ExtrinSplat: Decoupling Geometry and Semantics for Open-Vocabulary Understanding in 3D Gaussian Splatting

Supplementary Material

6. Implementation Details

6.1. Model Implementation Details

Data Preparation. Initially, we employ SAM with grid-based point prompting to acquire initial static object masks at varying granularities from the first input frame, I_0. Subsequently, these masks extracted from I_0 are utilized by the DAM2SAM [29] model to track the corresponding objects throughout the entire image sequence.

To ensure all objects appearing throughout the sequence are captured, we introduce a periodic new-object detection mechanism. This check is performed at a fixed interval of $\Delta t = 10$ frames. At each check, we first compute the total area of all tracked masks in the current frame, A_t. We then trigger a full re-segmentation on this frame using SAM to get a candidate mask area, A_{cand}. A potential new object event is flagged if the ratio A_t/A_{cand} falls below a threshold $\tau_{area} = 0.9$. When triggered, we identify a mask from the candidate set as a "new" object if its maximum Intersection over Union (IoU) with any existing tracked mask is below a threshold of $\tau_{iou} = 0.6$. Once identified, these new objects are added to the tracking pool and propagated by DAM2SAM henceforth.

Existing research [18] suggests that such a detection mechanism can introduce two potential drawbacks: (1) tracking failures for some objects, resulting in incomplete object tracks, and (2) re-appearing objects being misidentified as new after their tracking has been lost, leading to a single object being assigned multiple instance IDs. Our model, however, does not need to overcome these issues during the data preparation stage.

Regarding the first issue, we simply discard views with empty masks (i.e., where object tracking has failed) during our object-level grouping stage. As demonstrated in Appendix 7.2, our model achieves robust performance even with a reduced number of views per object. Consequently, this issue has a negligible impact on the overall model accuracy.

Regarding the second issue, the emergence of multiple instances for a single object is handled by our matching process. The matching between open-vocabulary queries and instance point clusters is a one-to-many operation based on similarity. In the event of multiple matches, we take the union of their results as the final output. Therefore, the presence of multiple instances for the same object does not degrade the final matching accuracy.

In summary, our model imposes minimal requirements on the data preparation stage and functions effectively even with partial mask information for each object. This demonstrates the robustness of our approach to imperfections in the input data.

Object-Level Grouping. The object-level grouping process is accomplished within a single forward rendering pass. In our implementation, we simply accumulate the contribution weights of all participating 3D Gaussians during the forward pass of the 3D Gaussian Splatting render. Throughout this process, the contribution weight of each Gaussian is naturally aggregated, obviating the need for auxiliary data structures or redundant computations. By leveraging the highly optimized volumetric projection inherent to 3D Gaussian Splatting, our method achieves exceptional computational efficiency while maintaining semantic coherence. For the subsequent neutral point processing, we use fixed thresholds across all experiments to ensure robustness and consistency. The semantic entropy threshold is set to $\tau_h = 0.9$, and the opacity threshold for filtering is set to $\tau_\alpha = 0.1$. A detailed sensitivity analysis for these hyperparameters is provided in Appendix 7.2.

Instance Feature Extraction. We acquire features for each object instance as follows. First, we identify the three largest masks for the instance based on pixel area. For each selected mask, we highlight the corresponding object on the original image with a green bounding box, creating three distinct input images. These images are then processed by a VLM, which generates a set of five nouns that describe the instance.

To match an instance against a user's text query, we compute the cosine similarity between the CLIP feature embedding of the query and the CLIP embeddings of the five nouns associated with that instance. This design allows a single query to potentially match multiple instances. A match is deemed successful if the similarity score for *any* of an instance's five candidate nouns surpasses a predefined threshold of $\eta = 0.9$.

The specific prompt template used to elicit these nouns from the VLM is defined as follows:

"In the images, identify the object that is enclosed by a bright green outline. Provide five distinct and appropriate nouns to describe ONLY that specific object. Return ONLY the five nouns separated by slashes (e.g., car/automobile/vehicle/motorcar/transport). Do not add any other explanatory text, titles, or formatting."

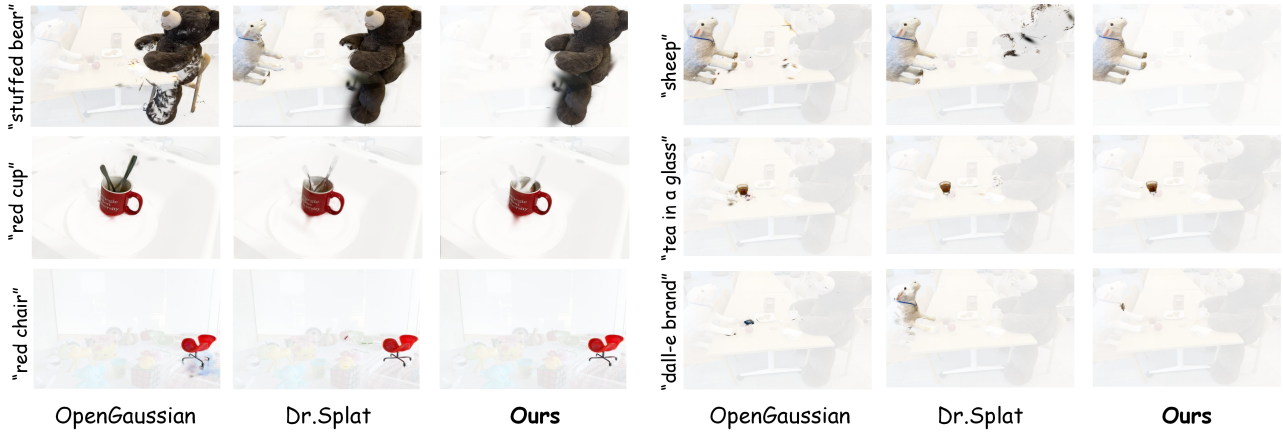

Figure 5. Additional qualitative results for open-vocabulary object selection on the LERF dataset.

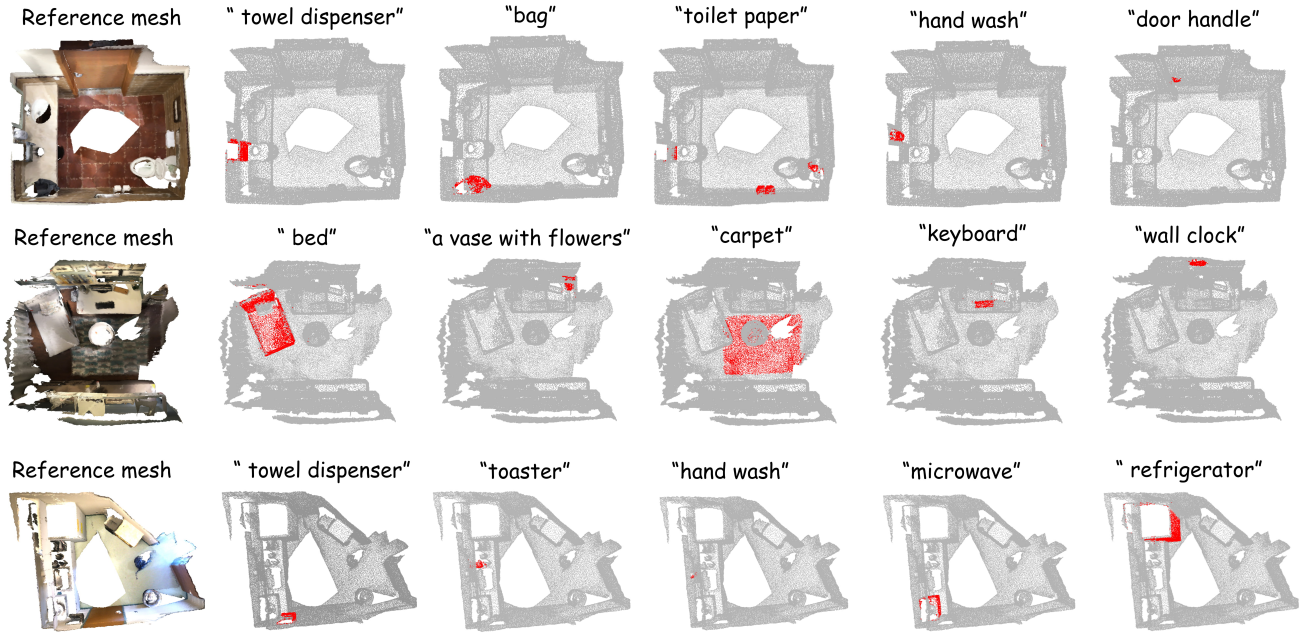

Figure 6. Additional qualitative results for open-vocabulary 3D semantic segmentation on the ScanNet dataset.

6.2. Evaluation Details

LERF Dataset Evaluation We evaluate our model on the LERF dataset, using annotations from LangSplat. Due to the absence of 3D ground truth, we follow the 2D-based evaluation protocol from OpenGaussian. This protocol measures 3D understanding by computing the multi-view IoU accuracy between rendered occupancy masks from our selected 3D Gaussians and the ground-truth masks, which were manually annotated and provided by OpenGaussian for a set of text queries.

ScanNet Dataset Evaluation For evaluation on the ScanNet dataset, we select the same 10 scenes as used in OpenGaussian: scene0000_00, scene0062_00, scene0070_00, scene0097_00, scene0140_00, scene0200_00, scene0347_00, scene0400_00, scene0590_00, and scene0645_00. The 19 categories defined by ScanNet used for text queries are: wall, floor, cabinet, bed, chair, sofa, table, door, window, bookshelf, picture, counter, desk, curtain, refrigerator, shower curtain, toilet, sink, and bathtub. 15 categories are without picture, refrigerator, shower curtain, bathtub; 10 categories are further without cabinet, counter, desk, curtain, sink.

7. Additional Experimental Results

7.1. Additional Qualitative Results

Fig. 5 presents additional qualitative results for the task of object selection in 3D space on the LERF dataset. Fig. 6 showcases more results of our model on the open-vocabulary 3D semantic segmentation task on the ScanNet dataset. These results were not included in the main manuscript due to space limitations. Consistent with our previous observations, both OpenGaussian and InstanceGaussian exhibit limitations in handling object boundaries and in fine-grained semantic understanding. In contrast, our model yields results with significantly sharper and more accurate semantic interpretations.

7.2. Additional Ablation Studies

Scene Understanding with Limited Mask Supervision. Our method leverages a mask-matching mechanism for semantic understanding, a characteristic that enables it to perform 3D segmentation from only a sparse set of 2D masks. To validate this capability, we conduct experiments using progressively sparser subsets of 2D masks (corresponding to $1/2$, $1/4$, $1/8$, $1/16$, and $1/32$ of the total available views), while all other model settings are held constant. Finally, we perform an open-vocabulary 3D object extraction task and qualitatively evaluate the results. As illustrated in Fig. 7, our method exhibits high robustness to the number of provided masks. Even with masks from only $1/8$ of the views, it maintains high-quality segmentation. This demonstrates our model's high data efficiency and its ability to generalize from sparse supervision. However, when the number of masks becomes excessively sparse, such as at $1/16$ or $1/32$, a portion of the 3D Gaussians may not be observed by any masked camera view. This lack of supervision results in noticeable artifacts. Notably, the $1/32$ subset often corresponds to merely 5–10 foreground masks. While these extreme cases produce artifacts, the ability to generate a coherent result from such minimal data underscores our method's low reliance on dense supervision and corroborates its strong generalization capabilities.

Ablation Study on Neutral Point Thresholds. On the LERF dataset, we investigate the influence of the entropy threshold τ_h and the opacity threshold τ_α in our two-stage neutral point processing module. The results of this sensitivity analysis are presented in Tab. 8. The baseline configuration, which bypasses entropy-based filtering by setting $\tau_h = 1.0$, achieves an mIoU of 53.0. A notable improvement is observed when τ_h is lowered to 0.9, underscoring the efficacy of pruning points with high semantic ambiguity. The necessity of the subsequent opacity-based filtering is also validated. With $\tau_h = 0.9$, setting $\tau_\alpha = 0$ removes all high-entropy points indiscriminately and degrades performance to 53.2 mIoU. This suggests that high-entropy points with high opacity are geometrically significant and should be retained. Peak performance is achieved at $(\tau_h, \tau_\alpha) = (0.9, 0.1)$. This configuration strikes a favorable trade-off between removing ambiguous transitional points and preserving geometrically salient structures. While the model demonstrates reasonable robustness to other settings, further reductions in τ_h to 0.8 or 0.5 yield diminished returns, likely due to the erroneous exclusion of valid surface points. Based on these findings, we adopt $\tau_h = 0.9$ and $\tau_\alpha = 0.1$ for all main experiments.

Instance Feature Extraction. The core of our instance feature extraction module is a VLM that grounds textual queries to 3D visual features. The representational capacity of the VLM is therefore a critical determinant of performance. To investigate this dependency, we ablate the VLM component with three different pre-trained models on the LERF dataset: SenseNova 6.5 Pro, InternVL3-78B [3], and Gemini 2.5 Pro [5] . The results, presented in Tab. 6, reveal a strong positive correlation between the representational power of the VLM and final segmentation accuracy. More specifically, employing VLMs known for more robust vision-language grounding consistently yields substantial gains in mIoU. This indicates that the quality of the semantic features provided by the VLM is a critical determinant of performance in this task. Therefore, the performance ceiling of our model is not static; it is set to rise in tandem with the ongoing evolution of Vision-Language Models.

We further analyze the method's sensitivity to the number of descriptive text prompts used for instance matching on the LERF dataset. As shown in Tab. 7, the relationship between prompt quantity and segmentation accuracy is non-monotonic. Starting from a single prompt, performance improves as the number of descriptors increases to five. This suggests that a richer set of semantic cues helps the VLM disambiguate instances, particularly for concepts too nuanced to be captured by a single term. However, increasing to 10 prompts leads to performance degradation. We hypothesize that an excessive number of prompts may introduce semantic noise or redundant information, thereby interfering with the VLM's feature matching process. Consequently, we use five descriptive prompts, as this configuration strikes a favorable balance between semantic richness and feature ambiguity.

7.3. Open-Vocabulary 3D Object Editing

Our method enables open-vocabulary editing of objects in 3DGS scenes by mapping a language query to corresponding instance IDs and then applying targeted manipulations. Fig. 8 demonstrates the scene editing capabilities of our method. Starting from an original scene reconstructed via 3DGS, we can select an object to perform operations such as **removal** (Fig 8(a)), **translation** (Fig. 8(b)), or **stylization** (Fig. 8(c)).

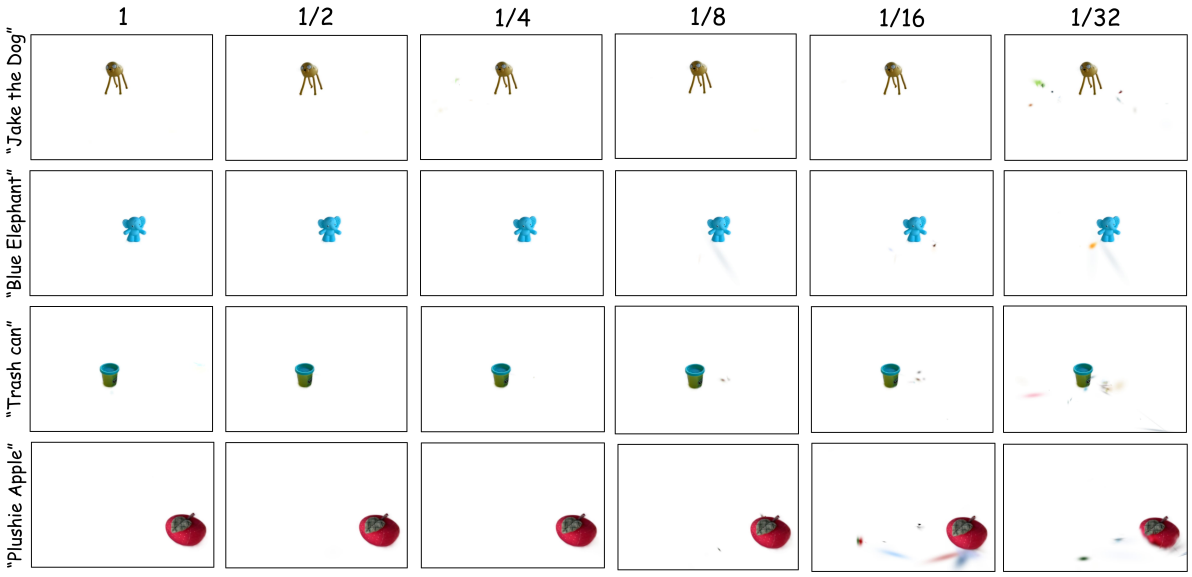

Figure 7. Open-Vocabulary 3D Object Extraction from Sparse Masks. We perform an open-vocabulary 3D object extraction task on the `figurines` scene from the LERF dataset, providing a progressively smaller subset of 2D masks as supervision. The results demonstrate that our model's accuracy experiences negligible degradation when using $\geq 1/4$ of the total masks. With only $1/8$ of the masks, it still exhibits a strong capability to capture the object's geometry. Even in the extreme case with as few as $1/32$ of the masks, our model can still recover the object's coarse shape.

Table 6. Ablation on the choice of VLM.

Model	mIoU↑
SenseNova 6.5 Pro	47.0
InternVL3-78B	50.2
Gemini 2.5 Pro	**54.3**

Table 7. Ablation on number of prompts.

Number of Prompts	mIoU↑
1	44.0
3	50.9
5	**54.3**
10	53.6

Table 8. Ablation on neutral point processing thresholds τ_h and τ_α.

τ_h	τ_α	mIoU↑
1.00	/	53.0
0.99	0.1	53.8
0.90	0.5	53.8
0.90	0.1	**54.3**
0.90	0.01	54.2
0.90	0.0	53.2
0.80	0.1	53.8
0.50	0.1	53.1

7.4. Open-Vocabulary Object Extraction in Complex and Real-World Scenes

To assess its practical applicability, we validate our method on a real-world scene. We captured an office environment using a standard mobile phone and tasked our model with open-vocabulary object extraction. The qualitative results, presented in Fig. 9, demonstrate that our model performs robustly on this in-the-wild data. This highlights the method's strong generalization capabilities and its potential for real-world applications.

To evaluate our model's comprehension capabilities in complex scenes, we conduct experiments on the Grasp-Net dataset [7]. This dataset is characterized by challenging object arrangements, including overlapping, adjacent, and contained instances. Despite the close proximity between instances, our model successfully distinguishes and segments them. As shown in Fig. 10, our method produces sharp, well-defined rendering boundaries, demonstrating its effectiveness in such challenging scenarios.

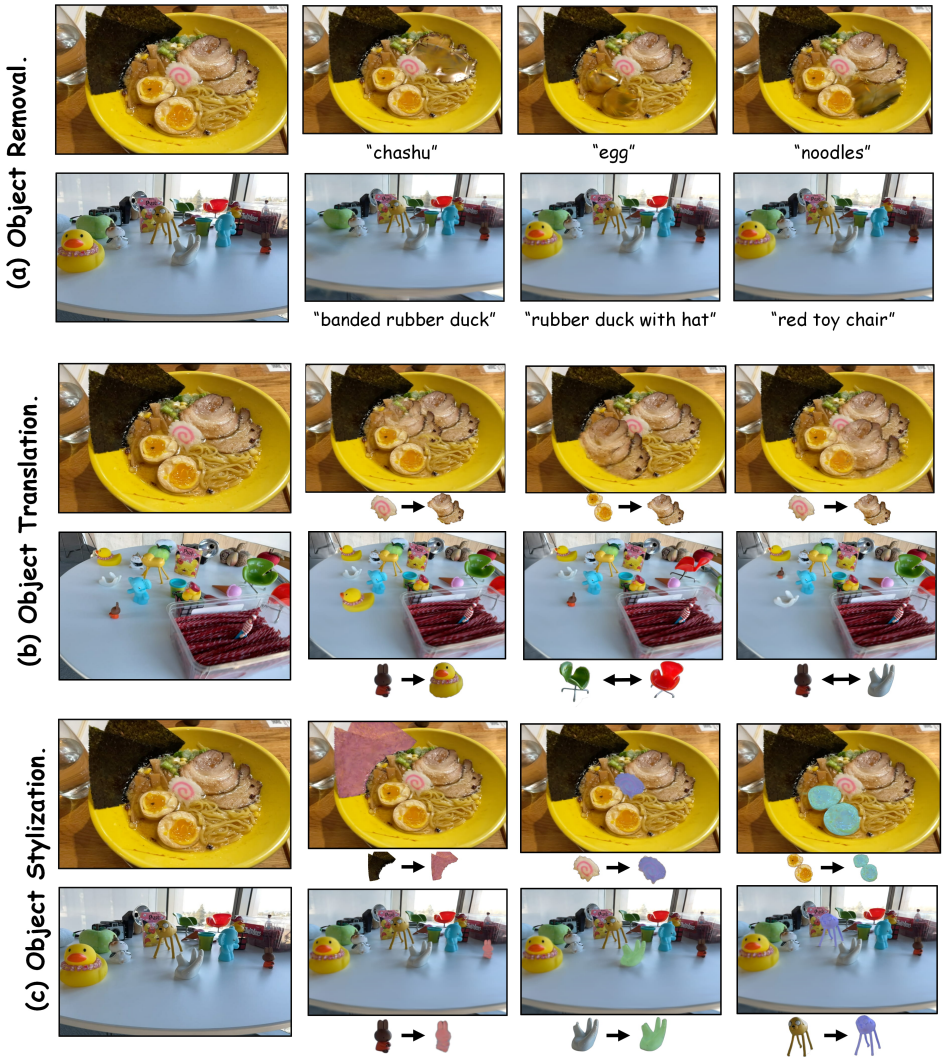

Figure 8. Demonstration of our scene editing capabilities. (a) Object Removal. (b) Object Translation. (c) Object Stylization. All manipulations are applied directly to the 3D scene rather than on the 2D rendered images.

8. Efficiency Analysis

To dissect our method's efficiency, we provide a detailed component-wise runtime breakdown in Tab. 9, based on the `teatime` scene in the LERF dataset, which contains 131 distinct instance categories. The total end-to-end processing time for this complex scene is approximately 9.25 minutes (555.14s), including all computational and I/O stages. The results clearly identify the primary computational bottlenecks, with three stages accounting for over 99% of the total computational workload: VLM Text Feature Acquisition (37.7%), Backward Matching (32.0%), and the initial Mask Acquisition (29.7%). The analysis also highlights the efficiency of the neutral point processing module, which constitutes only 0.1% of the total computational cost. This low figure indicates that the boundary refinement step is achieved with minimal performance overhead.

Notably, despite the aforementioned bottlenecks, our method's runtime holds a significant advantage over mainstream methods, which require hours of processing. For instance, in our evaluation on the LERF dataset, we found that InstanceGaussian [14] requires approximately 140 minutes for the 3D Gaussian training phase alone. Furthermore, our model offers potential for even greater speed. In principle, it processes each category independently, allowing for significant acceleration through parallelization. However, as a key design goal is to ensure deployability on consumer-grade hardware, this imposes a constraint on the model's total memory footprint. Consequently, we did not pursue further parallelization in the current implementation.

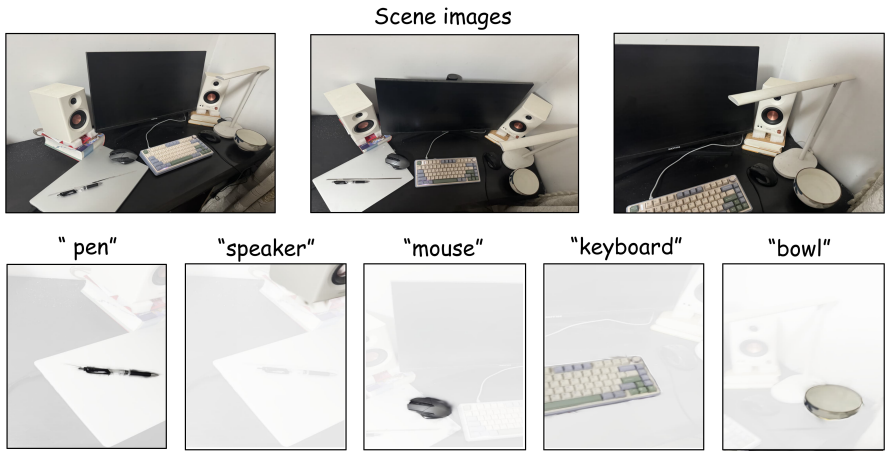

Figure 9. Qualitative results for the open-vocabulary object extraction task on a real-world scene captured with a mobile phone.

Figure 10. Qualitative results for the open-vocabulary object extraction task on the Grasp-Net dataset.

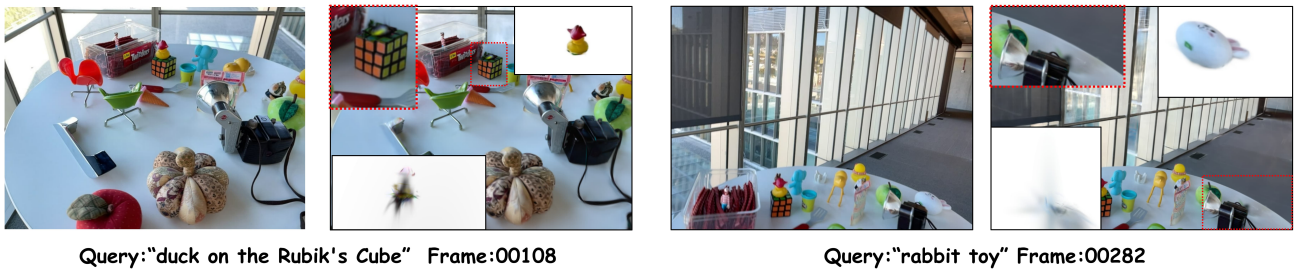

Figure 11. Qualitative results for rendering foreground, neutral, and background points on the `figurines` scene from the LERF dataset.

Table 9. Component-wise runtime breakdown for our method on the `teatime` scene in the LERF dataset. The analysis highlights that VLM inference and backward matching are the primary computational bottlenecks. All timings are in seconds, measured on a single NVIDIA Tesla V100 (32GB) GPU.

Component	Time (s)	Time / Cat. (s)	Compute %
Computational Stages			
Mask Acquisition	156.99	1.1984	29.7%
Backward Matching	169.23	1.2919	32.0%
Neutral Point Processing	0.54	0.0041	0.1%
Text Feature Acquisition	199.67	1.5242	37.7%
CLIP Feature Extraction	2.63	0.0201	0.5%
Total Computation	529.06	4.0386	100.0%
I/O Stages			
Data Loading	16.12	–	–
Saving Output	9.96	–	–
Grand Total (incl. I/O)	555.14	–	–

9. Analysis of Failure Cases

Impact of Mask Inaccuracy. Our method demonstrates considerable robustness to sporadic segmentation errors, provided that the initial masks generated by DAM2SAM are generally accurate. However, when these masks suffer from large-scale or frequent inaccuracies, our model can produce erroneous foreground-background distinctions during the backward weight accumulation process. This, in turn, adversely affects the final segmentation accuracy, as illustrated in a failure case in Fig. 12(a).

Mismatches from the VLM. Incorrect matching can also arise from the VLM itself, attributable to two primary sources, as shown in Fig. 12(b). First, ambiguous segmentation masks or challenging viewing angles in the input images can provide misleading guidance to the VLM. Second, inherent limitations in the VLM's comprehension capabilities can lead to incorrect judgments even with clear inputs. Either type of error can result in incorrect category assignments, ultimately causing the point clusters to be mismatched with the intended text query.

10. Discussion

10.1. Neutral Points

Prior work on so-called "boundary points" [15, 34] has primarily focused on refining their positions through dedicated training strategies to enhance semantic understanding. However, while repositioning these boundary points can improve semantic segmentation accuracy, it often compromises the realism and fidelity of the final rendering. This trade-off arises because boundary points include a special

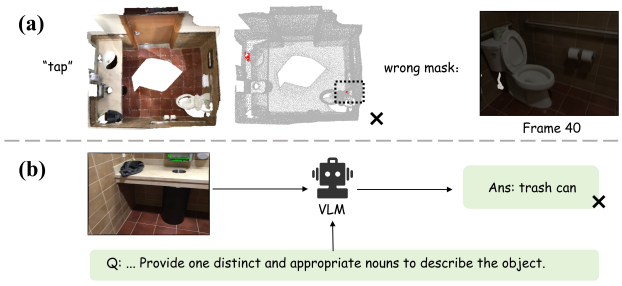

Figure 12. Examples of Failure Cases. (a) Inaccurate Masks: The segmentation model outputs incorrect 2D object masks. (b) VLM Misunderstanding: The VLM provides an incorrect object name for the given input images.

subset of points that belong neither to the foreground nor the background. These points serve as transitional elements that are crucial for ensuring rendering realism but lack specific semantic meaning. We term these as **neutral points**.

Neutral points are abundant in 3DGS scenes, making them non-negligible for semantic understanding. Nevertheless, accurately identifying and removing these neutral points in an unsupervised manner remains a significant challenge. In our implementation, precisely filtering out these points during the matching stage is difficult due to computational efficiency constraints. In Fig. 11, we present the visualization of neutral points from our model on the LERF dataset. Developing more effective methods to model and eliminate neutral points is a key direction for future improvement of our method.

10.2. Diversity of Semantic Categories

Prior work has noted that a single Gaussian point can belong to multiple semantic categories [22, 25, 26]. To verify this phenomenon, we conduct a statistical analysis of the semantic categories for all 3D Gaussian points within the `teatime` scene of the LERF dataset, as illustrated in Fig. 13. Our analysis reveals that approximately 25% of all visible 3D points exhibit multi-dimensional semantic attributes. In the context of our model, this means a substantial portion of 3D Gaussian points inherently possess multiple semantic labels simultaneously. For instance, a single point on a tree branch may belong to the categories of "branch", "tree", and "vegetation" all at once.This phenomenon is consistent with how humans perceive 3D environments.

This semantic diversity suggests that relying on a single semantic label is often insufficient to comprehensively describe the properties of a point. Therefore, this inherent polysemy must be fully considered when performing 3D semantic understanding.

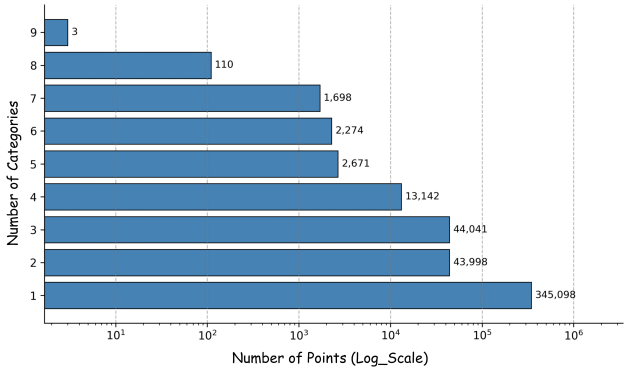

Figure 13. Category distribution of visible Gaussian points in the `teatime` scene from the LERF dataset.

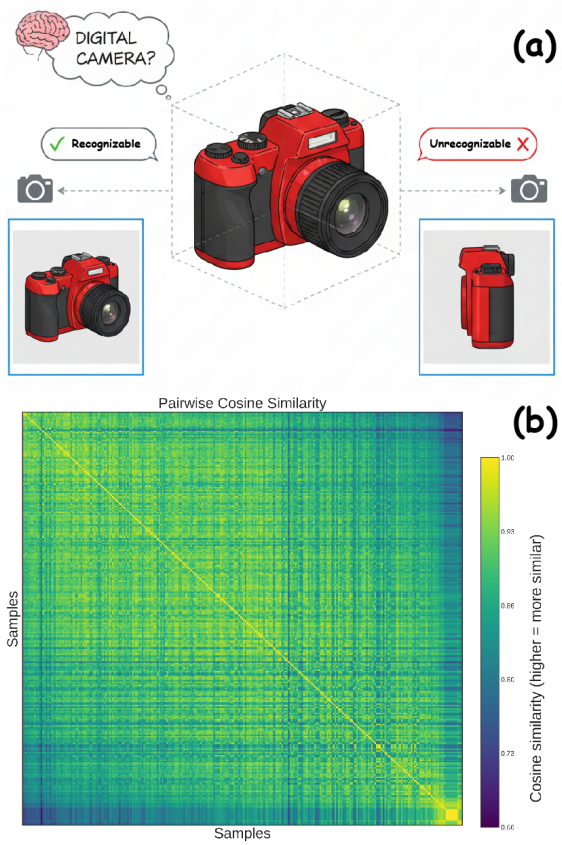

Figure 14. Illustration of the inconsistency of semantic features across different viewpoints. (a) The same object can present different semantic characteristics from different viewpoints. (b) Visualization of feature similarity for the "Jake the Dog" object in the `figurines` scene. The plot shows the cosine similarity scores between feature vectors from different views; a higher value (closer to 1) indicates that the features are more similar.

10.3. Inconsistency of Semantic Features

Our work diverges from the common practice in related literature of feeding masked object regions into a CLIP image encoder to obtain semantic features. This decision is based on the observation that for the same object, its semantic features can exhibit significant variations across different viewpoints [1]. As shown in Fig. 14(a), acquiring accurate CLIP image features becomes more challenging from certain angles. Due to the existence of such views, strategies like selecting the features from the view with the largest mask area or averaging the features across all views inevitably introduce errors.

To validate this phenomenon, we selected the "Jake the Dog" object from the `figurines` scene in the LERF dataset and extracted its CLIP image features from multiple viewpoints. A visualization of these features is presented in Fig. 14(b). The figure clearly shows that even for the same object, the semantic features vary noticeably with the observation angle. This feature inconsistency suggests that conventional strategies based on single-mask or averaged-mask feature extraction can lead to information loss, thereby degrading matching performance. In contrast, our VLM-based feature extraction approach alleviates this issue to a certain extent, enhancing the stability and robustness of the semantic representation.