Object-centric grouping for a better understanding of 3d scene

Yongjun Choi
ccyjun123@unist.ac.kr

Junhee Lee
junhee98@unist.ac.kr

Seungoh Han
sohan@unist.ac.kr

Artificial Intelligence of Graduate School, UNIST

1. Introduction

People live in a three-dimensional world, and language is the key interaction for communication. To better understand our surroundings, it has been a long-standing problem –facilitating language guidance to represent and manipulate the environments. Aligning language knowledge with the 3D scene is a prerequisite for scene interactions, but it is not easy to consider language features and the 3D scene simultaneously.

LERF [5] allows pixel-aligned queries by distilling the off-the-shelf CLIP [12] without any handcrafted feature and fine-tuning. Due to slow training and evaluation, it still has difficulty adapting to the real-world scenario. LangSplat [11] tackles these difficulties to embed compressed language features into 3D Gaussian Splatting [4]. It also resolves point ambiguity by extracting language features with object-wise masks leaning against the recent powerful segmentation model known as SAM [6].

It requires three independent branches along the SAM [6] hierarchy, causing three times more training and inference resources. We empirically found out there's a trade-off between obtaining better details in 3D space and having a more compact representation from hierarchy-based training. We only use the object-level group, a high-level branch out of three. This helps us solve the multi-view inconsistency and speeds up evaluation by avoiding the hassle of training multiple times.

Even though LangSplat shows a breakthrough in pulling up the performance compared to previous works, it still has suffered from ambiguities between 3D world and 2D off-the-shelf models such as CLIP and SAM. Extracting knowledge per view causes multi-view inconsistency because it does not consider 3D information. To alleviate this, we introduce tracking-based contrastive learning with object-wise language features inspired by ContrastiveLift [1]. It leads that the object-wise features get closer in their feature space for positive pairs indicating the same instance and pull the embedding of negative pairs further apart. Finally, our proposed methods show qualitatively better as well as almost 2x faster evaluation speed although the performance metric slightly degrades.

Our contributions are summarized as follows:

- We first map CLIP features into object-level segments to reduce computational resources, achieving 2x faster inference time.
- We propose object-centric contrastive loss to track language embedding across views based on a video tracker.
- The experimental results show better qualitative performance. In addition, language-guided object removal is implemented, representing applicable ability from our proposed pipeline.

2. Related Works

2.1. 3D Gaussian Models

3D Gaussian Splatting [4] was recently introduced that reconstructs 3D scenes in real-time by representing them as 3D Gaussians. Due to the success of 3D Gaussian Splatting, many extension works have emerged. Several works [9, 15, 16] extended 3D Gaussian Splatting to model 3D dynamic scenes. Dynamic 3d gaussians [9] proposed adapting the 3D Gaussian approach to dynamic scenes by modeling these 3D Gaussians over various time steps. Meanwhile, Some works [3, 13, 18] combined 3D Gaussian splatting with diffusion models to generate 3D contents. DreamGaussian [13] proposed a generative 3D Gaussian Splatting model to generate 3D contents. Unlike these methods, Our proposed method utilizes 3D Gaussians with object-wise language features to generate object-wise semantic representation in 3D scenes.

2.2. 3D Language Fields

Several works [7, 14] were constructed to 3D feature fields by utilizing learned LSeg [8] or DINO [2] features into

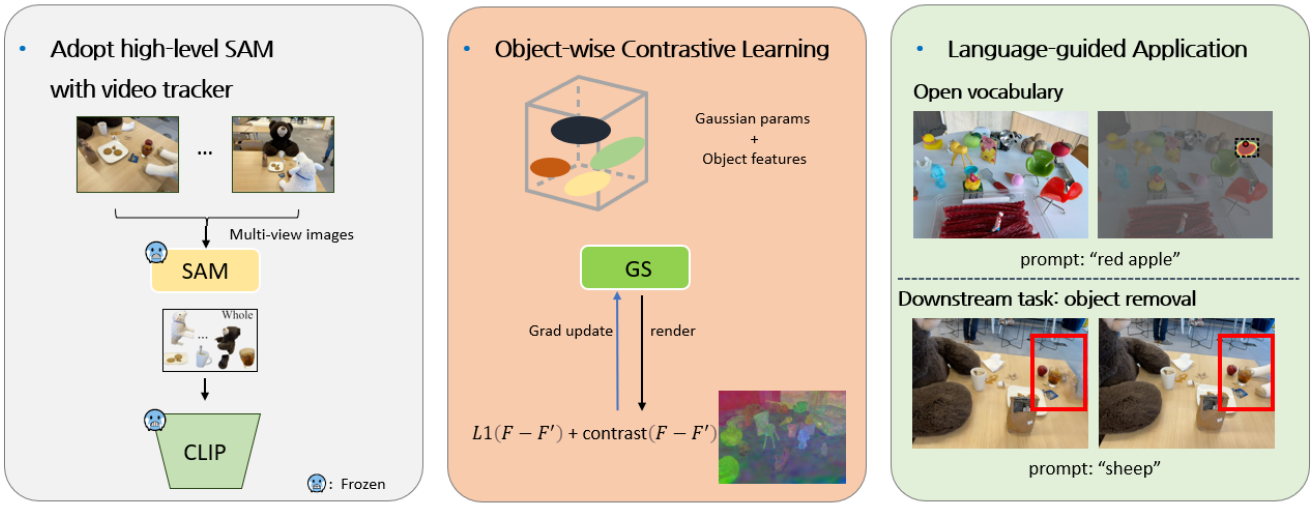

Adopt high-level SAM with video tracker
...
Multi-view images
SAM
Whole
CLIP
: Frozen
Object-wise Contrastive Learning
Gaussian params
+
Object features
GS
Grad update
render
$L1(F - F') + \text{contrast}(F - F')$
Language-guided Application
Open vocabulary
prompt: “red apple”
Downstream task: object removal
prompt: “sheep”

Figure 1. **Overall pipeline.**

a NeRF [10]. LERF [5] was the initial work to learn 3D scenes by embedding CLIP [12] features into NeRF. However, the NeRF-based method suffers from an expensive rendering process. Recently, 3D Gaussian Splatting proposed real-time 3D scene reconstruction. Feature 3DGS [19] presented the method to embed features from the 2D foundation model into each Gaussian representation. However, Feature 3DGS requires significant computational resources to embed features in each Gaussian. LangSplat [11] introduced SAM [6]-based CLIP features and autoencoder to compressed features to enhance training efficiency. Gaussian Grouping [17] confirms that each object has the same ID in a multi-view setting. In contrast, Our proposed method utilizes Contrastive lift [1] to learn tracking-based contrastive learning for learning object-centric grouping.

3. Method

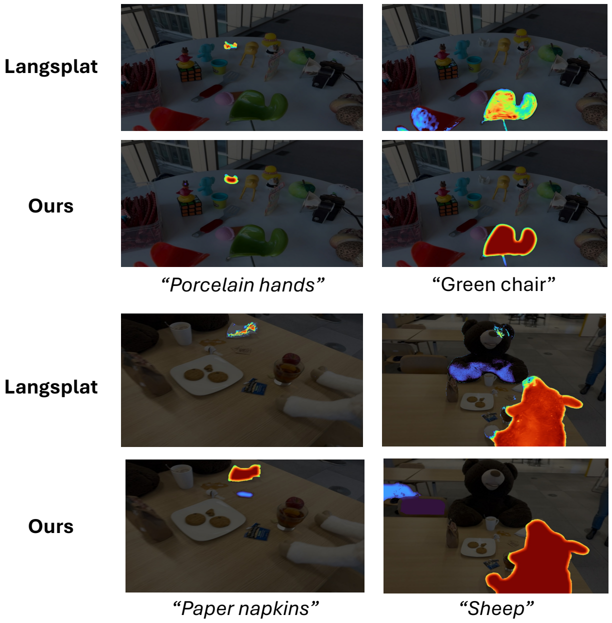
In this section, we describe object centric grouping method, which map with CLIP feature to learn object wise semantic representation for 3D scenes. We first describe our object wise CLIP mapping method in Sec. 3.1 in Sec. 3.2, we introduce contrastive loss for better alignment of 3D gaussians. Finally, in Sec. 3.3, we show downstream task which is one of the benefits of mapping language features to instance compared to distilling language features in each gaussian directly. Figure 1 represents overall pipeline.

3.1. Object wise CLIP feature mapping

In this work, instead of using all three levels of SAM's masks, we encode only at the object level into the CLIP feature. although SAM's segmentation performance is promised, it does not produce the consistent mask for the same object in multi views. To track the object's ID, we leverage Track Anything model as following Gaussian Grouping [17] to ensure that the same object has the same ID in multi view. For k objects, each objects are turned into CLIP image feature using object awared masks. To ensure recognizing object wise feature consistently, we concatenated the features in order based on the object's ID. Since not all objects can be contained in a single image due to occlusion or view angle limitations, we added a zero vector when object is not in the image. In this way, we can easily access the language feature of an object through its ID. To predict each object's ID, we assign ID features in each gaussians like Gaussian Grouping [17] that can predict each object's ID. This mapping procedure has the advantage of simplifying the learning pipeline as it does not require additional autoencoder training, and reducing open vocabulary query process time as only one model is evaluated.

3.2. Contrastive learning

To better align id features, we used the contrastive learning method inspired by Contrastive lift [1]. By sampling pixels within the same image, features which are in same mask make closer together and ones which are in different push further apart in Euclidian space. We applied slow-fast scheme like Contrastive lift [1]. Unlike NeRF based methods which the number of parameter is fixed during training, Gaussian splatting presents a challenge for introducing slow models since densification during training causes the total number of parameters in the slow and fast models to be different. To solve this problem, similar to LangSplat [11], we pretrained the vanilla 3D Gaussians before training.



“Porcelain hands” “Green chair”

“Paper napkins” *“Sheep”*

Figure 2. **Visualization of qualitative results.**

3.3. Downstream task

In LangSplat [11], language feature’s dimension is reduced Significantly, which makes not only lose information but also not be able to edit each Gaussians. In contrast, as we do not distill the language features directly into 3D Gaussians but access language features through the predicted object ID, there is no need for dimensionality reduction. Therefore, as long as the object IDs are well predicted, we can enjoy CLIP’s utility without information loss. This allows for a variety of downstream tasks, such as filtering the Gaussians to remove objects, or changing specific object’s color.

4. Experiments

4.1. Settings

To measure open vocabulary segmentation and localization, we train and evaluate our method on LERF dataset same as our baseline, LangSplat [11]. We measured performance on two scenes (Figurines, Teatime) from a given dataset and compared them to our baseline. also, to learn the object ID, We followed Gaussian Grouping [17]. we also assign an 16 dimension ID feature in each Gaussians and select k=256 for training classifier. for comparison, we add both of the language and object ID feature at cuda rasterization module. Since adding a 3D consistency loss cause OOM issue, we used only L1 loss and contrastive loss to learn features. we implement object removal for downstream task. for recognizing specific object IDs when text query given, we measure similarity score of text and image encoded features. after predict correspond objects, we filter specific Gaussians.

Scene	Figurines		Teatime	
	mIoU	Loc	mIoU	Loc
LangSplat	0.518	0.786	0.560	**0.864**
LangSplat + (1)	**0.522**	**0.804**	0.561	**0.864**
(2) + (3)	0.451	0.714	**0.573**	0.797
(2) + (3) + (4)	0.381	0.714	0.539	0.814
(2) + (3) + (4) + (5)	0.408	0.714	0.558	0.797

Table 1. **Comparison of Open Vocabulary segmentation and localization performance on LERF dataset.** each numbered method is, **(1)**: language feature contrastive loss, **(2)**: grouping, **(3)**: language mapping, **(4)**: instance level contrastive loss, **(5)**: slow-fast modeling. Those in bold are the highest performance, and underlined ones are the next highest performance.

Method	Figurines	Teatime
	eval time	eval time
LangSplat	around 10-11 min	around 11 min
Ours	around 6-7 min	around 6 min

Table 2. **Open Vocabulary evaluation time measurement.** our method is around 2 times faster than LangSplat

4.2. Result

Quantitative Results We first compare our method with our baseline and modified method on the LERF dataset. Table 1 shows the open vocabulary segmentation and localization results on two dataset. our method’s performance is lower than LangSplat, which utilizes SAM’s hierachical three levels and select along the best although our method only uses the object level.

Qualitative Results To show our model’s open vocabulary performance, we visualize open vocabulary segmentation result in Figure 2. note that LangSplat has more harsh boundary inside objects while our method gives perfect object shape thanks to mapping instance and language feature.

Evaluation Time In Table 2, we measure the time consumption to perform the open vocabulary task. As our method only inferences on one model, it is able to process the query about twice as fast as LangSplat.

4.3. Analysis

In this part, we’ll analyze why the performance has decreased from what we expected and look at the cases where it failed.

Why IoU Drop? As we can see in Figure 3, even though

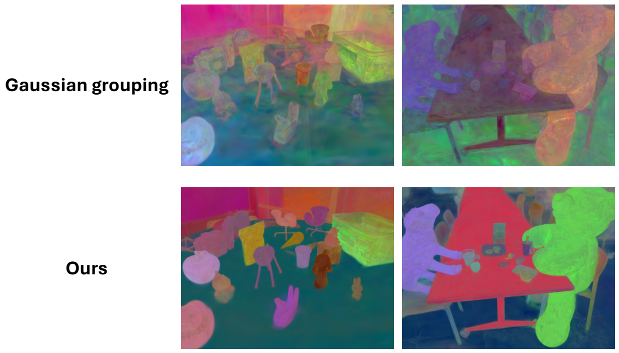


Figure 3. **Visualization of feature space compared to Gaussian Grouping.** overall feature space in each objects are much more aligned thanks to contrastive loss

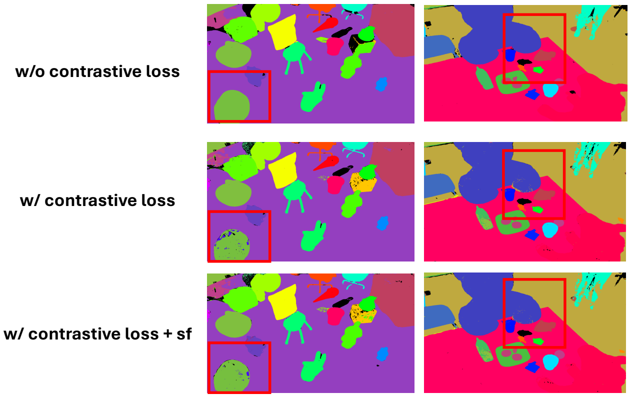


Figure 4. **Ablation study of contrastive loss's affectness.** even though contrastive loss seems to make whole feature field smoothly, it can affect classifier's performance when training together. you can see the artifact in the red boxes.

instance feature field aligned well it may affect classifier's performance. we can see segmentation prediction result after rendering in Figure 4. when introducing contrastive loss, there are several noise inside the object compared to the original one. slow fast learning can relieve this effect, but, still not using contrastive loss produce better result. we can try contrastive loss only after classifier train in future works.

Failure case in Figure 5, we can see some failure cases. Typical failures have two main characteristics. The first case is when the view changes dramatically. we observe errors in the scenes where the view changes from the front to the side or back of the object, which is likely because CLIP has been learning primarily from the front of the object when learning the image. CLIP model's view inconsistency causes incorrectly predict the object which is viewed from the back or heavily occluded views even though SAM mask consistently follows the object. The second case is when given text queries that are more granular than the object level. Since our model relies on object-level masked supervision, it will capture the whole object even if refering to something more granular.

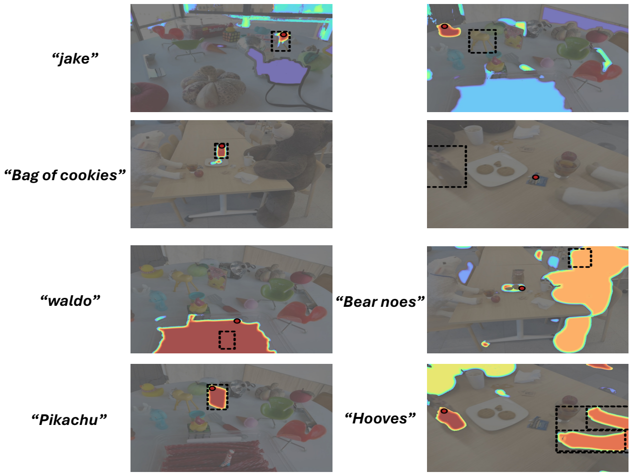


Figure 5. **Visualization of failure case.** The first two rows illustrate the problem of the CLIP feature encoding the same object differently due to a dramatic change in camera view. last two rows show the limitations of mapping CLIP features object-wise. you can see that for text queries more granular than the object level obtained in the pre processing step, it is not able to refine so that capture the entire object.

5. Conclusion

To relieve the complex pipeline of LangSplat, we proposed a way to handle language at the object level. not putting language features directly into Gaussians, we instead link object features with languages, which can improve LangSplat's complex training process and significantly reduce evaluation time. also, as we do not compress language features, we can leverage downstream task by selecting specific objects and link them with language feature space. but our solution has some limitations as we describe in Sec 4.3. SAM's view inconsistency can be solved to some extent by using object aware video tracker, but in order to connect 3D objects with text well, we need to solve CLIP's multi view inconsistency problem too. also, wrapping segments in object level causes a loss of granularity, which need a way to string segments based on the context of the text. we hope to solve these issues in future works.

References

[1] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. *arXiv preprint arXiv:2306.04633*, 2023. 1, 2

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1

[3] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21401–21412, 2024. 1

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 1

[5] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 1, 2

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 2

[7] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 1

[8] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 1

[9] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 1

[10] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2

[11] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 1, 2, 3

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[13] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 1

[14] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 International Conference on 3D Vision (3DV)*, pages 443–453. IEEE, 2022. 1

[15] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 1

[16] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 1

[17] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. *arXiv preprint arXiv:2312.00732*, 2023. 2, 3

[18] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529*, 2023. 1

[19] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 2