OpenInsGaussian: Open-vocabulary Instance Gaussian Segmentation with Context-aware Cross-view Fusion

Supplementary Material

A. 3D Class-agnostic Segmentation

A.1. Related Work

SA3D [2] uses user-provided prompts, such as points or bounding boxs, to generate segmentation masks in reference views, which are then used to train a neural field for object segmentation. Similarly, Spin-NeRF [12] employs a video-based segmenter [1] to generate multi-view masks. GARField [8] addresses inconsistencies in SAM-generated masks across different views by incorporating a scale-conditioned feature field. OmniSeg3D [16] introduces hierarchical contrastive learning to refine 2D SAM masks into a feature field, achieving fine-grained segmentation through adaptive cosine similarity thresholds. However, above methods rely on NeRF-based structures, which impose high computational costs during rendering, limiting their real-time applicability.

A.2. Contrastive Feature Learning

After obtaining segmentation masks from SAM and the corresponding mask-level language embeddings $\{F_t \mid t = 1, \ldots, T\}$, $\{M_t \mid t = 1, \ldots, T\}$, we learn class-agnostic instance features by modeling the relationship between 3D points and 2D pixels. For simplicity, we denote the fused context-aware language feature $F_{\text{fuse}}(t)$ as F_t in the following discussion. Following OpenGaussian and other class agnostic segmentation work [3, 8, 15, 15, 16], we train instance features for 3D Gaussians using segmentation masks.

Each Gaussian is assigned a low-dimensional instance feature $f \in \mathbb{R}^6$. To enforce multi-view consistency, we apply contrastive learning, bringing Gaussians within the same mask instance closer while pushing those from different instances apart. The instance feature map $M^f \in \mathbb{R}^{6\times H\times W}$ is obtained via alpha-blending, with binary masks B_i defining object instances:

$$\{B_0, B_1, \ldots, B_i\} = \{\mathbb{I}(M = i) \mid i \in M_t\}, \tag{1}$$

To ensure feature consistency within an instance, we compute the mean feature within each mask:

$$\bar{M}_i^f = \frac{B_i \cdot M^f}{\sum B_i} \in \mathbb{R}^6. \tag{2}$$

The intra-mask smoothing loss encourages all pixels within an instance to align with their mean feature:

$$\mathcal{L}_s = \sum_{i=1}^{m}\sum_{h=1}^{H}\sum_{w=1}^{W} B_{i,h,w} \cdot \left\| M^f_{:,h,w} - \bar{M}_i^f \right\|^2. \tag{3}$$

To enhance feature distinctiveness across instances, we define the inter-mask contrastive loss:

$$\mathcal{L}_c = \frac{1}{m(m+1)} \sum_{i=1}^{m}\sum_{j=1, j\neq i}^{m} \frac{1}{\left\| \bar{M}_i^f - \bar{M}_j^f \right\|^2}, \tag{4}$$

where m is the number of masks, and $\bar{M}_i^f, \bar{M}_j^f$ are mean features of different instances.

These losses ensure cross-view consistency for the same object while maintaining feature distinctiveness across different objects.

A.3. Two-Level Codebook Feature Discretization

After training instance features on 3D Gaussians, we apply a two-level coarse-to-fine clustering [15] to segment objects.

At the coarse level, we cluster Gaussians using both 3D coordinates $X \in \mathbb{R}^{n\times 3}$ and instance features $f \in \mathbb{R}^{n\times 6}$, ensuring spatially aware segmentation:

$$\begin{aligned} f \in \mathbb{R}^{n\times 6}, X \in \mathbb{R}^{n\times 3} \rightarrow \{C_{\text{coarse}} \in \mathbb{R}^{k_1\times(6+3)}, \\ I_{\text{coarse}} \in \{1, \ldots, k_1\}^n\}, \quad k_1 = 64. \end{aligned} \tag{5}$$

At the fine level, we further refine clusters using only instance features:

$$\begin{aligned} f \in \mathbb{R}^{n\times 6} \rightarrow \{C_{\text{fine}} \in \mathbb{R}^{(k_1\times k_2)\times 6}, \\ I_{\text{fine}} \in \{1, \ldots, k_2\}^n\}, \quad k_2 = 10. \end{aligned} \tag{6}$$

where $\{\text{C}, \text{I}\}$ means quantized features and cluster indices at each level of codebook.

We use K-means clustering [11] at both stages, with k_1 clusters at the coarse stage and $k_1 \times k_2$ clusters at the fine stage. This hierarchical approach preserves geometric integrity, ensuring that spatially unrelated objects are not grouped together.

During instance feature learning, supervision is limited to binary SAM masks. In the codebook construction stage, clustered instance features act as pseudo ground truth, replacing mask-based losses. The new training objective minimizes the difference between rendered pseudo-ground-truth features M^p and quantized features M^c:

$$\mathcal{L}_p = \|M^p - M^c\|_1, \tag{7}$$

This process refines instance segmentation while maintaining feature consistency and geometric structure in the 3D Gaussian representation.

A.4. Instance-Level 3D-2D Association

To establish a robust link between 3D Gaussian instances and multi-view 2D masks, we adopt an instance-level 3D-2D association strategy inspired by OpenGaussian [15]. Unlike prior methods that require additional networks for compressing language features or depth-based occlusion testing, our approach retains high-dimensional, lossless linguistic features while ensuring reliable associations.

Specifically, given a set of 3D clusters obtained from the discretization process (Sec. A.3), we render each 3D instance to individual views, obtaining single-instance maps $M^i \in \mathbb{R}^{6\times H\times W}$. These maps are compared with SAM-generated 2D masks $B^j \in \{0,1\}^{1\times H\times W}$ using an Intersection over Union (IoU) criterion. The SAM mask with the highest IoU is initially assigned to the corresponding 3D instance. However, to address occlusion-induced ambiguities, we further refine the association by incorporating feature similarity.

Instead of relying on depth information for occlusion testing, we populate the boolean SAM mask B^j with pseudo-ground truth features, forming a feature-filled mask $P^j \in \mathbb{R}^{6\times H\times W}$. We then compute a unified association score:

$$S_{ij} = \text{IoU}(\pi(M^i), B^j) \cdot (1 - \|M^i - P^j\|_1), \quad (8)$$

where $\pi(\cdot)$ denotes a binarization operation, ensuring IoU alignment, while the second term penalizes large feature discrepancies. The mask with the highest score is then associated with the 3D instance, allowing us to bind multi-view CLIP features effectively to 3D Gaussian objects.

By integrating both geometric alignment and semantic consistency, our method ensures precise and robust language embedding associations across multiple views.

B. Implementation Details

B.1. SAM and Clip Backbone

At the preprocessing stage, we utilize SAM-LangSplat, which is a modified version of SAM [9] for LangSplat [13] that automatically generates three levels of masks: whole, part, and sub-part. We select level 3 SAM masks (whole) [9] and use the ViT-H SAM model checkpoint for segmentation.

For feature extraction, we adopt Convolutional CLIP [10], a CNN-based variant of CLIP that empirically demonstrates better generalization than ViT-based CLIP [5] when handling large input resolutions [17] and better intermediate feature for our global feature extraction. Since competing methods use the ViT-B/16 checkpoint, we select the ConvNeXt-Base checkpoint, which has a comparable ImageNet zero-shot accuracy, ensuring a fair comparison of 2D backbone architectures.

Table 1. Ablation study on ScanNet with different feature aggregation weights. Metrics are reported as mIoU and mAcc for 10, 15, and 19 class settings.

Context Feature Weight α	mIoU (10, 15, 19)	mAcc (10, 15, 19)
0	0.42, 0.33, 0.33	0.60, 0.50, 0.49
0.2	0.51, 0.38, 0.38	0.69, 0.55, 0.54
0.4	0.51, 0.39, 0.38	0.68, 0.56, 0.55
0.6	0.50, 0.38, 0.38	0.68, 0.56, 0.54
0.8	0.47, 0.36, 0.35	0.65, 0.54, 0.52
1	0.50, 0.37, 0.37	0.68, 0.55, 0.53

B.2. Training Strategy

We follow OpenGaussian [15] general training settings. For the ScanNet dataset [4], that keep point cloud coordinates fixed and disable 3D Gaussian Splatting (3DGS) densification [6]. For the LeRF dataset [7], we optimize point cloud coordinates and enable 3DGS densification, which is stopped after 10k training steps.

B.3. Training Time

All experiments are conducted on a single NVIDIA RTX 4090 GPU (24GB). For the LeRF dataset, each scene consists of approximately 200 images and requires around 60 minutes for training. For the ScanNet dataset, scenes contain 100–300 images, with an average training time of 30 minutes per scene.

B.4. ScanNet Dataset Setup

We align our Scannet Dataset test dataset with [15] on 10 randomly selected ScanNet scenes, specifically: `scene0000_00`, `scene0062_00`, `scene0070_00`, `scene0097_00`, `scene0140_00`, `scene0200_00`, `scene0347_00`, `scene0400_00`, `scene0590_00`, `scene0645_00`.

For text-based queries, we utilize 19 ScanNet-defined categories:

- **19 categories:** wall, floor, cabinet, bed, chair, sofa, table, door, window, bookshelf, picture, counter, desk, curtain, refrigerator, shower curtain, toilet, sink, bathtub
- **15 categories:** wall, floor, cabinet, bed, chair, sofa, table, door, window, bookshelf, counter, desk, curtain, toilet, sink
- **10 categories:** wall, floor, bed, chair, sofa, table, door, window, bookshelf, toilet

Training images are downsampled by a factor of 2, and we use the cleaned point cloud that is processed by OpenGaussian.

B.5. Addional Results

We have conducted an ablation study (Tab. 1) to evaluate the impact of different weighting strategies on performance. This analysis demonstrates the robustness of our approach and highlights the sensitivity of the final segmentation quality to

Table 2. Per-scene performance of 3D point cloud semantic segmentation on the ScanNet dataset based on text query at different class splits (10 / 15 / 19 classes).

Scene ID	10-class		15-class		19-class	
	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑
scene0000_00	0.4744	0.7469	0.4054	0.6208	0.4230	0.6149
scene0062_00	0.4103	0.6476	0.2907	0.5372	0.2923	0.5372
scene0070_00	0.5227	0.6100	0.3899	0.4497	0.3498	0.4086
scene0097_00	0.5607	0.7321	0.3419	0.5689	0.3620	0.5658
scene0140_00	0.5781	0.7134	0.3422	0.4249	0.2985	0.3718
scene0200_00	0.4767	0.6554	0.4336	0.5452	0.4341	0.5452
scene0347_00	0.5587	0.6516	0.4018	0.5599	0.4467	0.5926
scene0400_00	0.5169	0.6865	0.4066	0.5902	0.4067	0.5902
scene0590_00	0.6052	0.7445	0.4517	0.6253	0.3952	0.5609
scene0645_00	0.4388	0.7269	0.3502	0.6075	0.3416	0.6509
Mean	**0.5142**	**0.6915**	**0.3814**	**0.5530**	**0.3750**	**0.5438**

Table 3. Per-scene performance of open vocabulary 3D object selection and rendering on Lerf dataset with mIoU and mAcc at different thresholds.

Scene	mIoU ↑	mAcc@0.25 ↑	mAcc@0.5 ↑
figurines	0.5375	0.7679	0.5893
ramen	0.2638	0.4366	0.1972
teatime	0.5855	0.7797	0.6441
waldo_kitchen	0.3178	0.5000	0.4091
Mean	**0.4262**	**0.6211**	**0.4599**

the fusion ratio. We also provide per scene evaluation results on Scannet (Tab. 2) and Lerf dataset (Tab. 3).

B.6. Efficiency

Regarding inference time, storage memory, feature extraction time, and training memory cost aspects: previous methods like LangSplat[13] and LEGaussians[14] perform text-query localization by rendering a 2D compressed language embedding map, which is then decoded to match the text query embeddings—this process is slow. In contrast, our approach and OpenGaussian[15] can directly localize text queries in 3D by searching the codebook. Additionally, both LangSplat[13] and LEGaussians[14] need to maintain the autoencoder decoder network, which requires larger storage memory. Since our context feature extraction only requires one additional forward pass of CLIP, we do not introduce significant additional computation. In our Attention-Driven Feature Aggregation module, we reuse preloaded multi-view features, incurring no extra memory cost. All methods are compatible with the 4090 GPU. Moreover, in previous methods, the feature extraction process requires running the CLIP model on each segmentation crop at each granularity level, which is very time-consuming. Our context-aware feature extraction module can be downgraded to solely global feature extraction, which significantly improves efficiency while sacrificing very little accuracy, as shown in our ablation results. Therefore, our proposed method can achieve substantial improvements without a decrease in efficiency.

References

[1] 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 (ICCV)*, pages 9650–9660, 2021. 1

[2] Jingwen Cen, Ziyao Zhou, Jiemin Fang, Wenguan Shen, Lingxi Xie, Dongdong Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[3] Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. Click-gaussian: Interactive segmentation to any 3d gaussians. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024. 1

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 2

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance

field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 2

[7] 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 (ICCV)*, pages 19729–19739, 2023. 2

[8] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21530–21539, 2024. 1

[9] 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. 2

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[11] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 1

[12] Arian Mirzaei, Tovi Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, and Anton Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20669–20679, 2023. 1

[13] 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. 2, 3

[14] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024. 3

[15] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2025. 1, 2, 3

[16] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024. 1

[17] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems*, 36:32215–32234, 2023. 2