C-SAGS: A Unified Framework for Text-Driven 2D-3D Segmentation using Informed 3D Gaussian Splatting

Liu, Jian[a,*], Yu, Zhen[b]

[a]School of Computer and Artificial Intelligence, Zhengzhou University, No.100 Science Avenue, Zhengzhou, Henan, 450001, China
[b]Department of Electrical and Computer Engineering, California State Polytechnic University, Pomona, CA 91768, USA

ARTICLE INFO

Keywords:
2D-3D Segmentation
3D Gaussian Splatting
Segment Anything Model (SAM)
Contrastive Language Image Pre-training (CLIP)

ABSTRACT

In computer vision, 3D segmentation is crucial for scene understanding. This study introduces C-SAGS, a comprehensive framework integrating CLIP, SAM, and 3DGS models for zero-shot, cross-modal 3D segmentation using text and geometric prompts. C-SAGS leverages informed 3D Gaussian Splatting to achieve high-quality 3D scene representation, enabling efficient and accurate segmentation. Experimental results demonstrate improved performance and reduced storage requirements compared to state-of-the-art methods. Our framework supports individual manipulation of segmented objects, facilitating 3D scene editing and reconstruction. We demonstrate a significant enhancement in IoU and accuracy, positioning our work within a broader context of 3D scene comprehension. The code is available at: https://gitee.com/liujian-0819/3DGS.

1. Introduction

Exploring methods for object segmentation is crucial in scene comprehension. Whereas image classification [1][2] and object detection [3][4] concentrate on object-level information, image segmentation addresses classification at the pixel level [5]. Semantic segmentation [6][7][8] involves predicting the class of each pixel within an image. On the other hand, instance segmentation [9][10] focuses on detecting and segmenting pixels that relate to individual object instances. In recent years, a growing number of applications, such as autonomous driving [11][12] , human-computer interaction [13], image search engines [14], and medical image analysis [15], have taken advantage of the knowledge inferred from image segmentation. Despite progress in traditional computer vision and machine learning techniques for image segmentation, more and more attention has been paid to deep learning methods with the flourish of Convolutional Neural Networks (CNN) [16], due to their significant superiority over their counterparts in terms of accuracy and efficiency. The Segment Anything Model (SAM) [17] provides a foundational framework for segmenting 2D images. As illustrated in **Fig.**1, SAM segments everything within a scene without interactive prompts provided by the user. In fact, SAM is trained on a dataset that includes 11 million images and 1.1 billion masks, showing strong zero-shot capabilities in a variety of real-world segmentation tasks. After fine-tuning, SAM can be customized for special fields such as medical image segmentation [18][19] and remote sensing imagery [20][21], among others.

Contrastive Language Image Pre-training (CLIP) [22] represents a foundational approach for aligning text and images through contrastive learning. Trained on billions of text-image pairs, this model excels in zero-shot visual recognition. The visual component of CLIP enhances its semantic understanding, allowing it to grasp extensive semantic contents in images. However, it struggles with tasks necessitating intricate predictions, like segmentation. While SAM leans more towards geometry, emphasizing locality and spatial ability, CLIP leans more toward language, emphasizing wholeness and semantic ability. Therefore, SAM-CLIP [23] aims to develop a robust foundational model capable of comprehending both images and text through the integration of two models, leveraging their respective strengths and compensating for their weaknesses. SAM-CLIP achieves outstanding results in tasks related to image classification and semantic segmentation. This sets a paradigm for multimodal research and underscores the benefits of combining spatial and semantic comprehension in downstream applications.

*Corresponding author
liujian10@zzu.edu.cn (L. Jian)
ORCID(s): 0000-0003-2981-2128 (L. Jian)
[1]Code repository: https://gitee.com/liujian-0819/3DGS
[2]Dataset warehouse: available upon request

Figure 1: SAM segments everything within a scene without interactive prompts from users.

3D segmentation is vital for understanding the spatial geometry of a scene. In computer vision, segmenting images in 3D space is inherently more challenging than in 2D due to the scarcity of 3D-supervised data, the intricacy of algorithms, and the significant computational demands. To address these challenges, various methods have been suggested for extending knowledge from 2D segmentation to a 3D space by learning continuous 3D segmentation from sparse 2D masks. Nonetheless, these methods necessitate an initial representation of the 3D scene. Neural Radiance Fields (NeRF) [24] and 3D Gaussian Splatting (3DGS) [25] are two popular methods known for their effectiveness in 3D representation. NeRF uses a multilayer perceptron (MLP) to implicitly represent a 3D scene and synthesize novel views through differentiable volume rendering. On the other hand, 3DGS explicitly encodes scenes with 3D Gaussians, the inherent properties of Gaussians can greatly improve the speed of both rendering and optimization.

Considering the advantages of 3DGS regarding computational efficiency, we proposed a universal 3D segmentation framework (C-SAGS), which seamlessly integrating CLIP, SAM, and 3DGS models, enabling a zero-shot, cross-modal segmentation in a 3D space by using text and geometric prompts. Specifically, our main contributions can be summarized as follows.

- We developed an integrated framework for 2D-3D segmentation by combining CLIP, SAM, and 3DGS. SAM-CLIP facilitates 2D segmentation for multi-view images based on textual or geometric prompts. Meanwhile, 3DGS is capable of learning a continuous representation of 3D segmentation from the generated 2D masks.
- We employed an informed Gaussian labeled by object ID to differentiate Gaussian ellipsoids in the splatting process. This enables the independent manipulation of each segmented object, thereby facilitating the process of editing 3D scenes.
- Different from a typical Gaussian, we integrate multiple spherical harmonic functions associated with style labels into the Gaussian. This integration allows various style scenes to be represented using just one point cloud, considerably decreasing storage needs.

The rest of this paper is organized as follows. Firstly, **Section 2** introduces the latest advances in both 2D and 3D segmentation methods. **Section 3** concisely outlines the CLIP, SAM, and 3DGS models. **Section 4** provides a detailed description of the proposed model. **Section 5** explains the experimental configuration and the assessment criteria. **Section 6** provides an in-depth qualitative and quantitative evaluation of the experimental results. **Section 7** wraps up the paper with a summary and discussion.

2. Related Works

2D Segmentation: Typically, the image segmentation task is divided into three subcategories: semantic segmentation, instance segmentation, and panoptic segmentation. Instance segmentation offers more comprehensive details about an image compared to semantic segmentation, including the location and number of segmented objects. Over the past few years, deep neural networks (DNN) have significantly contributed to numerous studies in the field of image segmentation.[26][27]. Among the fully-supervised deep learning methods, the single-stage strategy accomplished object positioning and mask generation at the same time, whereas the two-stage or multi-stage strategy achieved them in a certain sequence. At present, two-stage methods are predominant in object segmentation tasks due to the high performance and accuracy. For example, Mask R-CNN [10] is one of the most representative

works. Nevertheless, both two-stage and multi-stage approaches come with significant computational expenses. In contrast, single-stage segmentation methods such as SSD [28], YOLO [29] and YOLACT [30][31] strive to achieve segmentation in a straightforward way, realizing an effective balance between speed and accuracy. The integration of Transformers [32][33][34] into segmentation methods has pioneered a novel segmentation framework, enabling end-to-end segmentation while significantly reducing non-maximum suppression (NMS) during the inference process. For example, Segment Anything Model (SAM) [17][35] establishes a foundational model for high-performance image segmentation, which is based on the Visual Transformer (VIT) [36] .

3D Segmentation: 3D segmentation is an essential task in computer vision [37], significantly contributing to the comprehension of 3D scenes and having a wide range of applications, such as robotics [38], navigation [39], and AR/VR [40][41]. There are several formats available for representing 3D scene, including depth images [42][43][44], meshes [45][46], voxels [47][48], and point clouds.[49][50]. Point clouds are the most prevalent nondiscretized data type within 3D applications, which can be directly captured by 3D scanners or derived from stereo or multiple-view imagery. Deep neural networks [51] have proven superior to traditional geometry-based methods for point cloud segmentation by extracting robust features and providing more reliable geometric information. PointNet, for instance, stands as a point-based methods; however, its major limitation in learning point clouds is the failure to leverage the structural details available within the local point neighborhoods. Consequently, later studies have focused on refining the use of sampling strategies and feature extraction methods. Inspired by the implementation of attention mechanisms, Transformer-based networks [52][53][54] can act as a powerful backbone for various 3D segmentation tasks. Transformer-based methods have shown greater effectiveness in extracting features compared to other architectures, despite these approaches demanding considerable computational resources.

Scene Representation in 3D: Different methods have explored 3D segmentation by employing a range of 3D representations, which can be roughly classified into explicit representations (mesh, point cloud, voxel, volume) and implicit representations (SDF, Occupancy fields, NeRF). The discrete nature of the explicit representation results in artifacts because of inadequate refinement, and it also demands relatively high memory usage. On the other hand, implicit representation employs a function to describe the scene geometry, which is a continuous representation applicable to high resolution and generally does not require 3D supervision. For example, NeRF [24] introduced a 5D neural radiation field method to achieve an implicit representation of complex scenes, using a multilayer perceptron (MLP) to learn volumes, achieving photorealistic image synthesis. However, NeRF suffers from significant drawbacks, including prolonged training times and slow rendering speeds. In contrast, 3D Gaussian Splatting (3DGS) [25] employs Gaussian ellipsoids to represent 3D scenes and renders them using splatting. 3DGS achieves a balance between discrete and continuous characteristics: Within a sphere, it remains continuous and differentiable, whereas across the entire space, each sphere is treated as discrete.

Segmentation with Radiance Fields: Given NeRF's achievements in implicitly representing 3D scenes, a large amount of research has been directed towards 3D segmentation methods based on radiance fields. Zhi et al. [55] introduced Semantic-NeRF, showcasing NeRF's capabilities in semantic propagation and enhancement. NVOS [56] presented an interactive method for selecting 3D objects by training an MLP with specifically tailored 3D features. Other approaches such as N3F [57], DFF [58], LERF [59] and ISRF [60] use self-supervised 2D models to transform 2D visual features into 3D features employing additional feature fields during their training. Moreover, integrating CLIP has led to the development of several methods [61][62] for open-vocabulary 3D segmentation. With SAM becoming more popular, several research efforts [63][64][65][66] are investigating methods to extend its segmentation capabilities for 3D applications. For example, SA3D [63] uses an iterative pipeline to enhance 3D mask grids with SAM. OmniSeg3D [65] utilizes a layered contrastive learning technique to autonomously acquire hierarchical segmentation using multiview 2D masks derived from SAM. SA3D-L [66] introduces an efficient 3D segmentation model that integrates SAM and NeRF, allowing the implicit representation of multiple segmented objects within a single MLP.

The aforementioned 3D segmentation methods often struggle with obtaining 3D-supervised data or depend on intricate algorithms which require significant computational expenses. Employing NeRF to learn 3D segmentation from 2D masks seems a viable alternative; nevertheless, it falls short of real-time segmentation because of NeRF's lengthy training and inference time. To tackle this issue, we present a comprehensive 2D-3D segmentation framework (SAGA) that integrates CLIP, SAM, and 3DGS models to enable multi-scale, cross-modal 3D segmentation with zero-shot or few-shot.

3. Preliminaries

SAM [17] represents a foundational vision model, known for its versatility and precision, allowing diverse input prompts, including points, boxes, and masks. Integrating CLIP [22] with SAM, SAM-CLIP [23] provides a unified open-vocabulary segmentation framework that can use text prompts. Moreover, 3DGS [25] can synthesize new views with 3D consistency, which facilitates the expansion of 2D segmentation into the 3D space. This part offers a brief summary of SAM, CLIP, and 3DGS, respectively, providing essential context for our proposed method.

3.1. Segment Anything Model

SAM is based on the Transformer framework [32] and operates by processing an image $\mathbf{I}$ with user-provided prompts $\mathcal{P}$ to generate masks M that assign a label to each pixel, as described by the equation: $\mathbf{M} = \mathbb{SAM}(\mathbf{I}, \mathcal{P})$. Transformer is a deep learning framework designed to process sequential data, including text, audio, and images. To apply the Transformer for image processing, SAM initially divides the image into smaller segments, each corresponding to a block of pixels. The color values for each block are subsequently transformed into vectors, together providing the input sequence for the Transformer. In the Transformer, each pixel block functions akin to a word, with the complete image resembling a sentence.

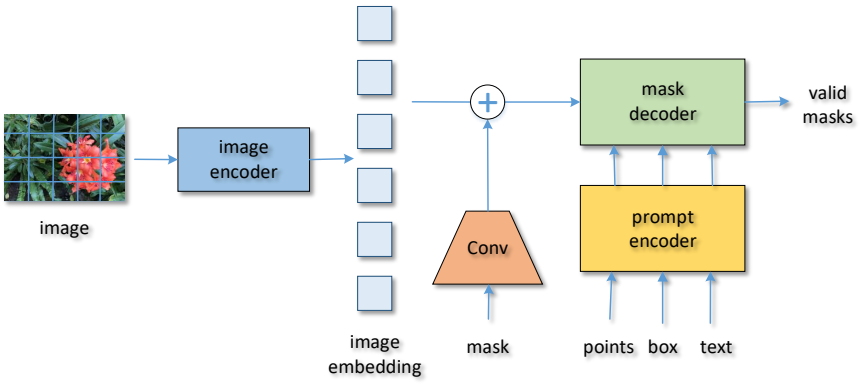

Figure 2: The three main components of Segment Anything Model (SAM), including an image encoder, a prompt encoder, and a mask decoder.

As shown in **Fig.** 2, SAM consists of three primary parts: the image encoder that transforms the image into a vector form, the prompt encoder that changes the prompt into a vector, and the mask decoder that generates segmentation masks. During the inference stage, the image and prompt vectors are merged to serve as input for the mask decoder, which generates segmentation masks.

Image Encoder: The image encoder employs a Vision Transformer (VIT) that has undergone training with MAE to process images. The encoder retrieves features from images that can then be combined with input prompts to predict the masks.

Prompt Encoder: SAM introduces natural language processing (NLP) prompts into image segmentation tasks. The prompt encoder transforms various prompts into a uniform vector, which is then combined with the image vector to create an input sequence for the Transformer.

Mask Decoder: The mask decoder is simultaneously trained with the Transformer using aligned image embeddings and prompts, further improved by incorporating embeddings from four additional tokens. The confidence scores of the segmentation results are determined using the IOU token through an MLP network.

3.2. Contrastive Language Image Pre-training

CLIP is a powerful and scalable model of learning using natural language supervision. During its pre-training phase, CLIP acquires the ability to execute a diverse array of tasks such as OCR, geo-location, and action recognition. It surpasses the top publicly accessible ImageNet models in terms of performance, offering enhanced computational efficiency. The core of CLIP is to use a large amount of paired data of images and text for pre-training to learn the

alignment relationship between images and text. The CLIP model has two modalities; one is the text modality and the other is the visual modality, including two main parts: text encoder and image encoder. In the inference phase, CLIP generates predictions by calculating the cosine similarity between the text and image vectors. This model is particularly suitable for zero-shot learning tasks, where the model can make predictions without requiring new training examples of images or text. **Figure** 3 illustrates the process of contrastive pre-training to align images with text.

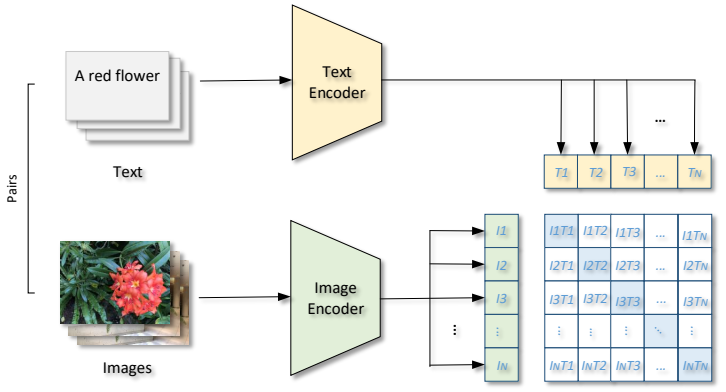

Figure 3: The scenario of contrastive pre-training to align images and text in CLIP.

3.3. 3D Gaussian Splatting

3DGS is an explicit scene representation method, which can be applied for real-time radiation field rendering, achieving visual quality comparable to or even better than previous methods with minimal optimization time. 3DGS consists of three key elements: Firstly, a point cloud-derived camera calibration was employed to optimize the 3D Gaussian representation of the scene through continuous volume path tracking; Secondly, a stable perception optimization/density control algorithm was designed to control 3D Gaussians, significantly improving rendering efficiency and consistency; Finally, a flexible perceptual rendering algorithm was developed that supports dynamic viewpoints and scene element changes, achieving good real-time rendering quality and speed. **Fig.**4 illustrates an overview of the 3DGS workflow.

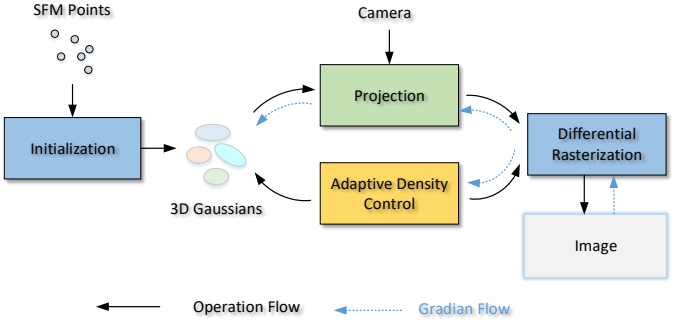

Figure 4: The overall workflow of 3D Gaussian Splatting (3DGS).

Employing a stochastic gradient descent (SGD) optimizer, 3DGS is trained on a dataset comprised of 2D images with camera poses. During the training process, 3DGS attempts to optimize a collection of 3D Gaussian functions. $G = \{G_1, G_2, ..., G_N\}$, where N denotes the number of 3D Gaussians in the scene. The mean μ of a Gaussian denotes its location, and the covariance Σ indicates its scale. Subsequently, 3DGS utilizes a differentiable rasterization to

improve both training efficiency and rendering. Given a certain camera pose, 3DGS projects the 3D Gaussian onto a 2D plane and computes the color $C(p)$ of a pixel p by α-blending a set of ordered Gaussians G_p overlapping the pixel. The color $C(p)$ is specified by **Equation** 1.

$$C(p) = \sum_{i=1}^{|G_p|} c_{G_i} \alpha_{G_i} \prod_{j=1}^{i-1} (1 - \alpha_{G_j}) \tag{1}$$

Where c_{G_i}, α_{G_i} denotes the color and opacity of the i-th Gaussian in $|Gp|$ associated with a specific pixel, respectively.

4. Method

SAM exhibits impressive results in high-resolution image segmentation and localization but struggles with semantic understanding. However, CLIP provides a robust image framework for semantic understanding. Thus, the combination of SAM and CLIP enhances their strengths while compensating for weaknesses in each model. This composed model demonstrates significant prompt-based capabilities, such as zero-shot segmentation from text and geometric prompts. 3DGS learns an explicit 3D scene representation utilizing Gaussian ellipsoids, facilitating the transfer of 2D segmentation expertise into a 3D environment. Consequently, we develop a comprehensive framework that integrates CLIP, SAM, and 3DGS models to accomplish 3D segmentation based on text and geometric prompts.

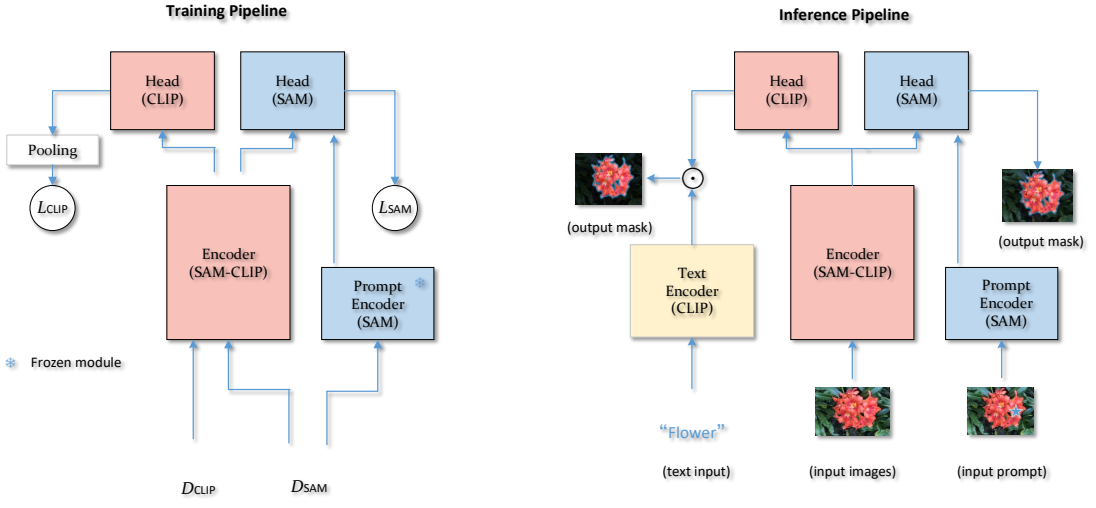

Figure 5: The multi-head architecture for SAM-CLIP, which includes five components: Head(SAM), Head(CLIP), Encoder(SAM-CLIP), PromptEncoder(SAM) and TextEncoder(CLIP).

4.1. Text-Driven Segment Anything Model

In the proposed framework, we employed SAM-CLIP [23] as the paradigm for 2D object segmentation. SAM-CLIP operates as a cross-modal open-vocabulary model driven by prompts including text, points, boxes, and masks. During the training phase, SAM-CLIP does not utilize any textual data directly. Instead, it derives all semantic information from a pre-trained CLIP model. To integrate the SAM and CLIP vision encoders into a single backbone, SAM serves as the base model in the design architecture, with the CLIP model functioning as the auxiliary model. These two models create an intricate pair that can be effectively applied in various tasks and demonstrate complementary strengths. In **Fig.** 5, SAM-CLIP employs a multi-head configuration. The base model (SAM) comprises an image encoder, a prompt encoder, and a mask decoder. The auxiliary model (CLIP) consists of an image encoder and a text encoder. $Head_{\text{SAM}}$ is initialized with the mask decoder of SAM and $Head_{\text{CLIP}}$ is initialized with random weights. In this scenario, the subsets D_{SAM} and D_{CLIP} pertain to SAM and CLIP, respectively. The image encoders of SAM and CLIP are integrated into a unified $Encoder_{\text{SAM-CLIP}}$ through the process of knowledge distillation (KD). In addition, $\mathcal{L}_{\text{CLIP}}$

and $\mathcal{L}_{\text{SAM}}$ denote the cosine distillation loss [67]. The training process is divided into two stages. Initially, head probing is performed to identify the parameters of $Head_{\text{CLIP}}$. This is followed by the joint training of $Head_{\text{SAM}}$, $Head_{\text{CLIP}}$, and $Encoder_{\text{SAM-CLIP}}$ using a multitask distillation loss: $\mathcal{L} = \mathcal{L}_{\text{CLIP}} + \lambda\mathcal{L}_{\text{SAM}}$.

SAM-CLIP is a multitask framework that integrates the SAM and CLIP heads. This combination facilitates zero-shot semantic segmentation. Upon providing the image to $Encoder_{\text{CLIP-SAM}}$, $Head_{\text{CLIP}}$ generates low-resolution mask predictions based on text prompts. Next, the point prompts are derived from the mask prediction, and both the mask prediction and point prompts are fed into the prompt encoder together. $Head_{\text{SAM}}$ takes embeddings from both the prompt encoder and the image encoder to generate high-resolution mask predictions. Generally, SAM predictions often result in segmenting only a part of the object indicated by point prompts, instead of segmenting the entire semantic object class. This suggests that the CLIP-head is essential in semantic segmentation as it provides the SAM-head with semantic understanding, whereas the point-prompting method utilized in SAM does not suffice for semantic segmentation. Moreover, point prompting is unsuitable for zero-shot semantic segmentation as it relies on points provided by humans. In contrast, SAM-CLIP needs only text prompts representing each object class (e.g., "person", "flower", "car") to autonomously produce semantic masks.

4.2. Informed 3D Gaussian Splatting

In this study, we implemented an informed 3D Gaussian splatting by assigning labels of the object and the style to each Gaussian ellipsoid, allowing for differentiation of them during image rendering. In addition, the informed 3D Gaussian can learn representations of same scene in different styles by employing a set of Gaussians, thereby significantly decreasing the storage requirements for the point cloud.

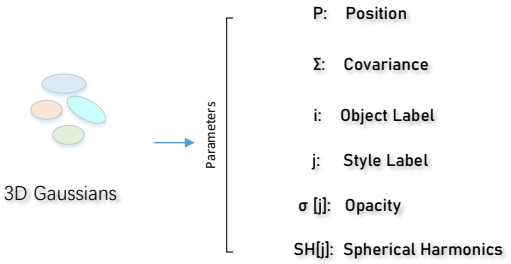

Figure 6: The parameters of the informed Gaussian functions. Here, i and j are the labels of object and style, respectively.

3D Gaussian Representation: Employing a 3D Gaussian representation allows for effective handling of entire scenes, even those with intricate backgrounds and significant depth variations. Moreover, this method can achieve high-quality rendering of structural and depth complexity without the need for normal information, which is particularly important for sparse point-cloud data where estimating normals is highly challenging. Gaussian functions are frequently utilized to represent fuzzy points or a small area with volume in space. 3D Gaussian is characterized by its mean (position), covariance (shape and scale), and sometimes additional parameters. The parameters of the informed Gaussian functions are shown in **Fig.** 6. Here, P is the center point of a Gaussian ellipse, which is its mean in a 3D space. The covariance matrix Σ describes the degree of diffusion or ambiguity of a point, as well as its extension in different directions, which can describe the curvature of an object's surface or the blurriness in volume rendering. σ[j] denotes an opacity array, while SH[j] is a spherical harmonic function array, both indexed with the style label j. Spherical harmonic functions (SH) are employed to model the light distribution and reflection characteristics in a given space, operating as color functions that vary with the observer's perspective.

Differential 3D Gaussian Splatting: Starting from a sparse SfM point cloud, 3DGS achieves high-quality novel view synthesis (NVS). To accomplish this, it is essential to have a representation that preserves the properties of a differentiable volume representation, while also having unstructured and explicit attributes. The 3D Gaussian function is chosen because it is differentiable and can be easily projected onto a 2D plat, facilitating fast α-blending in rendering. The Gaussian function is defined by a 3D covariance matrix Σ, which is centered on a point μ in the world coordinate system. For a given view transformation W, the covariance matrix Σ' in the camera coordinate system is described in

Equation 2.

$$\Sigma' = JW\Sigma W^T J^T \tag{2}$$

Where J is the Jacobian matrix associated with the affine approximation of the projection transformation. The covariance matrix Σ resembles the description of the shape of a Gaussian ellipsoid. Given a scaling matrix S and a rotation matrix R, the corresponding Σ is derived in **Equation** 3.

$$\Sigma = RSS^T R^T \tag{3}$$

To independently optimize these two factors, which are stored separately: a 3D vector S for scaling and a quaternion q for rotation. These can be easily converted into their respective matrices and combined to ensure normalization q to obtain effective unit quaternions. This anisotropic covariance representation method enables the optimization of 3D Gaussian to capture geometric structures of various shapes in the scene, ultimately obtaining a fairly compact representation.

Optimization with Adaptive Density Control: Initially, starting from the set of sparse points obtained from SfM algorithms, the model gradually adjusted the density of the Gaussian function set, allowing for a transition from an initial sparse dataset to a denser representation. To address the problem of insufficient reconstruction of small-scale geometric structures, a strategy of replicating and splitting Gaussian functions was adopted. These operations enable the optimization process to fine-tune the distribution of Gaussian functions to more accurately match the geometric structure of the scene. This adaptive density control strategy not only increases the richness of details, but also helps to avoid excessive consumption of computing resources. The dynamic adjustment of the α-value and the reasonable distribution of Gaussian functions enable the model to progressively converge over successive iterations, leading to a concise and accurate 3D scene representation.

Fast Differentiable Rasterizer: The rasterizer can quickly process a large number of 3D Gaussians in the scene, making it suitable for real-time rendering. The design of the rasterizer allows it to consider the position, size, and shape of Gaussian functions during rendering, as well as how they are projected onto the view plane. The differential capability of the rasterizer means that it can adjust the parameters of 3D Gaussians in reverse based on pixel errors in the final image. This is achieved by calculating gradients during the rasterization process and backpropagating these gradients to the corresponding Gaussian parameters. In order to improve computational efficiency and reduce computation time, this method may include a custom CUDA kernel, which enables the gradient calculation process to fully utilize the parallel processing capabilities of modern GPUs.

Training and Inference: The images within the dataset, along with their corresponding masks derived from SAM-CLIP, are employed for model training. The original point cloud is generated using the COLMAP tool. Each image is linked to a class label as well as a style label, which can be allocated to the Gaussian function during the training process. Multiple style scenes are successively trained using the same Gaussian set. After the initial scene training has been finished, only the parameters associated with SH and σ are optimized in subsequent training sessions, with the number and geometry of the Gaussian remaining intact. The Gaussian's adaptive density control has been disabled.

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \tag{4}$$

The objective of the training is to create a dense Gaussian set that effectively represents the scene for novel view synthesis. Besides position P, opacity α, and covariance Σ, we also optimized the SH coefficients K that represent Gaussian color to accurately capture the viewpoint-dependent appearance of the scene. The optimization of these parameters and the control of Gaussian density are interwoven together to better represent the scene. In fact, optimization is performed using random gradient descent, and the loss function $\mathcal{L}$ is defined in **Equation** 4. Here λ is the adjustment coefficient of the L1 loss and the D-SSIM loss.

4.3. 3D Segmentation and Scene Manipulation

Utilizing the pre-trained SAM-CLIP, we can use textual or geometric prompts (such as points, boxes, and masks) to infer 2D masks for the specified objects in the images. Given a set of images captured from different angles, 3DGS enables the transformation of 2D segmentation into 3D space by learning from sparse 2D masks. The feature of 3DGS guarantees that the produced masks preserve 3D consistency over successive perspectives. Since each segmented object and the background in a scene can be individually represented using informed Gaussian point clouds, it enables diverse scene modifications, such as inserting, removing, and manipulating objects, among other changes.

5. Experiments

5.1. Datasets

For the experiments, we utilized three NERF datasets: Synthetic(I) , LLFF(II) and LLFF-360(III). The Synthetic dataset comprises eight fabricated scenes, each scene presented in four different styles (real, normal, depth, mask), whereas the LLFF dataset contains eight real-world scenes, each scene converted into four diverse styles (real, gray, hot jet, inverse). In addition, the LLFF-360 dataset consists of images captured by a camera encircling the real scene.

5.2. Baselines

We compared our model with some baseline models to evaluate the performance of 3D segmentation. Within these SAM-based models, SA3D [63] introduces an approach for 3D segmentation utilizing NeRFs by employing inverse mask rendering and self-prompting from diverse perspectives. SAGA [64] is an interactive method for 3D segmentation that combines SAM and 3DGS. SA3D-L [66] is a lightweight framework designed to learn 3D segmentation from 2D masks generated by SAM.

5.3. Evaluation Metrics

The peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [68] are commonly employed to assess the similarity between predicted images and their ground truth. Moreover, the intersection-over-union ratio (IoU) is used in 2D segmentation to measure the degree of overlap between the annotated mask and the predicted mask. In this study, Precision, Recall, and Accuracy metrics are utilized for the quantitative evaluation of segmentation performance, with an IoU detection threshold set at 85%.

5.4. Experiment Settings

The model was developed and trained using a single NVIDIA A4500 GPU, and the average convergence time ranged from 20,000 to 30,000 iterations. For real scenes, the Gaussian point cloud was initialized from a sparse point cloud generated by the COLMAP software, whereas for synthetic scenes, it was randomly initialized. The Gaussians were iteratively optimized through the SGD during the training process. The Gaussian densification started from 500 iterations and concluded at 15,000 iterations.

6. Results

6.1. 3D-Aware Image Generation using Informed Gaussians.

In 3DGS, 3D scenes are represented explicitly by Gaussian ellipsoids, and their parameters are optimized during the training process. Consequently, each scene must be individually depicted by a distinct Gaussian point cloud. It is apparent that storing point clouds requires substantial memory, especially for large-scale scenes. Therefore, we developed an informed Gaussian that integrates object and style labels for the classification of Gaussian ellipsoids. Unlike a typical Gaussian, the informed Gaussian utilizes multiple Spherical Harmonics (SH) functions to generate a color array indexed by the style label. Leveraging a set of Gaussian ellipsoids, it is possible to learn the same scene featuring various styles. During the rendering process, object labels are utilized to differentiate Gaussian ellipsoids associated with the segmented objects.

The quantity evaluations in terms of PSNR for the generated images on NERF synthetic (I) and LLFF(II) datasets are illustrated in **Fig.**7. The scores are grouped by classes or styles, respectively. In **Fig.**8, the model was trained on the Synthetic dataset (I). The training started with a randomly initialized point cloud, and the representation of the real scene (j=0) was learned by optimizing the parameters within the Gaussian ellipsoids. The adaptive regulation of the 3D Gaussian density was achieved through cloning and splitting strategies. For the same scene with different styles (j=1,2,3,...), only the opacity σ[j] and the Spherical Harmonics function SH[j] were trainable, leaving other parameters intact. During the training process, adaptive 3D Gaussian density control was disabled. In **Fig.** 9, the model was trained on the LLFF dataset (II) in the same manner. The camera poses and the initial sparse point cloud for the real images were derived from the COLMAP libraries.

6.2. Novel Styles Synthesis via α-Blending.

Gaussian splatting involves projecting a 3D Gaussian (ellipsoid) onto a 2D image space (ellipse) for rendering. The color of the Gaussian is represented using a spherical harmonics (SH) function. Incorporating multiple SH functions

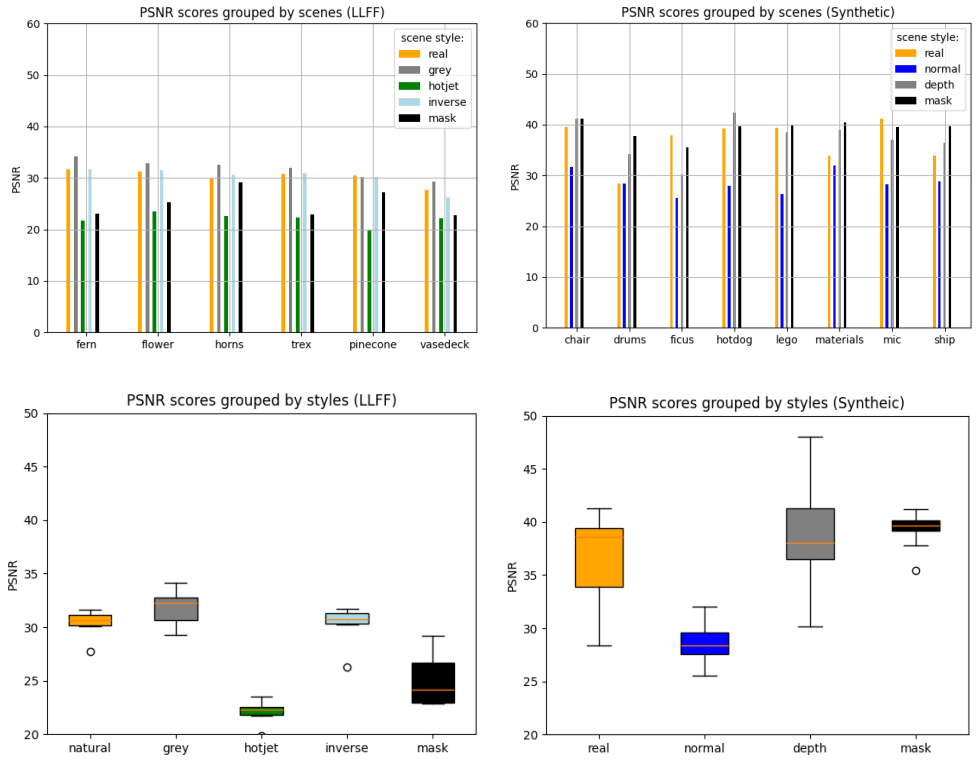

Figure 7: The quantity evaluation in terms of PSNR for the generated images on NERF synthetic (I) and LLFF(II) datasets.

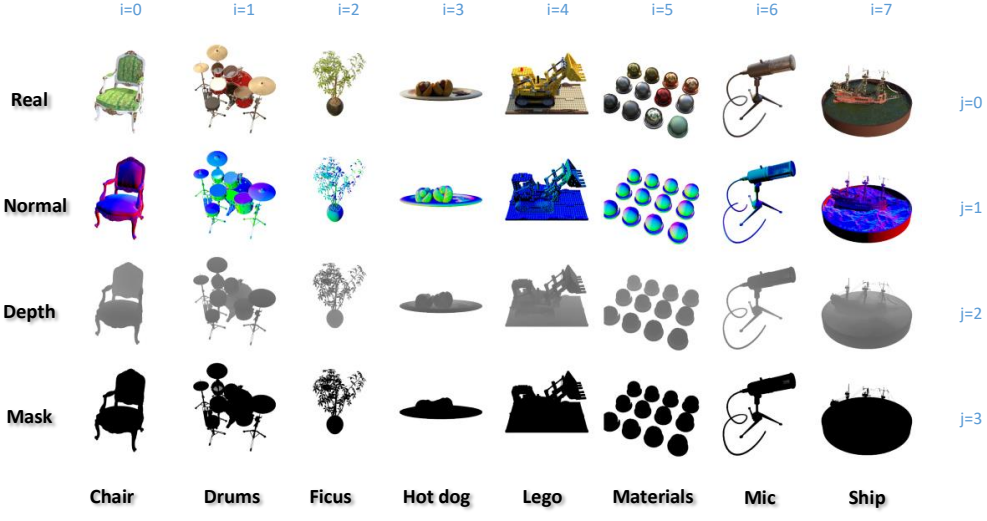

Figure 8: The generated images with different styles (j=0,1,2,3) on **Synthetic** dataset (I).

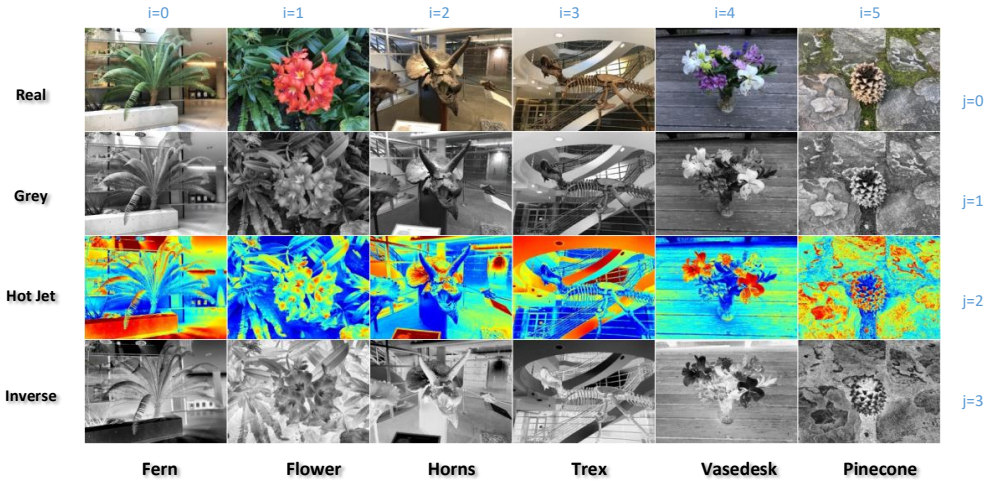

Figure 9: The generated images with different styles (j=0,1,2,3) on the **LLFF** dataset (II).

into the Gaussian allows it to output an array of colors c[j], with each color indexed by style labels. As shown in **Fig.**10, the fusion of the colors of the Gaussian produces a new color. The α-blending is defined in **Equation** 5.

$$c = (1.0 - \alpha) \cdot c[i] + \alpha \cdot c[j] \tag{5}$$

Here, α represents the blending coefficient, which varies between 0 and 1.0. Besides colors, other features such as texture, material, and illumination can also be synthesized using the α-blending method.

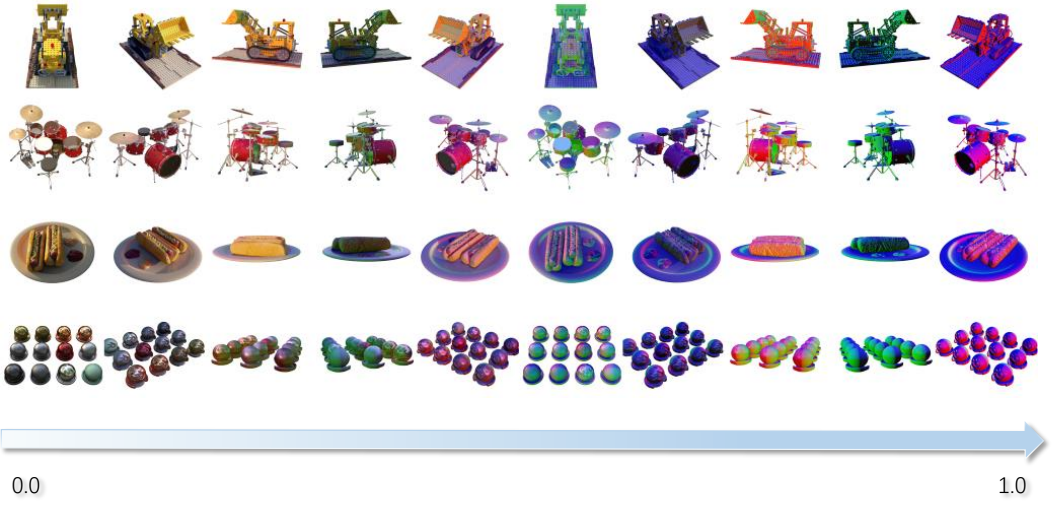

Figure 10: Novel style synthesis by α-blending the predicted colors of Gaussians. α was adjusted gradually from 0 to 1.0.

Table 1
Quantitative analysis comparing our model with other methods on the LLFF dataset.

Model /Scene	SA3D-L[66]		3DGS[25]		OURS	
	(psnr)	(ssim)	(psnr)	(ssim)	(psnr)	(ssim)
flower	23.57	0.978	22.03	0.975	**25.24**	**0.980**
fern	18.84	0.947	18.31	0.943	**23.06**	**0.969**
trex	19.04	0.957	20.85	0.962	**22.86**	**0.966**
horns	24.06	0.971	25.34	0.984	**29.15**	**0.989**
pinecone	22.87	0.967	27.00	0.986	**27.68**	**0.985**
vasedeck	20.56	0.952	**22.77**	**0.974**	21.07	0.953
Mean	21.49	0.962	22.72	0.971	**24.84**	**0.974**

6.3. Ablation Studies

Our model and SA3D-L both employ SAM for 2D segmentation, where SAM acts as a baseline to generate ground truth masks. These SAM-based models adopt different approaches to lift 2D to 3D segmentation. SA3D-L utilizes NERF as a backbone to generate new masks while maintaining 3D consistency, whereas 3DGS and our model use 3D Gaussian and informed 3D Gaussian techniques, respectively. To prove its superiority, we performed some ablation studies as follows. Table 1 provides a quantitative comparison across different models, indicating that our model achieves the highest quality in PSNR and SSIM metrics. Furthermore, as seen in **Fig.**11, the masks generated by our model present superior visual quality when compared to alternative approaches.

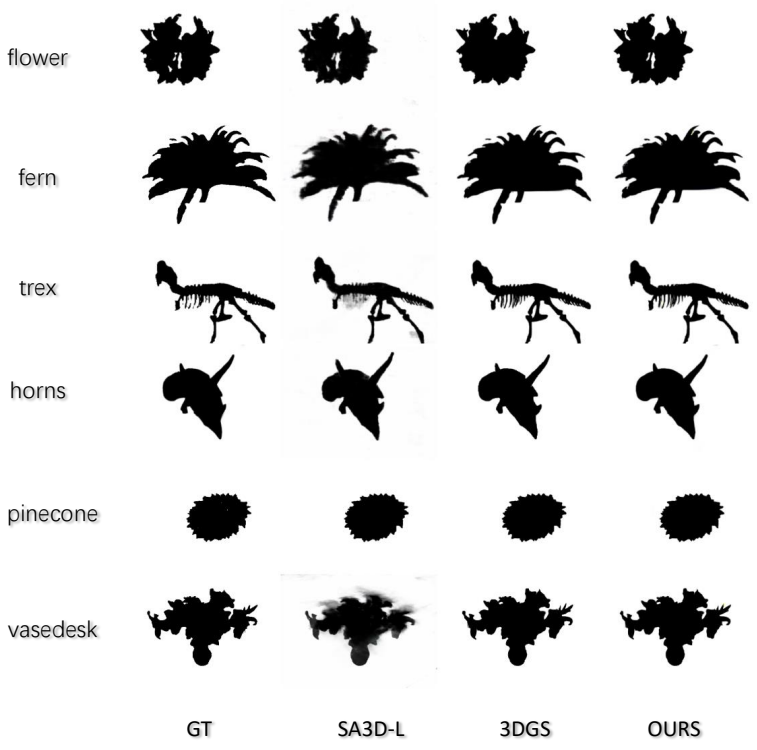

Figure 11: Visual comparison of the inference masks from these models against Ground Truth (SAM).

Table 2
Comparison of our model with SA3D-L in terms of storage requirements and computational costs.

Metrics /Models	Styles Num. (n)	Model size (MB)	Training time (hours)	Inference time (seconds)
SA3D-L w/NERF	n=1	14.12	8.23	6.931
	n=2	14.12	8.52	7.875
	n=3	14.12	7.68	7.573
	n=4	14.12	8.08	7.658
OURS w/Info.GS	n=1	154.5	0.06↓	0.004↓
	n=2	154.5	0.13↓	0.003↓
	n=3	154.5	0.19↓	0.005↓
	n=4	154.5	0.26↓	0.004↓

Table 3
Quantitative comparison of our model against state-of-the-art Models on the LLFF dataset.

Model	SA3D [63]		SA3D-L[66]		3DGS [64]		OURS	
/Scene	IoU%	Acc%	IoU%	Acc%	IoU%	Acc%	IoU%	Acc%
Fern	97.1	99.6	94.8	99.2	80.5	96.4	92.4	98.4
Orchids	83.6	96.9	90.5	98.3	94.4	99.5	94.0	98.5
Horns	94.5	99.0	94.7	99.4	95.8	99.4	95.2	99.7
Fortress	98.3	99.8	98.4	99.8	96.4	99.2	97.4	99.2
Room	88.2	98.3	93.2	98.1	92.5	98.6	94.5	98.6
Leaves	97.2	99.9	90.6	98.5	95.3	99.5	96.7	99.6
Mean	93.2	98.9	91.6	98.5	94.5	99.1	**95.0**	**99.5**

The performance of our model surpasses that of the conventional 3DGS, as the latter depends exclusively on training with 2D masks, whereas our model is trained together with a real-world scene, utilizing a shared Gaussian point cloud. In **Table** 2, we compared our model against SA3D-L in regard to memory requirements and computational costs. While our model greatly accelerates both training and inference speeds, it demands more storage capacity. Moreover, we note that as the number of scenes with different styles increases, the storage demand for our model remains invariable. It is because the multiple scenes actually share the same point cloud, which depicts the geometric structure of the scene.

6.4. Comparison of the 3D Segmentation Performance with SOTA Methods.

Up to now, various approaches such as SA3D, SA3D-L, and 3DGS have been developed to extend 2D segmentation into 3D space. These methods begin by using SAM to segment objects in 2D images. The generated sparse masks are subsequently employed to train 3D representation models such as NERF and 3DGS, enabling the production of new masks across a range of continuous perspectives. In **Table** 3, we present a quantitative comparison of our model against state-of-the-art methods, demonstrating a noticeable improvement in the segmentation capability of our model when compared to other models.

6.5. Modifying 3D Scenes by Adjusting the Segmented Objects

In 3DGS, scenes are represented through a Gaussian point cloud. Each Gaussian ellipsoid in the point cloud is assigned a class label inferred from multi-view 2D masks. Consequently, we can independently manipulate a subset of the Gaussian point cloud during rendering. As illustrated in **Fig.**12, the appearance of the segmented objects and their backgrounds can be modified separately, enabling the composition of a novel scene not present in the original dataset. Moreover, as illustrated in **Fig.**13 and **Fig.**14, we can insert or remove the segmented objects within the scene.

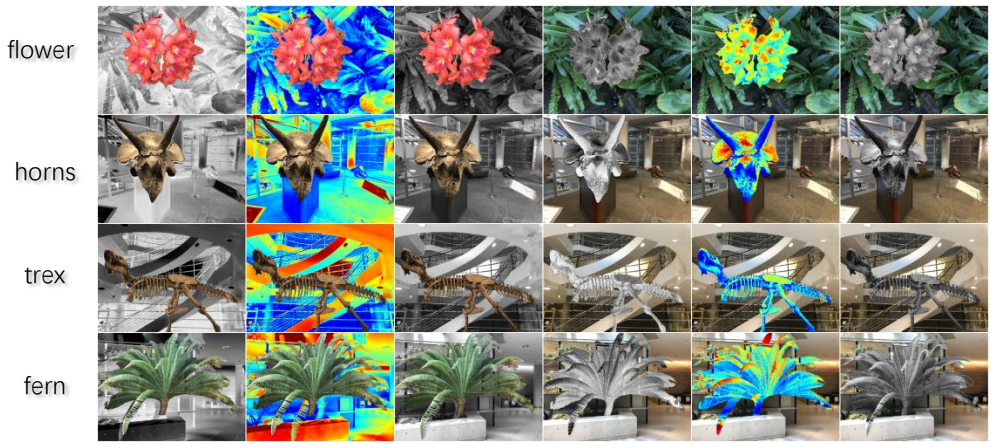

Figure 12: The styles of the segmented objects and backgrounds are manipulated individually.

Figure 13: The segmented object can be added progressively into a scene.

Figure 14: The segmented object can be removed progressively from a scene.

7. Conclusion

The progress of 3D segmentation is greatly hindered by the scarcity of 3D-supervised data and the high demands on computational resources. To address this issue, we presented a text-driven 2D-3D segmentation framework that leverages Gaussian splatting to extend 2D segmentation expertise into 3D space. SAM-CLIP is an integrated model that combines SAM with CLIP, responsible for generating 2D masks from textual or geometric prompts. Subsequently, 3DGS can learn a continuous representation of 3D segmentation from 2D masks with sparse perspectives. The objective of this work was to enhance 3D scene representation by integrating extra parameters into Gaussian, allowing for better differentiation of Gaussian ellipsoids during the rendering process. The informed Gaussian outputs various colors linked with style labels by including multiple SH functions in Gaussian, significantly reducing the storage needs for

the same scene with different styles. The experimental results revealed that the proposed method outperformed state-of-the-art methods in terms of IoU and accuracy. Additionally, it supports 3D scene editing and reconstruction through the separate manipulation of segmented objects and their backgrounds. This method is applicable for 3D comprehension in a variety of domains, such as medical imaging, remote sensing, autonomous driving, robotics, and virtual/augmented reality.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this work.

CRediT authorship contribution statement

Liu, Jian: Conceptualization of this study, Methodology, Programming. **Yu, Zhen:** Writing - Original draft revision.

Acknowledgments

This work is supported by the National Supercomputing Center in Zhengzhou.

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE*, 2016.

[3] Tsung Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] Wei Wei, Yu Cheng, Jiafeng He, and Xiyue Zhu. A review of small object detection based on deep learning. *Neural Computing & Applications*, 36(12), 2024.

[5] Wenchao Gu, Shuang Bai, and Lingxing Kong. A review on 2d instance segmentation based on deep neural networks. *Image and vision computing*, (Apr.):120, 2022.

[6] Liang Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 2018.

[7] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. 2017.

[8] Yifei Zhang, Désiré Sidibé, Olivier Morel, and Fabrice Mériaudeau. Deep multimodal fusion for semantic image segmentation: A survey. *Elsevier*, 2021.

[9] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. *Springer, Cham*, 2014.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2017.

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2012.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *Computer Science*, 2016.

[14] Ji Wan, Dayong Wang, Steven C. H. Hoi, Pencheng Wu, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the ACM International Conference on Multimedia*, 2014.

[15] Yan Xu, Yang Li, Mingyuan Liu, Yipei Wang, Maode Lai, and I Chao Chang. Gland instance segmentation by deep multichannel side supervision. *IEEE Transactions on Biomedical Engineering*, PP(99):1–1, 2016.

[16] Sakshi Indolia, Anil Kumar Goswami, S. P. Mishra, and Pooja Asopa. Conceptual understanding of convolutional neural network- a deep learning approach. *Procedia Computer Science*, 132:679–688, 2018.

[17] Xueyan Zou el al. Segment everything everywhere all at once. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition CVPR'23*, 2023 https://doi.org/10.48550/arXiv.2304.06718.

[18] Wu et al. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv e-prints*, 2023. https://arxiv.org/abs/2304.12620.

[19] Jiayuan Zhu, Yunli Qi, and Junde Wu. Medical sam 2: Segment medical images as video via segment anything model 2. *arXiv e-prints*, 2024. https://arxiv.org/abs/2408.00874.

[20] Lucas Prado Osco, Qiusheng Wu, Eduardo Lopes de Lemos, Wesley Nunes Gonçalves, Ana Paula Marques Ramos, Jonathan Li, and José Marcato. The segment anything model (sam) for remote sensing applications: From zero to one shot. *International Journal of Applied Earth Observation and Geoinformation*, 124:103540, 2023.

[21] Keyan Chen, Chenyang Liu, Hao Chen, Haotian Zhang, Wenyuan Li, Zhengxia Zou, and Zhenwei Shi. Rsprompter: Learning to prompt for remote sensing instance segmentation based on visual foundation model. *IEEE Transactions on Geoscience and Remote Sensing*, page 62, 2024.

[22] Alec et al. Radford. Learning transferable visual models from natural language supervision. 2021.

[23] Haoxiang Wang et al. Sam-clip: Merging vision foundation models towards semantic and spatial understanding, 2024.

[24] et al. Mildenhall. Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv e-prints*, 2020. https://arxiv.org/abs/2003.08934.

[25] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[26] Y Lecun, Y Bengio, and G Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[27] Schmidhuber and Jürgen. Deep learning in neural networks: An overview. *Neural Netw*, 61:85–117, 2015.

[28] Liu Wei, Anguelov Dragomir, Erhan Dumitru, Szegedy Christian, Reed Scott, Fu Cheng-Yang, and Alexander C Berg. Ssd: Single shot multibox detector. *Springer, Cham*, 2016.

[29] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv e-prints*, 2018.

[30] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019.

[31] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact++ better real-time instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 44(2):1108–1121, 2022.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv*, 2017.

[33] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Ying Shan, Bin Feng, and Wenyu Liu. Tracking instances as queries. 2021.

[34] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Solq: Segmenting objects by learning queries. 2021.

[35] Nikhila et al. Ravi. Sam 2: Segment anything in images and videos. *arXiv e-prints*, 2024. https://arxiv.org/abs/2408.00714.

[36] Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations ICLR'21*, 2021. https://arxiv.org/abs/2010.11929.

[37] Sun et al. Attention is all you need. *Vis. Intell.*, 2(14), 2024. https://doi.org/10.1007/s44267-024-00046-x.

[38] R. Rusu, Zoltán-Csaba Márton, Nico Blodow, Mihai Emanuel Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics Auton. Syst.*, 56:927–941, 2008.

[39] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning, 2016.

[40] Florian Wirth, Jannik Quchl, Jeffrey Ota, and Christoph Stiller. Pointatme: Efficient 3d point cloud labeling in virtual reality. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[41] Jianwei Li, Wei Gao, Yihong Wu, Yangdong Liu, and Yanfei Shen. High-quality indoor scene 3d reconstruction with rgb-d cameras: A brief review. *Computational Visual Media*, 8(3):369–393, 2022.

[42] Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. 2018.

[43] Max Schwarz, Anton Milan, Arul Selvam Periyasamy, and Sven Behnke. Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter. *The International Journal of Robotics Research*, 2017.

[44] Yajie Xing, Jingbo Wang, and Gang Zeng. Malleable 2.5d convolution: Learning receptive fields along the depth-axis for rgb-d scene parsing. 2020.

[45] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the International Conference on Computer Vision ICCV'20*, 2020. https://arxiv.org/abs/1909.00321.

[46] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision ECCV'18*, 2018. https://arxiv.org/abs/1804.01654.

[47] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, 2020.

[48] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems 32, Volume 2 of 20: 32nd Conference on Neural Information Processing Systems (NeurIPS 2019).Vancouver(CA).8-14 December 2019*, 2020.

[49] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *IEEE*, 2017.

[50] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. 2017.

[51] Guo et al. Deeplearning for 3d point clouds: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2020.

[52] Zhao et al. Point transforme. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[53] Wu et al. Pointatme: Efficient 3d point cloud labeling in virtual reality. In *In Proceedings of the 35th international conference on neural information processing systems*, 2022.

[54] Lai et al. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2022.

[55] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. *Proceedings of the IEEE International Conference on Computer Vision ICCV'21*, 2021. https://arxiv.org/abs/2105.01928.

[56] Ren et al. Neural volumetric object selection. In *Procceding of the Conference on Computer Vision and Pattern Recognition CVPR'22*, 2022. https://arxiv.org/abs/2205.14929.

[57] Vadim et al. Neural feature fusion fields:3d distillation of self-supervised 2d image representations. *International Conference on 3D Vision 3DV'22*, 2022. https://doi.org/10.48550/arXiv.2209.03494.

[58] Sosuke et al. Decomposing nerf for editing via feature field distillation. *The conference on Neural Information Processing Systems NeurIPS'22*, 2022. https://arxiv.org/abs/2205.15585.

[59] Kerr et al. Lerf:language embedded radiance fields. *arXiv e-prints*, 2023. https://arxiv.org/abs/2303.09553.

[60] Rahulet al. Interactive segmentation of radiance fields. *arXiv e-prints*, 2022. https://arxiv.org/abs/2212.13545.

[61] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, A. E. Saddik, C. Theobalt, Eric P. Xing, and Shijian Lu. 3d open-vocabulary segmentation with foundation models. *ArXiv*, abs/2305.14093, 2023.

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

[63] Cen et al. Segment anything in 3d with nerfs. *Preceedings of the conference on Neural Information Processing Systems NeurIPS'22*, 2022. https://arxiv.org/abs/2304.12308.

[64] Cen et al. Segment any 3d gaussians. *Preceedings of the conference on Neural Information Processing Systems NeurIPS'23*, 2023. https://arxiv.org/abs/2312.00860.

[65] Ying et al. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. *IEEE*, 2023.

[66] Jian Liu and Zhen Yu. Sa3d-l: A lightweight model for 3d object segmentation using neural radiance fields. *Neurocomputing*, 623:129420, 2025. https://doi.org/10.1016/j.neucom.2025.129420.

[67] Jean Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, and Mohammad Gheshlaghi Azar. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[68] Z. Wang, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux. Image quality assessment: From error visibility to structural similarity. 2013. https://ieeexplore.ieee.org/document/1284395.

JIAN LIU received the B.S. degree in chemical engineering from Nanjing Science and Technology University, in 1990, and the M.S. degree in computer science from California State University Long Beach, in 2003. He was a Software Engineer with Astrophysics Inc., one of the famous X-ray security inspection machine manufacturers in the USA, from 2005 to 2007. He came to the computer and artificial intelligence school, Zhengzhou University, in spring 2008. His research interests focus on artificial intelligence, virtual reality/augmented reality, and SLAM. He is also teaching courses in programming languages to undergraduate students.

ZHEN YU received the PhD. from the University of California, Irvine. In 2006 she became a Lead Nanofabrication Engineer at RF Nano Corporation. Since 2014, she has been a faculty member in the Department of Electrical Computer Engineering, California State Polytechnic University, Pomona, CA, USA. Her current research interests include nanotechnology featuring nanomaterial synthesis, low-cost and robust manufacturing processes, artificial intelligence, robotics technology, and application of nanotechnology to wireless communication and energy storage devices for UAVs.