Gaga: Group Any Gaussians via 3D-aware Memory Bank

Weijie Lyu[1], Xueting Li[2], Abhijit Kundu[3],
Yi-Hsuan Tsai[3], and Ming-Hsuan Yang[1,3]

[1] University of California, Merced
[2] Nvidia
[3] Google

https://www.gaga.gallery

Fig. 1: *Gaga* groups any Gaussians in an open-world 3D scene and renders multi-view consistent segmentation (pixels of the same 3D region across views are represented with the same color). By employing a 3D-aware memory bank, we eliminate the label inconsistency that exists in 2D segmentation and assign each mask across different views a universal group ID. This enables the process of lifting 2D segmentation to a consistent 3D segmentation. *Gaga* produces accurate 3D object instance segmentation, achieving high-quality results for downstream applications such as scene manipulation (*e.g.* changing the color of the cushion on the footstool to maroon).

Abstract. We introduce *Gaga*, a framework that reconstructs and segments open-world 3D scenes by leveraging inconsistent 2D masks predicted by zero-shot segmentation models. Contrasted to prior 3D scene segmentation approaches that heavily rely on video object tracking, *Gaga* utilizes spatial information and effectively associates object masks across diverse camera poses. By eliminating the assumption of continuous view changes in training images, *Gaga* demonstrates robustness to variations in camera poses, particularly beneficial for sparsely sampled images, ensuring precise mask label consistency. Furthermore, *Gaga* accommodates 2D segmentation masks from diverse sources and demonstrates robust

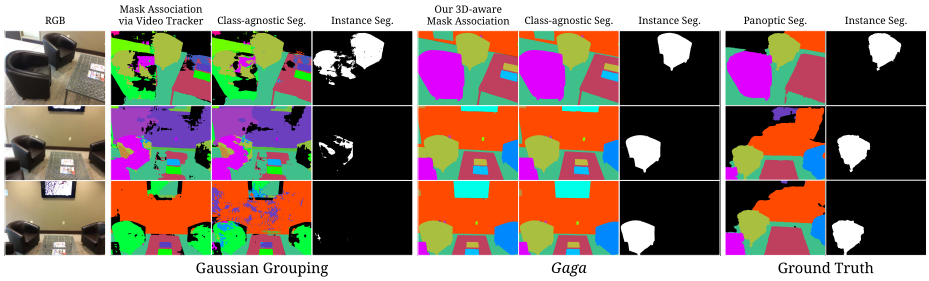

Fig. 2: Comparison of rendered segmentation. Previous methods [9, 26] adopt an off-the-shelf video object tracker for mask association. Results on the ScanNet dataset [8] show that they frequently misidentify objects, especially when similar objects are present in the scene (*e.g.* the leather sofas), and struggle to handle significant changes in camera perspective. In contrast, *Gaga* integrates 3D information to precisely locate objects and associate 2D masks, leading to multi-view consistent class-agnostic segmentation and precise instance segmentation rendering. We adopt the ground truth panoptic segmentation of the ScanNet dataset for comparison as it is visually the same as class-agnostic segmentation.

performance with different open-world zero-shot segmentation models, significantly enhancing its versatility. Extensive qualitative and quantitative evaluations demonstrate that *Gaga* performs favorably against state-of-the-art methods, emphasizing its potential for real-world applications such as scene understanding and manipulation.

Keywords: 3D Open-world Segmentation · Gaussian Splatting · Scene Understanding

1 Introduction

Effective open-world 3D segmentation is essential for scene understanding and manipulation. Despite notable advancements in 2D segmentation techniques, exemplified by Segment Anything (SAM) [14] and EntitySeg [21], extending these methodologies to the realm of 3D encounters the challenge of ensuring consistent mask label assignment across multi-view images. Specifically, masks of the same object across different views may have different mask label IDs, as the multi-view images are processed by the 2D segmentation model individually. Naively lifting these inconsistent 2D masks to 3D introduces ambiguity and leads to inferior results in 3D scene segmentation. Hence, we argue that it is crucial to assign each mask a multi-view consistent universal mask ID before lifting them to 3D. We refer to this task as mask association.

Prior research efforts [9, 26] built upon the recent advance in 3D reconstruction, Gaussian Splatting [11], attempt to solve this task by treating multi-view image datasets as video sequences and adopting an off-the-shelf video object tracking method [6]. Nevertheless, this design relies on the assumption of minimal view changes between multi-view images, a condition that may not consis-

tently hold in real-world 3D scenes. Consequently, these approaches struggle with similar objects or occluded objects that intermittently disappear and reappear in the sequence, as shown in Fig. 2.

As such, we analyze the fundamental disparity between the 3D mask association and video object tracking tasks: the utilization of inherent 3D information. Specifically, masks of the same object across different views shall correspond to the same group of 3D Gaussians. Hence, we can assign two masks from different views with the same universal mask ID if there is a large overlap between the two groups of 3D Gaussians that are splatted to them.

Based on this intuition, we propose *Gaga*, which groups any 3D Gaussians and renders consistent 3D segmentation across different views. Given a collection of posed RGB images, we first employ Gaussian Splatting to reconstruct a 3D scene and extract 2D masks using an open-world segmentation method. Subsequently, we iteratively build a 3D-aware memory bank that collects and stores Gaussians grouped by category. Specifically, for each input view, we project each 2D mask into 3D space using camera parameters and search the memory bank for the category with the largest overlap with the deprojected mask. Depending on the degree of overlap, we either assign the mask to an existing category or create a new one. Finally, following the mask association process described above, we leverage the consistent 2D masks to learn a feature on each Gaussian for rendering segmentation.

Our approach, *Gaga*, is capable of 1) synthesizing novel view images of RGB and segmentation with inherent 3D consistency; 2) grouping 3D Gaussians based on their 2D segmentation masks and providing accurate 3D instance segmentation for scene manipulation; 3) accommodating any 2D segmentation methods without additional mask pre-processing. Our contributions are summarized as follows:

- We propose a framework that reconstructs and segments 3D scenes using inconsistent 2D masks generated by open-world segmentation models.
- To resolve the inconsistency of 2D masks across views, we design a 3D-aware memory bank that collects Gaussians of the same semantic group. This memory bank is then employed to align 2D masks across diverse views.
- We show that the proposed method can effectively leverage any 2D segmentation masks, making it easily applicable for synthesizing novel view images and segmentation masks.
- We conduct comprehensive experiments on diverse datasets and challenging scenarios, including sparse input views, to demonstrate the effectiveness of the proposed method both qualitatively and quantitatively.

2 Related Work

Segment and Track Anything in 2D. Segment Anything (SAM) [14] and EntitySeg [21] demonstrate the effectiveness of large-scale training in image segmentation, thus establishing a pivotal foundation for open-world segmentation methods. Subsequent studies [6, 7, 25] further extend the applicability of SAM

to video data by leveraging video object segmentation algorithms to propagate the masks generated by SAM. Conversely, acquiring data for training their 3D counterparts poses a challenge, given that existing 3D datasets with annotated segmentation [8, 23] primarily focus on indoor scenarios.

NeRF-based 3D Segmentation. Neural Radiance Fields (NeRFs) [19] model scenes as continuous volumetric functions, learned through neural networks that map 3D coordinates to scene radiance. This approach facilitates the capture of intricate geometric details and the generation of photorealistic renderings, offers novel view synthesis capabilities.

Semantic-NeRF [27] initiates the incorporation of semantic information into NeRFs and enables the generation of semantic masks for novel views. Note that lifting semantic segmentation masks to 3D does not face the challenge of ambiguous mask IDs across views. Building upon it, subsequent researches expand the scope by introducing instance modeling and matching instance masks relying on 3D bounding boxes [10, 18], solving cost-based linear assignment during training [22, 24] or directly training instance-specific MLPs [15]. However, most of these methods are developed based on ground truth segmentation and tailored for scene modeling within specific domains. They often entail high computational costs and lack substantial evidence of their performance in open-world scenarios.

Leveraging SAM's open-world segmentation capability, SA3D [3] endeavors to recover a 3D consistent mask by tracing 2D masks across adjacent views with user guidance. Similarly, Chen et al. [5] distill SAM encoder features into 3D and query the decoder. In contrast, *Gaga* achieves multi-view consistency without user intervention, and offers segmentation for all objects rather than an instance. Concurrent work GARField [13] densely samples SAM masks and trains a scale-conditioned affinity field supervised on the scale of each mask deprojected to 3D. Whereas *Gaga* does not require the preprocessing process to obtain densely sample masks of different scales. Meanwhile, [13] groups Gaussians as clusters and multiple 3D instances may have the same group ID, which differs from the 3D segmentation task.

Gaussian-based 3D Segmentation. As an alternative to NeRF and its variants [1, 4, 20], Gaussian Splatting [11] has recently emerged as a powerful approach to reconstruct 3D scenes via real-time radiance field rendering. By representing the scene as 3D Gaussians, Gaussian Splatting achieves photorealistic novel view synthesis with high reconstruction quality and efficiency. Additionally, manipulating 3D Gaussians for scene editing is more straightforward than NeRF's representation.

SAGA [2] renders a 2D SAM feature map and uses a SAM guidance loss to learn 3D segmentation from the ambiguous 2D masks. Similar to [3], this method requires user input and only provides segmentation for one instance at a time. Feature 3DGS [28] distills LSeg [16] and SAM features to 3D Gaussians and decodes rendered features to obtain segmentation. However, it fails to provide consistent segmentation across views. Gaussian Grouping [26] and CoSSegGaussians [9] use 2D segmentation masks as pseudo labels and use a video object tracker [6] to associate masks across different views before lifting them to 3D.

However, in scenarios with significant changes in camera poses between frames, such approaches struggle to maintain accuracy.

3 Proposed Method

3.1 Preliminaries

Gaussian Splatting. Recently, Gaussian Splatting [11] has significantly advanced the 3D representation field by combining the benefits of implicit and explicit 3D representations. Specifically, a 3D scene is parameterized as a set of 3D Gaussians $\{G_i\}$. Each Gaussian $G_i = \{p_i, s_i, q_i, \alpha_i, c_i\}$ is defined by its position $p_i = \{x, y, z\} \in \mathbb{R}^3$, scale $s_i \in \mathbb{R}^3$, orientation $q \in \mathbb{R}^4$, opacity $\alpha \in \mathbb{R}$ and color features c encoded by spherical harmonics (SH) coefficients.

For image rendering, Gaussian Splatting employs the splatting rendering pipeline, wherein 3D Gaussians are projected onto the 2D image space using the world-to-frame transformation matrix corresponding to each camera pose. Gaussians projected to the same coordinates (x, y) are blended in depth order and weighted by their opacities to produce the color $c_{x,y}$ of each pixel:

$$c_{x,y} = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \tag{1}$$

Identity Encoding. Identity encoding [26] aims to assign a universal label to each 3D Gaussian for segmentation rendering. It is a 16-dimensional feature attached to each Gaussian, which is subsequently decoded to a segmentation mask ID $m_{x,y}$ for each pixel (x, y) through a classifier L, *i.e.* a combination of linear and SoftMax layers:

$$m_{x,y} = \arg\max\{L(\sum_{i \in N} e_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j))\}. \tag{2}$$

The resulting mask IDs are supervised using 2D segmentation masks.

3.2 Group Any Gaussians via 3D-aware Memory Bank

Given a set of posed images, we aim to reconstruct a 3D scene with semantic labels for segmentation rendering. To this end, we first leverage Gaussian Splatting for scene reconstruction. We then employ open-world 2D segmentation methods such as SAM [14] or EntitySeg [21] to predict class-agnostic segmentation for each input image. However, because the segmentation model processes each input image independently, the resulting masks are not naturally multi-view consistent. To resolve this issue, [9,26] assume that nearby input views are similar and apply a video tracker to associate inconsistent 2D masks of different views. Yet, this assumption may not hold for all 3D scenes, especially when the input views are sparse, as demonstrated in Fig. 2.

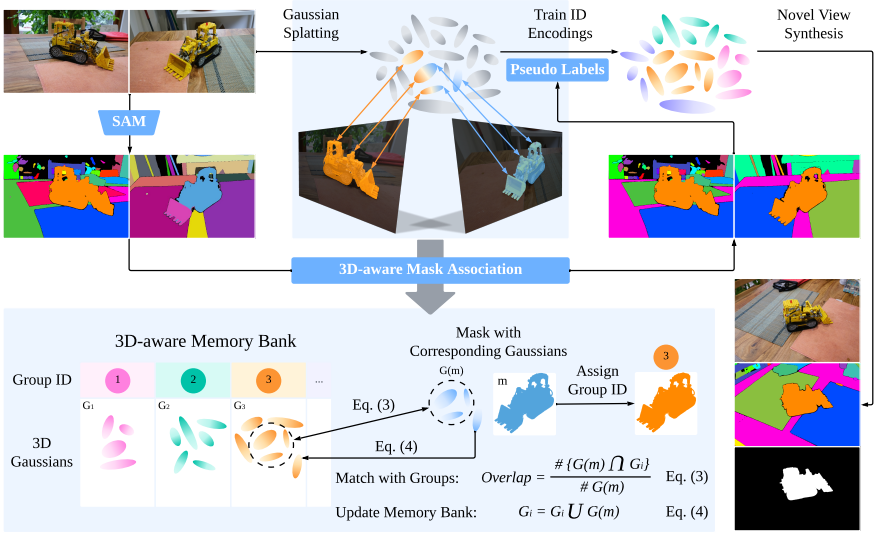

Fig. 3: Overview of *Gaga*. *Gaga* reconstructs 3D scenes with Gaussian Splatting and adopts an open-world model to generate 2D segmentation masks. To eliminate the 2D mask label inconsistency, we design the 3D-aware mask association process, where a 3D-aware memory bank is employed to assign a consistent group ID across different views to each 2D mask based on the 3D Gaussians projected to that mask (Sec. 3.2). Specifically, we find the corresponding Gaussians projected to each 2D mask and assign the mask with the group ID in the memory bank with the maximum overlapped Gaussians (Eq. 3). After the 3D-aware mask association process, we use masks with multi-view consistent group IDs as pseudo labels to train an identity encoding on each 3D Gaussian for segmentation rendering.

Gaga is inspired by the fundamental disparity between the task of mask association across multiple views and tracking objects in a video: the incorporation of 3D information. To reliably generate consistent masks across different views, we propose a method that leverages 3D information without relying on any assumptions about the input images. Our key insight is that masks belonging to the same instance in different views shall correspond to the same Gaussian group in the 3D space. Consequently, these Gaussians should be grouped together and assigned an identical group ID.

Corresponding Gaussians of a Mask. Based on this intuition, we first associate each 2D segmentation mask with its corresponding 3D Gaussians. Specifically, we splat all 3D Gaussians onto the camera frame given the camera pose of each input image. Subsequently, for each mask within the image, we identify which 3D Gaussians are projected within that mask. Those Gaussians should be identified as representatives of the mask in 3D and can be used as guidance for us to associate masks from different views.

Notably, segmentation masks typically describe the shape of foreground objects under the current camera pose. However, as Fig. 4 (a) shows, a significant

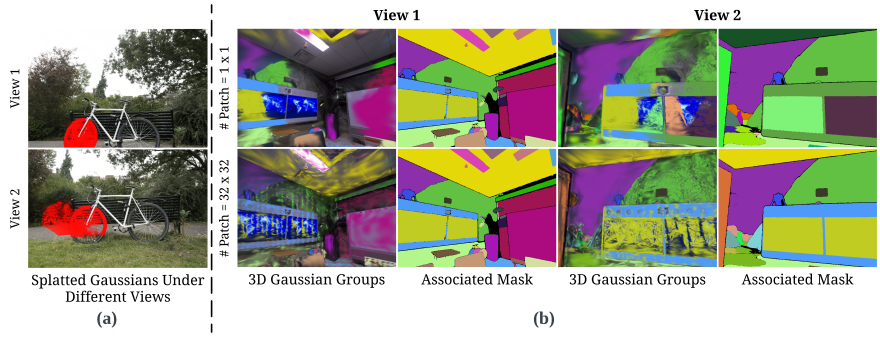

Fig. 4: Illustrations of finding corresponding Gaussians. (a) Motivation to choose front $x\%$ of Gaussians. We select $x\%$ of Gaussians which are closest to the camera frame because many Gaussians splatted to mask in view 1 represent objects from behind, as shown in view 2. **(b) The significance of mask partition.** We color the Gaussians in the 3D-aware memory bank based on their groups, displayed as "3D Gaussian Groups" in columns 1, and 3. When images aren't partitioned (row 1), the front $x\%$ of Gaussians concentrate in a confined area, failing to accurately represent the mask's shape, resulting in mismatched masks (columns 2, 4).

proportion of Gaussians do not contribute to the pixels in the 2D segmentation mask, as they represent objects situated behind. To address this, we select the front $x\%$ percentage of 3D Gaussians that are closest to the camera frame as the corresponding Gaussians of the mask. Here, x serves as a hyperparameter that can be adjusted based on the nature of the current 3D scene. As shown in Fig. 4 (b) row 1, selecting corresponding Gaussians based on the entire mask will inaccurately represent the shape of the mask for masks of large regions, and fail to associate masks across different camera poses. To resolve this issue, we propose a strategy wherein we partition an input image into patches and calculate corresponding Gaussians within each patch. Specifically, we begin by dividing the image into 32×32 patches. Subsequently, we identify the collection of top $x\%$ percentage of 3D Gaussians that are closest to the camera frame within each patch to be the corresponding Gaussians of mask m, denoted as $\mathcal{G}(m)$. As demonstrated in Fig. 4 row 2, this simple strategy effectively improves the consistency of associated masks across different views.

3D-aware Memory Bank. Next, to collect and categorize 3D Gaussians into groups and use them to associate masks across different views, we introduce a 3D-aware Memory Bank (see Fig. 3). Given a set of images, we initialize the 3D-aware Memory Bank by storing the corresponding Gaussians of each mask in the first image into an individual group and label the mask with a group ID the same as its mask label. For each 2D mask of the subsequent image, we first determine its corresponding Gaussians as outlined above. We then either assign these Gaussians to an existing group within the memory bank or establish a new one if they do not share similarities with existing groups in the memory bank. The details of this assignment process are elaborated in the following.

Group ID Assignment via Gaussian Overlap. To assign each mask a group ID, we aim to find if the current mask has a significant similarity with any group in the memory bank. Here, we define the similarity between two sets of 3D Gaussians based on their shared Gaussians ratio. Specifically, given the 3D Gaussians corresponding to a 2D mask m (denoted as $\mathcal{G}(m)$ as described above) and the Gaussians of group i (denoted as $\mathcal{G}_i$) in the memory bank, we identity their shared Gaussians as $\mathcal{G}(m) \cap \mathcal{G}_i$ (*i.e.* Gaussians of the same indices), we then compute the overlap as the ratio of the number of shared Gaussians to the number of all corresponding Gaussians of mask m:

$$Overlap(m, i) = \frac{\#(\mathcal{G}(m) \cap \mathcal{G}_i)}{\#\mathcal{G}(m)}. \tag{3}$$

If group i has the highest overlap with mask m among all groups in the memory bank, and this overlap value is above a threshold, we assign the group ID of mask m as i and add the non-overlapped Gaussians in the i_{th} group.

$$\mathcal{G}_i = \mathcal{G}_i \cup \mathcal{G}(m). \tag{4}$$

We establish a new group ID j if none of the existing groups contains an overlap with mask m above the overlap threshold. We add $\mathcal{G}(m)$ into this new group in the 3D-aware memory bank and assign mask m with the new group ID j. Note that we ensure each Gaussian will only be added to one group in the memory bank by recording the indices of all Gaussians that already exist in the memory bank.

3.3 3D Segmentation Rendering and Downstream Application.

After the group ID assignment process, masks projected by the same group of Gaussians are supposed to have the same group ID across different views. Similar to [26], we use those associated masks as pseudo labels and lift them to 3D by training the identity encoding. As we already obtain pre-trained Gaussians, we fix the other properties (e.g., location, opacity, etc.) of Gaussians for efficiency.

Our segmentation-aware 3D Gaussians can be readily used for various downstream applications. For instance, we can render segmentation masks of novel views that have consistent mask color for the same object across different camera poses. Gaussians can also be selected by their identity encoding for scene editing tasks including removal, color-changing, position translation, *etc.*, as demonstrated in Sec. 4.6.

4 Experiments

4.1 Experimental Setup

Datasets. We experiment with various datasets across diverse scenarios to demonstrate the performance of *Gaga*. For quantitative comparison, we use a scene understanding dataset LERF-Mask [26], along with two indoor scene

datasets: Replica [23] and ScanNet [8]. LERF-Mask is based on the LERF dataset [12] and annotated with tasks and ground truth by the author of [26]. It contains 3 scenes: figurines, ramen, and teatime. For each scene, 6-10 objects are selected as text queries, and Grounding DINO [17] is utilized to select the mask ID from the rendered segmentation. Indoor datasets Replica and ScanNet are commonly used to evaluate 3D scene understanding methods, as ground truth semantic segmentation and panoptic segmentation are provided. We employ 8 scenes from Replica rendered by the author of [27], each consisting of 180 training images and an equal number for testing. We utilize 7 scenes in ScanNet, processed similarly to [24]. Each scene contains over 300 training images and around 100 testing images. Note that all annotated segmentation masks are unavailable during training and only accessible during evaluation as ground truth. We present visual comparison results on the commonly used scene reconstruction dataset, MipNeRF 360 [1]. Additionally, we showcase the robustness of *Gaga* against variations in training image quantity by sparsely sampling Replica and MipNeRF 360 datasets. For all experiments, the quantitative and qualitative results are conducted on the test set, *i.e.*, novel view synthesis results.

Evaluation Metrics. mIoU and boundary IoU (mBIoU) are used for evaluation on the LERF-Mask dataset. For Replica and ScanNet datasets, we evaluate using ground truth panoptic segmentation, disregarding class information. To handle differences between predicted and ground truth mask labels, we calculate the best linear assignment based on IoU. Moreover, with IoU = 0.5 as the criterion, we report precision and recall to further evaluate the accuracy of predicted masks.

Implementation Details. We use SAM [14] the default version and EntitySeg [21] with the Hornet-L backbone to obtain open-world 2D segmentation. We preprocess the generated raw masks following the method outlined in [21], prioritizing those with higher confidence scores by ranking them accordingly. Masks with confidence scores below 0.5 are discarded. For all experiments, we train the vanilla Gaussian Splatting for 30K iterations and train the identity encoding for 10K iterations with all other parameters frozen. We choose the front 20% 3D Gaussians that are closest to the camera frames as the corresponding Gaussians of a mask. We set the overlap threshold for declaring a new group ID as 0.1. For fair comparisons, we train Gaussian Grouping [26] for 40K iterations, with all parameters for training 3D Gaussians with their identity encoding remaining the same as the default setting in [11] and [26]. We use the official transcript of [11] to obtain camera poses and initial point clouds.

4.2 Open-vocabulary 3D Query on LERF-Mask Dataset

We experiment on the LERF-Mask dataset following the evaluation process in [26]. Tab. 1 illustrates that *Gaga* achieves superior results in mIoU and mBIoU compared to previous methods [12,26]. especially when utilizing EntitySeg as the 2D segmentation method, resulting in approximately a 9% advantage for both mIoU and mBIoU. In Fig. 5, we present visualizations of the rendered segmentation, rendered feature field, and 3D query results. *Gaga* yields more precise

Table 1: Quantitative results for open-vocabulary 3D query tasks on LERF-Mask dataset. *Gaga* outperforms previous approaches, showcasing favorable performance in terms of mIoU and BIoU with both segmentation models. * denotes the results are reported in [26].

Model	2D Seg. Method	mIoU(%)	mBIoU(%)
LERF [12]*	/	37.17	29.30
Gaussian Grouping [26]	EntitySeg	54.10	50.90
Gaga(Ours)		62.44	60.28
Gaussian Grouping [26]*	SAM	72.79	67.58
Gaga(Ours)		**74.71**	**72.19**

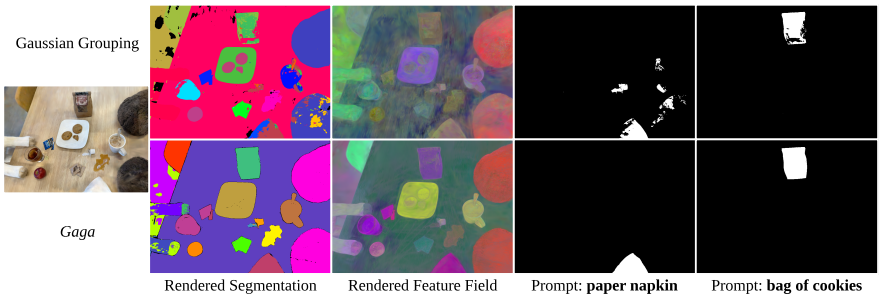

Fig. 5: Visual comparison on LERF-Mask dataset. Our rendered segmentation exhibits fewer artifacts and delivers more accurate instance segmentation results. Visualization of the rendered feature field via PCA demonstrates that *Gaga* obtains superior identity encoding features.

segmentation with fewer artifacts and empty regions, indicating that our 3D-aware mask association method provides multi-view consistent 2D segmentation with less label ambiguity as pseudo labels for training the identity encoding. The visualized identity encoding feature field of *Gaga* exhibits much cleaner results, further supporting this observation. Hence, *Gaga* can provide a more precise instance segmentation for queried 3D objects.

Table 2: Quantitative results on Replica and ScanNet datasets. *Gaga* performs well with both 2D segmentation methods on two datasets. Notice that the performance of Gaussian Grouping varies significantly with different 2D segmentation methods, whereas *Gaga* consistently delivers stable performance.

Model	2D Seg. Method	Replica IoU(%)	Replica Precision(%)	Replica Recall(%)	ScanNet IoU(%)	ScanNet Precision(%)	ScanNet Recall(%)
Gaussian Grouping [26]	EntitySeg	35.90	14.07	31.57	39.54	6.88	36.56
Gaga(Ours)		41.08	**63.06**	46.14	42.56	**33.89**	**47.63**
Gaussian Grouping [26]	SAM	21.76	25.00	19.72	34.24	18.70	32.61
Gaga(Ours)		**46.50**	41.52	**52.50**	**44.87**	18.61	45.94

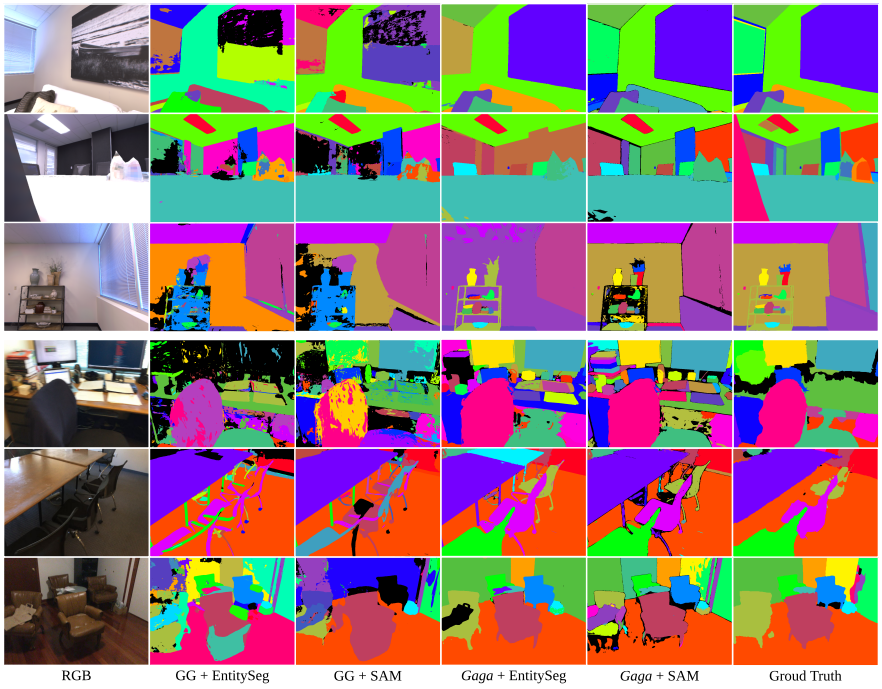

Fig. 6: Qualitative results on Replica and ScanNet datasets. *Gaga* provides high-quality segmentation masks that are more similar to the ground truth. Gaussian Grouping often covers the same object with different masks (rows 1, 4, 6), creates large empty regions (rows 1-4), and misidentifies similar instances (rows 5, 6).

4.3 3D Segmentation on Replica and ScanNet Datasets

Tab. 2 presents the quantitative comparison results on the Replica and ScanNet datasets. *Gaga* exhibits better performance on both datasets regardless of the 2D segmentation model utilized, showcasing its stability across different datasets and models. Qualitative results are shown in Fig. 6. Rows 1-3 depict the visualizations from the Replica dataset, while rows 4-6 showcase results from the ScanNet dataset. Gaussian Grouping [26] frequently assigns different mask IDs to the same object, resulting in inconsistent mask colors and empty regions. Rows 5 and 6 illustrate that Gaussian Grouping struggles to distinguish similar objects, whereas our proposed *Gaga* accurately identifies each object by leveraging 3D information.

4.4 3D Segmentation with Limited Data on Replica Dataset

To demonstrate the robustness of *Gaga* against changes in training image quantity, we sparsely sample the Replica training set with ratios of 0.3, 0.2, 0.1, and 0.05. As depicted in Tab. 3, *Gaga* consistently exhibits superior performance in terms of IoU, with approximately a 10% advantage using EntitySeg and a

Table 3: Quantitative results on Replica dataset with limited training data. *Gaga* consistently outperforms Gaussian Grouping with both 2D segmentation methods. The percentage of IoU drop indicates that *Gaga* exhibits greater robustness against reductions in training data.

2D Seg. Method		EntitySeg		SAM	
Model	Training Data	IoU(%) ↑	IoU Drop(%) ↓	IoU(%) ↑	IoU Drop(%) ↓
Gaussian Grouping [26]	30%	28.42	20.85	17.02	21.78
Gaga(Ours)		37.98	7.57	41.79	10.11
Gaussian Grouping [26]	20%	24.56	31.35	16.02	26.38
Gaga(Ours)		37.25	9.33	40.27	13.40
Gaussian Grouping [26]	10%	20.62	42.56	13.97	35.78
Gaga(Ours)		31.93	22.27	35.61	23.40
Gaussian Grouping [26]	5%	10.00	72.15	6.77	68.87
Gaga(Ours)		20.59	49.88	22.79	50.98

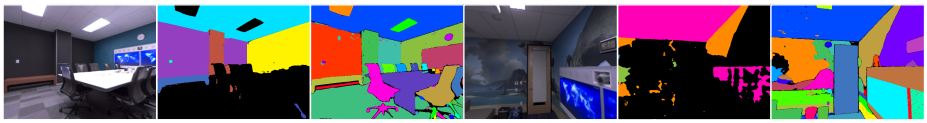

Fig. 7: Qualitative results on Replica dataset with limited training data. The visualizations depict samples when using only 5% of the training data (9 training images). Even with limited data, *Gaga* consistently produces high-quality segmentation. In contrast, Gaussian Grouping struggles to track objects accurately and leaves significant empty regions.

20% advantage using SAM. Remarkably, when utilizing SAM, *Gaga* surpasses fully trained Gaussian Grouping with just 5% of the training data (22.79% *vs.* 21.76%). We also compute the IoU drop compared to using all training images as follows:

$$IoU\ Drop(x) = \frac{IoU(100\%) - IoU(x\%)}{IoU(100\%)}, \tag{5}$$

where $IoU(x\%)$ denotes the IoU achieved when $x\%$ of the training data is used. Compared to Gaussian Grouping, *Gaga* exhibits less sensitivity to decreases in the number of training images, as evidenced by smaller values in IoU drop. Visualization results are shown in Fig. 7. With just 5% of the training data, *Gaga* can still deliver accurate segmentation masks, whereas Gaussian Grouping fails to provide masks for a significant portion of objects due to inaccurate tracking.

4.5 3D Segmentation on MipNeRF 360 Dataset

We further showcase the performance of *Gaga* on diverse scenarios, *i.e.*, the MipNeRF 360 dataset, with SAM as 2D segmentation method. We provide visualization comparison with Gaussian Grouping [26] in Fig. 8, rows 1-3. We display two images for each scene to assess the consistency across different views. *Gaga* offers

Fig. 8: Qualitative results on the MipNeRF 360 dataset. Rows 1-3 show that *Gaga* provides superior segmentation with finer details (rows 1, 2), fewer artifacts (row 1), and more consistent instance segmentation across different views (bicycle in row 3). Rows 4-6 present results with sparsely sampled data. While Gaussian Grouping can not accurately track objects (bicycle in row 4, bulldozer in row 5, stump and flowers in row 6), *Gaga* consistently provides precise segmentation.

more detailed segmentation, while segmentation masks generated by Gaussian Grouping exhibit severe artifacts. Additionally, inconsistency across two views exists in the rendering results of Gaussian Grouping, as shown in row 3.

We sparsely sample the MipNeRF 360 dataset with sample step = 3 and visualize the results in Fig. 8, rows 4-6. Similar to results in Fig. 7, Gaussian Grouping can not accurately track objects with limited training data, resulting in empty regions in rendered segmentation. Conversely, *Gaga* maintains accurate segmentation, even for those tiny leaves in the "stump" scene in row 6.

4.6 Application: Scene Manipulation

Gaga achieves high-quality and multi-view consistent 3D segmentation, beneficial for tasks like scene manipulation, as we can accurately segment the Gaussians of a 3D object and edit their properties. Using a pre-trained 3D Gaussian model with identity encoding, we employ the classifier trained with identity encoding to predict mask labels for each 3D Gaussian. Subsequently, we select 3D Gaussians sharing the same mask label as the target object and edit their properties for tasks like object coloring, removal, and position translation.

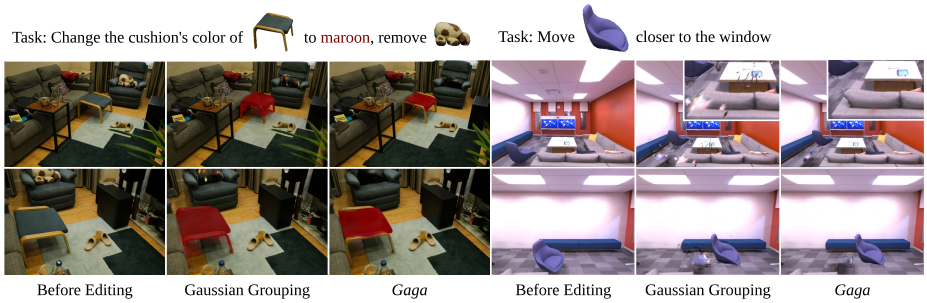

Fig. 9: Scene manipulation results on MipNeRF 360 and Replica datasets. *Gaga* accurately identifies the cushion part of the footstool, whereas Gaussian Grouping colors it entirely. For object removal and translation tasks, *Gaga* generates more precise 3D entities with fewer artifacts, resulting in better visual performance.

We demonstrate the effectiveness of this application on MipNeRF 360 and Replica datasets (see Fig. 9). In the "room" scene of MipNeRF 360, we change the color of the cushion in the footstool and remove the stuffed animal on the armchair. *Gaga* accurately identifies 3D Gaussians representing the cushion, while Gaussian Grouping [26] fails, colors the entire footstool maroon along with part of the sofa and some floating Gaussians. *Gaga* also effectively groups and removes the entire part of the stuffed animal on the sofa with minimal artifacts, whereas Gaussian Grouping leaves many floating Gaussians. Similar results are observed in the experiment involving the position shifting of a chair in the "office 3" scene of the Replica dataset. Notice that Gaussian Grouping also creates artifacts at positions far away from the target object.

4.7 Ablation Study on Mask Association Method

We conduct ablation studies to evaluate the effectiveness of the proposed mask association method on the Replica dataset. The baselines for comparison include:

1. *w/o. Mask Association*: Lifting inconsistent 2D masks to 3D.
2. *Video Object Tracker*: [26] is employed as a representative method.
3. *Memory Bank (w. All Gaussians)*: Associating masks in the same manner as *Gaga*, except that it selects all Gaussians splatted to the mask as its corresponding Gaussians.

Table 4: Ablation study on different mask association methods. Our mask association method with 3D-aware memory bank surpasses the previous video tracker baseline on both IoU, Precision, and Recall.

Baseline	IoU (%)	Precision (%)	Recall (%)
w/o. Mask Association	8.81	3.19	2.16
Video Object Tracker [26]	21.76	25.00	19.72
Memory Bank (w. All Gaussians) (Ours)	42.26	40.19	45.95
Memory Bank (w/o. Mask Partition) (Ours)	46.08	27.88	50.67
Memory Bank (Ours)	**46.50**	**41.52**	**52.50**

Fig. 10: Visual comparison of different mask association methods. *Gaga* with 3D-aware memory bank achieves a superior visual quality, and closer to the ground truth. Notice that the Video Object Tracker baseline mislabels the wall and floor, Memory Bank (w. All Gaussians) mislabels the floor, and Memory Bank (w/o. Mask Partition) baseline creates artifacts on the table and chairs.

4. *Memory Bank (w/o. Mask Partition)*: Associating masks in the same manner as *Gaga*, except that it does not partition the image and masks into patches.
5. *Memory Bank*: *i.e.*, *Gaga*.

Quantitative results in Tab. 4 indicate that *Gaga* with the 3D-aware memory bank achieves superior performance with a 24.74%, 16.52%, and 32.78% improvement on IoU, precision on recall, respectively, compared to the previous method with video object tracker. Comparison with the Memory Bank (w. All Gaussians) and Memory Bank (w/o. Mask Partition) baselines demonstrate the effectiveness of our well-designed process for finding corresponding Gaussians of each mask. We also show the visual comparison in Fig. 10.

5 Conclusions

We introduce *Gaga*, a framework that reconstructs and segments open-world 3D scenes by utilizing inconsistent 2D masks predicted by zero-shot segmentation models. *Gaga* employs a 3D-aware memory bank to store the indices of pre-trained 3D Gaussians and establishes mask association across different views by identifying the overlap between Gaussians that are projected to each mask. Results on various datasets demonstrate that *Gaga* outperforms previous methods with superior segmentation accuracy, multi-view consistency, and reduced artifacts. Additionally, application in scene manipulation highlights *Gaga*'s high segmentation accuracy and practical utility.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. ICCV (2021)
2. Cen, J., Fang, J., Yang, C., Xie, L., Zhang, X., Shen, W., Tian, Q.: Segment any 3d gaussians. arXiv preprint arXiv:2312.00860 (2023)
3. Cen, J., Zhou, Z., Fang, J., Yang, C., Shen, W., Xie, L., Jiang, D., Zhang, X., Tian, Q.: Segment anything in 3d with nerfs. In: NeurIPS (2023)
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: ECCV (2022)

5. Chen, X., Tang, J., Wan, D., Wang, J., Zeng, G.: Interactive segment anything nerf with feature imitation (2023)
6. Cheng, H.K., Oh, S.W., Price, B., Schwing, A.G., Lee, J.: Tracking anything with decoupled video segmentation. In: ICCV (2023)
7. Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., Yang, Y.: Segment and track anything. arXiv preprint arXiv:2305.06558 (2023)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
9. Dou, B., Zhang, T., Ma, Y., Wang, Z., Yuan, Z.: Cosseggaussians: Compact and swift scene segmenting 3d gaussians with dual feature fusion. arXiv preprint arXiv:2401.05925 (2024)
10. Fu, X., Zhang, S., Chen, T., Lu, Y., Zhu, L., Zhou, X., Geiger, A., Liao, Y.: Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In: International Conference on 3D Vision (3DV) (2022)
11. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM TOG (2023)
12. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: ICCV (2023)
13. Kim, C.M., Wu, M., Kerr, J., Tancik, M., Goldberg, K., Kanazawa, A.: Garfield: Group anything with radiance fields. In: arXiv (2024)
14. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W., Dollár, P., Girshick, R.B.: Segment anything. In: ICCV (2023)
15. Kundu, A., Genova, K., Yin, X., Fathi, A., Pantofaru, C., Guibas, L., Tagliasacchi, A., Dellaert, F., Funkhouser, T.: Panoptic neural fields: A semantic object-aware neural scene representation. In: CVPR (2022)
16. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: ICLR (2022)
17. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
18. Liu, Y., Hu, B., Huang, J., Tai, Y.W., Tang, C.K.: Instance neural radiance field. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
19. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
20. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG (2022)
21. Qi, L., Kuen, J., Shen, T., Gu, J., Li, W., Guo, W., Jia, J., Lin, Z., Yang, M.: High quality entity segmentation. In: ICCV (2023)
22. Siddiqui, Y., Porzi, L., Bulò, S.R., Müller, N., Nießner, M., Dai, A., Kontschieder, P.: Panoptic lifting for 3d scene understanding with neural fields. In: CVPR (2023)
23. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
24. Wang, B., Chen, L., Yang, B.: Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. In: ICLR (2023)
25. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos (2023)

26. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023)
27. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.J.: In-place scene labelling and understanding with implicit scene representation. In: ICCV (2021)
28. Zhou, S., Chang, H., Jiang, S., Fan, Z., Zhu, Z., Xu, D., Chari, P., You, S., Wang, Z., Kadambi, A.: Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. CVPR (2024)

Appendix

In this appendix, we provide further experimental results, including a qualitative comparison with GARField [13], additional results on scene manipulation and sparse view setting in Sec. B. We then delve into more experimental details of the datasets, metrics, and implementation in Sec. C. Additional ablation studies are shown in Sec. D and limitations are discussed in Sec. E.

A Supplementary Video

Please watch the supplementary demo video for a comprehensive introduction and visual comparison between our method *Gaga* and the current state-of-the-art methods. The video features additional qualitative comparisons and an animation illustration of our method.

B Supplementary Experiments

B.1 Results Compared with GARField

We provide comparison results with GARField in Fig. 11. GARField follows a hierarchical grouping pipeline. It extracts densely sampled segmentation masks from SAM [14] and trains a feature field using contrastive loss for grouping. If two rays fall into the same SAM mask, their features will be pulled together. Otherwise, features are pushed apart.

We use the default setting to train GARField. For a fair comparison, *Gaga* also uses the 2D segmentation masks provided by SAM. Visualization results show that *Gaga* provides segmentation masks with better quality and multi-view consistency. Whereas GARField does not provide multi-view consistent segmentation, and it also has inferior RGB rendering results.

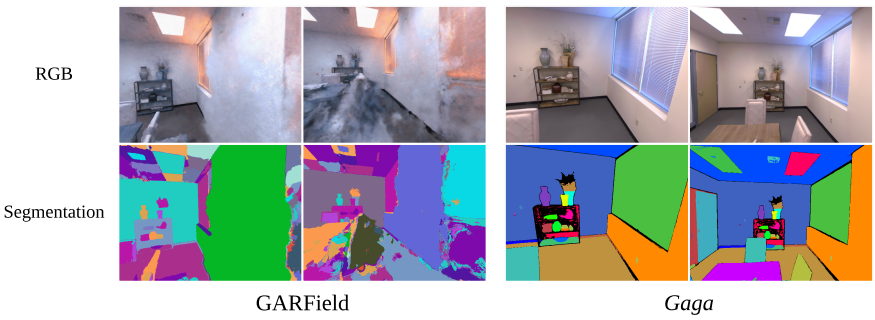

Fig. 11: Qualitative comparison with GARField on Replica dataset. *Gaga* renders higher-quality RGB and segmentation masks in significantly less time. It's worth noting that in the segmentation masks generated by GARField, the same colors are used multiple times for different masks, meaning one mask label may contain multiple groups representing different 3D instances.

Fig. 12: Scene manipulation results on MipNeRF 360 and LERF-Mask datasets. *Gaga* accurately identifies the flowerpot without affecting the color of the plant. Notice that Gaussian Grouping creates a cyan region on the wooden door behind. For the object removal and duplication tasks, *Gaga* can also provide more accurate results with fewer artifacts.

After training, GARField employs a hierarchical grouping pipeline to cluster each pixel into groups and generates segmentation masks. This hierarchical structure comprises 41 levels, and it takes approximately 20 minutes to output segmentation masks for a single image. In contrast, *Gaga* renders segmentation for one image under 0.5 seconds.

B.2 Additional Results on Scene Manipulation

We provide additional results for the downstream scene manipulation task to further demonstrate the prospect of applying *Gaga* to real-world scenarios. On the "counter" scene of the MipNeRF 360 dataset [1], we change the color of the flowerpot to cyan and duplicate the glass jar. Gaussian Grouping [26] can not differentiate the plant and flowerpot, whereas *Gaga* generates a more accurate segmentation mask. Additionally, *Gaga* produces a clearer boundary and avoids artifacts on the iron tray when duplicating the glass jar.

In the "figurines" scene of the LERF-Mask dataset [26], we transform the yellow duck to blue and remove the red toy chair. *Gaga* precisely changes only the duck's color without affecting other objects, and achieves a more thorough removal of the red toy chair.

B.3 Additional Results on Sparsely Sampled Replica Dataset

We provide additional qualitative results for the experiment on the sparsely sampled replica dataset in Fig. 13. As the number of training images decreases, Gaussian Grouping produces more empty regions, *e.g.* the sofa and the wall, due to difficulties in accurate tracking under sparse views. Whereas *Gaga* exhibits a more robust performance against reductions in the number of images.

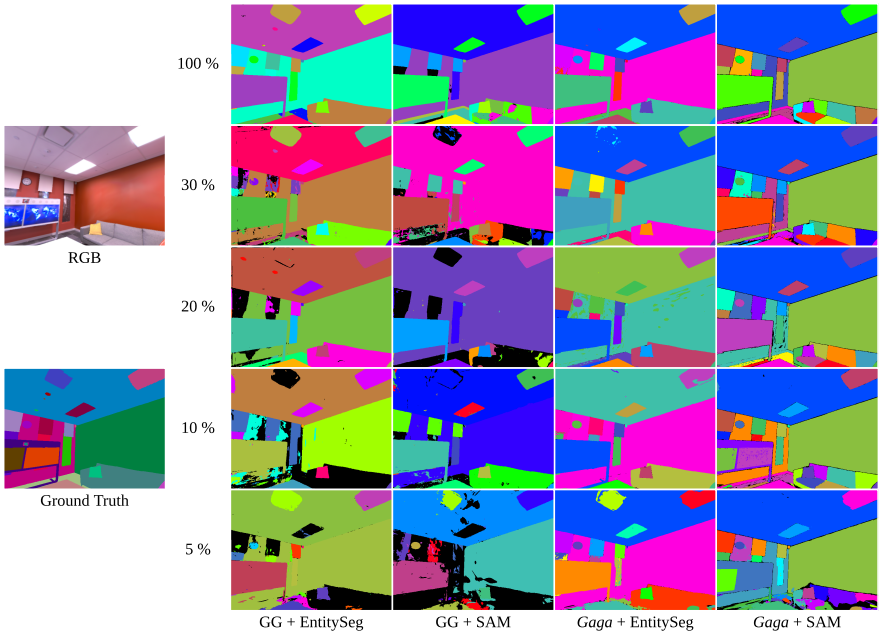

Fig. 13: Qualitative results on the sparsely sampled Replica dataset. We showcase the novel view synthesis segmentation rendering results provided by Gaussian Grouping and *Gaga* as the percentage of training images employed decreases from 100% to 5%. Gaussian Grouping cannot correctly track the sofa under sparse views and fails to differentiate ceiling and wall, whereas *Gaga* consistently provides high-quality segmentation results.

C Experimental Details

C.1 Details on Datasets

Replica Dataset [23]. We select 8 scenes from the entire Replica dataset the same as [27]. We use the rendered results provided by authors of [27] and follow their data processing process: for each scene, we uniformly select 20% images as training data and 20% images as test data from all rendered RGB images. This results in 180 training images and 180 test images for each scene.

Sparsely Sampled Replica Dataset. For the same 8 scenes as the previous experiment, we randomly sample 30%, 20%, 10%, and 5% of the total 180 training images, resulting in 54, 36, 18, and 9 training images for each task, respectively. The number of test images remains at 180.

ScanNet Dataset [8]. DM-NeRF [24] selects 8 scenes from the entire ScanNet dataset. Each scene has approximately 300 images for training and about 100 images for testing. We utilize 7 out of the 8 scenes, excluding "scene 0024_00" due to the subpar 3D reconstruction results in both Gaussian Splatting [11] and Gaussian Grouping [26].

Table 5: Selected scenes in Replica and ScanNet datasets. We select 8 scenes from the Replica dataset following [27], and 7 scenes from the ScanNet dataset following [24].

Dataset	Scene Name			
Replica [23]	office 0	office 1	office 2	office 3
	office 4	room 0	room 1	room 2
ScanNet [8]	scene 0010_00	scene 0012_00	scene 0033_00	scene 0038_00
	scene 0088_00	scene 0113_00	scene 0192_00	

MipNeRF 360 Dataset [1]. We downsample the images by a factor of 4 to accommodate the large size of the original images. For novel view synthesis evaluation, we set the sample step at 8, the same as the setting in [11].

We employ the official script from Gaussian Splatting [11] for colmap to acquire camera poses and the initial point cloud. Consequently, the actual number of images utilized in the experiment might be lower than expected due to colmap process failures. Please refer to Tab. 5 for the scene names used in the Replica and ScanNet datasets.

C.2 Details on Evaluation Metrics

Given the disparate mask label assignments between the ground truth segmentation and the predicted segmentation for 3D objects, we find the best linear assignment between the labels based on IoU for quantitative evaluation. Subsequently, we employ IoU > 0.5 as the criterion for precision and recall calculations. We outline the pseudocode for the evaluation procedure in Algorithm 1.

C.3 Further Implementation Details

For training vanilla 3D Gaussians, we maintain the same parameter setting as [11]. To train the identity encoding, we freeze all the other attributes of Gaussians and use the same parameter setting as [26]. The identity encoding has 16 dimensions, and the rendered 2D identity encoding is in the shape of 16 $\times\ h \times\ w$, where h and w represent the height and width of the image. The same classifier is utilized for predicting mask ID given the 2D identity encoding and selecting Gaussians for editing given the 3D identity encoding. It has 16 input channels and the number of output channels equals the number of groups in the 3D-aware memory bank after the mask association process for all images. All datasets are trained on a single NVIDIA RTX 6000 Ada GPU.

D Supplementary Ablation Studies

We conduct additional ablation studies on three parameters involved in the process of mask association and find corresponding Gaussians of a mask. These ablation studies are performed on the Replica dataset [23], utilizing SAM [14]

Algorithm 1 Evaluation Metrics

Input *pred_masks* and *gt_masks* are represented in binary format with shape (n_{image}, n_{mask}, h, w), where n_{image} is the number of test images, n_{mask} is the number of predicted or ground truth masks, h, w are the height and width of test images.

We use `scipy.optimize.linear_sum_assignment` to solve the linear assignment problem.

Function `evaluate`(pred_masks, gt_masks)
  Input: pred_masks (torch.bool), gt_masks (torch.bool)
  Output: iou (torch.float), precision (torch.float), recall (torch.float)

assert len(gt_masks) == len(pred_masks)
$n_{image} \leftarrow$ len(gt_masks)
$n_{pred} \leftarrow$ pred_masks.shape[1]
$n_{gt} \leftarrow$ gt_masks.shape[1]
iou_matrix $\leftarrow$ torch.zeros((n_{gt}, max(n_{gt}, n_{pred})))
for i **in** n_{gt} **do**
  for j **in** n_{pred} **do**
    iou_list $\leftarrow$ []
    for k **in** n_{image} **do**
      iou_list.append(`IoU`(gt_masks[k][i], pred_masks[k][j]))
    end for
    iou_matrix[i][j] $\leftarrow$ iou_list.mean()
  end for
end for
$gt\_indices$, $pred\_indices \leftarrow$ `linear_assignment`(iou_matrix)
paired_iou $\leftarrow$ iou_matrix[$gt\_indices$][$pred\_indices$]
iou $\leftarrow$ paired_iou.mean()
$n_{correct} \leftarrow$ torch.sum(paired_iou > 0.5)
precision $\leftarrow \frac{n_{correct}}{n_{pred}}$
recall $\leftarrow \frac{n_{correct}}{n_{gt}}$
return iou, precision, recall

Table 6: Ablation study on the percentage of front Gaussians. Results for selecting 10%, 20%, 30%, and 100% of front Gaussians as corresponding Gaussians of a mask are presented below. *Gaga* demonstrates stable performance across varying parameters, showcasing its robustness.

Perc. Front Gaussians (%)	IoU (%)	Precision (%)	Recall (%)
10	46.42	39.57	51.54
20 *	**46.50**	41.52	**52.50**
30	45.73	**42.31**	50.88
100	42.26	40.19	45.95

Table 7: Ablation study on image partition. We partition the entire image and its masks into patches to prevent the selected corresponding Gaussians from concentrating in a confined region. Comparison results show that *Gaga* can perform well as long as the partition process is employed.

Num. Patches	IoU (%)	Precision (%)	Recall (%)
1 × 1	46.08	27.88	50.67
16 × 16	46.11	38.22	51.62
32 × 32 *	**46.50**	**41.52**	**52.50**
64 × 64	44.72	40.65	49.14

as the 2D segmentation model. Parameters denoted with * are used as the default setting. We also provide additional visual comparison results for the mask association methods utilized by Gaussian Grouping [26] and *Gaga* in Sec. D.4.

D.1 Percentage of Front Gaussians

We present the ablation study on the percentage of front Gaussians selected as corresponding Gaussians in Tab. 6. We choose 10%, 20%, 30%, and 100% (*i.e.* selecting all Gaussians splatted to the mask as its corresponding Gaussians) as candidate parameters. The default setting (20%) has a better performance in general. *Gaga* shows stable performance for all candidate parameters, indicating its robustness and it does not rely on cautious parameter selection.

D.2 Number of Image Patches During Partition

We provide the ablation study on the number of image patches used during the image partition process in Tab. 7. Candidate parameters include 1 × 1 (without mask partition process), 16 × 16, 32 × 32, 64 × 64. Similar to the results in Tab. 6, *Gaga* remains insensitive to the choice of this parameter as long as the image partition process is in place. Without the mask partition process, there is a significant drop in precision.

D.3 Overlap Threshold

During the group ID assigning process, if none of the existing groups in the memory bank has a larger overlap with the current mask than the threshold,

Table 8: Ablation study on the overlap threshold. If the overlap between the current mask and all groups in the memory bank falls below this threshold, we add this mask to the memory bank as a new group. Results indicate that the default setting of 0.1 generally yields better outcomes.

Overlap Threshold	IoU (%)	Precision (%)	Recall (%)
0.01	43.86	**44.99**	48.98
0.1 *	46.50	41.52	**52.50**
0.2	**47.57**	34.77	52.40

we incorporate this mask into the memory bank as a new group, signifying the discovery of a new 3D object. We present the ablation study on overlap threshold in Tab. 8. When the threshold is set to 0.01, we rarely establish a new group and prefer to associate the mask with an existing group. It provides the best precision but at the expense of inferior IoU performance. Conversely, setting the threshold to 0.2 results in a frequent declaration of new group IDs, yielding the best IoU but a significant decrease in precision. Therefore, we set the threshold to 0.1 to strike a balance in performance across all three metrics.

D.4 Additional Comparison on Mask Association Methods

We present visual comparison results for two mask association methods, video object tracker [6] utilized by [26] and *Gaga*'s 3D-aware memory bank, in Fig. 14. In the "garden" scene of the MipNeRF 360 dataset, the video object tracker struggles to track objects in the background, whereas the 3D-aware memory bank provides associated results for each mask. For the scene in the ScanNet dataset, the video tracker fails to distinguish between four identical sofas, resulting in multiple masks for the same object. Additionally, it assigns different mask IDs to the table in two views. In contrast, the 3D-aware memory bank precisely locates each object, leading to improved mask association results and better pseudo labels for training segmentation features.

E Limitations

Though *Gaga* achieves state-of-the-art performance compared to existing works, there are a few limitations and future works. First, the optimization process of identity encoding and the rest of the Gaussian parameters are independent, this is because we need to first train 3D Gaussians to acquire their spatial location for mask association. While this pipeline allows for the utilization of any pre-trained 3D Gaussians as input without the need to re-train the entire scene, it does require additional training steps. We aim to enable the joint processing of mask association and identity encoding training in future works.

Secondly, artifacts may occur in the segmentation rendered by *Gaga* due to inherent inconsistency in the 2D segmentation. For example, an object might be depicted as two separate masks in the initial view but as one entire mask in

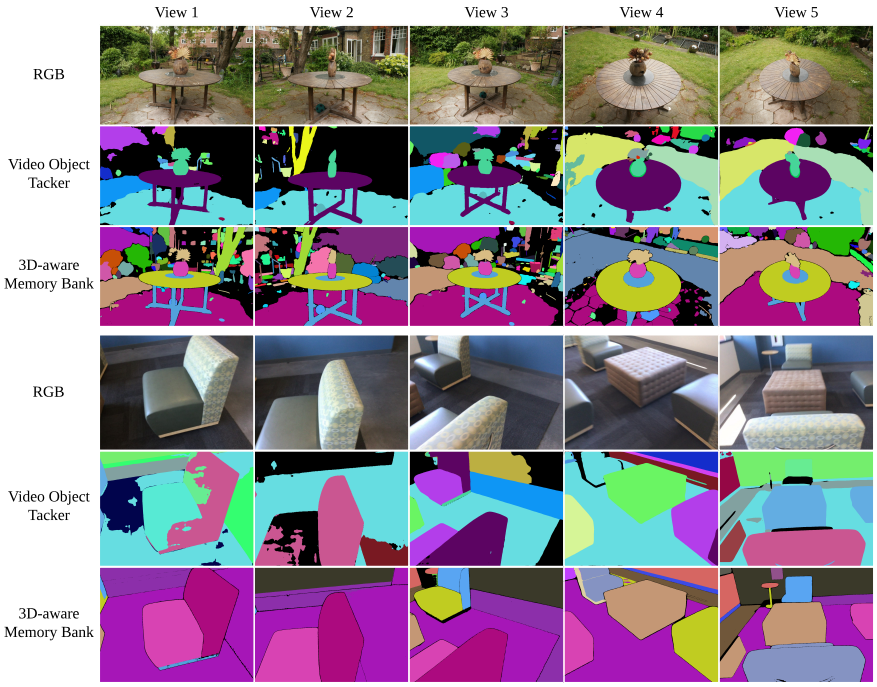

Fig. 14: Visual comparison between different mask association methods. Our 3D-aware memory bank offers more detailed associated masks. It accurately tracks identical objects in the scene and assigns them different mask IDs. Conversely, the video object tracker leaves empty regions in positions where it cannot track masks, and it struggles to provide consistent mask ID for the same object across views.

subsequent views. This ambiguity introduces challenges to our mask association process. Preprocessing steps such as dividing, merging, or reshaping the 2D segmentation masks could potentially resolve this issue and improve grouping results.