal{L}_{sem} = \mathcal{L}_{ce}(\hat{S}, S). \quad (6)$$

By enhancing 3D Gaussians with efficient semantic modeling, our approach enables efficient rendering while exhibiting effective segmentation results.

3.3. 3D Coherent Regularization

Having achieved efficient semantic representations through semantic attribute compactness, however, immutably employing view-inconsistent CLIP semantics to optimize Gaussians leads to ambiguous and subpar rendering results, as illustrated in (a) and (c) of Fig. 3. This limitation stems from the 2D CLIP model’s inherent difficulty in maintaining consistent object identities across different views, inadequately enforcing multi-view semantic consistency constraints. To tackle this limitation, we propose a **3D Coherent Regularization (3DCR)** approach to enhance semantic-consistent constraints at 2D and 3D levels.

Consistency Regularization for 2D Views. Inspired by the inherently consistent semantics rendered from trained 3D models, we leverage enhanced self-predicted semantics derived from trained 3D Gaussians as view-consistent supervision signals for imposing cross-view coherent semantic constraints.

Specifically, after several training iterations, we render a semantic map \hat{S}_k from the current view I_k . We then introduce a *coherent semantic regularization* by incorporating semantic cues from adjacent training views, to eliminate semantic ambiguity and establish view-consistent supervision signals for the current view I_k .

To achieve this, we treat the sequence of multi-view training images as temporally adjacent frames, akin to a video, and use a zero-shot tracker [4] to associate SAM masks across adjacent views. Next, we render semantic maps \hat{S}_{k-1} and \hat{S}_{k+1} at the preceding and subsequent views, respectively. For the q -th region R_k^q in the k -th view, we aggregate semantic information from corresponding regions (R_{k-1}^q, R_{k+1}^q) from adjacent views, and perform coherent regularization to unify the \hat{S}_k^q within the region R_k^q . This coherent regularization is achieved through a majority voting strategy. Intuitively, this strategy unifies the region-based semantic representation by selecting the most

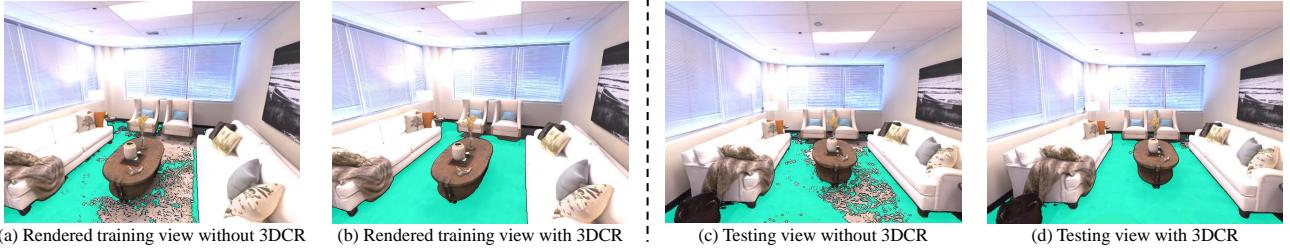


Figure 3. Illustration of rendered segmentation maps with the text query "Rug". (a) and (b) correspond to training views, while (c) and (d) pertain to testing views. In (a), the rendered result appears ambiguous when immutably employing 2D CLIP semantics for Gaussian optimization. Conversely, (b) shows that leveraging 3DCR can provide coherent semantic constraints to supervise Gaussians, leading to more precise results (d). For more details, refer to Sec. 3.3.

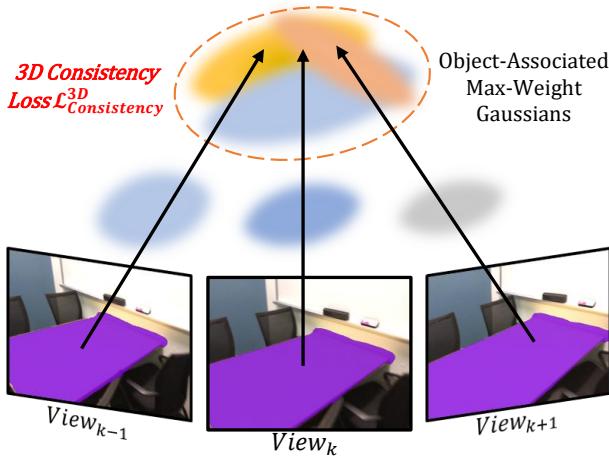


Figure 4. Illustration of the consistency regularization for 3D Gaussians in 3D Coherent Regularization (3DCR).

frequently occurring semantic index across views, thereby forming a consistent semantic representation Z_k^q , which can be formulated as:

$$Z_k^q = \operatorname{argmax}_c \sum_{v \in U_k^q} \mathbb{1}\{\operatorname{argmax}(\psi(\hat{S}_k^v)) = c\}, \quad (7)$$

where $U_k^q \equiv \{R_{k-1}^q, R_k^q, R_{k+1}^q\}$ denotes the set of corresponding region masks across the current and adjacent views, providing enhanced coherent semantic regularization. c indicates the c -th category. The indicator function $\mathbb{1}\{\cdot\}$ returns 1 if the condition matches category c at a given pixel, and 0 otherwise. Similarly, this coherent regularization is applied to all regions in the rendered semantic map \hat{S}_k . As shown in (b) of Fig. 3, this regularization effectively improves the semantic consistency, such as the rug area in the current view. Consequently, we utilize this view-consistent supervision signal Z to train 3D Gaussians, rather than vanilla 2D CLIP semantics:

$$\mathcal{L}_{consistency}^{2D} = \mathcal{L}_{ce}(\hat{S}, Z). \quad (8)$$

Moreover, in each iteration, this coherent semantic regularization is applied to a single training view, ensuring training efficiency. With progressively enhanced view-consistent constraints, the semantic consistency of 3D Gaussians improves over time, which, in turn, reinforces the regularization in subsequent iterations. As a result, this iterative refinement facilitates more coherent segmentation results, as evidenced in the experiments.

Consistency Regularization for 3D Gaussians. Beyond 2D constraints, we further introduce a feature similarity constraint among 3D Gaussian primitives associated with the same object. Since the rendering results are predominantly determined by maximally-weighted Gaussian primitives (as formulated in Eq. (3)), we encourage semantic similarity among these dominant Gaussians associated with the same object.

For instance, as shown in Fig. 4, given an object region R_k^q in k -th view I_k , we first gather pixel locations from the corresponding object masks $\{R_{k-1}^q, R_k^q, R_{k+1}^q\}$ across the current and adjacent views. For each of these pixels, we identify the maximally contributing Gaussian along the ray cast into 3D space, yielding a matched set of Gaussian primitives $\mathbf{G} = \{\Theta_i\}_{i=1}^n$ that intersect rays from all three views. These matching, maximally weighted Gaussians associated with the same object are then encouraged to have similar semantic representations. To implement this, we minimize the distance between their features $\mathbf{H} = \{\mathbf{f}_i \in \mathbf{G}\}_{i=1}^n$ and their corresponding cluster feature \mathbf{M} , computed via mean-pooling over set \mathbf{G} . The 3D feature similarity constraint is enforced through a KL-divergence-based loss as follows:

$$\mathcal{L}_{consistency}^{3D} = \mathcal{L}_{kl}(\mathbf{M} || \mathbf{H}) = \sum_{i=1}^n \mathbf{M} \log\left(\frac{\mathbf{M}}{\mathbf{f}_i}\right). \quad (9)$$

This loss encourages 3D Gaussians associated with the same object to share similar semantic features by minimizing their divergence from a pooled semantic anchor, thereby enhancing 3D semantic consistency. In summary, the overall 3D coherent regularization loss contains 2D and 3D-

level constraints, formulated as:

$$\mathcal{L}_{consistency} = \mathcal{L}_{consistency}^{2D} + \mathcal{L}_{consistency}^{3D}. \quad (10)$$

3.4. Overall Training with Progressive Densification Regulation Scheme

Our overall training scheme serves two objectives. First, to establish a semantic Gaussian radiance field for accurate and view-consistent semantic understanding of 3D scenes. Second, to regulate the quantity of Gaussian primitives for more efficient semantic Gaussian radiance field acquisition.

Specifically, the entire framework is trained end-to-end using reconstruction loss \mathcal{L}_{rgb} and semantic loss \mathcal{L}_s across two phases. In phase I, the semantic embeddings of 3D Gaussians are optimized utilizing \mathcal{L}_{sem} . In phase II, after training the 3D Gaussians a few iterations \mathcal{T} , the 3D coherent regularization loss $\mathcal{L}_{consistency}$ replaces the \mathcal{L}_{sem} for semantic consistency enhancement. The overall training process can be formulated as:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{rgb} + \mathcal{L}_{sem} & \text{Iteration} \leq \mathcal{T} \\ \mathcal{L}_{rgb} + \mathcal{L}_{consistency} & \text{Iteration} > \mathcal{T}. \end{cases} \quad (11)$$

In addition to achieving precise 3D semantic understanding, we introduce a **Progressive Densification Regulation (PDR)** strategy to regulate the amount of Gaussians, aiming to improve efficiency while upholding high-quality scene representations. Specifically, in vanilla 3DGS, loss computation initially operates at full image resolution with fixed densification parameters (e.g., threshold and interval) for 3D Gaussians. This fixed training scheme often results in a proliferation of Gaussian primitives at the early training process, as the densification strategy outlined in [18] rapidly causes new Gaussians to further split or clone before adequate optimization, leading to redundant Gaussian generation and compromising rendering efficiency.

To mitigate primitive proliferation, when employing our PDR, Gaussians undergo initial optimization at lower image resolutions with reduced densification frequency and elevated densification threshold. As training progresses, the image resolution and densification frequency gradually increase while the densification threshold decreases, ultimately reaching full resolution and default densification parameters. Experimental results show that PDR not only effectively prevents excessive proliferation of Gaussians in the early training stage to enhance rendering efficiency, but also constructs a more effective Gaussian radiance field.

4. Experiment

4.1. Experiment Setup

Evaluation Datasets. We evaluate our approach on three multi-view indoor scene datasets that are extensively used in 3D scene reconstruction and segmentation, including 3DOVS [33], ScanNet [8], and Replica [41].

- 3DOVS is a real-world dataset containing diverse objects captured in various poses and backgrounds. The scenes featured in 3DOVS consist of 28 to 37 images with a face-forward orientation, whose views are sampled in an “outside-in” manner, resulting in notable overlap across views. We conduct experiments on four scenes (Bed, Sofa, Lawn, and Bench) for evaluation. We follow the experimental protocol outlined in 3DOVS [33], using posed training images for optimizing semantic fields and semantic labels from the testing set for evaluation.
- ScanNet is a real-world dataset that offers a variety of indoor scenes and provides camera poses obtained via BundleFusion [9] and semantic segmentation labels. The experiments involve four scenes (Scene0004, Scene0389, Scene0494, and Scene0693), each with 233 to 289 images captured along predefined trajectories. Every 10-th is reserved for evaluation, with the remaining images used for training.
- Replica is a synthetic dataset that includes high-fidelity indoor scenes with photorealistic textures and per-primitive semantic classes. We select six scenes (Room0, Room1, Room2, Office0, Office2, and Office4) for evaluation. For each scene, images are captured along a defined trajectory, with every 10-th image designated for evaluation and the rest used for training.

To further evaluate the robustness of our method, we introduce a *sparse-view* benchmark, specifically targeting challenging “inside-out” indoor scenarios in ScanNet and Replica. For each scene, we evenly sample 30 images. Of these, every 10-th image is designated as a test image, and the remaining images serve as training data.

Evaluation Metrics. We evaluate segmentation performance on novel views using two metrics: mean Intersection over Union (mIoU) and mean Pixel Accuracy (mAcc). Moreover, to assess the quality of novel view images, we leverage Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [51] to assess the quality of novel view images.

4.2. Implementation Details

Data Pre-processing. For CLIP feature extraction, we utilize the CLIP ViT-B/16 model [38] as the base experimental setting. Moreover, we provide CLIP-LSeg ViT-L/16 [24] that is used in Feature 3DGS [55], for more comprehensive comparisons. To acquire the SAM’s masks, we employ the SAM ViT-H model [20]. Specifically, we deploy SAM to automatically generate masks for each training image. Point prompts are sampled on a 32×32 grid, with a minimum mask region area set to 100 and a bounding box IoU threshold of 0.7. The CLIP features and SAM’s masks mentioned above are pre-computed offline.

Training. We implement our approach using PyTorch and

Table 1. Comparison with state-of-the-art methods on segmentation results of novel views across different scenes from the ScanNet dataset [8], under multi-view training data conditions. Our proposed approach demonstrates superior performance.

Method	Scene0004		Scene0389		Scene0494		Scene0693		Average	
	mIoU ↑	mAcc ↑								
LERF [19]	7.137	12.139	20.102	62.025	17.624	35.079	16.531	51.933	15.349	40.294
3DOVS [33]	9.763	15.666	20.596	64.099	18.556	36.223	22.292	54.141	17.802	42.532
Feature 3DGS [55]	20.907	52.557	19.978	72.902	26.204	52.191	12.236	39.614	19.831	54.316
LangSplat [37]	30.318	69.623	18.936	68.003	21.167	47.796	16.526	33.789	21.737	54.803
Ours	40.123	78.851	39.761	89.393	58.543	88.224	33.303	62.674	42.932	79.786

Table 2. Comparison with state-of-the-art methods on segmentation results of novel views across different scenes from the Replica dataset [41], under multi-view training data conditions. Our proposed approach demonstrates superior performance.

Method	Room0		Room1		Room2		Office0		Office2		Office4		Average	
	mIoU ↑	mAcc ↑												
LERF [19]	7.615	35.346	14.043	30.218	6.552	22.468	3.369	4.887	7.077	16.884	11.056	22.948	8.285	22.125
3DOVS [33]	7.733	36.667	16.016	35.957	7.547	25.675	3.843	6.487	7.070	18.409	12.276	20.435	9.081	23.938
Feature 3DGS [55]	8.794	39.916	10.226	32.024	11.909	40.558	8.120	21.588	11.218	41.776	13.537	43.260	10.634	36.520
LangSplat [37]	11.932	45.703	15.706	47.586	19.658	72.577	9.968	36.708	17.805	55.518	16.445	51.356	15.252	51.575
Ours	27.249	74.129	28.358	72.140	23.082	76.659	20.932	38.489	40.312	90.452	29.882	56.991	28.302	68.143

Gaussian Splatting [18] on an NVIDIA A100 GPU, adhering to the default parameter settings in Gaussian Splatting for scene reconstruction. To enable scene understanding, we augment each 3D Gaussian with learnable semantic parameters of dimension $d = 3$ and modify the CUDA kernel to incorporate semantic rasterization while maintaining reconstruction quality.

During training, the learning rates for the semantic parameters and convolution layer are set to $2.5e^{-3}$ and $5e^{-4}$, respectively. Adam optimizer is used to train our model for 30k iterations, with each scene taking approximately 20 minutes to train. The similarity-matching computation in SAC is conducted offline to generate semantic index maps before training. The hyperparameter \mathcal{T} is set to 15k to leverage the 3D coherent regularization method. For PDR, we start with a training image resolution scale factor of 0.5, a densification interval of 200 iterations, and a densification threshold scale factor of 1.5. We then gradually increase the resolution to 1.0, while the densification interval and threshold are adjusted toward their default values following a cosine scheduling scheme. The resolution increase occurs during the first 7k iterations, and the densification adjustment takes place within the initial 4k iterations.

4.3. Comparison with State-of-the-art Methods

4.3.1 Results on the ScanNet Dataset

We compare our approach with NeRF-based methods: LERF [19] and 3DOVS [33], and 3DGS-based methods: Feature 3DGS [18] and LangSplat [37].

We present the quantitative results in Table 1. Our

method consistently outperforms other competitors in segmentation accuracy across diverse datasets. Notably, our approach exhibits a significant improvement in mIoU of 21.20% over the second-best method. These results indicate that our method effectively leverages semantic-consistent constraints for 3D Gaussian optimization, achieving precise 3D semantic understanding.

The qualitative analysis, as presented in the first two rows of Fig. 5, demonstrates that most current methods struggle to accurately segment object boundaries. While LangSplat employs SAM-generated region masks to enhance the object boundaries (e.g., *Table*), it easily yields ambiguous and noisy segmentation results due to the lack of view-consistent semantic constraints. In contrast, our approach exhibits spatially continuous and view-consistent results, stemming from our 3D coherent regularization approach, which effectively incorporates view-consistent semantic constraints to ensure coherent and accurate results.

4.3.2 Results on the Replica Dataset

We further evaluate the open-vocabulary segmentation performance through testing on synthetic scenes from Replica.

Table 2 reports the quantitative results on the Replica datasets. It can be seen that our method also achieves superior performance in segmentation accuracy across various scenes. Specifically, our approach achieves remarkable mIoU improvements of 13.05% compared to the second-best method. These findings demonstrate the effectiveness of our method for accurate 3D semantic understanding.

As shown in the last four rows of Fig. 5, we present

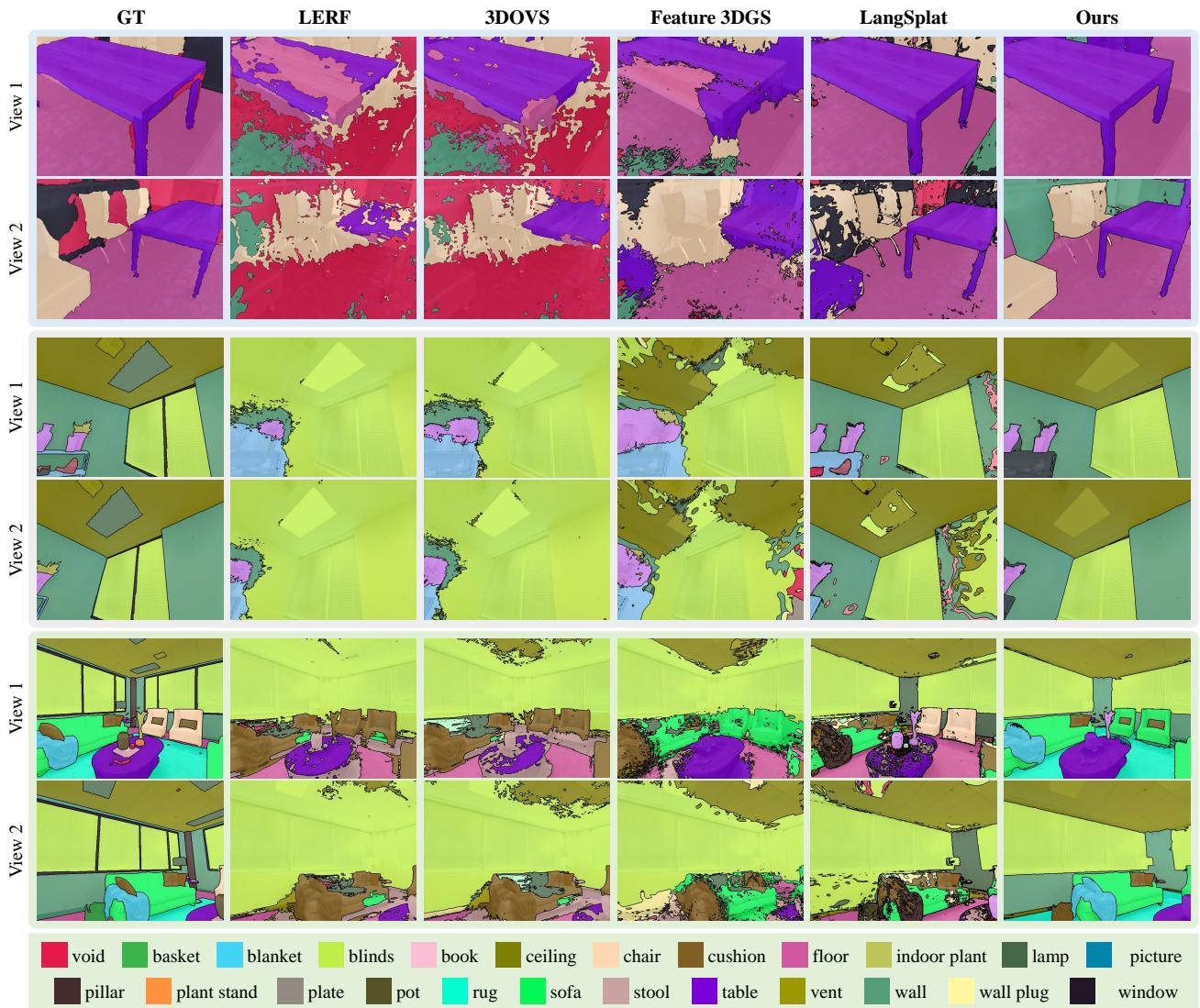


Figure 5. Visual segmentation results of novel view under multi-view training data conditions. While current methods produce ambiguous segmentation results due to the absence of 3D consistency constraints, our method achieves more precise and view-consistent results across various views.

qualitative results produced by our method and other approaches. It can be seen that existing methods face challenges in delivering precise and complete segmentation results (e.g., *Blinds*), our method consistently produces accurate and coherent segmentation results. These improvements stem from our 3D coherent regularization approach, which effectively enforces 3D consistency constraints to improve cross-view semantic consistency.

4.3.3 Results on the 3DOVS Dataset

Additionally, we evaluate our method on diverse face-forwarding scenes from the 3DOVS dataset. Quantitative

comparisons presented in Table 3, coupled with qualitative visualizations in Fig. 6, demonstrate that our approach achieves superior performance, obtaining precise and finer boundaries for each object. These results substantiate the stable generalization performance of our approach when dealing with face-forwarding scenes.

4.3.4 Results on the Sparse-view Benchmark Datasets

We introduce a sparse-view evaluation benchmark, a setup that trains the model solely using sparse training views, to evaluate the robustness of methods.

We report the sparse-view quantitative comparisons in

Table 3. Comparison with state-of-the-art methods on segmentation results of novel views across various scenes from the 3DOVS dataset [33]. Our proposed approach demonstrates superior open-world segmentation performance.

Method	Bed		Sofa		Lawn		Bench		Average	
	mIoU ↑	mAcc ↑								
FFD [21]	56.6	86.9	3.7	9.5	42.9	82.6	6.1	42.8	27.3	55.5
LERF [19]	73.5	86.9	27.0	43.8	73.7	93.5	53.2	79.7	56.9	76.0
3DOVS [33]	89.5	96.7	74.0	91.6	88.2	97.3	89.3	96.3	85.3	95.5
LangSplat [37]	92.5	99.2	90.0	97.9	96.1	99.4	94.2	98.6	93.2	98.8
Ours	97.2	99.3	94.1	98.4	96.5	99.4	94.8	98.7	95.6	99.0

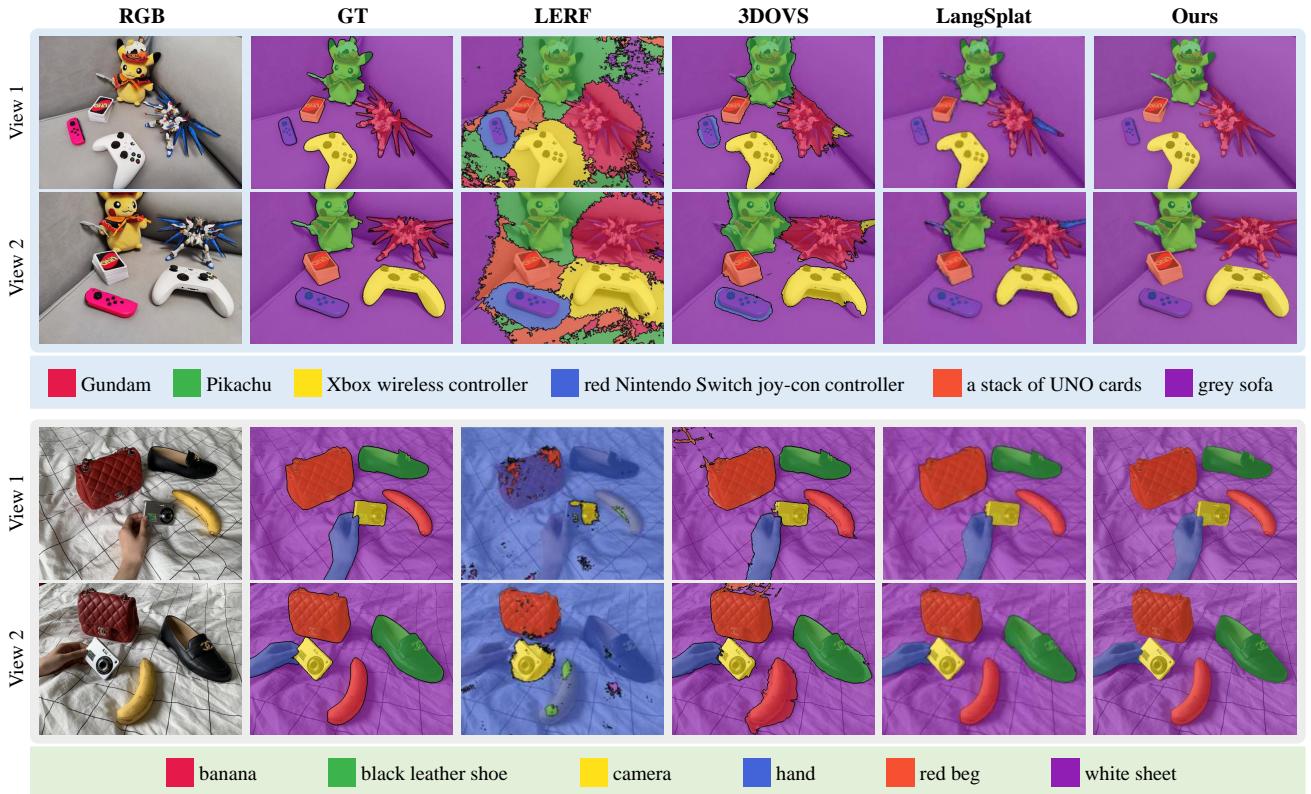


Figure 6. Visual segmentation results on various scenes from the 3DOVS dataset [33]. Our approach maintains accurate and stable generalization performance in face-forwarding scenes.

Table 4 and 5. Our approach, despite being trained using sparse views, consistently achieves superior performance in segmentation precision. Concretely, our method surpasses the second-best method by 23.66% and 14.56% in mIoU on ScanNet and Replica, respectively. Existing methods fail to produce accurate semantic results, because they suffer from insufficient view-consistent constraints, making them sensitive to sparse inputs. Moreover, LangSplat’s approach of fixing the position, scaling, and rotation of 3D Gaussians during semantic parameter optimization inhibits flexible adjustment of semantic representations, further impeding accuracy.

Fig. 7 presents visual comparisons of various methods under sparse and under-constrained conditions. Specifically, the 3DOVS method tends to produce ambiguous visual results, and Feature 3DGS exhibits coarse semantic outputs due to the lack of sufficient semantic-consistent constraints under the sparse input conditions. Moreover, LangSplat encounters significant difficulties in sparse-input scenarios. This limitation arises because, during semantic parameter optimization, LangSplat inherits the inferior Gaussian representations (including position, scaling, and rotation) obtained from sparse-input scene reconstruction using vanilla 3DGS, and subsequently optimizes only se-

Table 4. Comparison with state-of-the-art methods on segmentation results of novel views across different scenes from the ScanNet dataset [8], under sparse-view training data conditions. Our proposed approach maintains results.

Method	Scene0004		Scene0389		Scene0494		Scene0693		Average	
	mIoU ↑	mAcc ↑								
LERF [19]	8.838	9.259	15.528	74.926	20.594	30.911	11.276	39.841	14.059	38.734
3DOVS [33]	12.726	30.569	12.754	62.287	14.989	24.847	16.438	44.634	14.227	40.584
Feature 3DGS [55]	22.414	51.851	19.341	66.565	23.914	46.779	10.742	30.438	19.103	48.908
LangSplat [37]	16.448	42.199	7.724	32.094	18.962	44.276	7.443	20.192	12.644	34.690
Ours	25.849	53.013	41.507	84.122	59.041	85.188	44.656	71.825	42.763	73.537

Table 5. Comparison with state-of-the-art methods on segmentation results of novel views across different scenes from the Replica dataset [41], under sparse-view training data conditions. Our proposed approach maintains superior results.

Method	Room0		Room1		Room2		Office0		Office2		Office4		Average	
	mIoU ↑	mAcc ↑												
LERF [19]	6.302	33.592	5.578	24.181	6.604	20.152	0.435	1.568	0.822	4.847	6.130	18.140	4.312	17.080
3DOVS [33]	6.723	35.868	6.374	35.095	7.543	23.014	0.542	1.807	1.289	6.000	4.845	14.353	4.553	19.356
Feature 3DGS [55]	6.737	36.329	8.502	38.607	10.803	42.202	6.844	25.242	9.190	34.651	15.428	52.440	9.584	38.245
LangSplat [37]	2.518	15.338	4.654	24.760	4.469	22.782	1.858	7.469	3.591	13.013	5.349	22.837	3.740	17.700
Ours	13.952	57.019	27.632	76.072	30.761	76.606	11.432	29.752	32.612	87.034	28.494	76.444	24.147	67.154

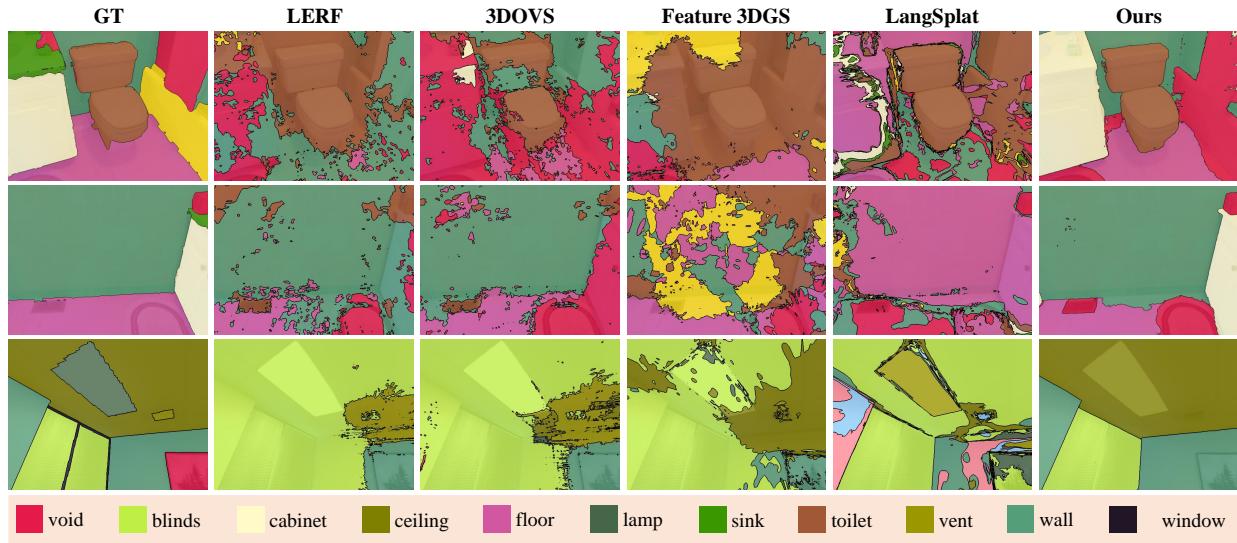


Figure 7. Visual segmentation results of novel view, under sparse-view training data conditions. Our approach obtains more robust and accurate segmentation performance.

mantic parameters. Consequently, these optimization constraints lead to suboptimal semantic understanding performance. In contrast, our method shows a more complete structure across various scenes. This can be attributed to our 3D coherent regularization strategy, which effectively integrates semantic information from adjacent views and enforces 3D consistency constraints, ensuring robustness even with sparse inputs.

4.3.5 Results using CLIP-LSeg Model

We further present experimental results using various vision-language foundation models, such as CLIP-LSeg [24], which was previously used in by Feature 3DGS [55] for optimizing Gaussian semantic attributes. The quantitative results presented in Table 6 and 7, indicate that our approach consistently achieves superior segmentation results across diverse scenes. Additionally, the visual comparison results shown in Fig. 8, demonstrate that Feature

Table 6. Performance comparison of segmentation results in novel views from the ScanNet dataset [8], utilizing multi-view training data and CLIP-LSeg [24] to optimize the Gaussian radiance field. Our proposed approach achieves superior results.

Method	Scene0004		Scene0389		Scene0494		Scene0693		Average	
	mIoU ↑	mAcc ↑								
Feature 3DGS [55]	39.595	85.641	41.677	88.598	50.353	85.575	48.942	76.832	45.142	84.162
Ours	48.051	88.727	47.697	92.773	56.136	89.087	60.801	80.863	53.171	87.862

Table 7. Performance comparison of segmentation results in novel views from the Replica dataset [41], utilizing multi-view training data and CLIP-LSeg [24] to optimize the Gaussian radiance field. Our proposed approach maintains superior results.

Method	Room0		Room1		Room2		Office0		Office2		Office4		Average	
	mIoU ↑	mAcc ↑												
Feature 3DGS [55]	25.169	71.148	25.297	72.115	29.746	80.812	24.226	65.128	34.516	88.565	33.743	81.298	28.783	76.514
Ours	26.043	73.994	27.366	74.608	30.314	82.814	35.263	78.388	39.315	90.007	41.097	82.374	33.233	80.364

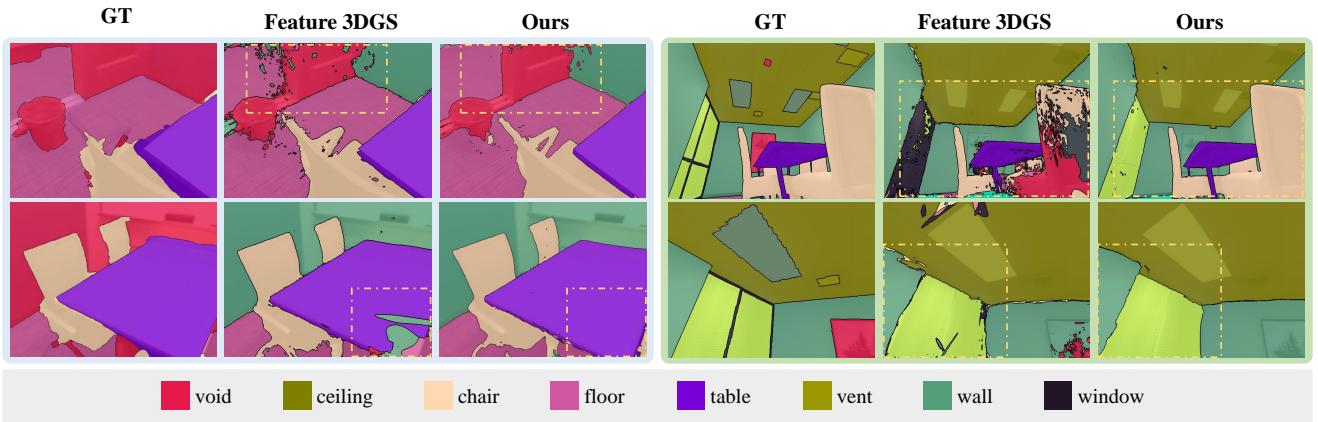


Figure 8. Visual segmentation results of novel views using CLIP-LSeg [24] for semantic Gaussian attribute optimization. Our approach demonstrates more accurate and coherent segmentation results.

3DGS easily produces ambiguous semantics due to a lack of view-consistent constraints, whereas our approach delivers more accurate and coherent segmentation results. These results underscore the effectiveness and generalizability of our method under different CLIP models.

4.3.6 Scene Reconstruction Results

We also report the reconstruction quality to demonstrate that our method not only delivers accurate segmentation results, but also achieves high-quality scene reconstruction.

As shown in Table 8, our method achieves reconstruction quality on par with other 3DGS-based methods and suppresses NeRF-based approaches under both multi-view and sparse-view training conditions. Furthermore, we provide qualitative results in Fig. 9 and Fig. 10. We can observe that our approach consistently renders photo-realistic details across diverse synthetic and real-world scenes. In summary, these results demonstrate the effectiveness of our method in simultaneously reconstructing and semantically

understanding 3D scenes.

4.3.7 Efficiency Comparison

We report the training time and inference results for efficiency comparison, as summarized in Table 9. All evaluations are conducted using a single NVIDIA A100 GPU. The evaluation highlights the superior efficiency of our approach, which outperforms competing methods in both training and inference speed. Specifically, NeRF-based methods suffer from slow speeds due to the computational demands of volume rendering. Although Feature 3DGS adopts a fast splatting technique, it faces an efficiency bottleneck caused by embedding high-dimensional semantic features into 3D Gaussians, which particularly hampers training efficiency. Similarly, LangSplat’s efficiency is limited by its time-consuming autoencoder pre-training and post-processing upsampling processes. In contrast, our approach achieves superior efficiency by modeling compact semantic Gaussian representations through our SAC strat-

Table 8. Quantitative comparison on reconstruction results of novel views in Replica and ScanNet datasets. Our proposed approach maintains reconstruction quality on par with the 3D Gaussian Splatting (3DGS) method [18].

Method	Multi-view Training Data						Sparse-view Training Data					
	Replica [41]			ScanNet [8]			Replica [41]			ScanNet [8]		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
LERF [19]	31.034	0.904	0.103	24.921	0.778	0.392	17.509	0.697	0.484	20.819	0.719	0.429
3DOVS [33]	31.373	0.908	0.091	24.915	0.780	0.389	17.923	0.708	0.477	21.414	0.723	0.422
3DGS [18]	35.477	0.955	0.090	28.265	0.857	0.258	26.333	0.880	0.196	22.713	0.752	0.354
Feature 3DGS [55]	35.439	0.955	0.090	28.453	0.863	0.250	26.313	0.881	0.193	22.224	0.741	0.355
LangSplat [37]	35.490	0.955	0.090	28.441	0.860	0.257	26.247	0.881	0.195	22.125	0.751	0.354
Ours	35.519	0.955	0.090	29.021	0.866	0.250	26.662	0.885	0.191	22.760	0.759	0.350

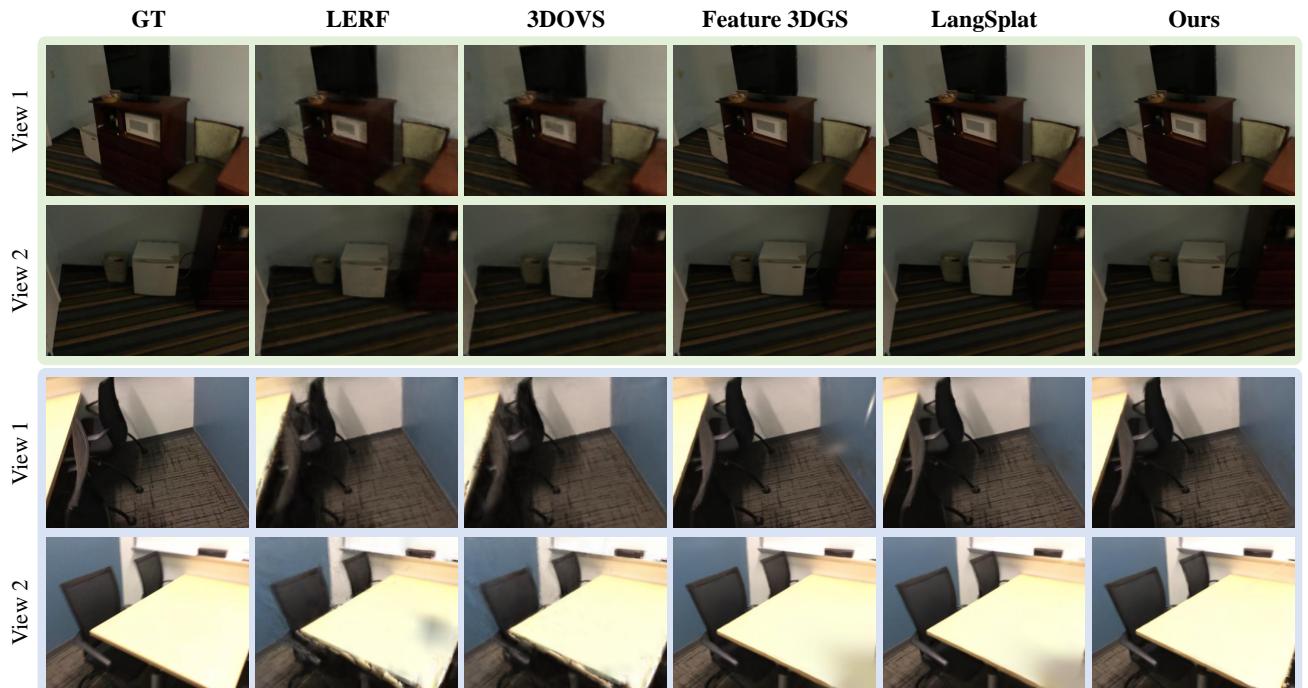


Figure 9. Visual reconstruction results of novel view, under multi-view training data conditions. Our method achieves photo-realistic rendering quality comparable to the Gaussian-based method, outperforming NeRF-based approaches.

egy, enabling a faster rendering speed (>100 FPS).

4.4. Ablation Studies

In this section, we present ablation experiments to evaluate the effectiveness of each component in our approach.

Effectiveness of proposed components. We adopt Feature 3DGS [55] as the experimental baseline. As shown in Table 10, we gradually employ different components, including Semantic Attribute Compactness (SAC), 3D Coherent Regularization (3DCR), and Progressive Densification Regulation (PDR), to evaluate their effectiveness.

- Compared to baseline (a), adding SAC significantly improves inference efficiency. This enhancement is due to SAC’s capacity to embed compact semantic information

into 3D Gaussians. Moreover, SAC’s region-level unified processing refines object boundaries, leading to more precise segmentation results.

- As exhibited in (c) and (d), further applying 3DCR achieves notable performance improvements without adding extra computational overhead during inference. Moreover, combining both 2D and 3D constraints, as demonstrated in (e), leads to better performance, highlighting their complementary nature. These results validate that 3DCR introduces essential semantic-consistent constraints, effectively mitigating ambiguous and coarse semantics and improving the overall coherence of the segmentation results. We further analyze the impact of different regularization weights on the 3DCR loss. As

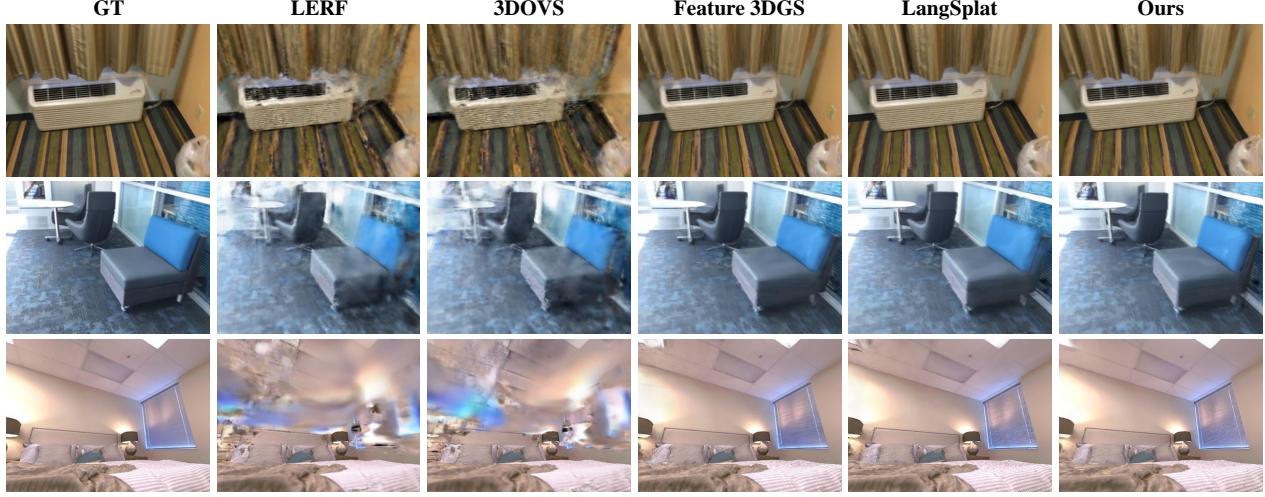


Figure 10. Visual reconstruction results of novel view, under sparse-view training data conditions. Our approach shows robust reconstruction quality across various scenes.

Table 9. Comparison with current methods on training time and inference speed. More details refer to Sec. 4.3.7.

Method	NeRF-based		3DGS-based		
	LERF [19]	3DOVS [33]	Feature 3DGS [55]	LangSplat [37]	Ours
Training Time ↓	~ 2h 10 mins	~ 2h	~ 14h 50 mins	~ 1h 24mins	~ 20mins
Inference FPS ↑	0.2	0.3	2.5	45	190

Table 10. Ablation studies for our approach. SAC: Semantic Attribute Compactness. 3DCR: 3D Coherent Regularization. PDR: Progressive Densification Regulation. "w/o coherent" denotes excluding semantic information integration from adjacent views. Settings (b1)-(b2) assess the impact of different feature dimensions of the semantic parameter, while (c1)-(c4) evaluate the effect of introducing 3DCR at various stages of training. The final configuration is denoted by **Bold**.

Index	Setting	FPS	Room0		Scene0494	
			mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑
(a)	Baseline	2.5	8.794	39.916	25.789	52.006
(b)	(a) + SAC	150	19.920	63.125	36.480	73.529
(c)	(b) + 3DCR ($\mathcal{L}_{consistency}^{2D}$)	150	25.564	71.918	54.023	85.636
(d)	(b) + 3DCR ($\mathcal{L}_{consistency}^{3D}$)	150	23.986	68.810	53.750	83.900
(e)	(b) + 3DCR ($\mathcal{L}_{consistency}^{2D}, \mathcal{L}_{consistency}^{3D}$)	150	26.852	73.542	58.293	87.366
(f)	(e) + PDR (Ours)	190	27.249	74.129	58.543	88.224
(g)	Ours w/o coherent	190	25.445	71.564	56.633	86.863
(h)	Ours w 3DCR ($0.5 * \mathcal{L}_{consistency}^{3D}$)	190	27.055	74.044	58.483	88.037
(i)	Ours w 3DCR ($1.5 * \mathcal{L}_{consistency}^{3D}$)	190	27.739	74.166	58.152	88.005
(b1)	Ours ($d=3$)	190	27.249	74.129	58.543	88.224
(b2)	Ours ($d=8$)	175	27.888	72.995	58.729	88.174
(c1)	Ours ($\mathcal{T}=10k$)	190	24.794	71.036	57.810	87.467
(c2)	Ours ($\mathcal{T}=15k$)	190	27.249	74.129	58.543	88.224
(c3)	Ours ($\mathcal{T}=20k$)	190	26.034	73.068	57.585	86.801
(c4)	Ours ($\mathcal{T}=25k$)	190	24.309	70.265	56.561	86.007

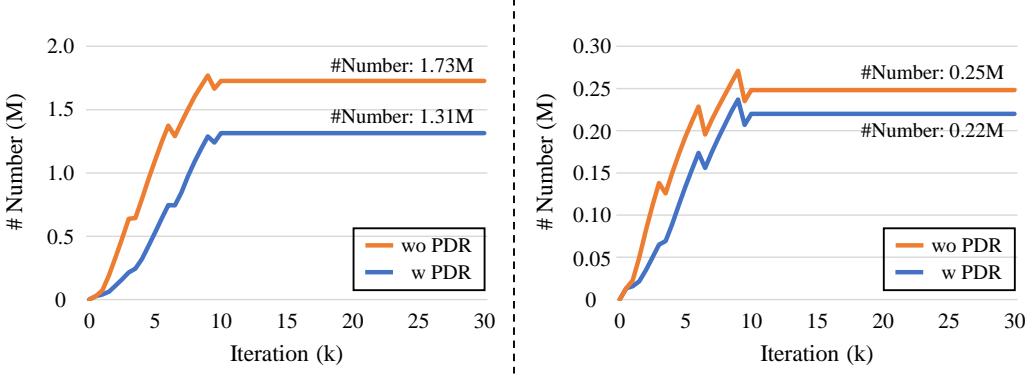


Figure 11. Visualization of the varying count of Gaussians in the Room0 scene (left) and Scene0494 scene (right) during training. The vertical axis indicates the number of Gaussians (“# Number”), while the horizontal axis represents the training iteration (“# Iteration”). These results show that PDR effectively regulates the number of Gaussian, improving rendering efficiency.

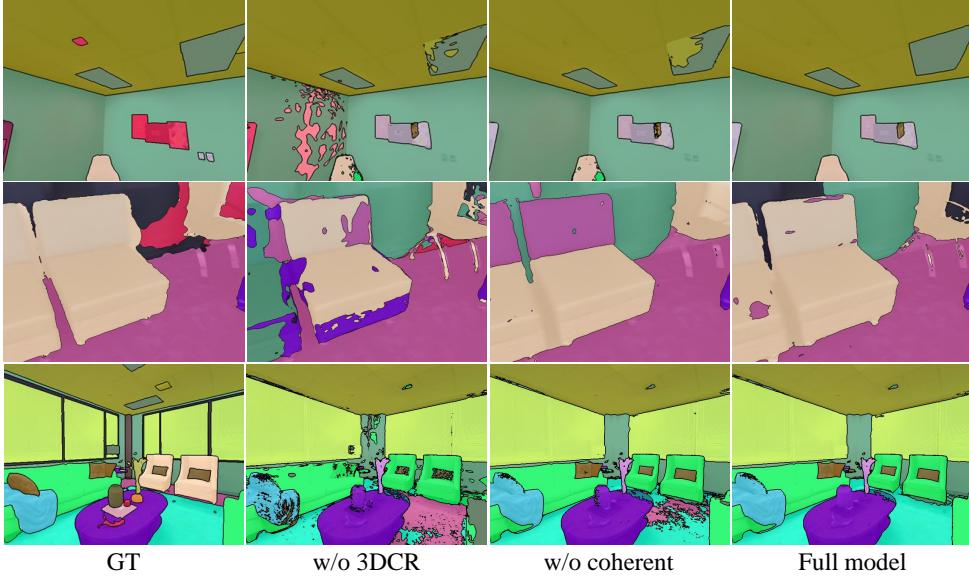


Figure 12. Visual comparison of ablation experiments. As illustrated, incorporating the 3D coherent regularization (2nd column to 4th column) effectively mitigates segmentation ambiguities and improves coherence. Besides, the integration of semantic information from adjacent views for regularization (3rd column to 4th column) enhances object completeness, leading to more precise segmentation results.

- shown in (h) and (i), both weaker and stronger regularization weights maintain robust performance, demonstrating the stability and effectiveness of our 3DCR mechanism across a range of settings.
- As shown in (e), incorporating PDR further boosts computational efficiency and delivers a modest accuracy improvement. This improvement stems from PDR’s ability to regulate the quantity of Gaussian primitives, which not only accelerates rasterization but also facilitates more effective semantic Gaussian radiance field construction. Moreover, Fig. 11 also highlights the effectiveness of PDR in regulating Gaussian quantities.

Additionally, we present visual ablation results in Fig.

12. It can be seen that integrating the 3D coherent regularization significantly reduces semantic ambiguity and enhances consistency compared to removing it (“w/o 3DCR”). Furthermore, incorporating semantic information from adjacent views improves scene coherence compared to omitting it (“w/o coherent”). These results show the effectiveness of our proposed strategies in achieving precise and coherent 3D semantic understanding.

Effectiveness of coherent in 3DCR. As shown in (f) of Table 10, “w/o coherent” denotes the absence of semantics from adjacent views for regularization. Instead, the semantics are only regularized using the current view’s masks from SAM. The regularized semantics are then applied as

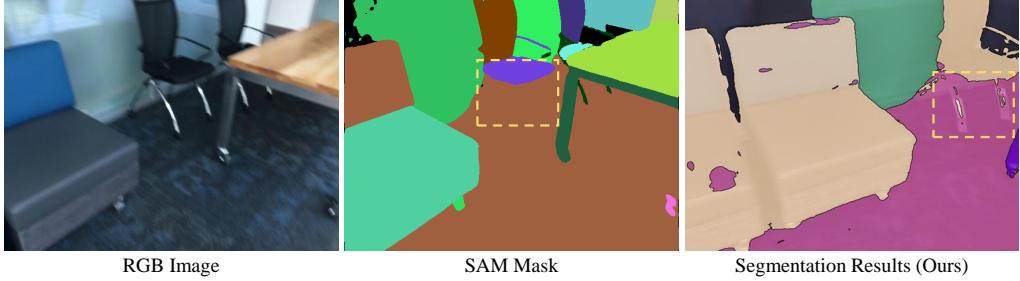


Figure 13. Illustration of a SAM failure case caused by motion blur. The chair leg (yellow dashed box) is incorrectly merged with the floor region in the generated mask, leading to imprecise segmentation.

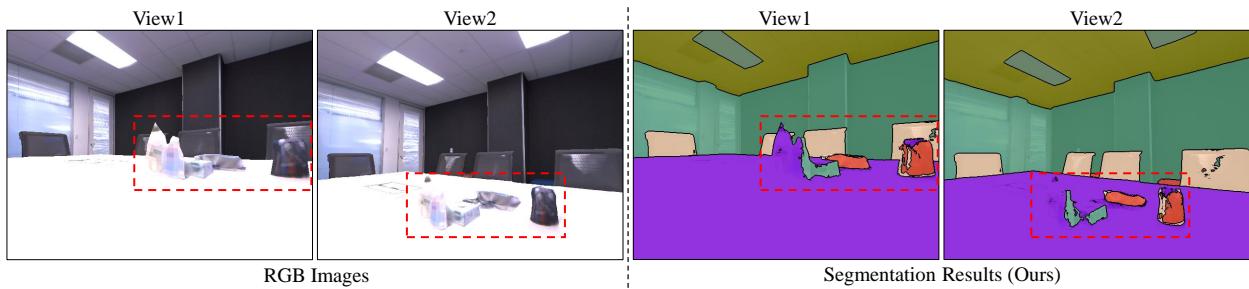


Figure 14. A failure case in a cluttered, low-contrast region (red dashed box), where weak CLIP features result in inaccurate semantic predictions.

Z in Equation (8). The decrease in performance verifies that incorporating semantics from adjacent views improves semantic consistency.

Analysis of d in SAC. We analyze the impact of different d in SAC, illustrated in (b1) - (b2). As d increases, the inference speed declines, coupled with a marginal improvement in mIoU. Therefore, we opt for $d = 3$ as it delivers efficient yet competitive results.

Analysis of \mathcal{T} in 3DCR. We examine the influence of different settings of \mathcal{T} , as depicted in (c1) - (c4). Specifically, $\mathcal{T} = 10k$ denotes the application of our 3D coherent regularization strategy after 10,000 training iterations. When \mathcal{T} is set to 15k, the model achieves overall superior results, making it the chosen final configuration.

4.5. Discussion

Real-World Applications Potential. The proposed CLIP-GS, with its efficient, view-consistent, language-driven semantic understanding capability, holds significant promise for various real-world applications. In robotics, precise 3D scene understanding and text-driven object segmentation can improve navigation, manipulation, and grasp planning while eliminating the need for costly manual annotations. In augmented and virtual reality, CLIP-GS enables scene understanding and natural language querying, supporting immersive interaction and context-aware content placement. In summary, CLIP-GS paves the way for more intelligent

and responsive 3D applications in robotics and AR/VR environments.

Limitations. While our approach achieves precise and efficient 3D semantic understanding, we acknowledge potential limitations caused by the performance of 2D foundation models, such as SAM and CLIP. On the one hand, the quality of object-level region masks generated by SAM may occasionally be imperfect in some complex indoor environments, affecting the results of our method. For example, as shown in Fig. 13, motion blur in the input image can cause a chair’s leg to be incorrectly merged with the floor, resulting in inaccurate mask generation. Such imperfect masks may affect the semantic learning of 3D Gaussians and lead to incomplete segmentation in novel views. Future work may explore enhanced region segmentation strategies or adaptive mask refinement techniques to mitigate this issue. On the other hand, as illustrated in Fig. 14, our method encounters challenges in cluttered or low-contrast regions, where accurate object segmentation becomes difficult. This limitation arises because the semantic supervision relies on CLIP features, which may be weaker in such cases. Future work will integrate more robust vision-language foundation models to enhance robustness in challenging scenarios.

5. Conclusion

In this work, we present CLIP-GS, a novel approach utilizing CLIP-informed 3D Gaussian Splatting for achiev-

ing real-time, view-consistent 3D semantic understanding in indoor Scenes. CLIP-GS incorporates Semantic Attribute Compactness (SAC) to embed compact semantic information into 3D Gaussians, allowing for highly efficient rendering. Additionally, the proposed 3D Coherent Regularization (3DCR) enhances semantic-consistent constraints across multiple viewpoints, facilitating more coherent 3D segmentation. Experimental results demonstrate that our approach significantly outperforms SOTA methods on synthetic and real-world indoor scenes. Furthermore, our method maintains superior performance even under sparse input conditions, substantiating its robustness.

References

- [1] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 3d gaussian splatting: Survey, technologies, challenges, and opportunities. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 3
- [2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 1
- [3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision*, pages 333–350, 2022. 3
- [4] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1316–1326, 2023. 5
- [5] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 3
- [6] Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. Click-gaussian: Interactive segmentation to any 3d gaussians. In *Proceedings of the European Conference on Computer Vision*, pages 289–305. Springer, 2024. 3
- [7] Xuangeng Chu and Tatsuya Harada. Generalizable and animatable gaussian head avatar. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 57642–57670, 2024. 3
- [8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 7, 8, 11, 12, 13
- [9] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics*, 36(4):1, 2017. 7
- [10] Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 3d gaussian splatting as new era: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 3
- [11] Tobias Fischer, Jonas Kulhanek, Samuel Rota Bulò, Lorenzo Porzi, Marc Pollefeys, and Peter Kotschieder. Dynamic 3d gaussian fields for urban areas. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 80466–80494, 2024. 3
- [12] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022. 1
- [13] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting. *arXiv preprint arXiv:2403.15624*, 2024. 3
- [14] Xu Hu, Yuxi Wang, Lue Fan, Junsong Fan, Junran Peng, Zhen Lei, Qing Li, and Zhaoxiang Zhang. Semantic anything in 3d gaussians. *arXiv preprint arXiv:2401.17857*, 2024. 3
- [15] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 10608–10615, 2023. 1
- [16] Dong In Lee, Hyeongcheol Park, Jiyoung Seo, Eunbyung Park, Hyunje Park, Ha Dam Baek, Shin Sangheon, Sangpil Kim, et al. Editsplat: Multi-view fusion and attention-guided optimization for view-consistent 3d scene editing with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3
- [17] Siyu Jiao, Haoye Dong, Yuyang Yin, Zequn Jie, Yinlong Qian, Yao Zhao, Humphrey Shi, and Yunchao Wei. Clip-gs: Unifying vision-language representation with 3d gaussian splatting. *arXiv preprint arXiv:2412.19142*, 2024. 3
- [18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 1, 3, 4, 7, 8, 13
- [19] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 1, 3, 8, 10, 11, 13, 14
- [20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2, 4, 5, 7
- [21] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 23311–23330, 2022. 3, 10

- [22] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, pages 29–43. Wiley Online Library, 2021. 4
- [23] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91, 2024. 3
- [24] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *Proceedings of the International Conference on Learning Representations*, 2022. 1, 3, 7, 11, 12
- [25] Guibiao Liao and Wei Gao. Rethinking feature mining for light field salient object detection. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(10):1–24, 2024. 1
- [26] Guibiao Liao, Wei Gao, Qiuping Jiang, Ronggang Wang, and Ge Li. Mmnet: Multi-stage and multi-scale fusion network for rgb-d salient object detection. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2436–2444, 2020.
- [27] Guibiao Liao, Wei Gao, Ge Li, Junle Wang, and Sam Kwong. Cross-collaborative fusion-encoder network for robust rgb-thermal salient object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7646–7661, 2022. 1
- [28] Guibiao Liao, Jiankun Li, and Xiaoqing Ye. Vlm2scene: Self-supervised image-text-lidar learning with foundation models for autonomous driving scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3351–3359, 2024. 1
- [29] Guibiao Liao, Kaichen Zhou, Zhenyu Bao, Kanglin Liu, and Qing Li. Ov-nerf: Open-vocabulary neural radiance fields with vision and language foundation models for 3d semantic understanding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 1
- [30] Guibiao Liao, Qing Li, Zhenyu Bao, Guoping Qiu, and Kanglin Liu. Spc-gs: Gaussian splatting with semantic-prompt consistency for indoor open-world free-view synthesis from sparse inputs. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11264–11274, 2025. 1
- [31] Bingzheng Liu, Jianjun Lei, Bo Peng, Chuanbo Yu, Wan-qing Li, and Nam Ling. Novel view synthesis from a single unposed image via unsupervised learning. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(6):1–23, 2023. 1
- [32] Caixia Liu, Dehui Kong, Shaofan Wang, Jinghua Li, and Baocai Yin. A spatial relationship preserving adversarial network for 3d reconstruction from a single depth view. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18(4):1–22, 2022. 1
- [33] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulkotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 53433–53456, 2023. 1, 2, 3, 7, 8, 10, 11, 13, 14
- [34] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3
- [35] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, pages 405–421, 2020. 1, 3
- [36] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 3
- [37] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 1, 2, 3, 8, 10, 11, 13, 14
- [38] 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 *Proceedings of the International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 4, 7
- [39] Qihong Shen, Xingyi Yang, and Xinchao Wang. Flashsplat: 2d to 3d gaussian splatting segmentation solved optimally. In *Proceedings of the European Conference on Computer Vision*, pages 456–472. Springer, 2024. 3
- [40] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. *arXiv preprint arXiv:2311.18482*, 2023. 3
- [41] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 7, 8, 11, 12, 13
- [42] Guoxing Sun, Rishabh Dabral, Heming Zhu, Pascal Fua, Christian Theobalt, and Marc Habermann. Real-time free-view human rendering from sparse-view rgb videos using double unprojected textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3
- [43] Xingrui Wang, Cuiling Lan, Hanxin Zhu, Zhibo Chen, and Yan Lu. Gsemssplat: Generalizable semantic 3d gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2412.16932*, 2024. 3
- [44] Yuxin Wang, Qianyi Wu, Guofeng Zhang, and Dan Xu. Learning 3d geometry and feature consistent gaussian splatting for object removal. In *Proceedings of the European Conference on Computer Vision*, pages 1–17, 2024. 3
- [45] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *Proceedings of the European Conference on Computer Vision*, pages 55–71, 2024. 3

- [46] Sizhe Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. Clipself: Vision transformer distills itself for open-vocabulary dense prediction. In *Proceedings of the International Conference on Learning Representations*, 2024. 1, 2
- [47] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. In *Proceedings of the European Conference on Computer Vision*, pages 156–173, 2024. 3
- [48] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *Proceedings of the European Conference on Computer Vision*, pages 162–179. Springer, 2024. 3
- [49] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024. 1
- [50] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1020–1031, 2023. 1
- [51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7
- [52] Ruida Zhang, Chengxi Li, Chenyangguang Zhang, Xingyu Liu, Haili Yuan, Yanyan Li, Xiangyang Ji, and Gim Hee Lee. Street gaussians without 3d object tracker. *arXiv preprint arXiv:2412.05548*, 2024. 3
- [53] Xiaoyu Zhang, Guibiao Liao, Wei Gao, and Ge Li. Tdrnet: Transformer-based dual-branch restoration network for geometry based point cloud compression artifacts. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 1
- [54] Weichao Zhao, Hezhen Hu, Wengang Zhou, Li Li, and Houqiang Li. Exploiting spatial-temporal context for interacting hand reconstruction on monocular rgb video. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(6):1–18, 2024. 1
- [55] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 1, 2, 3, 5, 7, 8, 11, 12, 13, 14
- [56] Wojciech Zielenka, Stephan J Garbin, Alexandros Lattas, George Kopanas, Paulo Gotardo, Thabo Beeler, Justus Thies, and Timo Bolkart. Synthetic prior for few-shot drivable head avatar inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3
- [57] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *arXiv preprint arXiv:2401.01970*, 2024. 3
- [58] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, pages 29–538. IEEE, 2001. 4