OpenGaussian++ - Refining OpenGaussian: The 3D Point-Level Understanding Framework with Open Vocabulary

Anonymous CVPR submission

Paper ID *****

Abstract

OpenGaussian is limited by inconsistent per-view SAM segmentations and a fragile, single-view semantic association. We independently evaluate two solutions: multi-view mask refinement and a more effective multi-view feature fusion framework. On ScanNet, our visibility-weighted fusion model improves average mIoU from 0.299 to 0.318, producing cleaner object boundaries. This demonstrates that overcoming single-view dependencies with a robust, visibility-aware multi-view consensus is critical for reliable 3D semantic understanding.

1. Introduction

3D Gaussian Splatting (3DGS) [1] has become the leading neural rendering method for its speed and quality. OpenGaussian [5] extends 3DGS for view-independent 3D point-level understanding using cluster-based feature embedding with SAM masks and CLIP features.

We present three modifications to enhance OpenGaussian's semantic consistency and training efficiency: (1) Multi-view SAM mask refinement enforcing consistency across overlapping views, (2) Multi-view CLIP feature fusion that averages features across views where each Gaussian is visible, and (3) Visibility-weighted fusion using camera angles to weight view contributions.

2. Related Works

2.1. Neural Rendering

3DGS achieves superior performance over Neural Radiance Fields (NeRFs) [2] through rasterization-based rendering rather than slow volume rendering.

2.2. 3D Point-Level Understanding

LangSplat [3] and LEGaussian [4] assign compressed CLIP features directly to Gaussians, leading to information loss and 2D-to-3D mapping challenges. OpenGaussian's cluster-based approach with mask-level feature consistency provides more stable learning signals but trades point-wise granularity for distinct segmentation.

3. Method

OpenGaussian assigns semantic understanding to 3D Gaussians through three steps: (1) learning 6-dimensional feature maps for each Gaussian using SAM mask consistency, (2) clustering Gaussians by feature similarity, and (3) assigning CLIP values to clusters. We propose three modifications to enhance this framework.

3.0.1 Modification 1: Consistent SAM-masks

- **Problem:** SAM generates per-frame masks independently, causing geometric and semantic inconsistencies for the same object across different viewpoints. Object boundaries are imprecise, and many pixels remain unclassified.
- **Proposed solution:** We utilize multi-view consistency of Gaussian splats to refine SAM masks through a two-stage approach that synchronizes object IDs across views and expands segment coverage:
  1. **Cross-view ID synchronization for consistent object identification:**
     We address the problem that identical objects receive different segment IDs across camera views. For each Gaussian splat g_i with opacity $\alpha_i > \tau_{\text{opacity}}$ (we use $\tau_{\text{opacity}} = 0.8$), we render the individual splat in each camera view to obtain its 2D footprint and weight distribution. We apply depth-based visibility testing: a splat is considered visible its projection lies in image bounds and $|d_{\text{projected}} - d_{\text{rendered}}| < \tau_{\text{depth}}$ where $d_{\text{projected}}$ is the Euclidean distance from the image plane to the splat and d_{rendered} is the depth from the rendered depth map.
     For each visible footprint, we compute the most dominant segment ID by aggregating pixel IDs inside the footprint weighted by the opacity function of the pro-

jected splat: $\text{ID}_{\text{dominant}} = \arg\max_k \sum_{p \in \text{footprint}} w_p \cdot \mathbf{1}[\text{SAM}(p) = k]$, where w_p is the value of the opacity function evaluated at pixel p. Finally, we constrain object IDs across multiple views to the same new global identifier by applying it to all pixels belonging to the object mask.

2. **Mask expansion using weighted Gaussian footprints:**
To reduce regions classified as "void" and enhance the boundary quality between segments, we expand segmented regions with projected Gaussian splats. For each pixel in the image, we maintain a look-up table to accumulate values for different IDs. For each splat, we compute its underlying object ID and collect votes for the winning ID across multiple views. For those viewpoints where the ID of the projected splat matches the winning ID, we increase accumulated weights for this ID in all pixels belonging to the base segment. We then apply additional splat expansion for the splat pixels that don't belong to the object segment - extension_mask $= (w_p > 0) \wedge \neg(\text{segment\_mask}_k)$, updating $\mathbf{V}^{(c)}_{p,k} += w_p$. For the viewpoints where the computed ID of the projected splat doesn't match the winning ID, we introduce the multi-view constraint. Instead of accumulating the weights for the ID that belongs to the splat in the viewpoint over the base object segment, we replace its ID with the winning ID and then accumulate the weights in the same way as above. By doing so, we indicate that the splat resides on the edge between different segments and can therefore be classified differently when rendered in different viewpoints. Therefore, we aim to invert the direction of expansion of the misclassified splat projection according to the direction seen in the majority of other viewpoints.

Due to efficient mask usage that is highly parallelizable on the GPU, the additional SAM masks refinement pipeline introduces relatively small overhead compared to the overall time needed for OpenGaussian to process a scene.

3.1. Two-Level Code Book Discretization

In this step, the goal is to cluster Gaussians with similar features. Initially, the 6-dimensional feature embedding is concatenated with the x, y, and z coordinates to create a 9-dimensional feature embedding. The underlying rationale for this design choice is that Gaussians in proximity within Euclidean space are likely to exhibit semantic similarity. The codebook is then initialized by randomly selecting feature embeddings from $k = 64$ distinct Gaussians, forming $\mathbf{C} \in \mathbb{R}^{k \times 9}$. Subsequently, each Gaussian is assigned to a cluster based on Euclidean distance. Following cluster assignment, the Gaussian-specific features undergo further refinement with a modified optimization objective. Specifically, the optimization aims to minimize the intra-cluster variance by drawing feature embeddings within each cluster toward their respective cluster centroid. This is achieved by minimizing the following loss:

$$\mathcal{L}_p = \|M_p - M_c\|_1 \tag{1}$$

In this case, the maps M_p and M_c do not contain the x,y, and z coordinates of the Gaussians. They are strictly used for the original clustering and are dropped once assigned. There now exists a one-level code book that discretizes the Gaussians. Now, each cluster in the code book is split up into $k = 10$ clusters, and the same process is repeated. For this fine level of clustering, the x,y, and z coordinate for each Gaussian are not used for the clustering.

3.2. Instance-Level 2D-3D Association without Depth Test

The culmination of our 3D instance learning pipeline is the endowment of each geometric cluster with rich semantic meaning. We achieve this by associating each 3D cluster with a high-dimensional language feature vector from a pre-trained Vision-Language Model (VLM), specifically CLIP. This critical step bridges the gap between abstract 3D geometry and open-vocabulary language understanding. The process requires establishing a robust and accurate mapping between the 3D world, as represented by our instance clusters, and the 2D observations from which semantic features are derived. In the following sections, we describe the baseline single-view association method and present our more robust multi-view fusion framework.

3.2.1 Original Method: Single-View Association

The baseline method employs a fragile "winner-takes-all" strategy. It associates each 3D cluster $\mathcal{C}_i$ with the CLIP feature from a single best-matching 2D SAM mask found across all rendered views, discarding all other data. This reliance on a single observation makes the process highly susceptible to errors from occlusion or poor viewing angles. By ignoring the rich, consensus-building information available in other views, this method leads to unstable and noisy semantic assignments.

3.2.2 Modification 2 & 3: Aggregation of CLIP-features

To overcome the inherent fragility of the single-view approach, we introduce a multi-view fusion framework designed for robustness and stability. Our paradigm shifts from finding the single best view to building a reliable semantic consensus from multiple high-quality observations. The framework consists of a shared initial association stage followed by two distinct feature aggregation strategies.

View-wise Association and Filtering. The foundational step of our framework is to first evaluate the quality of association for a 3D cluster $\mathcal{C}_i$ in *every* visible view v. For each view, we find the best possible match among the local 2D SAM masks $M_{v,j}$. The quality of this match is quantified by a holistic joint score $S_{i,v,j}$ that considers both geometric and feature-space alignment:

$$S_{i,v,j} = \text{IoU}(\pi(\mathcal{C}_i, v), M_{v,j}) \times \underbrace{(1 - d_{\text{L1}}(\mathbf{f}_{\mathcal{C}_i,v}, \mathbf{f}_{M_{v,j}}))}_{\text{Feature Similarity}}, \tag{2}$$

where:

- $\pi(\mathcal{C}_i, v)$ is the 2D silhouette obtained by rendering the 3D cluster $\mathcal{C}_i$ from the perspective of camera v.
- $\mathbf{f}_{\mathcal{C}_i,v}$ and $\mathbf{f}_{M_{v,j}}$ are the mean instance features of the rendered cluster and the 2D mask, respectively.
- The IoU term ensures strong geometric alignment, while the feature similarity term (scaled to $[0, 1]$) penalizes feature-space dissimilarity. A perfect feature match yields a similarity of 1, while diverging features drive it towards 0.

For each view v, we determine the best local match by finding $S_{i,v} = \max_j S_{i,v,j}$.

This process yields a set of candidate associations, one for each view. As a crucial quality control step, we filter this set by retaining only the associations whose score $S_{i,v}$ exceeds a confidence threshold τ (e.g., $\tau = 0.2$). This filtering discards low-confidence matches that could arise from spurious overlaps or feature mismatches. The result of this stage is a curated set of high-quality views $\mathcal{V}_i^*$ for each 3D cluster, along with their corresponding best-match CLIP features $\{\mathbf{f}_{i,v}^{\text{CLIP}}\}_{v \in \mathcal{V}_i^*}$.

Feature Aggregation Strategies. Having curated a set of high-confidence 2D semantic observations, we aggregate them to produce a single, robust feature vector for the 3D cluster. We propose two strategies for this final step.

- **Modification 2: Democratic (Unweighted) Aggregation.** This strategy is founded on the principle that every validated viewpoint should have an equal say in determining the final semantic identity. The definitive feature $\mathbf{F}_i$ is computed as the arithmetic mean of the CLIP features from all views in the high-quality set $\mathcal{V}_i^*$:

$$\mathbf{F}_i = \frac{1}{|\mathcal{V}_i^*|} \sum_{v \in \mathcal{V}_i^*} \mathbf{f}_{i,v}^{\text{CLIP}}. \tag{3}$$

  By averaging, this method effectively mitigates the impact of random, uncorrelated noise present in individual views. The normalization by $|\mathcal{V}_i^*|$ ensures that the magnitude of the final feature vector is not dependent on the number of available high-quality views, promoting stability across different instances. Its principal strength lies in its simplicity and effectiveness at noise reduction.

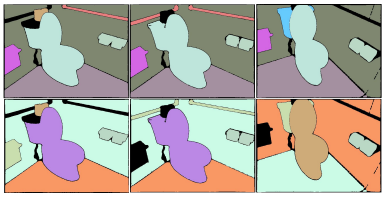

Figure 1. SAM mask consistency before (bottom) and after (top) our cross-view ID synchronization.

- **Modification 3: Confidence-and-Visibility-Weighted Aggregation.** This more refined strategy is founded on the intuition that not all "good" views are equally "great". A view where an object is large, central, and unoccluded should contribute more to the final identity than a view where it is small, peripheral, or partially occluded. We therefore introduce a weighting scheme that prioritizes more salient and confident observations. The weight $w_{i,v}$ for each view $v \in \mathcal{V}_i^*$ is defined as:

$$w_{i,v} = \underbrace{S_{i,v}}_{\text{Match Confidence}} \times \underbrace{V_{i,v}}_{\text{Visibility/Salience}}, \tag{4}$$

  where $S_{i,v}$ is the association score representing the *confidence* of the 2D-3D match, and $V_{i,v}$ is a *visibility* score. As implemented in our code, $V_{i,v}$ is operationally defined as the total pixel count of the rendered silhouette, naturally giving more weight to views where the object appears larger. The final feature $\mathbf{F}_i$ is the weighted average:

$$\mathbf{F}_i = \frac{\sum_{v \in \mathcal{V}_i^*} w_{i,v} \cdot \mathbf{f}_{i,v}^{\text{CLIP}}}{\sum_{v \in \mathcal{V}_i^*} w_{i,v}}. \tag{5}$$

  This ensures that the final semantic embedding is anchored by the most informative and unambiguous perspectives available in the dataset, leading to a more accurate and nuanced representation that reflects the quality of the underlying observations.

Both modifications provide a significant improvement in robustness over the baseline, with the weighted approach offering a more sophisticated and potentially more accurate fusion of multi-view semantic information.

4. Results

We evaluate our methods on 9 challenging scenes from the ScanNet dataset presented in Table 1.

Regarding *modification 1*, due to computational constraints, Table 1 presents results from a preliminary implementation using point approximations for Gaussian projections. These results fall within the margin of error relative to baseline. Evaluation of our current implementation on scene0062_00 yielded mIoU: 0.3217 and mAcc.: 0.5224, showing no significant improvement over baseline.

Scene	Baseline		M1 (Ours)		M2 (Ours)		M3 (Ours)	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
scene0000_00	0.318	0.458	0.3285	0.4855	0.3192	0.4609	**0.3304**	**0.4941**
scene0062_00	0.345	0.555	**0.3972**	**0.6591**	0.3300	0.5682	0.3667	0.5767
scene0070_00	0.228	**0.339**	0.2163	0.3110	**0.2367**	0.3343	0.2131	0.3033
scene0097_00	0.361	0.521	**0.3792**	**0.5496**	0.3041	0.4636	0.3593	0.5453
scene0140_00	0.227	0.401	0.2497	0.4336	0.2241	0.4004	**0.2562**	**0.4447**
scene0200_00	0.402	0.550	0.3879	0.5235	0.4164	0.5595	**0.4384**	**0.5802**
scene0347_00	0.291	0.406	**0.3218**	**0.4437**	0.2948	0.4118	0.2934	0.4213
scene0590_00	0.298	0.432	0.2991	0.4048	0.3028	0.4185	**0.3332**	**0.4471**
scene0645_00	0.299	0.488	0.2919	0.4703	**0.3140**	**0.4910**	0.2851	0.4712
Average	0.299	0.451	0.312	0.466	0.305	0.456	**0.318**	**0.477**

Table 1. ScanNet per-scene semantic segmentation results.

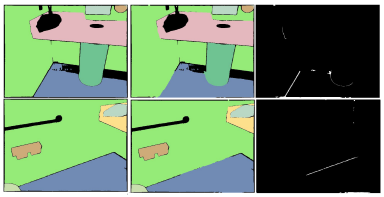

Figure 2. Mask expansion stages: original SAM mask, after ID synchronization, after mask expansion, and difference map.

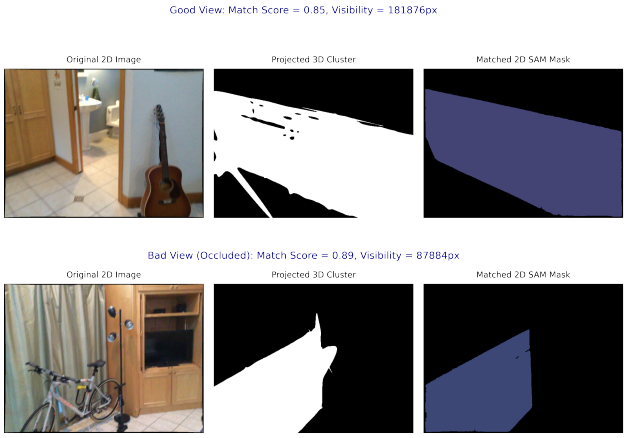

Figure 3. Mechanism of the Visibility-Weighted Fusion. Our method determines each view's contribution to the final semantic feature by calculating a weight, which is a product of its Match Score and Visibility. This figure contrasts two views of the same 3D cluster. (Top) The "Good View" combines a strong Match Score (0.85) with high Visibility (181,876px), resulting in a large overall weight. (Bottom) The partially occluded "Bad View" has a high Match Score (0.89), but its Visibility is drastically reduced by occlusion (87,884px)

Qualitative analysis demonstrates enhanced mask consistency across viewpoints in the initial stage, as shown in Figure 1. Toilet objects and structural elements maintain consistent IDs across different viewing angles after cross-view synchronization.

Figure 2 shows the secondary refinement stage reduces unclassified regions and improves boundary definition. However, segmentation quality correlates inversely with splat sampling density, as discussed in Section 5.

The baseline's single-view approach is prone to artifacts like semantic bleeding (Fig. 4b). In contrast, our multi-view fusion framework improves robustness. While unweighted averaging (M2) mitigates noise through consensus (Fig. 4c), our visibility-weighted strategy (M3) achieves optimal performance. By prioritizing high-quality, unoccluded views (Fig. 3), M3 yields the most accurate segmentation with the sharpest boundaries (Fig. 4d). This validates our hypothesis that a weighted multi-view consensus is critical for robust 3D semantic understanding.

5. Conclusion

Several technical challenges merit further investigation. The initial challenge concerns the observed phenomenon wherein larger Gaussian splats, particularly those representing substantial structural elements such as walls, demonstrate dominant characteristics that effectively occlude smaller splats corresponding to less volumetrically significant objects. This occlusion mechanism results in suboptimal representation of smaller entities within the refined segmentation masks. One possible solution could be a splat sampling strategy that is less penalizing for smaller objects. The second challenge pertains to the cross-view constraint methodology illustrated in Figure 1. While this approach demonstrates efficacy in enhancing segmentation

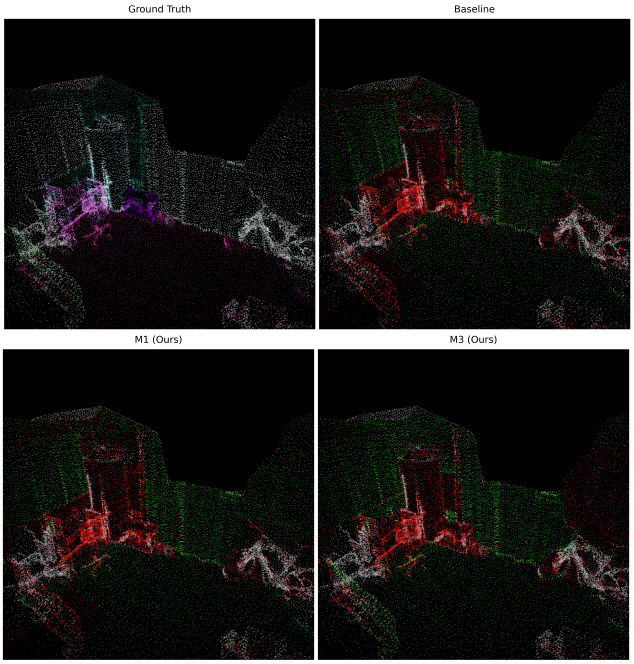

Figure 4. Qualitative Comparison of Semantic Segmentation. This figure illustrates the semantic segmentation of a complex scene. (a) Ground Truth: Shows distinct, clean segments for different objects. (b) Baseline: The "winner-takes-all" approach suffers from significant noise and semantic bleeding, misclassifying the floor (green) as part of the central object (red). (c) M2 (Ours): Our unweighted averaging method significantly cleans up the segmentation, improving the distinction between objects. (d) M3 (Ours): Our visibility-weighted averaging method provides the most accurate result, yielding sharper boundaries and a segmentation that most closely resembles the ground truth.

consistency across multiple perspectives, implementation difficulties were encountered when scaling the algorithm to accommodate both comprehensive scene-wide application and diminutive object instances. These implementation constraints may potentially be mitigated through parametric optimization of point sampling density and opacity threshold configurations. An additional algorithmic constraint worth consideration involves the selective exclusion of splats exhibiting elevated variance ratios along their principal axes, as these elements frequently generate projections spanning multiple segmentation boundaries. These identified limitations constitute priority areas for subsequent research iterations.

References

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

[2] 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, 2020. 1

[3] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. *arXiv preprint arXiv:2312.16084*, 2023. 1

[4] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. *arXiv preprint arXiv:2311.18482*, 2023. 1

[5] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding, 2024. 1