

GAGA: GROUP ANY GAUSSIANS VIA 3D-AWARE MEMORY BANK

Anonymous authors

Paper under double-blind review

SUPPLEMENTARY MATERIAL

In this supplementary document, we provide further experimental results, including a qualitative comparison with GARField (Kim et al., 2024), more 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. More 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 *Gaga*.

B SUPPLEMENTARY EXPERIMENTAL RESULTS

B.1 ADDITIONAL RESULTS COMPARED WITH GARFIELD

We provide comparison results with GARField in Fig. 1. GARField follows a hierarchical grouping pipeline. It extracts densely sampled segmentation masks from SAM (Kirillov et al., 2023) 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

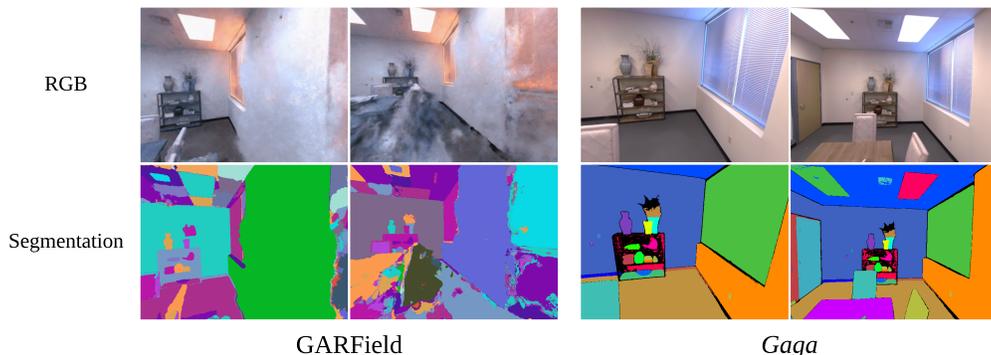


Figure 1: **Qualitative comparison with GARField on Replica dataset.** *Gaga* renders higher-quality RGB and segmentation masks in significantly less time. It’s worth noting that in the segmentation masks generated by GARField, the same colors are used multiple times for different masks, meaning one mask label may contain multiple groups representing different 3D instances. This is because, essentially, GARField performs a clustering task rather than a segmentation task.

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107



Figure 2: **Scene manipulation results on MipNeRF 360 and LERF-Mask dataset.** *Gaga* accurately identifies the flowerpot without affecting the color of the plant. Notice that Gaussian Grouping (Ye et al., 2024) 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.

with better quality and multi-view consistency. Whereas GARField does not provide multi-view consistent segmentation, and they also have inferior RGB rendering results.

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

B.2 ADDITIONAL RESULTS ON SCENE MANIPULATION

Gaga 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.

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 (Barron et al., 2021), we change the color of the flowerpot to cyan and duplicate the glass jar. Gaussian Grouping (Ye et al., 2024) 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 (Ye et al., 2024), 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. 3. As the number of training images decreases, Gaussian Grouping produces more empty regions, e.g. the sofa, due to difficulties in accurate tracking under sparse views. Whereas *Gaga* exhibits a more robust performance against reductions in the number of images.

C EXPERIMENTAL DETAILS

C.1 DETAILS ON DATASETS

We employ the official script from Gaussian Splatting (Kerbl et al., 2023) for colmap to acquire camera poses and the initial point cloud. Consequently, the actual number of images utilized in the

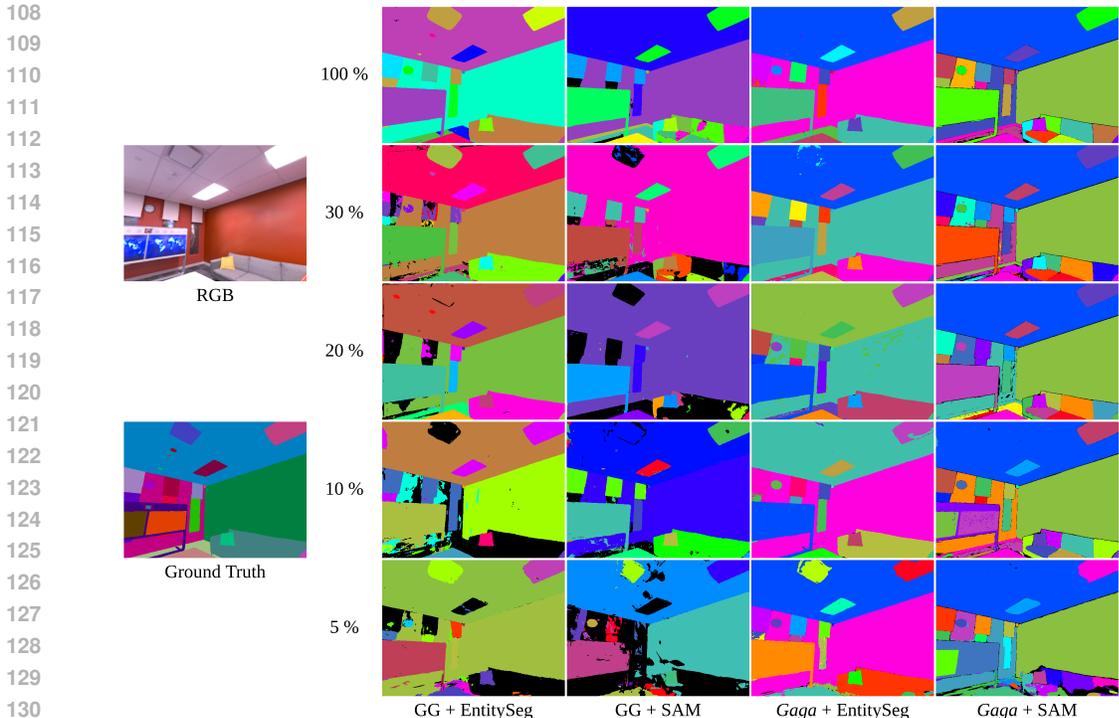


Figure 3: **Qualitative results on the sparsely sampled replica dataset.** We showcase the novel view synthesis segmentation rendering results provided by Gaussian Grouping and *Gaga* as the percentage of training images employed decreases from 100% to 5%. Gaussian Grouping cannot correctly track the sofa under sparse views and fails to differentiate ceiling and wall, whereas *Gaga* consistently provides high-quality segmentation results.

experiment might be lower than expected due to colmap process failures. Please refer to Tab. 1 for the scene names used in the Replica and ScanNet datasets.

LERF-Mask Dataset (Ye et al., 2024). LERF-Mask is based on the LERF dataset (Kerr et al., 2023) and annotated with tasks and ground truth by the author of (Ye et al., 2024). It contains 3 scenes: figurines, ramen, and teatime. For each scene, 6-10 objects are selected as text queries, and Grounding DINO (Liu et al., 2023) is utilized to select the mask ID from the rendered segmentation.

Replica Dataset (Straub et al., 2019). We select 8 scenes from the entire Replica Dataset the same as (Zhi et al., 2021). We use the rendered results provided by authors of (Zhi et al., 2021) 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 (Dai et al., 2017). DM-NeRF (Wang et al., 2023) 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 (Kerbl et al., 2023) and Gaussian Grouping (Ye et al., 2024).

MipNeRF 360 Dataset (Barron et al., 2021). We downsample the images by a factor of 4, consistent with the setting in (Ye et al., 2024), to accommodate the large size of the original images. For

Table 1: **Selected scenes in Replica and ScanNet datasets.** We select 8 scenes from the Replica dataset following (Zhi et al., 2021), and 7 scenes from the ScanNet dataset following (Wang et al., 2023).

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

novel view synthesis evaluation, we set the sample step at 8, the same as the setting in (Kerbl et al., 2023).

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 $\text{IoU} > 0.5$ as the criterion for precision and recall calculations. We outline the pseudocode for the evaluation procedure in Algorithm 1. Note that all annotated segmentation masks are unavailable during training and are only accessible during evaluation as ground truth.

Algorithm 1 Evaluation Metrics

Input pred_masks and gt_masks are represented in binary format with shape $(n_{\text{image}}, n_{\text{mask}}, h, w)$, 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 ← len(gt_masks)
n_pred ← pred_masks.shape[1]
n_gt ← gt_masks.shape[1]
iou_matrix ← torch.zeros((n_gt, max(n_gt, n_pred)))
for i in n_gt do
  for j in n_pred do
    iou_list ← []
    for k in n_image do
      iou_list.append(IoU(gt_masks[k][i], pred_masks[k][j]))
    end for
    iou_matrix[i][j] ← iou_list.mean()
  end for
end for
gt_indices, pred_indices ← linear_assignment(iou_matrix)
paired_iou ← iou_matrix[gt_indices][pred_indices]
iou ← paired_iou.mean()
n_correct ← torch.sum(paired_iou > 0.5)
precision ←  $\frac{n_{\text{correct}}}{n_{\text{pred}}}$ 
recall ←  $\frac{n_{\text{correct}}}{n_{\text{gt}}}$ 
return iou, precision, recall

```

Table 2: **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 3: **Ablation study on image partition.** We partition the entire image and its masks into patches to prevent selected corresponding Gaussians from concentrating in a confined region of a mask. Comparison results show that *Gaga* can perform well when 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

C.3 FURTHER IMPLEMENTATION DETAILS

For training vanilla 3D Gaussians, we maintain the same parameter setting as (Kerbl et al., 2023). To train the identity encoding, we freeze all the other attributes of Gaussians and use the same parameter setting as (Ye et al., 2024). 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 classifier for predicting mask ID given the 2D identity encoding and selecting Gaussians for editing given the 3D identity encoding shares the same architecture, with 16 input channels. The number of output channels equals the number of groups in the 3D-aware memory bank after associating all images. All experiments are conducted 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 (Straub et al., 2019), utilizing SAM (Kirillov et al., 2023) 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 (Ye et al., 2024) 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. 2. 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. 3. Candidate parameters include 1×1 (without mask partition process), 16×16 , 32×32 , 64×64 . Similar to the results in Tab. 2, *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.

Table 4: **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

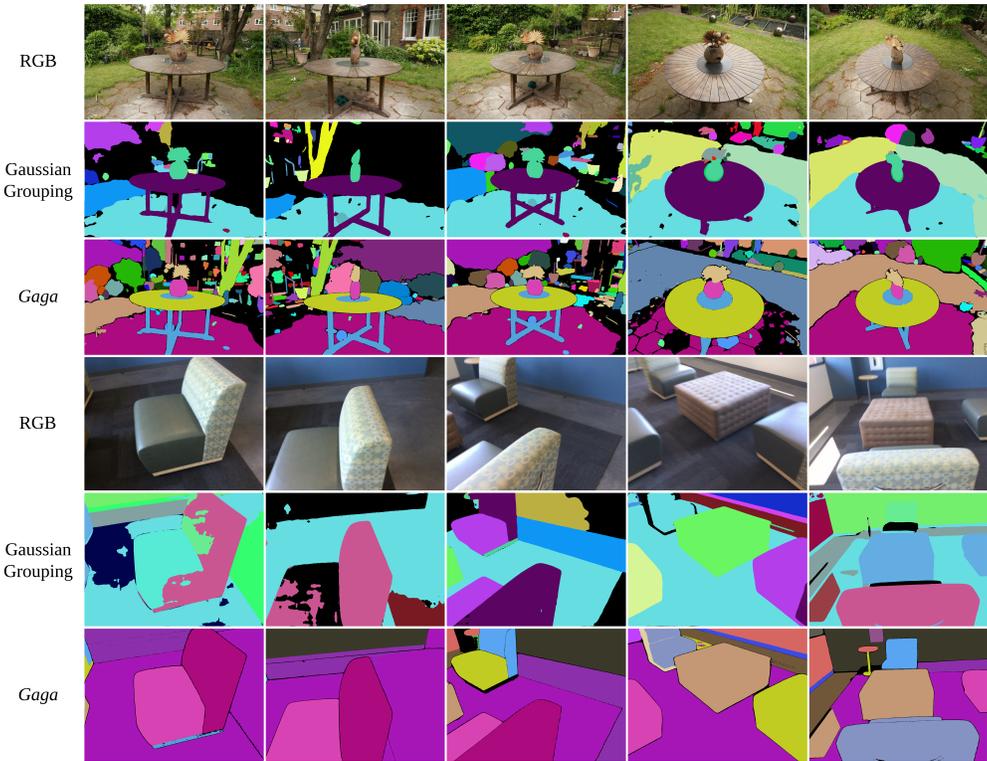


Figure 4: **Visual comparison between different mask association methods.** *Gaga* offers more detailed associated masks, accurately tracks identical objects in the scene and assigns them different mask IDs. Conversely, Gaussian Grouping leaves empty regions in positions where it cannot track masks, and it struggles to provide consistent masks for the same object across views.

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, 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. 4. 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 tracker (Cheng et al., 2023) utilized by (Ye et al., 2024) and *Gaga*'s 3D-aware memory bank, in Fig. 4. In the "garden"

324 scene of the MipNeRF 360 dataset, the video tracker struggles to track objects in the background,
325 whereas *Gaga* provides associated results for each mask. For the scene in the ScanNet dataset, the
326 video tracker fails to distinguish between four identical sofas, resulting in multiple masks for the
327 same object. Additionally, it assigns different mask IDs to the table in two views. In contrast, *Gaga*
328 precisely locates each object, leading to improved mask association results and better pseudo labels
329 for training segmentation features.

330 331 E LIMITATIONS 332

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

340 Secondly, artifacts may occur in the segmentation rendered by *Gaga* due to inherent inconsistency
341 in the 2D segmentation. For example, an object might be depicted as one mask in the initial view but
342 as two separate masks in subsequent views. This ambiguity introduces challenges to our mask as-
343 sociation process. Preprocessing steps such as dividing, merging, or reshaping the 2D segmentation
344 masks could potentially resolve this issue and improve grouping results.

345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377

REFERENCES

- Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021. 2, 3
- Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander G. Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *ICCV*, 2023. 6
- Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 3
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 2, 3, 4, 5
- Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lurf: Language embedded radiance fields. In *ICCV*, 2023. 3
- Chung Min* Kim, Mingxuan* Wu, Justin* Kerr, Matthew Tancik, Ken Goldberg, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *CVPR*, 2024. 1
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *ICCV*, 2023. 1, 5
- Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3
- Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 3, 5
- Bing Wang, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. In *ICLR*, 2023. 3, 4
- Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024. 2, 3, 5, 6
- Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. 3, 4