Prompt SA-GS: Enhanced 3D Segmentation in Optimized Gaussian Splatting

Kai Zhu*
kaizhu@cs.toronto.edu

Lakshya Gupta*
lgupta@cs.toronto.edu

Anannya Popat*
anannyap20@cs.toronto.edu

Shailesh Nanisetty*
shailesh005@cs.toronto.edu

University of Toronto, Department of Computer Science

Abstract

3D Gaussian Splatting has rapidly evolved as a potent alternative to Neural Radiance Fields (NeRFs) for 3D reconstruction and novel-view synthesis, delivering results that are both faster and more effective. However, the potential for further enhancements in the modification of Gaussians and optimization remains vast, promising to broaden its practical applications. This paper seeks to build upon and practically implement the innovative strategies from the recent SA-GS paper in the realm of 3D segmentation. We integrated a LangSAM model to facilitate prompt-based segmentation, allowing the selection of desired objects. A major feature is our prompt initialization strategy of sampling points from text-prompted masks on a sparse but diverse set of initial views, chosen using k-means clustering. Additionally, we incorporated DUSt3R-based Gaussian initialization to improve scene fidelity. This approach marks an evolution from the single-view point selection method using SAM, as proposed by SA-GS, aiming to simplify and enhance 3D segmentation. Our experimental results achieved comparable or greater IoU and accuracy scores with respect to SA-GS, while using 50% fewer projected views for mask generation to reduce compute and memory requirements.

1. Introduction

Despite extensive research in 3D reconstruction, efforts to understand and modify 3D scenes, particularly through segmentation, remain underdeveloped. Current segmentation methods are well-explored for Bird-Eye View (BEV) [9, 17–19, 24] and range images [4, 13], yet they are notably lacking in complex 3D reconstruction models like NeRFs [15, 20], which do not inherently support segmentation or detection due to their continuous volumetric representation.

To address this gap, SA3D [2] represents a pioneering effort aimed at merging detailed scene reconstruction—NeRF's forte—with practical applications such as object segmentation. This is achieved by augmenting a 2D foundational model, like SAM (Segment-Anything Model) [8], with 3D perception capabilities, allowing SA3D to project segmentation from a single 2D view across all views used in the NeRF model, thus ensuring consistent segmentation throughout the 3D scene. However, despite these advancements, segmentation capabilities are still missing in more efficient and advanced 3D reconstruction algorithms like Gaussian splatting [7].

This paper proposes to adapt and modify the concepts pioneered by SA-GS [5], which are derived from SA3D, to segment or edit Gaussians within a scene. To improve SA-GS from a usability aspect, we implement a LangSAM [12] model to allow prompt-based segmentation of the desired object and random point initialization for a small number of views carefully selected using k-means clustering algorithm to ensure variation of position and rotation within the views. While LangSAM is itself capable of generating masks across multiple views based on text prompts, the cost in compute makes it infeasible to apply directly onto a dense image set. We use SA-GS in combination with LangSAM is to streamline the segmentation process and improve its efficiency. The segmentation mask is then generated across multiple views to segment the desired object effectively from the 3D space, as proposed by SA-GS.

Moreover, we aim to enhance the implementation of SA-GS by optimizing camera parameter and point-cloud initialization through the use of DUSt3R [22]. Traditionally, Structure from Motion (SfM) has been employed for initializing point clouds and camera parameters in NeRFs and 3D Gaussian Splatting. Despite its effectiveness, SfM is cumbersome and highly reliant on sequential steps that can propagate errors, requiring a large number of images to produce satisfactory results. DUSt3R addresses these issues by utilizing a sparse set of images to generate optimized point clouds and camera parameters through a transformer architecture and a straightforward optimization problem. This streamlined approach not only enhances efficiency but also reduces the need for adaptive density control in the training of 3D Gaussian splatting.

The optimized SA-GS offers potential for a broad range of applications, from collision detection to augmented/virtual reality and film production, due to its rapid editing and segmentation capabilities without the need for extensive parameter training. Additionally, this methodology alleviates the labor-intensive process of manual annotation across multiple frames, requiring only a few target points from the user in a single 2D view. This innovation sets the stage for extensive future research. To summarize, the objectives of our project are outlined as follows:

1. Eliminate functions and files related to COLMAP initialization of point-cloud and camera parameters in the original Gaussian splatting [7] implementation, replacing them with custom functions and files that utilize DUSt3R outputs.

2. Employ the 2D segmentation model LangSAM [12] to enable prompt-based object detection and initialization of random points within the segmented objects for 4 views selected through k-means clustering to ensure that the views are guaranteed to be from a good variety.

3. Apply concepts inspired by SA-GS [5] to automatically produce multi-view masks, achieving consistent 3D segmentation through the proposed view-wise label assignment.

4. Finally, we contrast the SA-GS with our newly modified methodology which involves using multiple views for generating initial masks and sampling of random points, instead of a single-view.

2. Related Work

2.1. 3D Gaussian Splatting

3D Gaussian Splatting [7] has emerged as an innovative technique for scene reconstruction, rivaling Neural Radiance Fields (NeRF) by offering high-quality rendering combined with faster processing speeds suitable for real-time applications. It has been increasingly explored in recent research, particularly in areas like dynamic scene processing and integration with advanced generative models. Notably, Dynamic 3D Gaussians [11] enhance Gaussian Splatting by enabling it to accommodate dynamic scenes through the tracking of objects represented as sets of 3D Gaussians. Moreover, recent developments such as DreamGaussian [21], GaussianDreamer [25], and GSGEN [3] have successfully integrated Gaussian splatting with diffusion models, paving the way for generating high-quality 3D assets. Building on these advancements, we introduce an optimized version of SA-GS [5] by adapting the 2D foundational model-SAM [8] to 3D Gaussian spaces along with utilizing DUSt3R [22] for point-cloud initilization, achieving effective 3D object segmentation.

2.2. DUSt3R

This paper [22] represents a pioneering effort in optimizing the initialization of point clouds and camera parameters in 3D Gaussian splatting, leveraging a streamlined transformer architecture alongside a simplified optimization problem. The use of DUSt3R, inspired by NeRFmm [23], significantly enhances the process by generating a dense point cloud from sparse views within the same number of iterations required by COLMAP. This advancement eliminates the need for adaptive density control in traditional Gaussian splatting, thereby accelerating the overall process. While NeRFmm simplified NeRF model training by integrating camera parameters as learnable elements within the 3D scene representation, it struggled with camera parameter estimation in scenarios involving track-to-object motion and was less effective in scenes where the camera orientation was largely unidirectional. To circumvent these limitations, our approach incorporates the techniques developed in DUSt3R for 3D point cloud initialization, offering a more robust and versatile solution compared to the method employed by NeRFmm.

2.3. LangSAM

LangSAM (Language-Segmentation Anywhere Model) [12] is a novel implementation of the vanilla SAM [8] model in computer vision that merges natural language processing with image segmentation, allowing users to command segmentation tasks through textual inputs. This model interprets complex descriptions provided by users, applying these directly to segment specific features or objects within images. Such capabilities broaden LangSAM's utility across various applications. In medical imaging, it can assist clinicians by segmenting anatomical features described in text, enhancing diagnostic accuracy and treatment planning. In autonomous vehicles, LangSAM has the potential to improve object detection by processing verbal descriptions, refining environmental awareness and response systems. Moreover, in the digital arts and gaming, it can aid creators to construct scenes dynamically by describing elements verbally, streamlining the creative process and reducing development time. LangSAM thus exemplifies the fusion of human communicative efficiency with advanced image processing technologies and easy interactions across multiple domains, which we aim to leverage.

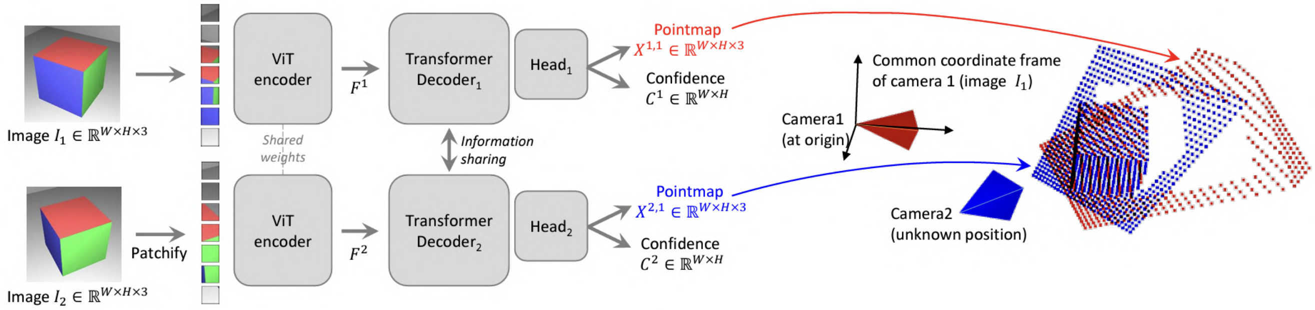
Image $I_1 \in \mathbb{R}^{W\times H\times 3}$
Image $I_2 \in \mathbb{R}^{W\times H\times 3}$
Patchify
ViT encoder
Shared weights
ViT encoder
F^1
F^2
Transformer Decoder$_1$
Transformer Decoder$_2$
Information sharing
Head$_1$
Head$_2$
Pointmap
$X^{1,1} \in \mathbb{R}^{W\times H\times 3}$
Confidence
$C^1 \in \mathbb{R}^{W\times H}$
Pointmap
$X^{2,1} \in \mathbb{R}^{W\times H\times 3}$
Confidence
$C^2 \in \mathbb{R}^{W\times H}$
Common coordinate frame of camera 1 (image I_1)
Camera1 (at origin)
Camera2 (unknown position)

Figure 1. Taken from DUSt3R [22]: Transformer architecture to generate relative pointmaps for each pair of image

2.4. Segmentation in NeRFs

Scene manipulation and segmentation in Neural Radiance Fields (NeRFs) have been focal points of recent advancements, aiming to extend the capabilities of NeRFs beyond mere scene reconstruction. Techniques such as those introduced in EditNeRF [10] and Semantic-NeRF [26] exemplify this shift. EditNeRF enables user-driven modifications of NeRF-generated scenes, allowing for real-time adjustments such as object removal and color changes. This flexibility opens up new possibilities for interactive applications. Meanwhile, SemNeRF focuses on semantic segmentation within NeRF-generated environments, facilitating the separation of different scene components based on their semantic categories. Expanding upon these innovations, SA3D [2] integrates 3D semantic capabilities into NeRF, enhancing the model's utility in practical applications such as VR, AR, and robotics. By augmenting a foundational 2D segmentation model to work within NeRF's 3D reconstructions, SA3D allows for dynamic interaction and precise segmentation of complex scenes.

These techniques typically necessitate alterations or retraining of the original NeRF models, or the training of additional specific parameters to achieve 3D segmentation. However, due to the limitations in representation and rendering speed of NeRF, applying these methods to practical applications poses a significant challenge. Thus, 3D segmentation in gaussian splatting would prove to be advantageous due to their obvious benefits in terms of rendering speed and efficiency compared to NeRFs. Additionally, due to the explicit representation of 3D gaussians, we can perform segmentation relatively easily without the need for any learnable parameters and training process, as proposed by SA-GS [5].

3. Theory

3.1. 3D Gaussian Splatting

3D Gaussian Splatting (3D GS) [7] is a developing technique for real-time rendering of radiance fields. It has demonstrated its effectiveness in creating new views with the same high quality as NeRF while achieving real-time speeds. This method models scenes using a collection of 3D Gaussians.Specifically, each 3D Gaussian is parameterized by a position ’$\mu \in \mathbb{R}^3$’, a covariance matrix ’Σ’, an opacity value ’α’ and spherical harmonics (SH).

The rendering process utilizes a splatting pipeline in which 3D Gaussians are projected onto a 2D image plane, transforming them into 2D Gaussians. These 2D Gaussians are then merged using the -blending algorithm to produce the final color.

$$c = \sum_{i\in\mathbb{N}} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{1}$$

In the -blending stage, only those 2D points within each 2D Gaussian that have a probability density exceeding a specific threshold are computed. Consequently, a 2D Gaussian can be intuitively viewed as a 2D ellipse, while a 3D Gaussian corresponds to a 3D ellipsoid.

In practice, the length of an axis of the ellipse is determined to be three times , where represents the square root of the variance along that axis.

3.2. DUSt3R

The DUSt3R [22] model is a novel approach for Dense Un- constrained Stereo 3D Reconstruction from un-calibrated and un-posed cameras. It uses a network that can regress a dense and accurate 3D representation from just a pair of images, without any prior knowledge about the scene or the cameras.

Generation of Relative Pointmaps
As illustrated in Fig. 1, the process begins by encoding two scene views (I^1, I^2) using a shared pre-trained ViT encoder, CroCo, in a Siamese configuration. The token representations obtained, F^1 and F^2, are subsequently processed by two transformer decoders that engage in continuous information exchange through cross-attention mecha-

nisms. At the end of the pipeline, two regression heads generate corresponding pointmaps $X^{1,1}$ and $X^{2,1}$ along with their associated confidence maps C^1 and C^2. Notably, both pointmaps are aligned within the coordinate frame of the first image, I^1. The network, denoted as F, is optimized using a straightforward regression loss (see Eq. (2) and Eq. (3)).

The regression loss for a valid pixel $i \in D^v$ in view $v \in (1, 2)$ is simply defined as the euclidean distance:

$$\ell_{\text{regr}}(v, i) = \left\| \frac{1}{Z} X_i^{v,1} - \frac{1}{\bar{Z}} X_i^{v,1} \right\| \tag{2}$$

To address the scale ambiguity between the predicted and the ground-truth data, both the predicted and the ground-truth pointmaps are normalized by scaling factors $z = norm(X^{1,1}, X^{2,1})$ and $\bar{z} = norm(\bar{X}^{1,1}, \bar{X}^{2,1})$, represent the average distance of all valid points to the origin.

In practice some 3D points are ill-defined, such as those in the sky or on translucent objects. Thus, certain areas within an image are typically more challenging to predict than others. Consequently, the transformer architecture has learned to predict a confidence score for each pixel, which reflects the network's reliability concerning that specific pixel. The final training objective incorporates this understanding through a confidence-weighted regression loss, as detailed in Eq. (3), applied across all valid pixels.

$$L_{\text{conf}} = \sum_{v \in \{1,2\}} \sum_{i \in D_v} C_i^{v,1} \ell_{\text{regr}}(v, i) - \alpha \log C_i^{v,1} \tag{3}$$

Here, $C_i^{v,1}$ is the confidence score for pixel i and α is the hyper-parameter controlling the regularization term. Additionally, to ensure a strictly positive confidence, we define $C_i^{v,1} = 1 + \exp\left(-C_i^{v,1}\right) > 1$.

Recovering Camera Parameters

The generated local point maps can be used to obtain the camera intrinsics using the Weiszfeld algorithm, as laid out in [22]:

$$f_1^* = \arg\min_{f_1} \sum_{i=0}^{W} \sum_{j=0}^{H} C_{i,j}^{1,1} \left\| \begin{pmatrix} i' \\ j' \end{pmatrix} - f_1 \begin{pmatrix} \frac{X_{i,j,0}^{1,1}}{X_{i,j,2}^{1,1}} \\ \frac{X_{i,j,1}^{1,1}}{X_{i,j,2}^{1,1}} \end{pmatrix} \right\| \tag{4}$$

where $i' = i - \frac{W}{2}$ and $j' = j - \frac{H}{2}$.. Here W,H is the width and height of the image respectively.

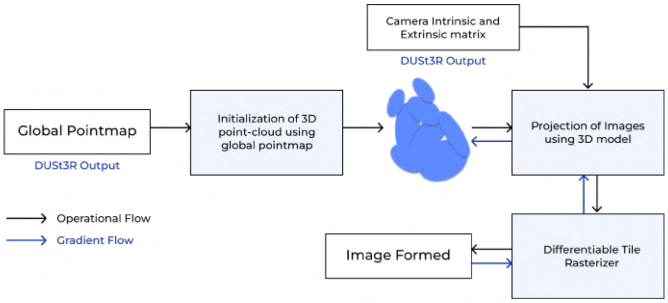
Camera Intrinsic and Extrinsic matrix
DUSt3R Output
Global Pointmap
DUSt3R Output
Initialization of 3D point-cloud using global pointmap
Projection of Images using 3D model
Operational Flow
Gradient Flow
Image Formed
Differentiable Tile Rasterizer

Figure 2. Modified Gaussian Splatting Architecture with DUSt3R Integrated

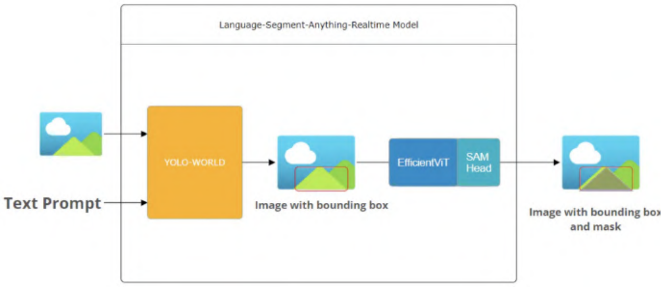
Language-Segment-Anything-Realtime Model
YOLO-WORLD
Image with bounding box
EfficientViT
SAM Head
Text Prompt
Image with bounding box and mask

Figure 3. Schematic Pipeline of the LangSAM model

Pair-wise to Globally-Aligned Poses

[22] also describes aligning the local pointmaps to globally aligned pointmaps by solving the following optimization problem:

$$\chi^* = \arg\min_{\chi, P, \sigma} \sum_{e \in E} \sum_{v \in e} \sum_{i=1}^{HW} C_i^{v,e} \left\| \chi_i^v - \sigma_e P_e \chi_i^{v,e} \right\| \tag{5}$$

Here, we write $v \in e$ for $v \in n, m$ if $e = (n, m)$. The idea is that, for a given pair e, the same rigid transformation P_e should align both point maps $X_{n,e}$ and $X_{m,e}$ with the world-coordinate point maps $\tilde{X} * n, e$ and $\tilde{X} * m, e$, since $X_{n,e}$ and $X_{m,e}$ are by definition both expressed in the same coordinate frame. To avoid the trivial optimum where $\sigma_e = 0$, for all $e \in E$, the constraint that the product of all σ_e equals one is imposed.

Recovering Camera Extrinsics

A straightforward extension to this framework enables to recover all cameras parameters. By simply replacing the calculated K_n and known depthmaps D_n in $P_n^{-1} \boldsymbol{h} \left(K_n^{-1} \begin{bmatrix} i \\ j \\ D_{i,j}^n \end{bmatrix} \cdot D_{i,j}^n \right)$, we can estimate the camera extrinsics/transformation matrix P_n for $n = 1 \ldots N$.

Fig. 2 gives an overview of the DUSt3R integrated Gaussian Splatting pipeline.

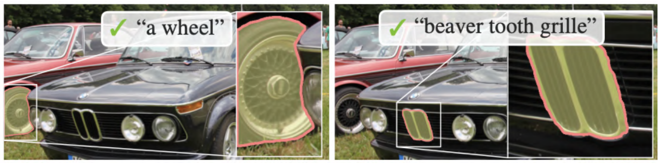

Figure 4. Qualitative results with simple and nuanced text prompts [8]

3.3. LangSAM

This model segments objects based on free-form text prompts. It harnesses the capabilities of instance segmentation combined with text prompts to create masks for specified objects within images. Built upon the Segment-Anything (SA) approach, it primarily utilizes the Grounding DINO detection model—an advanced object detection system designed for open-set environments, where it detects objects as defined by textual inputs. The model extends the DETR (DEtection TRansformer) framework, augmenting it with the ability to incorporate multi-level textual information through a process known as grounded pre-training. The three principal components of this model are the feature enhancer, language-guided query selection, and cross-modality decoder.

Initially, the LangSAM [12] model predicts the bounding boxes for the objects in the input image as described by the user's text prompts. Subsequently, it identifies the actual detected boxes (verified from the outputs) and loads a pre-trained SAM model, which could be any of the three variants: [VIT-H, VIT-L, VIT-B]. It then segments these boxes to generate masks on the input image. Therefore, this method is a zero-shot text-to-mask approach.

3.4. Problem Outline

Equipped with a set of calibrated 3D Gaussians obtained from the DUSt3R sampled-3DGS model, denoted by $\mathcal{G} = \{g_0, g_1, \ldots, g_n\}$ along with 4 initial viewpoints $\mathcal{V} = \{v_0, v_1, v_2, v_3\}$, users can input a text-based prompt for a single 2D view v_0—take ”truck” as an example. This prompt is translated across three additional views to generate segmented masks of the referenced object, ”truck.” These views are intentionally chosen using k-means clustering to ensure varied position and rotation of the desired object within the scene. Subsequently, we randomly sample 2D point prompts, denoted as $\mathbb{P}_{2D} = \{p_0, p_1, \ldots, p_m\}$ to locate the 2D representation of the object within the starting views $\mathcal{V} = \{v_0, v_1, v_2, v_3\}$. Our designed algorithm, inspired by SA-GS [5] is tasked with isolating the 3D object $\mathcal{O}$ within $\mathcal{G}$, which corresponds to the object identified by the text-based user prompt. Here, $\mathcal{O}$ represents a subset within the array of 3D Gaussians $\mathcal{G}$.

Following the methodology advocated by SA-GS, we assign m_i to denote the binary mask projection of the object $\mathcal{O}$ from the i-th viewpoint. An accurate segmentation of $\mathcal{O}$ is indicated by the congruence of m_i with the ground truth mask m_i^* across all viewpoints indexed by i in the set $\{0, 1, \ldots, n\}$. The ground truth mask m_i^* functions as the definitive segmentation template for $\mathcal{O}$ in the i-th viewpoint. Unlike the pre-defined truths common in traditional 2D image segmentation or 3D point cloud segmentation tasks, no such baselines exist for the segmentation of 3D Gaussians. Hence, our algorithm is tailored to minimize the divergence between the projected mask m_i and the ground truth mask m_i^*.

3.5. Utilizing 2D masks to Segment 3D Gaussians

3D prompts for Multiview Masks Generation As described in section 3.4, users start with an initial text prompt to segment the desired object for a few views. However, these sparse perspectives with 2D prompt points falls short for accurately delineating the object within its three-dimensional space. To remedy this, we generate masks from multiple viewpoints, serving as a foundational aid for 3D segmentation. These multi-view masks facilitate the object's isolation in three dimensions through the intersection of their respective view frustums. The critical phase of obtaining these masks hinges on the creation of 2D prompt points across varied viewpoints. Denoting the i-th 2D prompt point in the first given view v_0 as p_i^0, we define the corresponding 3D prompt p_i^{3D} as:

$$\arg\min_{\mu} \left\{ d(\mu), d(\mu) > 0 \mid \mu \in \mathcal{G}, \left\| P_0\mu - p_i^0 \right\|_1 < \varepsilon \right\} \tag{6}$$

Here, $d(\mu)$ signifies the depth at the center of Gaussian μ, and the product $P_0\mu$ computes the position of μ in the initial viewpoint v_0. According to Eq. 6, the 3D prompt corresponding to p_i is determined as the center of a certain 3D Gaussian. This center must meet two conditions: its projection should be proximate to p_i within a tolerance defined by ε in Manhattan distance, and among multiple qualifying Gaussian centers, the one with the least positive depth is selected. Applying Eq. 6 to all the 2D prompts of the remaining initial views v_1, v_2, v_3, yields a set of corresponding 3D prompts. For a view not in the set of initial views, v_i, these 3D prompts are projected onto the 2D plane, generating 2D prompts for that view. This approach aggregates 2D prompts across all views, with all resulting masks derived through the SAM process.

3.6. Label Assignment (View-wise)

With the entirety of the masks at hand, our next step involves giving binary labels upon each 3D Gaussian. For this task, we utilize a matrix denoted as L, which comprises elements L_{ij}. These elements are specified as follows in Eq. 7:

$$L_{ij} = \begin{cases} 1 & \text{if } P_j \mu_i \in m_j, \\ 0 & \text{if } P_j \mu_i \notin m_j, \end{cases} \tag{7}$$

Here, μ_i represents the center of the i-th Gaussian within the scene, while m_j is the foreground mask for the j-th view. The term P_j refers to the projection matrix corresponding to the j-th view.

3.7. Label Voting (Multi-View Setting)

To this point, each 3D Gaussian possesses an array of binary labels L_i, assigned as described earlier. Utilizing these assigned labels, we now employ a straightforward but efficient heuristic to decide whether a 3D Gaussian g_i is part of the intended 3D object. Specifically, we initially establish the confidence score s_i for g_i as follows (Eq. 8):

$$s_i = \frac{1}{N} \sum_{j=0}^{N-1} L_{ij}, \tag{8}$$

Here, N represents the total count of views. Subsequently, we employ a threshold τ and consider a Gaussian to be positive if its score surpasses τ.

4. Proposed Method

As depicted in Figure 5, our methodology starts with a collection of sparse views of a specific object within a scene. These views are strategically selected using a k-means clustering algorithm to ensure a comprehensive assortment of angles. For example, the views showcased in the referenced figure are grouped into four (n=4) clusters based on their positions and orientations. Each cluster's representative view is chosen based on its minimal distance to the cluster's centroid, ensuring maximum diversity and enhancing the accuracy of the eventual 3D segmented output.

The objects of interest within these sparse views are segmented using a user-provided text-based prompt through LangSAM, which simplifies the mask generation process, making it more accessible and user-oriented. A concise explanation of LangSAM's workflow is presented in Section 3.3. Subsequently, a selection of 2D prompt points is randomly sampled for each initial view.

For each selected 2D prompt point, a corresponding 3D gaussian is chosen based on Eq.(6), where the gaussian least far away from the camera is picked if multiple gaussian candidates pass the distance-based filter. Thereafter, these 3D prompts are re-projected into all 2D views using the projection matrix P_j for each view. The subsequent 2D prompts, derived from this re-projection, serve as the basis for mask generation through SAM for each view, as detailed in Section 3.5. This process is followed by view-wise label assignment, as described in Section 3.6, where binary labels are allocated to each of the 3D Gaussians. Moreover, a multi-view label voting mechanism ensures that the majority of the views ($> 60\%$) corroborate the positioning of the prompts within the object mask, as elucidated in Section 3.7.

In parallel, the Dust3R model, known for producing globally aligned point maps and camera intrinsic and extrinsic matrices, is integrated into the 3D Gaussian Splatting model. This integration is preferred over the use of the more complex COLMAP. The global point map substitutes for the 3D point cloud, and the matrices are utilized as the model parameters, streamlining the process in comparison to the conventional Structure from Motion (SfM) technique applied in the original 3D Gaussian Splatting.

Ultimately, we harness these multiple masks, obtained from different views, to segment a 3D object by intersecting the corresponding frustums through SAM, effectively delineating the target object in three dimensions.

5. Experiments

5.1. Datasets

To test our method across a variety of scenes, we use image sets from the LLFF [14], Mip-NeRF 360 [1], and 3D Gaussian Splatting [6] datasets. The specific selection of scenes from each dataset is based on the availability of corresponding segmented ground truth masks in the SPIn-NeRF dataset [16], which are used to evaluate the output of our system.

5.2. Setup

We base our experimental parameters based on results reported by previous work [5] and our own preliminary testing. We find that the previously cited label voting threshold of 0.6 produce good results. Additionally, for multi-view initial prompt generation, we empirically determine that 2 randomly sampled points from 4 initial views provide consistently effective prompting for mask generation and segmentation.

5.3. Analysis

We use IoU and accuracy scores to quantify our segmentation results using the SPIn-NeRF segmentation dataset as ground truth. Furthermore, quantitative and qualitative comparisons are performed for different rates (10%, 50%, and 100%) of view sampling for mask generation. We also perform an ablation study on the effect of initial view selection for prompting: sampling points from a single view (similar to SA-GS [5]), randomly-sampled multi-view, and k-means clustered multi-view (our method).

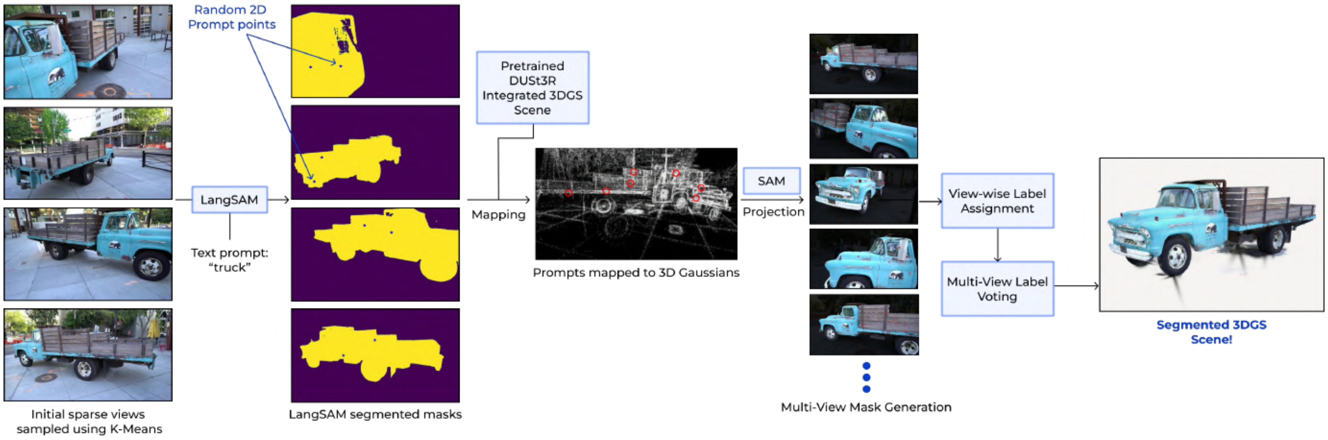
Random 2D Prompt points
Initial sparse views sampled using K-Means
LangSAM
Text prompt: "truck"
LangSAM segmented masks
Pretrained DUSt3R Integrated 3DGS Scene
Mapping
Prompts mapped to 3D Gaussians
SAM
Projection
Multi-View Mask Generation
View-wise Label Assignment
Multi-View Label Voting
Segmented 3DGS Scene!

Figure 5. Our proposed method: a few views were selected via k-means clustering to generate initial masks using LangSam based on a text prompt. Initial 2D prompt points are randomly sampled from the masked areas on each initial view and mapped to 3D Gaussians in the pre-trained scene. The resulting 3D prompts are projected into all views. Each set of projected prompts are used to generate masks through SAM for each view. Gaussians are labelled based on the multi-view masks and segmented through majority label-voting.

	Single View	Random Multiview	K-Means Multiview (Ours)
IoU	90.2	90.8	93.4
Acc	96.6	96.8	97.7

Table 1. Ablation on initial view sampling strategy. Single view is the strategy previously employed by 3DSA. Our method use k-means clustering to maximize the positional and angular variation of multiple views for initial sampling, yielding improved metrics over random multiview and single view initializations.

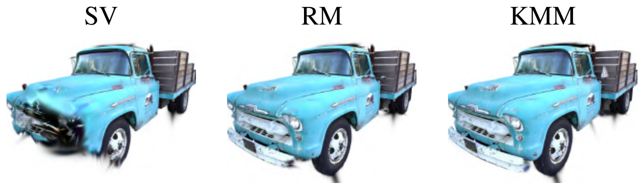

Figure 6. Comparative Segmentation Analysis using Different View Selection Strategies for Initial Prompt Sampling: (a) single view sampling (SV) (b) random multi-view sampling (RM) (c) K-means clustering-based multi-view sampling (KMM) based on views' rotation and position to pick most diverse views

5.4. Results

5.4.1 Initial View Selection

The adoption of a multi-view framework for initial point selection significantly improves the segmentation quality. This is illustrated in Figure 6, which shows the limitations of single-view approaches, notably the distorted front of the truck as a result of the method's high sensitivity toward the choice of initial view. As we transition to multi-view methods, these distortions are progressively rectified. The randomly selected multi-view sampling provides some improvements, but the K-Means method, which selects views based on diverse camera positions and orientations, captures object features more effectively. This approach not only refines feature delineation, such as on the truck's bumper, but also reduces occlusion risks, delivering sharper contrasts compared to the random sampling. The strategic diversity in view selection crucially enhances the overall segmentation quality.

5.4.2 Multi-view Sampling Rate

	Views		
	10% (25)	50% (126)	100% (251)
IoU	80.1	93.4	93.9
Acc	91.1	97.7	97.9

Table 2. Experiment on adjusting the number of projected views to generate masks on for the truck scene. Numbers within parentheses represent the count of sampled views. Our 50% view IoU (93.4) exceeded the previous SA-GS 50% view score of 92.1

Figure 7 demonstrates the impact of using varying percentages of available views on segmentation outcomes. Employing just 10% of views in majority voting leads to substantial over-segmentation with incorrect markings of the ground as part of the target object due to occlusions from limited perspectives. Increasing the view percentage mitigates this, with progressive improvements evident in 50% and optimal results in 100% columns, where all views are utilized. Although using 50% of the views achieves near-optimal results, enhancing efficiency without significantly

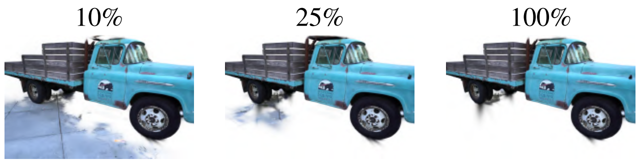

Figure 7. Effect of Multiview Sampling Rate on Segmentation Accuracy: (a) Segmentation with 10% of views, showing over-segmentation and misplacement on the ground. (b) Improved delineation using 50% of views, balancing efficiency and quality. (c) Optimal segmentation from employing 100% of the views, demonstrating the highest precision.

compromising segmentation quality, utilizing 100% of the views remains ideal for achieving the most precise segmentation, especially when computational resources are not a constraint.

5.4.3 Comparison with SA-GS

Scene	Text Prompt
Truck	”truck”
Orchid	”all of the flowers”
Fortress	”fortress”
Pinecone	”pine cone”

Table 3. Text prompts used in each segmented scene. Note the use of quantitative specifier in the orchid scene to select multiple objects.

	Ours (50% views)		SA-GS (100% views)	
	IoU	Acc	IoU	Acc
Truck	93.4	97.7	93.4	97.8
Orchid	76.0	95.7	82.2	96.8
Fortress	95.2	99.2	88.5	98.1
Pinecone	91.0	98.8	91.6	98.5
Mean				

Table 4. Comparison of scene-specific segmentation statistics between our proposed method and SA-GS, using 50% of available views compared to the latter’s 100%. Cited SA-GS statistics are based on the results without Gaussian decomposition [5]

Figure 8 illustrates the outcomes of our text-prompted segmentation using the K-Means Multiview (KMM) initialization approach, revealing several key observations. In the orchid scene, the segmented gaussians miss one flower and a connecting stem from the ground truth, reflecting limitations in the ’all flowers’ prompt interpretation by LangSAM, which struggled with specificity in a complex scene with multiple stems and flowers. Finally, in the fortress scene, employing our KMM approach with a 50% view sampling rate yielded significantly improved IoU and accuracy metrics than the SA-GS method.

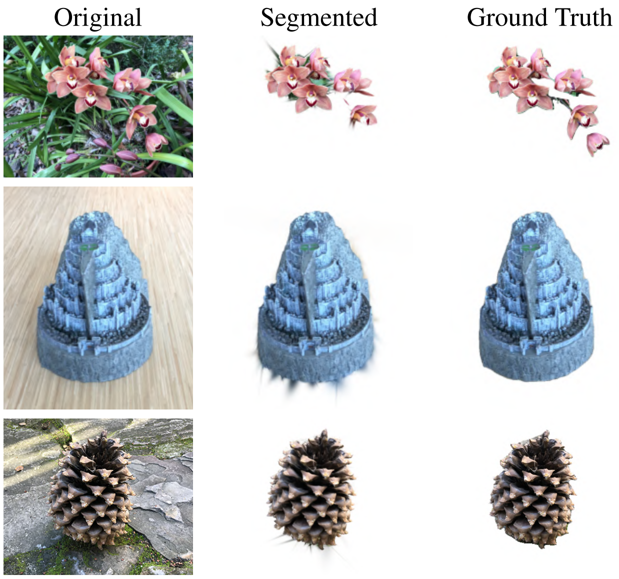

Figure 8. Text-prompted 3D segmentation results on the orchid, fortress, and pinecone datasets

6. Conclusion

We proposed a text-prompted segmentation method for 3D Gaussian splatting. By implementing components from the SA-GS paper [5], we identified the strengths and shortcomings of the previous 2D point-prompted technique. Our novel multiview initial sampling technique produced segmented results with metrics exceeding those presented by SA-GS using 50% fewer projected views, thereby halving the time and memory cost. Our method also demonstrated effective segmentation of single and multiple objects across a variety of scenes. Potential improvements to this method for future work include transmittance-weighed label voting to reduce the inclusion of occluded areas, boundary refinement through Gaussian decomposition [5], and more intelligent 2D prompt point sampling especially in cases where multiple objects are to be segmented.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 6

[2] Jiazhong Cen, Jiemin Fang, Zanwei Zhou, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment anything in 3d with radiance fields, 2024. 1, 3

[3] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting, 2024. 2

[4] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet:in defense of range view for lidar-based 3d object detection, 2021. 1

[5] Xu Hu, Yuxi Wang, Lue Fan, Junsong Fan, Junran Peng, Zhen Lei, Qing Li, and Zhaoxiang Zhang. Segment anything in 3d gaussians, 2024. 1, 2, 3, 5, 6, 8

[6] 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. 6

[7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. 1, 2, 3

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 1, 2, 5

[9] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers, 2022. 1

[10] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields, 2021. 3

[11] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis, 2023. 2

[12] et.al; Medeiros, Luca. Language segment-anything model. https://github.com/luca-medeiros/lang-segment-anything, 2023. 1, 2, 5

[13] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving, 2019. 1

[14] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 6

[15] 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, 2020. 1

[16] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 6

[17] Mong H. Ng, Kaahan Radia, Jianfei Chen, Dequan Wang, Ionel Gog, and Joseph E. Gonzalez. Bev-seg: Bird's eye view semantic segmentation using geometry and semantic point cloud, 2020. 1

[18] Cong Pan, Yonghao He, Junran Peng, Qian Zhang, Wei Sui, and Zhaoxiang Zhang. Baeformer: Bi-directional and early interaction transformers for bird's eye view semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9590–9599, June 2023. 1

[19] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs, 2022. 1

[20] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction, 2022. 1

[21] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation, 2024. 2

[22] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2023. 1, 2, 3, 4

[23] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters, 2022. 2

[24] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, Jie Zhou, and Jifeng Dai. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision, 2022. 1

[25] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models, 2023. 2

[26] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 3