3DISCO: 3D OPEN VOCABULARY INSTANCE SEGMENTATION WITH CLEAN OBJECTS VIA GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Segmenting 3D scenes into semantically meaningful instances is central to embodied perception. Leveraging high-quality 2D segmentations across multiple views is promising, but maintaining consistent identities across views, constraining Gaussians to lie within one object's boundaries, and assigning semantics based on highly informative views are challenging. We introduce 3DISCO, a 3D segmentation method based on Gaussian splatting that identifies objects using instance features learned via uncertainty-aware contrastive training. 3DISCO treats 2D masks as noisy labels, using predicted uncertainty to down-weight unreliable regions and guide the recovery of erroneous or missing annotations. To enable cleaner boundary delineation, 3DISCO prunes Gaussians that exhibit feature similarity to an instance but are inconsistent with that instance's 2D masks. 3DISCO then assigns semantics to each instance using features from only its most informative views, selected via a multimodal reranker, producing language-aligned 3D representations. Experiments on LERF and ScanNet show substantial improvements in semantic accuracy and boundary coherence.

1 INTRODUCTION

Understanding and segmenting objects in 3D scenes is fundamental for applications in robotics, AR/VR, and interactive scene editing. Gaussian splatting offers an efficient, photorealistic 3D representation with real-time rendering, making it an appealing substrate for such semantic understanding (Kerbl et al., 2023; Huang et al., 2024). Since 3D annotations are scarce, a strategy that offers a better shot at generalization is to lift supervision from 2D foundation models, using segmenters such as SAM and its successors (Kirillov et al., 2023; Ravi et al., 2024; Carion et al., 2025) for instance masks and vision-language models such as CLIP (Radford et al., 2021) for open-vocabulary recognition. However, these 2D segmenters operate independently on each view, and their errors accumulate in 3D as drifting instance identities, blurry object boundaries, and floating geometry.

We identify three sources of error in this 2D-to-3D lifting. First, automatically generated masks are noisy: boundaries are imprecise, instances merged, split, or missed, and masks lack cross-view identities. Second, alpha blending lets individual Gaussians contribute to extended image regions, so Gaussians bleed across object boundaries and their assignment becomes ambiguous. Third, approaches often reproject models to 2D for semantic recognition, which depends strongly on viewpoint; naïvely averaging vision-language features across observations (including occluded or uninformative views) and compressing each object into one embedding fails to resolve fine-grained distinctions between similar labels.

We present 3DISCO, which addresses each of these error sources in turn. To improve recall in the 2D segmentation, we query Qwen3-VL (Team, 2025) to name the objects in each image and prompt SAM3 (Carion et al., 2025) for text-guided instance masks. We then associate masks across views by assigning an instance feature to each Gaussian, trained with an uncertainty-aware contrastive objective, avoiding the need for matched mask identifiers across frames. After a first round of training, we use the 2D mask supervision and the uncertainty from our instance embeddings to identify overlooked objects, and query SAM (Kirillov et al., 2023) at those locations for a second round of training to boost recall. While this two-step process provides strong instance assignment,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: Renders of object instances automatically identified and segmented by 3DISCO. Objects represent semantically meaningful categories and have minimal or no artifacts.

Gaussians at this stage often bleed across object boundaries, creating inconsistent 3D geometries. We introduce a prototype-guided mask consistency (PMC) loss, which identifies and removes offending Gaussians: Gaussians assigned to each prototype are optimized to be opaque inside their 2D mask and transparent outside, driving ambiguous Gaussians toward transparency for pruning. The resulting 3D instances comprehensively capture the objects in the scene and represent geometrically accurate object models. To semantically label the retrieved instances, rather than relying on vanilla feature aggregation, we identify two sources of valuable information: isolated object renders, which remove clutter and occlusions, and contextual crops, which preserve disambiguating scene cues. Our approach uses a multimodal reranker (Zhang et al., 2025) to select the most informative views and resolve ambiguous labels, using fast embedding retrieval to scale to large vocabularies.

We treat 3D segmentation via a principled pipeline that requires executing multiple components, some of which are computationally expensive. However, our system as a whole improves the overall efficiency of the segmentation process compared to existing methods. In particular, the Qwen3-VL-guided proposals produce a smaller set of 2D masks, accelerating contrastive feature training, while selective-view semantic alignment enables faster processing than naïve aggregation.

Our main contributions are 1) an uncertainty-aware contrastive objective with multi-view mask correction for learning instance features from noisy 2D masks, 2) a PMC loss that prunes inconsistent Gaussians inconsistent, and 3) a retrieve-then-rerank formulation for open-vocabulary labeling over isolated and contextual views. As shown in Figure 1, the resulting instance segmentations are clean and nearly artifact-free, with precise boundaries even for the reflective porcelain hand resting on a table of similar color, or the eggs in the cluttered ramen bowl. Our approach achieves state-of-the-art 3D open-vocabulary segmentation on LERF-OVS and ScanNet++.

2 Related Work

Open vocabulary (OV) 3D segmentation aims to identify objects from arbitrary categories using language supervision (e.g., from CLIP Radford et al., 2021). A key challenge is reconciling the viewpoint-sensitive semantics of 2D features with the spatial consistency needed for 3D reasoning.

2.1 Gaussian Splatting Segmentation

3D Gaussian splatting offers an efficient, editable, explicit 3D representation with real-time rendering (Kerbl et al., 2023). Many 3D segmentation methods rely on 2D segmentation masks (e.g., from Kirillov et al., 2023; Ravi et al., 2024; Carion et al., 2025) as supervision for segmenting Gaussians.

Some transfer labels through max-contributor assignment without training (Chacko et al., 2025) or linear programming (Shen et al., 2024). Because these approaches operate per view, boundary Gaussians receive conflicting labels from different views, creating fragmented boundaries. Other work learns semantic embeddings directly on Gaussians to enforce multi-view consistency (Castillo et al., 2024). Gaussian Grouping trains per-Gaussian identity features supervised by 2D video-tracked masks and regularized via 3D spatial consistency (Ye et al., 2024), while GaGa constructs a 3D-aware memory bank to associate Gaussians into coherent groups across views (Lyu et al., 2024). However, when alpha blending causes one Gaussian to contribute to multiple objects during rendering, gradient-based learning receives mixed supervision, resulting in averaged features that do not commit to a single instance. Interactive segmentation enables user-driven instance manipulation through prompting (Cen et al., 2023; Choi et al., 2024), but inherits similar consistency challenges.

2.2 Open-vocabulary Gaussian splatting

To enable language-guided queries in neural 3D representations, LERF pioneered embedding CLIP features directly into radiance fields (Kerr et al., 2023). To adapt this idea to Gaussian splatting without excessive memory use, methods compress high-dimensional CLIP features through scene-specific autoencoders (Qin et al., 2024) or combine DINO and CLIP features with quantization schemes (Shi et al., 2024), trading expressivity for efficiency. SuperGSeg builds hierarchical representations by clustering Gaussians into super-Gaussians for multi-granularity semantics (Liang et al., 2024). OmniSeg3D targets multi-granularity segmentation through hierarchical contrastive learning that distinguishes instances at varying levels of detail (Ying et al., 2023). Alternative approaches aggregate features across views using probabilistic weighting (Cheng et al., 2025), visibility-aware gating (Wang et al., 2025), or pre-trained quantization codebooks (Jun-Seong et al., 2025). Recent work combines instance segmentation with OV understanding. Several approaches first establish 3D instances from 2D cues through instance-aware offset voting (Jiang et al., 2025), context-aware feature propagation and clustering (Sun et al., 2025), or contrastive instance learning in 3D (Piekenbrinck et al., 2025; Wu et al., 2024), and then match these instance representations to language features. InstanceGaussian instead adopts a progressive joint strategy that refines appearance and semantic features together during later training stages (Li et al., 2025).

Our 3D segmentation method builds upon existing contrastive instance learning approaches, addressing the core remaining difficulty that alpha blending causes boundary Gaussians to receive gradients from multiple objects, leading to ambiguous feature assignments. Operating directly on Gaussians—as opposed to rendered features, as done in prior work—enables our method to identify which Gaussians cause inconsistencies and constrain their 2D projections to remain consistent with instance masks across views, which in turn enables accurate 3D instance geometry. Moreover, while existing approaches operate on fixed 2D mask sets, our approach leverages uncertainty to identify missing objects and obtains fresh 2D masks to improve recall. Selectively identifying views with complementary information (isolated renders and contextual crops) maximizes the capability of current 2D recognition objects to semantically label the retrieved 3D instances.

3 Method

We reconstruct the scene from posed images with 2D Gaussian Splatting (2DGS; Huang et al., 2024), whose planar Gaussians $\mathcal{G} = \{g_i\}_{i=1}^{N}$ adhere more closely to surfaces than volumetric 3D Gaussians and thus benefit boundary-sensitive segmentation. Each Gaussian carries a learnable instance feature supervised by 2D instance masks as pseudo-labels; clustering the learned feature field yields 3D instances that are then labeled with open-vocabulary queries (Figure 2). 3DISCO addresses three sources of error. First, the masks are noisy, with incorrect boundaries, merged or split instances, and missing objects; we learn instance features with an uncertainty-aware contrastive objective and use the stabilized field to remove inconsistent masks and re-query SAM where masks are wrong or missing for retraining (§ 3.1–3.2). Second, alpha blending lets Gaussians bleed across object boundaries, so initial lifting may associate some Gaussians to the wrong instance; we render each instance's Gaussians, and our PMC loss penalizes opacity outside its 2D mask, so contradicting Gaussians become transparent and are pruned (§ 3.3). Third, reprojected recognition quality depends strongly on viewpoint; to maximize information content, we cast labeling as retrieval over isolated and contextual views of each object, followed by multimodal reranking (§ 3.4).

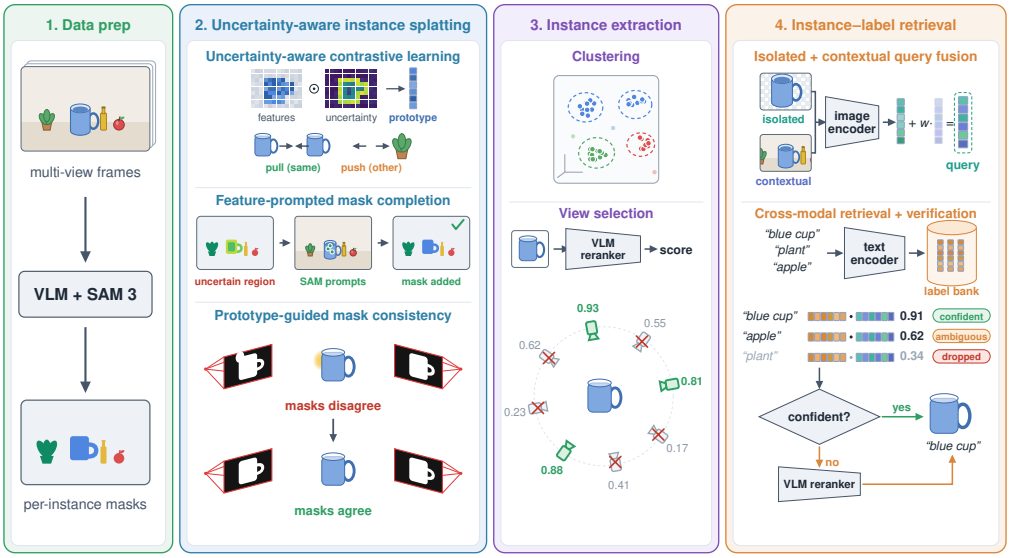


Figure 2: We reconstruct the scene with 2D Gaussian Splatting and learn a Gaussian feature field from 2D instance masks used as pseudo-labels. Uncertainty-aware feature learning reduces the influence of noisy masks, after which the learned field is used to identify and correct inconsistent or missing masks. We then enforce prototype-guided mask consistency to suppress Gaussians that contradict their mask constraints. Finally, the cleaned instances are clustered in 3D and assigned open-vocabulary semantic labels using retrieved object views and multimodal reranking.

3.1 Uncertainty-aware contrastive instance feature learning

Instance masks For each training image, the vision-language model Qwen3-VL (Team, 2025) names the visible objects, and we use these names to prompt SAM3 (Carion et al., 2025) for text-guided instance masks $\mathbf{M}_1, \ldots, \mathbf{M}_K$, which serve as pseudo-labels for the instance features. The masks require no predefined vocabulary, but they are noisy and cover the scene only partially. We collect the pixels outside all masks into an unlabeled region $\mathbf{U}$ and append it as an additional group:

$$\mathbf{U} = \left\{ \mathbf{p} \;\middle|\; \mathbf{p} \notin \textstyle\bigcup_{k=1}^{K} \mathbf{M}_k \right\}, \qquad \mathbf{M}_{K+1} = \mathbf{U} \;, \qquad \mathcal{R} = \{\mathbf{M}_1, \ldots, \mathbf{M}_{K+1}\} \;. \tag{1}$$

Because $\mathbf{U}$ may contain both true background and missed objects, the losses below treat it differently from the instance masks.

Features, uncertainty, and prototypes For compatibility with 2DGS training, each learned quantity is alpha-blended for each view using the 2DGS blending weights $\omega^{\mathbf{p}}$. The instance features $\mathbf{q} \in \mathbb{R}^D$ produce a feature render $Q(\mathbf{p}) = \sum_i \omega_i^{\mathbf{p}} \mathbf{q}_i$, which is then normalized to $\bar{Q}(\mathbf{p})$, where i indexes over Gaussians. Since mask quality varies spatially, with contradictory supervision at boundaries, occlusions, and mislabeled objects, we also learn an uncertainty estimate for the instance features. Each Gaussian carries a scalar log-variance s_i, similarly rasterized into a per-pixel log-variance $s(\mathbf{p})$ and precision $\rho(\mathbf{p}) = \exp(-s(\mathbf{p}))$. To support contrastive training, we summarize each region $\mathbf{M}_k \in \mathcal{R}$ by a prototype, computed as the precision-weighted mean of its normalized rendered features and rescaled to unit length:

$$\mathbf{u}_k = \frac{\sum_{\mathbf{p} \in \mathbf{M}_k} \mathrm{sg}(\rho(\mathbf{p}))\, \bar{Q}(\mathbf{p})}{\sum_{\mathbf{p} \in \mathbf{M}_k} \mathrm{sg}(\rho(\mathbf{p}))} \;, \qquad \mathbf{z}_k = \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|_2} \;, \qquad k = 1, \ldots, K+1 \;, \tag{2}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator used to prevent the uncertainty from moving a prototype through its weighting. Gradients propagate through the *instance* prototypes $\mathbf{z}_1, \ldots, \mathbf{z}_K$, but the *unlabeled* prototype $\mathbf{z}_{K+1}$ is detached from gradient computations, since $\mathbf{U}$ may mix several overlooked objects and should not be collapsed onto one learned representation. Disagreement between a pixel feature and a prototype is measured by the cosine residual $r_k(\mathbf{p}) = 1 - \bar{Q}(\mathbf{p})^\top \mathbf{z}_k$.

Intra-instance loss Contrastive learning will draw instance features close to each other when they correspond to the same 2D mask. To obtain uncertainty estimates in a principled way, we treat each mask's normalized prototype $\mathbf{z}_k$ as the ground-truth label and the normalized feature render $\bar{Q}(\mathbf{p})$ as the per-pixel prediction under an isotropic Gaussian likelihood, given by $\frac{1}{2}(\rho(\mathbf{p})\|\mathbf{z}_k - \bar{Q}(\mathbf{p}))\|^2 + s(\mathbf{p}))$ (Kendall & Gal, 2017). Since $\frac{1}{2}\|\mathbf{z}_k - \bar{Q}(\mathbf{p})\|^2 = r_k(\mathbf{p})$ for unit vectors, maximizing this likelihood pulls each feature toward its prototype with a strength given by the precision and fits the variance to the residual, so that pixels with contradictory supervision have high uncertainty. In practice, we found it more effective to treat feature fitting and variance estimation as separate loss terms, which allowed us to maintain non-negligible contrastive loss even for highly uncertain assignments. Concretely, our objective minimizes the feature residual weighted by a clipped precision, and minimizes the precision weighted by the full residuals:

$$\gamma(\mathbf{p}) = \mathrm{sg}(\max(\rho(\mathbf{p}), \nu)) \ , \qquad \ell_k(\mathbf{p}) = \gamma(\mathbf{p}) \cdot r_k(\mathbf{p}) + \frac{\mathrm{sg}(r_k(\mathbf{p})) \cdot \rho(\mathbf{p}) + s(\mathbf{p})}{n(\mathbf{p})} \ , \tag{3}$$

where $n(\mathbf{p}) = \sum_{j=1}^{K} \mathbf{1}[\mathbf{p} \in \mathbf{M}_j]$ is the number of instance masks that contain a pixel $\mathbf{p}$. Dividing by $n(\mathbf{p})$ prevents overlapping masks from counting a pixel's uncertainty repeatedly. The intra-instance loss averages ℓ_k over each instance mask and adds a term, weighted by $\eta > 0$, that attracts unlabeled pixels toward their detached prototype without uncertainty fitting:

$$\mathcal{L}_{\text{intra}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathbf{M}_k|} \sum_{\mathbf{p} \in \mathbf{M}_k} \ell_k(\mathbf{p}) + \frac{\eta}{|\mathbf{U}|} \sum_{\mathbf{p} \in \mathbf{U}} \gamma(\mathbf{p}) \cdot r_{K+1}(\mathbf{p}). \tag{4}$$

Inter-instance loss To complete our contrastive objective, we push apart features across instances. Let $\mathcal{N}_k = \mathcal{R} \backslash \{\mathbf{M}_k\}$ denote the negative groups for instance k, consisting of the other instance masks together with $\mathbf{U}$. We penalize pixels in these groups whose residual to $\mathbf{z}_k$ falls below a margin m:

$$\mathcal{L}_{\text{inter}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{N}_k|} \sum_{\mathbf{M} \in \mathcal{N}_k} \frac{1}{|\mathbf{M}|} \sum_{\mathbf{p} \in \mathbf{M} \backslash \mathbf{M}_k} \gamma(\mathbf{p}) \ \max\big(0, \ m - r_k(\mathbf{p})\big). \tag{5}$$

Since masks may overlap, pixels that also lie in $\mathbf{M}_k$ are excluded, so they never act as negatives for their own prototype. The full instance objective, $\mathcal{L}_{\text{inst}} = \lambda_{\text{intra}}\mathcal{L}_{\text{intra}} + \lambda_{\text{inter}}\mathcal{L}_{\text{inter}}$, is evaluated independently in each view and updates only the instance features and log-variances. Gradients do not flow through geometry or appearance, preventing noisy masks from distorting the reconstruction.

Geometric regularization Our method optimizes Gaussian geometry and appearance with the 2DGS photometric loss $\mathcal{L}_{\text{render}}$ (Huang et al., 2024). To make the reconstruction more amneable to instance grouping, we first encourage compact object surfaces by applying the depth distortion loss $\mathcal{L}_d$ of 2DGS only within instance masks, since unrelated surfaces and unobserved objects may overlap in $\mathbf{U}$. Moreover, highly anisotropic Gaussians can stretch across object boundaries and cover the interior of an instance only sparsely, which weakens the spatial connectivity used in clustering (§ 3.2). We bound the anisotropy of each Gaussian with a hinge on its log-space axis ratio:

$$\mathcal{L}_{\text{ar}} = \frac{1}{N} \sum_{i=1}^{N} \max\Big(0, \ \log \frac{\max_a \sigma_{i,a}}{\min_a \sigma_{i,a}} - \log \kappa\Big) \ , \tag{6}$$

where the $\sigma_{i,a}$'s are the scales of Gaussian i and κ is the maximum allowed ratio.

3.2 Instance clustering and uncertainty-gated mask correction

Two-stage clustering Because instance prototypes are computed independently in each view, we first group them into consistent objects. We use the following process to obtain clusters both for mask correction and (later) for semantic inference. The process performs two consecutive runs of DBSCAN (Ester et al., 1996). The first pass clusters instance prototypes $\mathbf{z}_1, \ldots, \mathbf{z}_K$ across views by cosine distance and assigns each Gaussian to the cluster whose mean prototype is closest to its feature. Clustering on features alone may erroneously merge distinct objects, because the inter-instance loss only separates objects that receive separate masks in some common view. To separate such instances, a second pass of DBSCAN uses Euclidean distance on the Gaussian positions within

each cluster, splitting objects that share features but are spatially separated. Each final cluster c is represented by a centroid embedding $\boldsymbol{e}_c$, the normalized mean of its Gaussian features, and its predicted region in view v consists of the pixels whose rendered features are similar to this centroid:

$$\mathcal{F}_c^v = \left\{ \mathbf{p} \;\middle|\; \cos\big(\bar{Q}(\mathbf{p}), \boldsymbol{e}_c\big) \geq \tau_{\text{pred}} \right\} \quad , \tag{7}$$

where $\bar{Q}(\mathbf{p})$ is rendered in view v and a single threshold τ_{pred} is shared by all objects.

Mask correction Uncertainty weighting limits the influence of ambiguous masks, but does not repair them by itself. Once the feature field has stabilized, each predicted region $\mathcal{F}_c^v$ provides an object mask estimate derived from the learned features, which we use to correct the supervision from the 2D masks from SAM 3 in a single pass. We compare each region $\mathcal{F}_c^v$ in one view with the overlapping SAM 3 masks, retaining matched masks that have an IoU with $\mathcal{F}_c^v$ above τ_{iou} (see Appendix A.3 for details of this matching). The presence of several masks inside one region indicates over-segmentation, while an overlapping mask with low IoU indicates under-segmentation; we drop supervision from SAM 3 masks corresponding to both cases. A region $\mathcal{F}_c^v$ with no corresponding mask suggests that SAM 3 overlooked an object in this view. If the region's mean log-variance $s(\mathbf{p})$ exceeds τ_s, we consider that additional supervision would improve the 3D segmentation quality. For each flagged region, we prompt SAM (Kirillov et al., 2023) with points sampled inside $\mathcal{F}_c^v$ to obtain a replacement mask. Predicted regions only locate errors and are never used as supervision, retaining SAM's boundary quality. We conduct one round of training with the updated mask set.

3.3 Prototype-guided mask consistency

Mask correction yields cleaner supervision, but outlier Gaussians remain. Because alpha blending lets a Gaussian influence a broad image region, residual mask errors can assign instance-compatible features to Gaussians whose projection extends beyond that instance's mask. Collectively, such Gaussians cause boundary leakage and floating geometry around otherwise well-segmented instances. Our PMC loss measures whether feature-based membership agrees with the masks. For each instance k in a training view, with a *frozen* prototype $\mathbf{z}_k$, we select the matching Gaussians $\Omega_k = \{i \mid \cos(\mathbf{q}_i, \mathbf{z}_k) \geq \tau_{\text{pmc}}\}$. We splat the matching Gaussians to obtain an accumulated opacity image $\hat{A}_k \in [0,1]^{H \times W}$, and treat the opacity as a trainable segmentation mask:

$$\mathcal{L}_{\text{pmc}} = \frac{1}{K} \sum_{k=1}^{K} \Big[\lambda_{\text{bce}}\, \text{BCE}\big(\hat{A}_k, \mathbf{M}_k\big) + \lambda_{\text{dice}}\, \text{Dice}\big(\hat{A}_k, \mathbf{M}_k\big) \Big], \tag{8}$$

where BCE is the binary cross-entropy averaged over pixels and Dice is the soft Dice loss (Milletari et al., 2016). A Gaussian whose feature matches instance k but whose contribution falls outside $\mathbf{M}_k$ produces spurious opacity in $\hat{A}_k$ and is driven toward transparency for subsequent pruning. Because the features are frozen, PMC removes contradicting Gaussians while maintaining instance membership. All instances share one projection and are rasterized in a single batched pass (Appendix A.4). The training schedule for our overall approach is given in Appendix A.5.

3.4 3D open-vocabulary panoptic segmentation

Semantic association as retrieval Given candidate instances with Gaussian sets $\mathcal{G}_c$, we next associate semantic embeddings to each cluster by 1) considering isolated and contextual views, 2) ranking candidate views according to their discriminative power, and 3) aggregating language-aligned vision embeddings across the top views. This overcomes the challenge faced by standard approaches that average vision-language features across all views for an instance, which treats every observation as equally trustworthy even though many views are occluded, truncated, or blurred. For each instance, we start by considering only views in which its predicted region $\mathcal{F}_c^v$ used as the context mask agrees with the silhouette rendered from $\mathcal{G}_c$ alone (Appendix A.6). For each such viewpoint, we yield two object-centric crops. The isolated crop $I_{c,v}^{\text{iso}}$ is rendered from $\mathcal{G}_c$, yielding a view that is completely free of clutter and occlusions, which is only possible because each object is an explicit, clean set of Gaussians. The contextual crop $I_{c,v}^{\text{ctx}}$ is cut from the original image around the context mask and retains the surrounding scene, which helps disambiguate categories with similar appearance in isolation. We next remove degenerate views view geometric filters. Then, we produce *cheap* embeddings for all candidate views using the Perception Encoder (Bolya et al., 2025)

Table 1: Text-based 3D object retrieval on LERF-OVS. As per standard protocol, Gaussians selected by each query are rendered without occlusion by other scene objects. Baseline results obtained from Wang et al. (2025) and Piekenbrinck et al. (2025). Best in red, second in orange, third in yellow.

Method	figurines		ramen		teatime		waldo_kitchen		**mean**	
	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑
LangSplat (Qin et al., 2024)	10.16	8.93	7.92	11.27	11.38	20.34	9.18	9.09	9.66	12.41
LEGaussians (Shi et al., 2024)	17.99	23.21	15.79	26.76	19.27	27.12	11.78	18.18	16.21	23.82
OpenGaussian (Wu et al., 2024)	39.29	55.36	31.01	42.25	60.44	76.27	22.70	31.82	38.36	51.43
SuperGSeg (Liang et al., 2024)	43.68	60.71	18.07	23.94	55.31	77.97	26.71	45.45	35.94	52.02
Dr. Splat (Jun-Seong et al., 2025)	54.42	80.36	24.33	35.21	57.35	77.97	37.05	63.64	43.29	64.30
InstanceGaussian (Li et al., 2025)	54.87	73.21	25.03	38.03	54.13	69.49	41.47	63.64	43.87	61.09
CAGS (Sun et al., 2025)	60.85	82.14	36.29	46.48	68.40	86.44	37.62	63.64	50.79	69.62
VoteSplat (Jiang et al., 2025)	68.62	85.71	39.24	61.97	66.71	88.14	25.84	33.68	50.10	67.38
Occam's LGS (Cheng et al., 2025)	52.90	78.57	32.01	54.92	61.02	93.22	42.95	72.72	47.22	74.84
VALA (Wang et al., 2025)	60.38	89.29	45.41	67.61	70.61	88.14	55.71	86.36	58.02	82.85
OpenSplat3D (Piekenbrinck et al., 2025)	60.71	85.71	49.20	76.06	73.27	88.14	55.63	77.27	59.70	81.79
LAGA (Cen et al., 2025)	64.1	–	55.6	–	70.9	–	65.6	–	64.0	–
ExtrinSplat (Ding et al., 2026)	66.4	–	45.6	–	64.4	–	40.9	–	54.3	–
LUDVIG (Marrie et al., 2025)	58.0	–	42.3	–	58.6	–	42.8	–	50.4	–
3DISCO$_{\text{3DGS}}$ (ours)	80.64	98.21	55.89	82.22	76.14	89.58	66.34	85.00	69.75	88.75
3DISCO$_{\text{2DGS}}$ (ours)	80.66	98.21	58.06	77.78	77.26	89.58	66.72	90.00	70.67	88.89

and use a slower (but more accurate) vision-language reranker to identify the most discriminative crops (Zhang et al., 2025). Concretely, the reranker scores the candidate crops against a text query describing a good recognition view, and so it favors views in which the object is clearly visible from an informative angle. We average the embeddings of the top T views $\mathcal{V}_c^T$ into $\bar{\mathbf{y}}_c^{\text{iso}}$ and $\bar{\mathbf{y}}_c^{\text{ctx}}$, and fuse them into the overall object embedding $\mathbf{y}_c = \bar{\mathbf{y}}_c^{\text{iso}} + \mu\, \bar{\mathbf{y}}_c^{\text{ctx}}$, where $\mu < 1$ emphasizes the isolated object over the context. Notably, we can obtain the embeddings $\mathbf{y}_c$ with the selected crops ahead of time and query the scene with new label sets without re-encoding any images.

Label retrieval and reranking At inference, each candidate label t is encoded with Perception Encoder into $\mathbf{y}_t$, and we compute the cosine similarity $S(c,t) = \cos(\mathbf{y}_c, \mathbf{y}_t)$ for each object. Labels with $S(c,t) < \tau_{\min}$ are rejected, and the top label is assigned when its score exceeds τ_{conf}. Instances between the two thresholds are ambiguous given the fixed aggregate semantic embeddings. For such cases, the reranker R scores the top T crops once more, but now against the query labels t:

$$t^* = \arg\max_t \frac{1}{T} \sum_{v \in \mathcal{V}_c^T} \Big[R\big(I_{c,v}^{\text{iso}}, t\big) + \beta\, R\big(I_{c,v}^{\text{ctx}}, t\big) \Big], \qquad \beta < 1 \ . \tag{9}$$

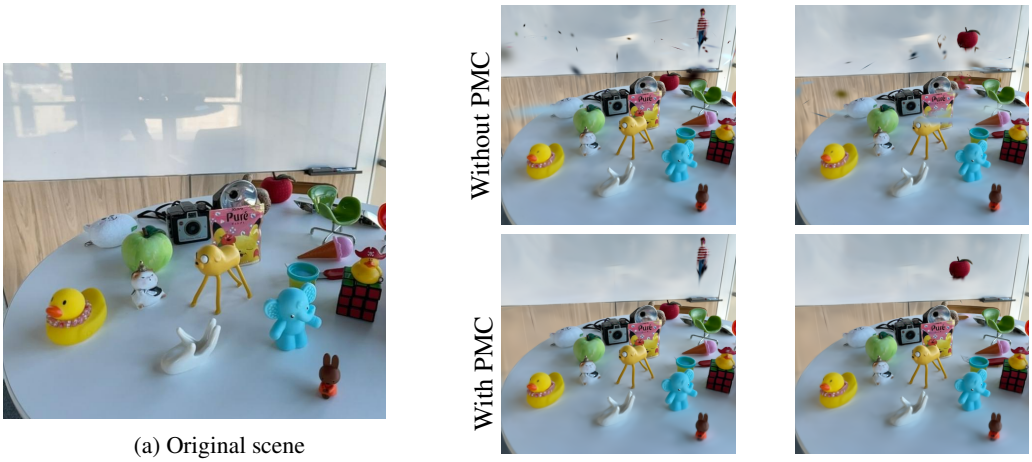
Because the reranker sees only T curated views and a shortlist of labels, and only for ambiguous instances, its cost stays bounded even for large vocabularies.

4 Experiments

4.1 Experimental setup

We evaluate OV 3D segmentation, retrieving and segmenting object instances from language queries, on two datasets. LERF-OVS, re-annotated by LangSplat (Qin et al., 2024), has object-centric and scene-level captures with OV masks; we evaluate only 3D retrieval, as 2D evaluation can hide Gaussians with misassigned features that surface only under 3D editing. ScanNet++ (Yeshwanth et al., 2023) provides high-fidelity indoor reconstructions with dense 3D annotations, allowing predictions to be evaluated directly on the mesh; each scene is queried with its annotated classes from the top-100 benchmark (Appendix B). The appendices cover implementation details (A), more qualitative results (C), and baseline comparisons on mask quality (D).

Metrics We report mean IoU (mIoU) and mean accuracy at a 25% IoU threshold (mAcc@25), averaged over scenes. On LERF-OVS, the Gaussians selected by each query are rendered by opacity into a binary mask and compared against the 2D ground truth. On ScanNet++, each ground-truth mesh vertex takes the class voted by nearby Gaussians, weighted by opacity and Mahalanobis distance, and IoU is computed per class over all vertices (Appendix B).



(a) Original scene

(b) Moved Waldo toy

(c) Moved toy apple

Figure 3: Interactive scene editing via click-and-drag. With PMC, objects move cleanly; without PMC, they carry and leave behind floating artifacts. Additional scenes in Figures E.1– E.3

4.2 Quantitative results

LERF-OVS As shown in Table 1, both 3DISCO variants achieve the highest mIoU in every scene and improve mean mIoU and mAcc by 6.7 and 6.0 points over the best baselines. Gains are largest in *figurines* (+12.0 mIoU) and smallest in *waldo kitchen* (+1.1), where sparser viewpoint coverage limits the benefit of PMC.

ScanNet++ Table 2 reports results on ScanNet++, where 3DISCO outperforms all baselines by a large margin. Baselines produce noisy language features, which cause false positives. Contrastive instance learning and PMC yield clean, multi-view-consistent instances, so each label attaches to a coherent object.

Table 2: ScanNet++ OV 3D segmentation. 10 randomly chosen scenes, top-100 categories.

Method	mIoU ↑	mAcc@25 ↑
LangSplat (Qin et al., 2024)	6.80	7.64
OpenGaussian (Wu et al., 2024)	12.73	20.48
Dr. Splat (Jun-Seong et al., 2025)	14.38	18.95
VALA (Wang et al., 2025)	13.90	19.22
3DISCO (ours)	**26.86**	**44.52**

Ablations Table 3 evaluates the key components of our method. Without PMC, Gaussians whose features match an instance but whose projections fall outside its masks are never removed, causing boundary leakage and floating geometry. These errors propagate to labeling. Spurious Gaussians pollute the isolated renders and yield out-of-distribution inputs for the vision-language model, which degrades retrieval. PMC thus both sharpens instance boundaries and keeps isolated renders clean enough for vision-language alignment. Without uncertainty weighting, contradictory supervision at boundaries, occlusions, and mislabeled objects enter the feature losses at full weight and blur the separation between instances. Mask correction mainly improves mAcc, since recovering missed objects and removing merged or split masks turns failed queries into successful ones rather than refining boundaries. It helps only when used together with uncertainty weighting. However, without it, the predicted regions are less reliable and missed-object detection loses its uncertainty gate, so correction discards valid masks and re-queries spurious regions.

Among the language-alignment ablations, removing isolated crops causes the largest drop. Intuitively, only these views rendered from the instance's Gaussians are free of occlusion and clutter. Removing contextual crops hurts less but degrades recognition of categories that look alike in isolation, where the surroundings disambiguate. Replacing 2DGS with 3DGS slightly reduces performance, as looser surface alignment harms feature learning and the rendered views used for labeling.

Replacing our masks with those used by LangSplat (Qin et al., 2024), which oversegment objects, causes PMC to treat most surface Gaussians as contradictory in some views and drive them toward



(a) Initial arm pose (b) Picking up the stapler (c) PCA of instance features

Figure 4: I2RT YAM arm manipulating a scene perceived via 3DISCO. (a) Initial arm pose. (b) The arm executing the query "pick the stapler". (c) Top-down view of the PCA projection of the scene's instance features.

transparency, so large parts of objects vanish. Therefore, our method requires moderately view-consistent masks, which SAM's automatic mask generation does not guarantee.

4.3 Qualitative results

3DISCO produces sharp, stable instance contours in cluttered scenes, with little boundary bleeding or floating geometry (Figure 1). PCA of the learned features (Figure E.4) shows that even small, tightly packed objects (e.g., spices and silverware in *waldo kitchen*) form well-separated clusters. Figures C.1–C.6 show multi-view renders, and Figures D.1–D.8 show baseline comparisons.

We observe two failure modes. First, 3DISCO only recovers instances the VLM names; parts never proposed as separate objects (e.g., the bear's nose in *teatime*) receive no masks. Second, recognition degrades for objects identifiable from few viewpoints: "Pikachu" in *figurines* is recognizable only from the front, so its embedding stays weak even though reranking prioritizes those views.

Table 3: Ablations on LERF-OVS. 'u': uncertainty, ' mc': mask correction.

Method	mIoU ↑	mAcc. ↑
No PMC	52.56	71.80
No Isolated	51.17	65.53
No Context	64.75	79.59
LangSplat Masks	63.10	78.19
No u	65.88	80.14
No mc	70.41	84.64
No u+mc	66.72	80.33
3DISCO (full)	**70.67**	**88.89**

4.4 Downstream applications

3D scene editing Our 2DGS models can be edited by manually moving groups of Gaussians. As shown in Figure 3, Gaussians selected by feature similarity to a clicked pixel move as a coherent unit with PMC; without PMC, objects carry and leave behind artifacts (see Appendix E.1 for details).

Robot manipulation Our models also support robot perception for manipulation. We integrate 3DISCO into TiPToP (Shen et al., 2026) and successfully execute the language query "pick the stapler" which is occluded from the initial view of the arm, as shown in Figure 4 (see Appendix E.2 for details).

5 Conclusion

We introduced 3DISCO, an instance-aware open-vocabulary Gaussian splatting framework that lifts noisy 2D masks into clean 3D instances. Uncertainty-aware contrastive learning and mask correction suppress mask noise, PMC prunes Gaussians that contradict their masks, and retrieval over isolated and contextual views with multimodal reranking labels instances directly in 3D. 3DISCO outperforms prior methods on LERF-OVS and ScanNet++ and enables clean interactive scene editing. Future work includes combining instances with finer per-Gaussian semantics in a hierarchical representation, enabling part-level queries and supervision from multi-granularity masks, and extending the framework to dynamic scenes.

6 AI DISCLOSURE

In this work, we used generative AI tools to help implement parts of the method. We have not used generative AI tools for any other tasks requiring disclosure (e.g., generating data, proposing hypotheses, designing experiments, or interpreting results), and mathematical claims and proofs are not applicable to this work. Additionally, we used generative AI tools to set up software environments, find relevant related work, draft the structure of the paper, and improve phrasing. All AI-assisted work has been verified by the authors, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network, 2025. URL https://arxiv.org/abs/2504.13181.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Myrna Castillo, Mahtab Dahaghin, Matteo Toso, and Alessio Del Bue. Contrastive gaussian clustering for weakly supervised 3D scene segmentation. In *27th International Conference on Pattern Recognition (ICPR-24)*, pp. 114–130, 2024.

Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. *arXiv preprint arXiv:2312.00860*, 2023.

Jiazhong Cen, Xudong Zhou, Jiemin Fang, Changsong Wen, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Tackling view-dependent semantics in 3d language gaussian splatting. In *ICML*, 2025.

Rohan Chacko, Nicolai Häeni, Eldar Khaliullin, Lin Sun, and Douglas Lee. Lifting by Gaussians: A simple fast and flexible method for 3D instance segmentation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV-25)*, pp. 3497–3507, 2025.

Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's LGS: An efficient approach for language Gaussian splatting. *arXiv preprint arXiv:2412.01807*, 2025.

Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. Click-Gaussian: Interactive segmentation to any 3D Gaussians. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *18th European Conference on Computer Vision (ECCV-24)*, pp. 289–305, 2024.

Jiayu Ding, Xinpeng Liu, Zhiyi Pan, Shiqiang Long, and Ge Li. Extrinsplat: Decoupling geometry and semantics for open-vocabulary understanding in 3d gaussian splatting, 2026. URL https://arxiv.org/abs/2509.22225.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 226–231, 1996.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.

Minchao Jiang, Shunyu Jia, Jiaming Gu, Xiaoyuan Lu, Guangming Zhu, Anqi Dong, and Liang Zhang. VoteSplat: Hough voting Gaussian splatting for 3D scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV-25)*, pp. 6456–6465, 2025.

Kim Jun-Seong, GeonU Kim, Kim Yu-Ji, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. splat: Directly referring 3D Gaussian splatting via direct language embedding registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-25)*, pp. 14137–14146, 2025.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30 (NeurIPS-17)*. Curran Associates, Inc., 2017.

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.

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-23)*, pp. 19729–19739, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Haijie Li, Yanmin Wu, Jiarui Meng, Qiankun Gao, Zhiyao Zhang, Ronggang Wang, and Jian Zhang. InstanceGaussian: Appearance-semantic joint gaussian representation for 3D instance-level perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-25)*, pp. 14078–14088, June 2025.

Siyun Liang, Sen Wang, Kunyi Li, Michael Niemeyer, Stefano Gasperini, Nassir Navab, and Federico Tombari. SuperGSeg: Open-vocabulary 3D segmentation with structured super-Gaussians. *arXiv preprint arXiv:2412.10231*, 2024.

Weijie Lyu, Xueting Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any gaussians via 3d-aware memory bank, 2024.

Juliette Marrie, Romain Menegaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes, 2025. URL https://arxiv.org/abs/2410.14462.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, 2016.

Jens Piekenbrinck, Christian Schmidt, Alexander Hermans, Narunas Vaskevicius, Timm Linder, and Bastian Leibe. OpenSplat3D: Open-vocabulary 3D instance segmentation using Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-25) Workshops*, pp. 5285–5294, 2025.

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 (CVPR-24)*, pp. 20051–20060, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML-21)*, pp. 8748–8763, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Qiuhong Shen, Xingyi Yang, and Xinchao Wang. FlashSplat: 2D to 3D Gaussian splatting segmentation solved optimally. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *18th European Conference on Computer Vision (ECCV-24)*, pp. 456–472, 2024.

William Shen, Nishanth Kumar, Sahit Chintalapudi, Jie Wang, Christopher Watson, Edward Hu, Jing Cao, Dinesh Jayaraman, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. TiPToP: A modular open-vocabulary planning system for robotic manipulation. *arXiv preprint arXiv:2603.09971*, 2026.

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 (CVPR-24)*, pp. 5333–5343, 2024.

Wei Sun, Yanzhao Zhou, Jianbin Jiao, and Yuan Li. CAGS: Open-vocabulary 3D scene understanding with context-aware Gaussian splatting. *arXiv preprint arXiv:2504.11893*, 2025.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Sen Wang, Kunyi Li, Siyun Liang, Elena Alegret, Jing Ma, Nassir Navab, and Stefano Gasperini. Visibility-aware language aggregation for open-vocabulary segmentation in 3D Gaussian splatting. In *ICCV 2025 Workshop on Wild 3D: 3D Modeling, Reconstruction, and Generation in the Wild*, 2025.

Shaoxiang Wang, Shihong Zhang, Christen Millerdurai, Rüdiger Westermann, Didier Stricker, and Alain Pagani. Inpaint360gs: Efficient object-aware 3d inpainting via gaussian splatting for 360deg scenes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 117–127, 2026.

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. In *Advances in Neural Information Processing Systems 37 (NeurIPS-24)*, pp. 19114–19138, 2024.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *18th European Conference on Computer Vision (ECCV-24)*, pp. 162–179, 2024.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. *arXiv preprint arXiv:2311.11666*, 2023.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Supplementary Material for "3DISCO: 3D Open Vocab Instance Segmentation With Clean Objects Via Gaussian Splatting"

Anonymous ICLR 2027 Submission

A Implementation details

A.1 Instance mask generation

Our method builds on 2DGS Huang et al. (2024). For each training frame, we prompt Qwen3-VL-8B-Instruct (FP8) Team (2025) to name the visible objects and pass these names as text prompts to SAM3 Carion et al. (2025), which returns an instance mask for each named object. To reduce redundancy, a mask is removed if more than 90% of its area lies inside a larger mask or if its IoU with another mask exceeds 0.85; in both cases, the larger mask is kept.

A.2 Feature learning

Instance features and log-variances are rasterized on detached Gaussian means, rotations, scales, and opacities, so that $\mathcal{L}_{\text{inst}}$ updates only the features and log-variances (section 3.1). The opacity used to normalize the log-variance is also detached. For numerical stability, the log-variance is clamped to $[-4, 4]$ when computing the precision $\rho(\mathbf{p})$, and the confidence $\gamma(\mathbf{p})$ is obtained by clipping the detached precision to $[0.1, 1]$.

A.3 Mask correction

Mask correction runs once, after the first 1k iterations of feature learning (section A.5). Each predicted region $\mathcal{F}_c^v$ is compared with the masks of view v associated with it, i.e., masks that have at least 50% of their own area inside $\mathcal{F}_c^v$ or an IoU of at least 0.25 with it. Masks that contain other masks, such as a table with several objects on it, are excluded to avoid ambiguous containment, so only leaf masks are considered. A mask whose IoU with the region exceeds τ_{iou} confirms the region and is retained. The region indicates over-segmentation if several masks are associated with it, and under-segmentation if a single mask is associated with it but has IoU below τ_{iou}; in both cases, the offending masks are removed. A region with no associated mask is treated as a missed object only if the mean log-variance $s(\mathbf{p})$ over $\mathcal{F}_c^v$ exceeds τ_s. For each flagged region, we select three point prompts inside $\mathcal{F}_c^v$: the first at the maximum of its distance transform, i.e., the interior pixel farthest from the region boundary, and the other two by farthest-point sampling among the interior pixels. We query SAM Kirillov et al. (2023) with these prompts for a replacement mask, which is added unless its IoU with a larger remaining mask exceeds 0.85, as in section A.1.

A.4 Prototype-guided mask consistency

The two terms of the PMC objective in eq. (8) are

$$\mathcal{L}_{\text{bce}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{HW}\sum_{\mathbf{p}}\text{BCE}\big(\hat{A}_k(\mathbf{p}), \mathbf{M}_k(\mathbf{p})\big), \tag{A.1}$$

$$\mathcal{L}_{\text{dice}} = 1 - \frac{1}{K}\sum_{k=1}^{K}\frac{2\sum_{\mathbf{p}}\hat{A}_k(\mathbf{p})\,\mathbf{M}_k(\mathbf{p})}{\sum_{\mathbf{p}}\hat{A}_k(\mathbf{p}) + \sum_{\mathbf{p}}\mathbf{M}_k(\mathbf{p})}, \tag{A.2}$$

where $\mathbf{M}_k(\mathbf{p}) \in \{0, 1\}$ indicates whether $\mathbf{p} \in \mathbf{M}_k$ and BCE is binary cross-entropy, so that $\mathcal{L}_{\text{pmc}} = \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}$.

For efficiency, all instances share one projection. We compute the Gaussian-to-prototype similarities in a single matrix product, reuse the projected means, covariances, and tile intersections, and batch the K alpha-only rasterizations with zero opacity outside each Ω_k, which is equivalent to rendering each subset separately.

Table A.1: Hyperparameters, shared across all scenes unless noted. *ScanNet++ values; on LERF-OVS, $\tau_{\min}$ and τ_{conf} are calibrated per scene (section A.7).

Stage	Hyperparameter	Value
Feature learning	feature dimension D	8
	margin m	0.5
	loss weights λ_{intra}, λ_{inter}	1.0, 1.0
	unlabeled-attraction weight η	10^{-5}
	log-variance clamp range	$[-4, 4]$
	confidence clip range	$[0.1, 1.0]$
Reconstruction	aspect-ratio weight λ_{ar}	1.0
	maximum axis ratio κ	3.0
Clustering	DBSCAN stage 1 (cosine): ϵ, min. samples	0.02, 3
	DBSCAN stage 2 (Euclidean): ϵ, min. samples	0.2, 25
	predicted-region threshold τ_{pred}	0.95
Mask correction	association: inside fraction, IoU	0.5, 0.25
	IoU threshold τ_{iou}	0.8
	log-variance gate τ_s	-1
	point prompts per region	3
	redundancy IoU	0.85
PMC	membership threshold τ_{pmc}	0.6
	loss weights λ_{bce}, λ_{dice}	0.2, 0.5
	relative pruning thresholds (opacity, scale)	0.5, 0.5
Labeling	view agreement τ_a	0.6
	views per object T (LERF-OVS / ScanNet++)	20
	context weights μ, β	0.2, 0.1
	retrieval thresholds $\tau_{\min}$, τ_{conf}	0.6, 0.615*
Editing	selection similarity threshold	0.6

A.5 Training schedule

Training proceeds in four sequential stages:

1. **Reconstruction:** 30k iterations of 2DGS with $\mathcal{L}_{\text{render}}$ and $\lambda_{\text{ar}}\mathcal{L}_{\text{ar}}$.
2. **Feature learning:** 1k iterations of $\mathcal{L}_{\text{inst}}$ on the fixed geometry, followed by mask correction (section A.3) and 3k further iterations on the corrected masks.
3. **Mask consistency:** with the features frozen, 1k iterations of $\mathcal{L}_{\text{pmc}}$. Gaussians are then pruned if their opacity α_i or scale falls below the absolute floor $\alpha_{\min}$ or $\sigma_{\min}$, respectively, or drops below half of its value at the start of this stage.
4. **Refinement:** 5k iterations of $\lambda_{\text{render}}\mathcal{L}_{\text{render}} + \lambda_{\text{ar}}\mathcal{L}_{\text{ar}} + \lambda_{\text{d}}\mathcal{L}_{d}$ to refine appearance.

A.6 Instance validation and view selection

After training, for each cluster c and view v, the isolated mask $\mathcal{S}_c^v$ is obtained by rendering only $\mathcal{G}_c$ and thresholding its accumulated opacity at 0.7; it is the silhouette produced by the object's own geometry, whereas the predicted region $\mathcal{F}_c^v$ is the extent implied by the feature field. A view is kept as a candidate observation if $\text{IoU}(\mathcal{F}_c^v, \mathcal{S}_c^v) \geq \tau_a$, and clusters without any candidate observation are discarded. As a final cleanup, residual Gaussians of $\mathcal{G}_c$ that contribute mostly outside the silhouette are removed: a Gaussian is discarded if, averaged over the candidate views, more than half of its contribution falls outside $\mathcal{S}_c^v$.

Before reranking, geometric filters remove degenerate views: we discard views in which the object is largely cut off by the image border (in-frame score below 0.8) and views in which its mask covers less than 10% of the largest mask area of that cluster across views. The reranker then scores each remaining crop against the fixed text query and the top T views are kept, with $T = 20$.

A.7 THRESHOLD CALIBRATION

On ScanNet++, we use $\tau_{\min} = 0.6$ and $\tau_{\text{conf}} = 0.615$ for all scenes. On LERF-OVS, both thresholds are calibrated per scene by a grid search on a small subset of the ground-truth masks (3), maximizing the IoU between these masks and the masks predicted by the retrieval and reranking procedure of section 3.4.

A.8 HYPERPARAMETERS

Table A.1 lists all hyperparameters. Unless noted otherwise, they are shared across all scenes and datasets.

A.9 RUNTIME

On one NVIDIA RTX 4090, from mask creation to evaluation takes an average takes 36.89 minutes.

B EVALUATION PROTOCOL

ScanNet++ We evaluate on 10 scenes (00dd871005, 02f25e5fee, 0373d65cce, 04d0dc245b, 062e5a23a6, 07f5b601ee, 07ff1c45bb, 0c5385e84b, 0caa1ae59a, 0fe8539661). On ScanNet++, our method uses 3DGS Kerbl et al. (2023) instead of 2DGS, because the label transfer described below assigns each mesh vertex to the Gaussians whose volume contains it, whereas the flat disks of 2DGS have no extent along their normals and therefore almost never contain a vertex. The label set is the class list of the official ScanNet++ top-100 instance segmentation benchmark with "window" removed, since our evaluation is object-focused, and each scene is queried with the classes annotated in it. Since predictions are made per Gaussian but the ground truth is annotated on the vertices of the scene mesh, we transfer the predictions to the vertices. For each exported Gaussian, we take its mean, scales, rotation, and opacity in ScanNet++ world coordinates together with its predicted class, treating Gaussians without a prediction as background. For each vertex, we compute its Mahalanobis distance d to each Gaussian under that Gaussian's covariance, which measures the offset relative to the Gaussian's extent along each of its principal axes, and collect all Gaussians with $d \leq 3$. Each of these Gaussians adds a weight of its opacity times $\exp(-d^2/2)$ to its predicted class, so that near-transparent Gaussians and Gaussians far from the vertex contribute little, and the vertex is assigned the class with the largest total weight. Because background votes compete with the other classes, a vertex can also be predicted as background. For vertices that no Gaussian covers in this way, e.g., in gaps between Gaussians, we instead aggregate the same weights over the 8 Gaussians with the smallest Mahalanobis distance, so that every vertex receives a prediction. Finally, for each queried class, we compute the 3D IoU between the set of vertices predicted as that class and the set annotated with it. We report mIoU, the mean of these IoUs over classes, and mAcc@25, the fraction of classes whose IoU exceeds 0.25, both averaged over scenes.

C Additional qualitative results

C.1 Learned feature visualization

Figure E.4 shows a PCA visualization of the learned instance features. Across all scenes, distinct objects form compact, well-separated clusters with clearly delineated boundaries, even when they are small and tightly packed. In *Figurines*, the many toys are spread out across the table, and each is clearly assigned its own distinct instance feature. The more challenging cases are cluttered scenes where objects are in close contact. In *Ramen*, the bowl is densely cluttered with overlapping ingredients, yet each individual component remains clearly distinguishable: the eggs, the noodles, and smaller toppings such as the corn are each assigned their own feature, and the boundaries between them are preserved despite their close contact. The learned features are not limited to salient foreground objects. In *Teatime*, the model separates fine details such as the teabag tag from the cup it hangs on. Beyond the objects on the table in the foreground, the background tables and couches are also captured as separate instances, indicating that the model decomposes the entire scene rather than only its most prominent elements. Similarly to *Ramen*, *Waldo Kitchen* contains many small, cluttered items, such as the spices on the rack, the silverware, and the dishes in the sink. Each of these appears as its own unique instance in the PCA, clearly separated from its neighbors as well as from the shelving and countertop behind it. Together, these results show that the learned features are consistent within each object and discriminative across objects, even in cluttered scenes with many small, closely spaced instances.

C.2 Effect of PMC on extracted instances

Figs. C.1–C.6 show 3D object instances identified and segmented by 3DISCO. To isolate the effect of the PMC loss across eight objects from the *figurines* scene, we extract instances with our clustering procedure (section 3.2) and render each instance from five viewpoints. Without PMC, stray Gaussians belonging to nearby objects float around the extracted instance, because spatial proximity causes incorrect feature alignment during contrastive learning. Applying the PMC loss substantially reduces these artifacts: instance boundaries tighten, and the spurious Gaussians are removed, leaving only the Gaussians that belong to the queried object.

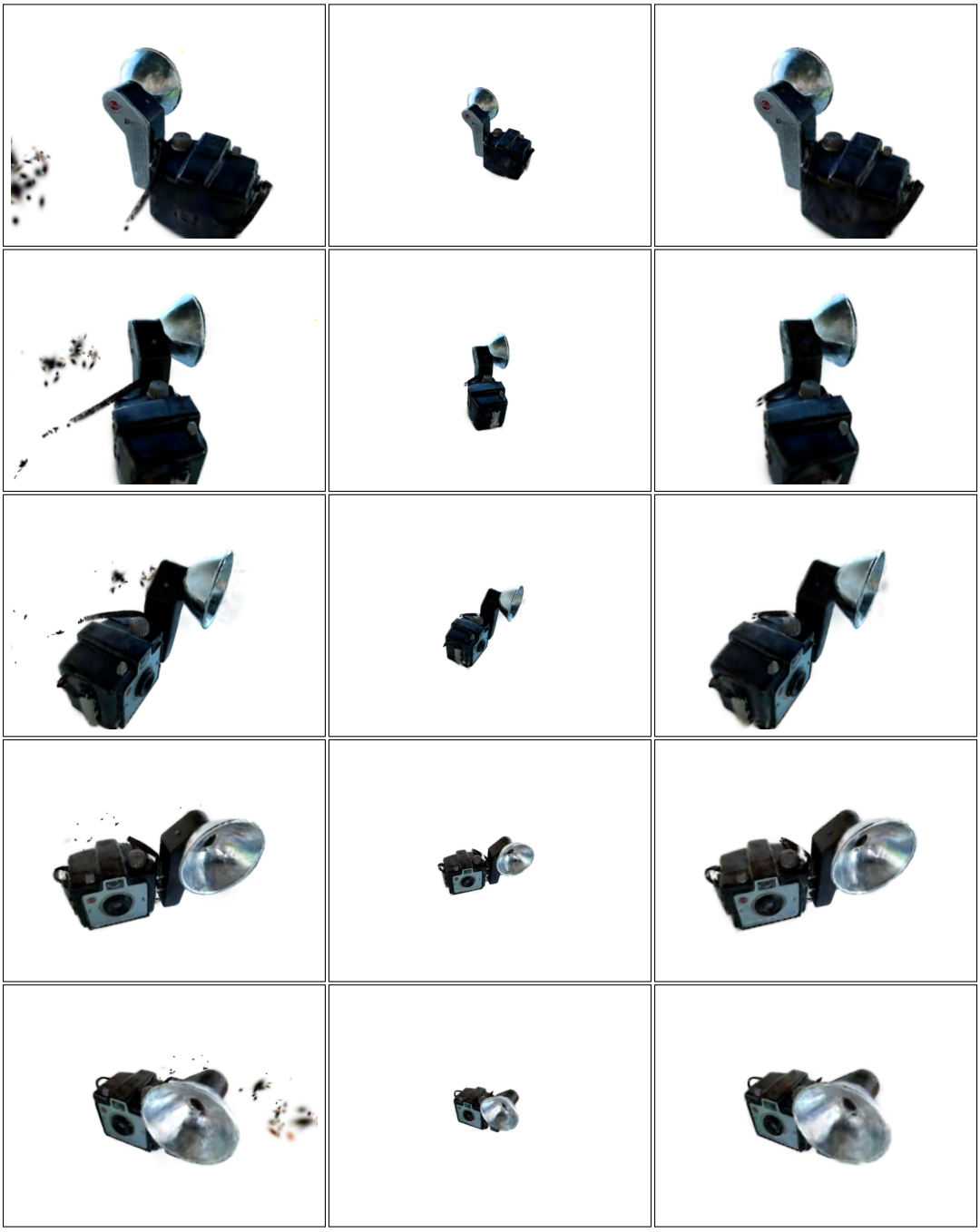
D Qualitative comparison with baselines

In Figs. D.1–D.8, we qualitatively compare 3DISCO to OpenGaussian Wu et al. (2024), InstanceGaussian Li et al. (2025), and Dr. Splat Jun-Seong et al. (2025) on open-vocabulary segmentation in LERF-OVS. The comparison highlights two capabilities: assigning Gaussians to the correct instance clusters, and retrieving these clusters when prompted with open-vocabulary queries. Among all methods, ours is the only one that consistently handles difficult retrieval cases and returns the intended Gaussian clusters without introducing unrelated objects.

This is most visible for small or visually subtle instances. In *waldo kitchen*, retrieving all four knives is particularly challenging: competing baselines either miss one or more knives or merge in Gaussians that do not correspond to the prompt, whereas our approach retrieves all four, with some minor artifacts. Baselines also struggle with the stainless-steel pots. OpenGaussian Wu et al. (2024) and InstanceGaussian Li et al. (2025) each retrieve part of the correct objects but also include a plastic container that does not belong to the prompt, whereas our retrieval matches the intended instances.

In *teatime*, the other methods have difficulty with the DALL-E sign, which hangs around the neck of a sheep and occupies a small printed region surrounded by competing textures. Only our approach retrieves the sign; Dr. Splat Jun-Seong et al. (2025) retrieves Gaussians around it without isolating the sign itself, and the other baselines miss it entirely.

These comparisons show that our method recovers the intended clusters in scenes where baselines either fail to retrieve the target instances or introduce unrelated objects.



(a) Without PMC (b) With PMC (c) Close-up with PMC

Figure C.1: Multi-view rendering of “camera”. Five views rendered without PMC (left), with PMC (center), and with PMC closer to the object (right).



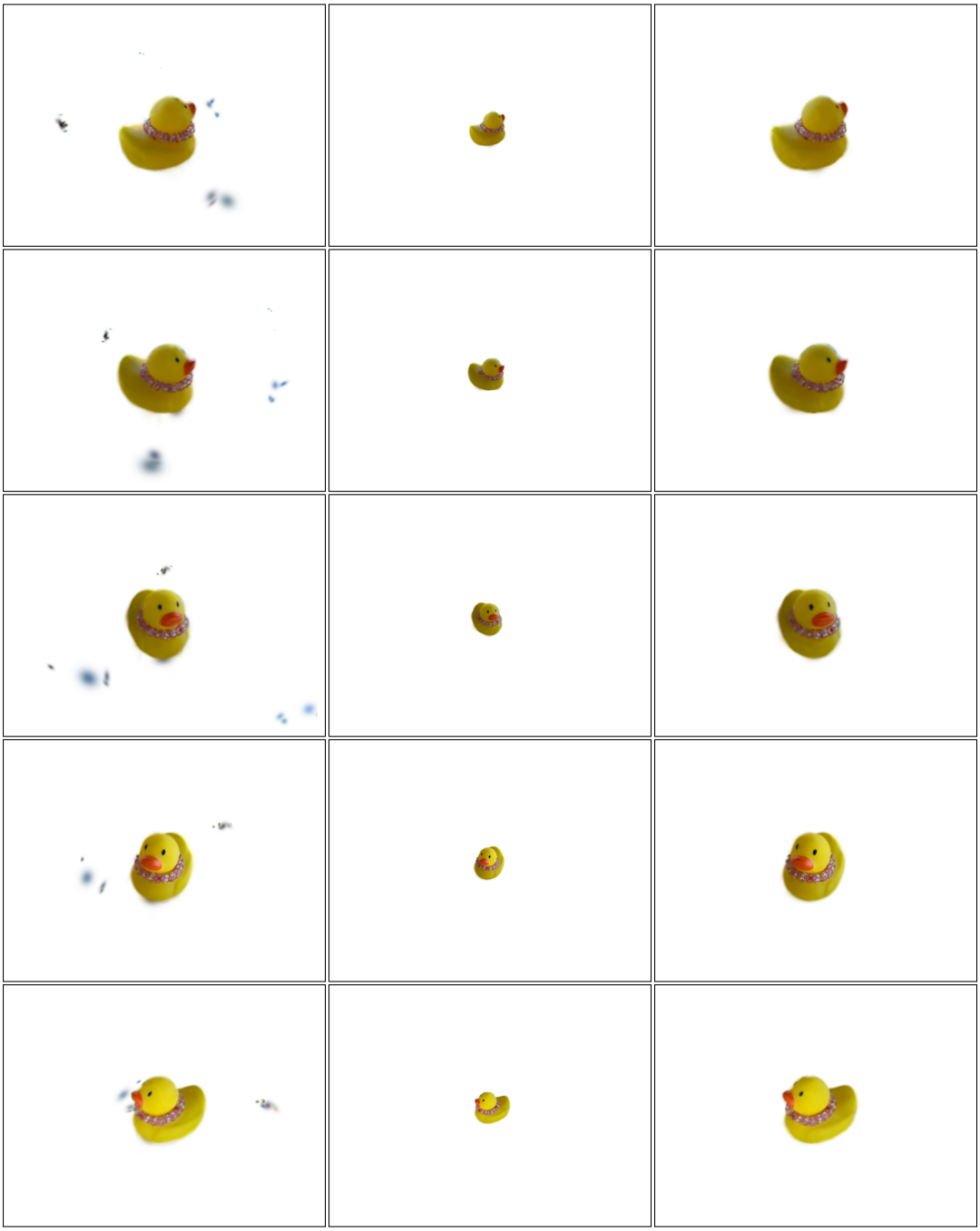
(a) Without PMC (b) With PMC (c) Close-up with PMC

Figure C.2: Multi-view rendering of “Rubik’s cube”. Five views rendered without PMC (left), with PMC (center), and with PMC closer to the object (right).



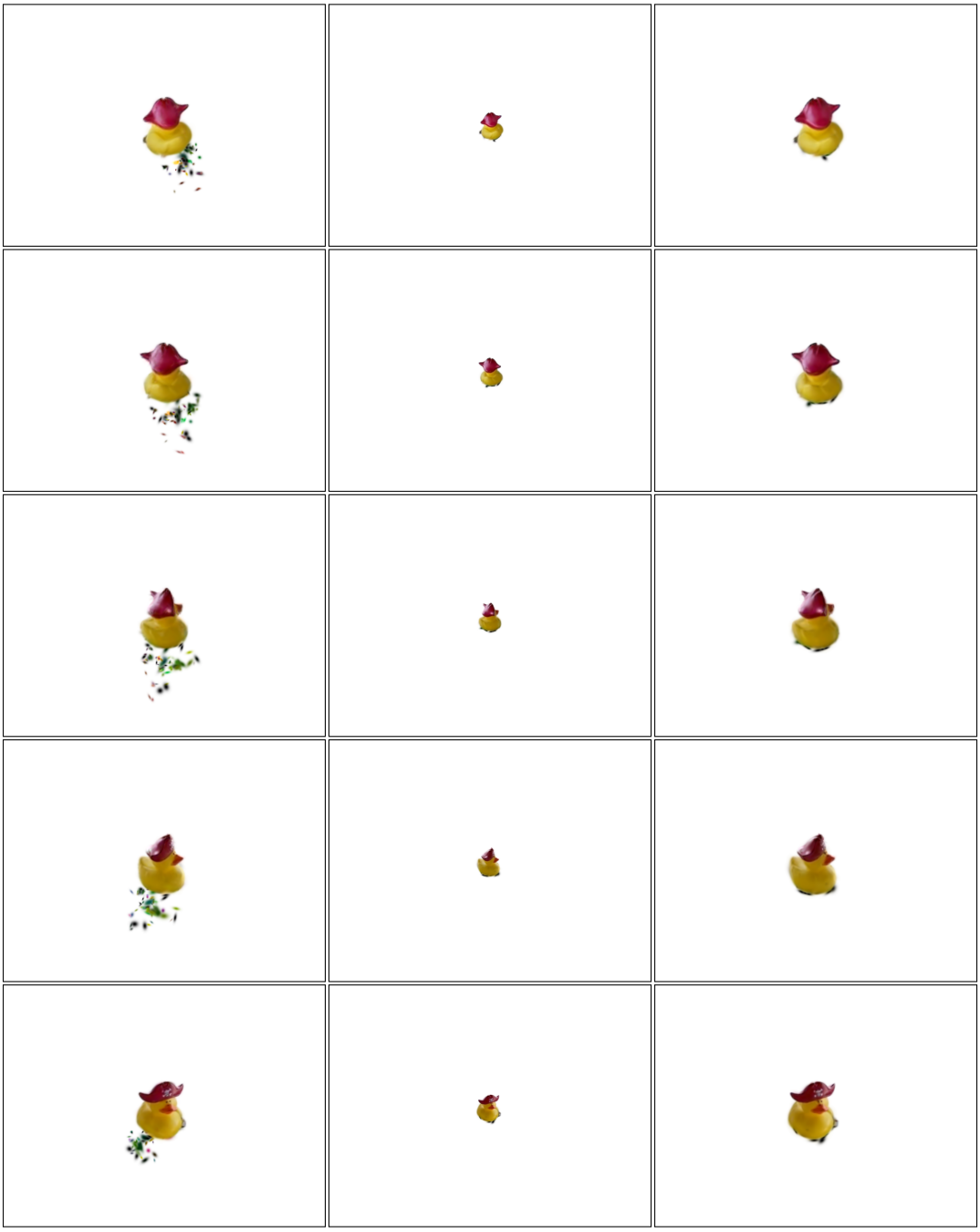
(a) Without PMC (b) With PMC (c) Close-up with PMC

Figure C.3: Multi-view rendering of “cup”. Five views rendered without PMC (left), with PMC (center), and with PMC closer to the object (right).



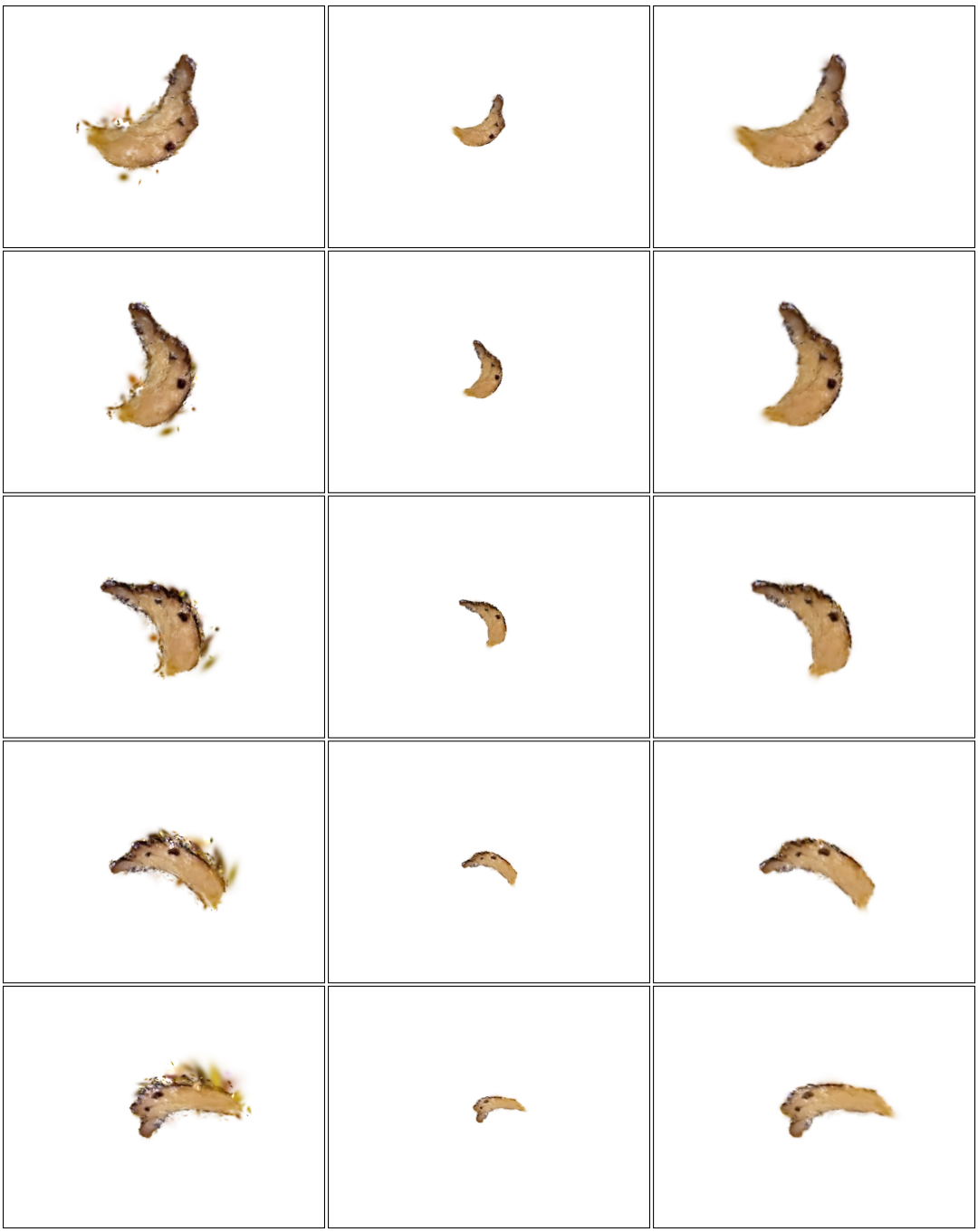
(a) Without PMC (b) With PMC (c) Close-up with PMC

Figure C.4: Multi-view rendering of "duck with buoy". Five views rendered without PMC (left), with PMC (center), and with PMC closer to the object (right).



(a) Without PMC (b) With PMC (c) Close-up with PMC

Figure C.5: Multi-view rendering of “duck with pirate hat”. Five views rendered without PMC (left), with PMC (center), and with PMC closer to the object (right).

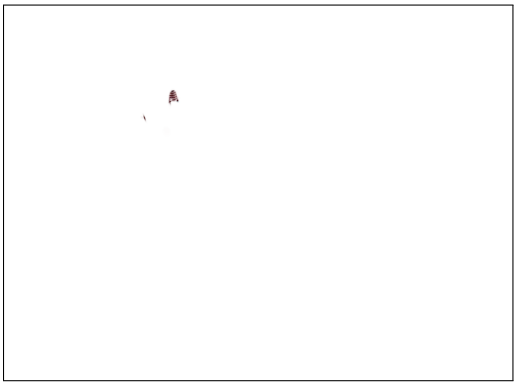


(a) Without PMC (b) With PMC (c) Close-up with PMC

Figure C.6: Multi-view rendering of “meat”. Five views rendered without PMC (left), with PMC (center), and with PMC closer to the object (right).



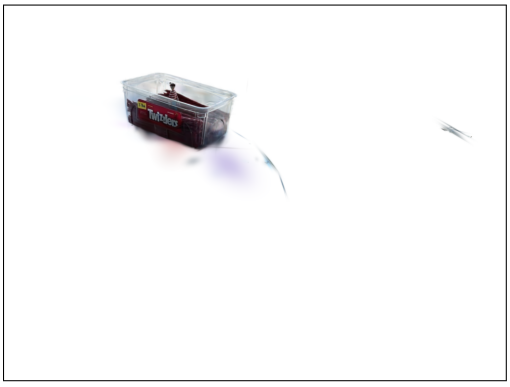
(a) Original



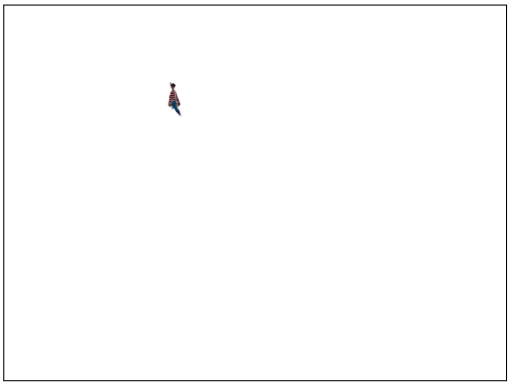
(b) Dr. Splat



(c) OpenGaussian



(d) InstanceGaussian

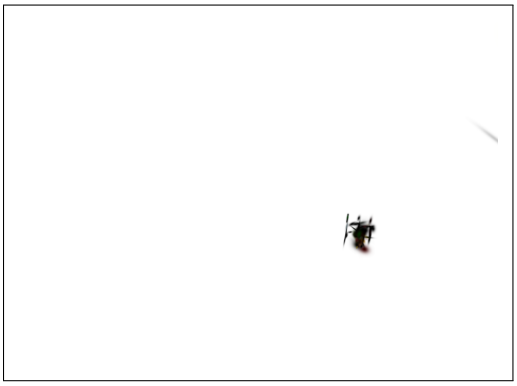


(e) 3DISCO (Ours)

Figure D.1: Qualitative 3D open-vocabulary semantic segmentation results for "Waldo" in the *Figurines* scene, comparing our approach with existing methods. We apply each method's evaluation setup as described by its authors.



(a) Original



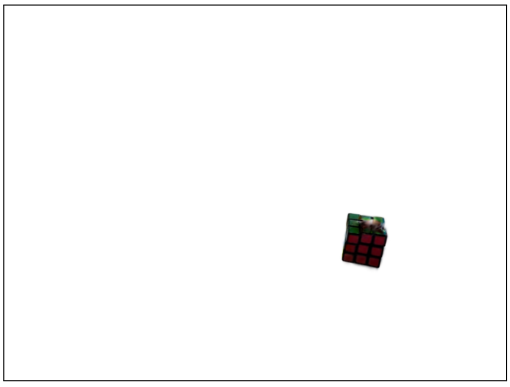
(b) Dr. Splat



(c) OpenGaussian



(d) InstanceGaussian



(e) 3DISCO (Ours)

Figure D.2: Qualitative 3D open-vocabulary semantic segmentation results for "Rubik's cube" in the *Figurines* scene, comparing our approach with existing methods. We apply each method's evaluation setup as described by its authors.



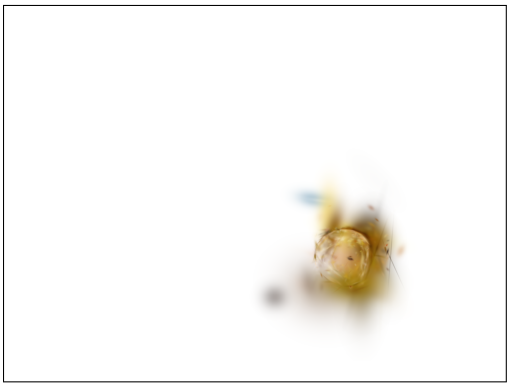
(a) Original



(b) Dr. Splat



(c) OpenGaussian



(d) InstanceGaussian



(e) 3DISCO (Ours)

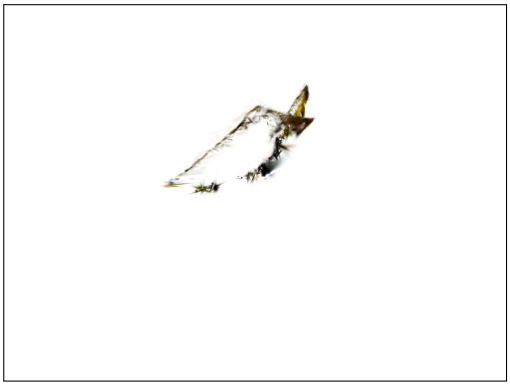
Figure D.3: Qualitative 3D open-vocabulary semantic segmentation results for "kamaboko" in the *Ramen* scene, comparing our approach with existing methods. We apply each method's evaluation setup as described by its authors.



(a) Original



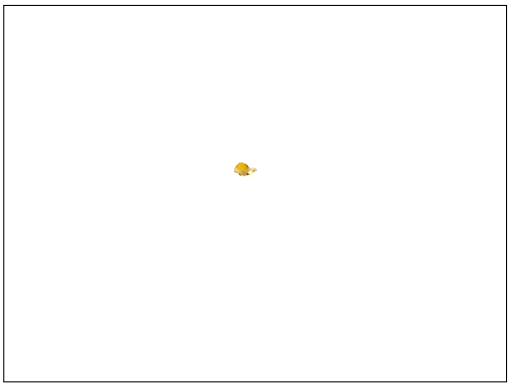
(b) Dr. Splat



(c) OpenGaussian



(d) InstanceGaussian



(e) 3DISCO (Ours)

Figure D.4: Qualitative 3D open-vocabulary semantic segmentation results for "corn" in the *Ramen* scene, comparing our approach with existing methods. We apply each method's evaluation setup as described by its authors.



(a) Original



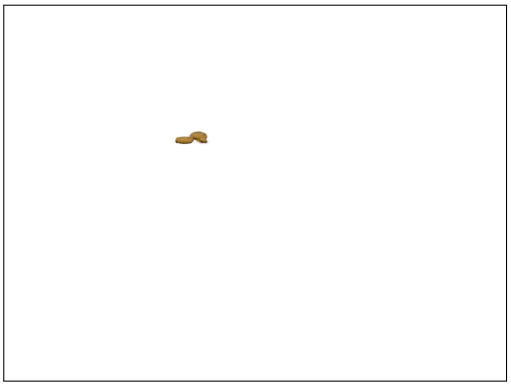
(b) Dr. Splat



(c) OpenGaussian



(d) InstanceGaussian



(e) 3DISCO (Ours)

Figure D.5: Qualitative 3D open-vocabulary semantic segmentation results for "three cookies" in the *Teatime* scene, comparing our approach with existing methods. We apply each method's evaluation setup as described by its authors.



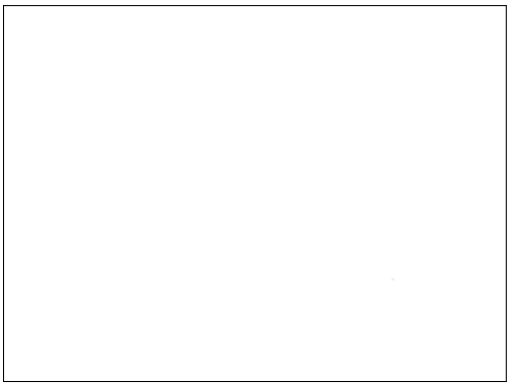
(a) Original



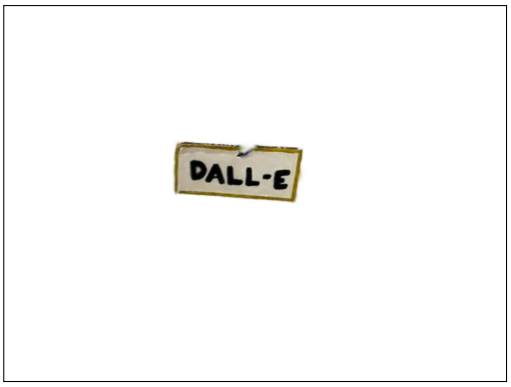
(b) Dr. Splat



(c) OpenGaussian



(d) InstanceGaussian



(e) 3DISCO (Ours)

Figure D.6: Qualitative 3D open-vocabulary semantic segmentation results for "DALL-E sign" in the *Teatime* scene, comparing our approach with existing methods. We apply each method's evaluation setup as described by its authors. InstanceGaussian assigned no Gaussians to this prompt that were visible in this view.



(a) Original



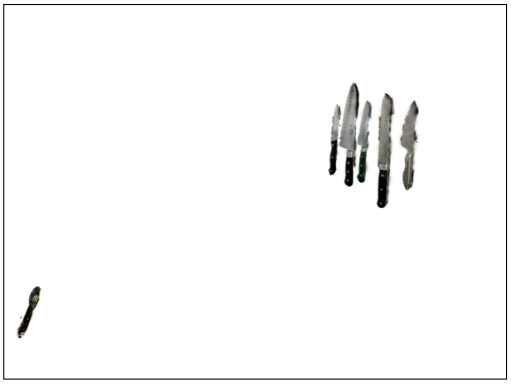
(b) Dr. Splat



(c) OpenGaussian



(d) InstanceGaussian

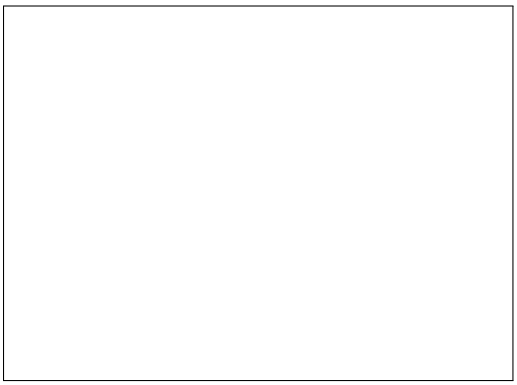


(e) 3DISCO (Ours)

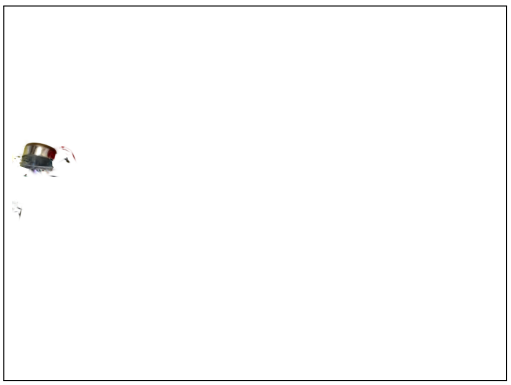
Figure D.7: Qualitative 3D open-vocabulary semantic segmentation results for “knife” in the *Waldo Kitchen* scene, comparing our approach with existing methods. We apply each method’s evaluation setup as described by its authors.



(a) Original



(b) Dr. Splat



(c) OpenGaussian



(d) InstanceGaussian



(e) 3DISCO (Ours)

Figure D.8: Qualitative 3D open-vocabulary semantic segmentation results for “stainless steel pots” in the *Waldo Kitchen* scene, comparing our approach with existing methods. We apply each method’s evaluation setup as described by its authors. Dr. Splat assigned no Gaussians to this prompt that were visible in this view.

E ADDITIONAL DETAILS ON THE DOWNSTREAM APPLICATIONS

E.1 INTERACTIVE SCENE EDITING

To evaluate the learned instance features independently of semantic labeling, we use them for interactive 3D scene editing (fig. 3). When a user clicks a pixel $\mathbf{p}$ on an object, we take its rendered feature $Q(\mathbf{p})$ and compute its cosine similarity to every Gaussian feature $\mathbf{q}_i$, selecting all Gaussians whose similarity exceeds 0.6. Objects that never appear in the same view are never contrasted and may therefore acquire similar features, so we additionally apply spatial DBSCAN to the selected Gaussians to isolate the clicked object. The selected Gaussians are then jointly translated or rotated in 3D. Selection thus requires neither the two-stage instance clustering of section 3.2 nor any mask propagation.

Figs. E.1–E.3 show edits in the *ramen*, *teatime*, and *waldo kitchen* scenes, where individual objects such as a spoon, chopsticks, a cookie, an apple, a knife, and a dish soap bottle are relocated. Because the learned features separate instances cleanly, the selection captures the whole object and nothing else, so relocating it leaves no residual Gaussians at its original location. The role of the PMC loss is visible in each edit: without PMC, noisy Gaussians from neighboring objects remain associated with the moved instance and appear as floating artifacts at both its new and its original location. With PMC, these artifacts are eliminated and the surrounding scene geometry is left intact.

By construction, moving a segmented 3D object leaves a hole where it originally sat, with 3DISCO or any other 3D segmentation approach. Existing scene-editing works fill these holes by inpainting the 2D renders Wang et al. (2026), which can be combined directly with our approach. Future work could use multiple inpainted images to fill the holes directly in the 3D Gaussian model.

Figs. E.2 and E.3 also show the strength of our approach in *instance* segmentation. The cookie in *teatime* and the knife in *waldo kitchen* both sit close to other objects of the same semantic class, yet our method cleanly separates them as individual instances, as the edits show.

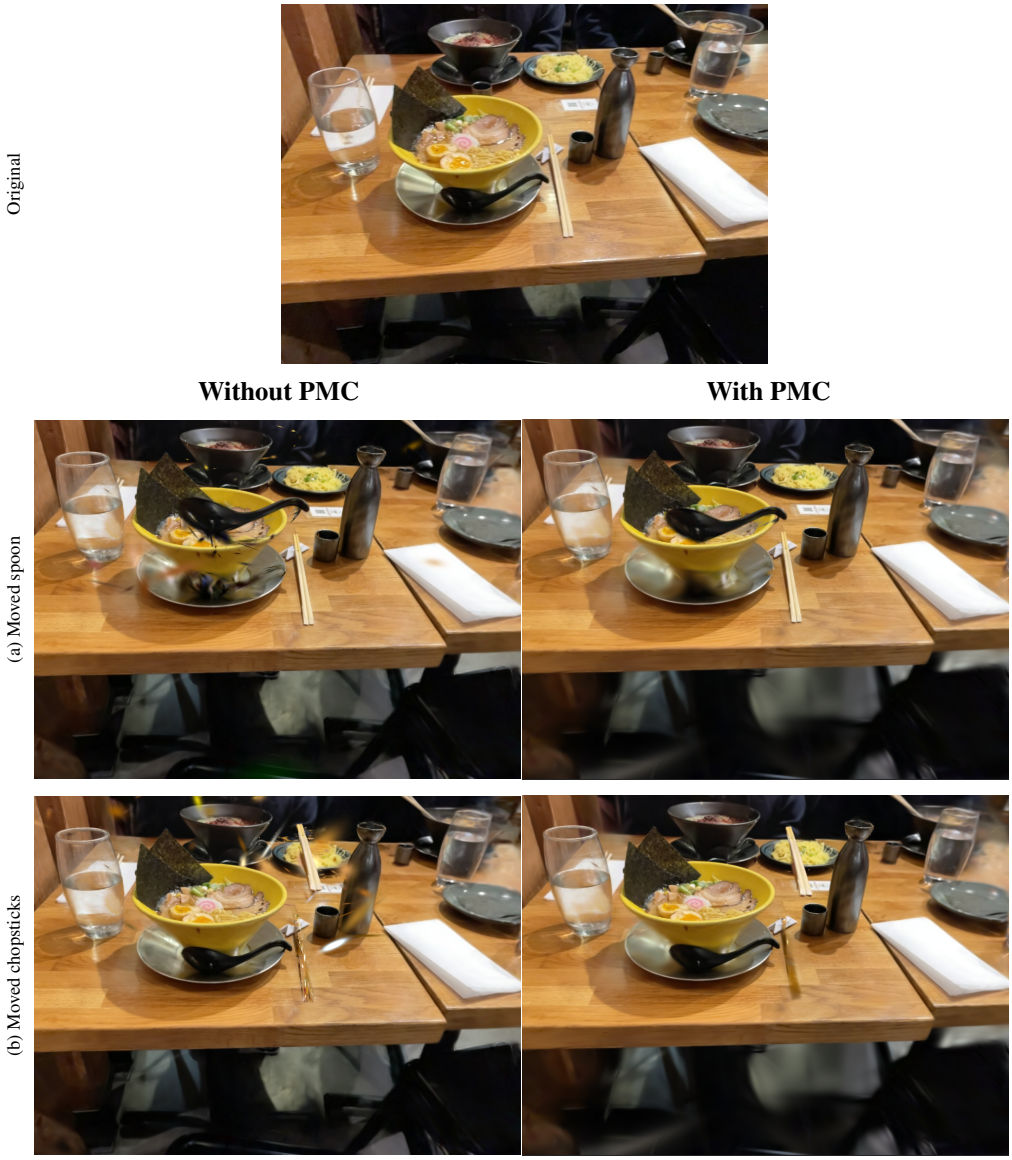


Figure E.1: 3D interactive scene editing on the *Ramen* scene. Top: original scene. Each row shows an edit without PMC (left) and with PMC (right).

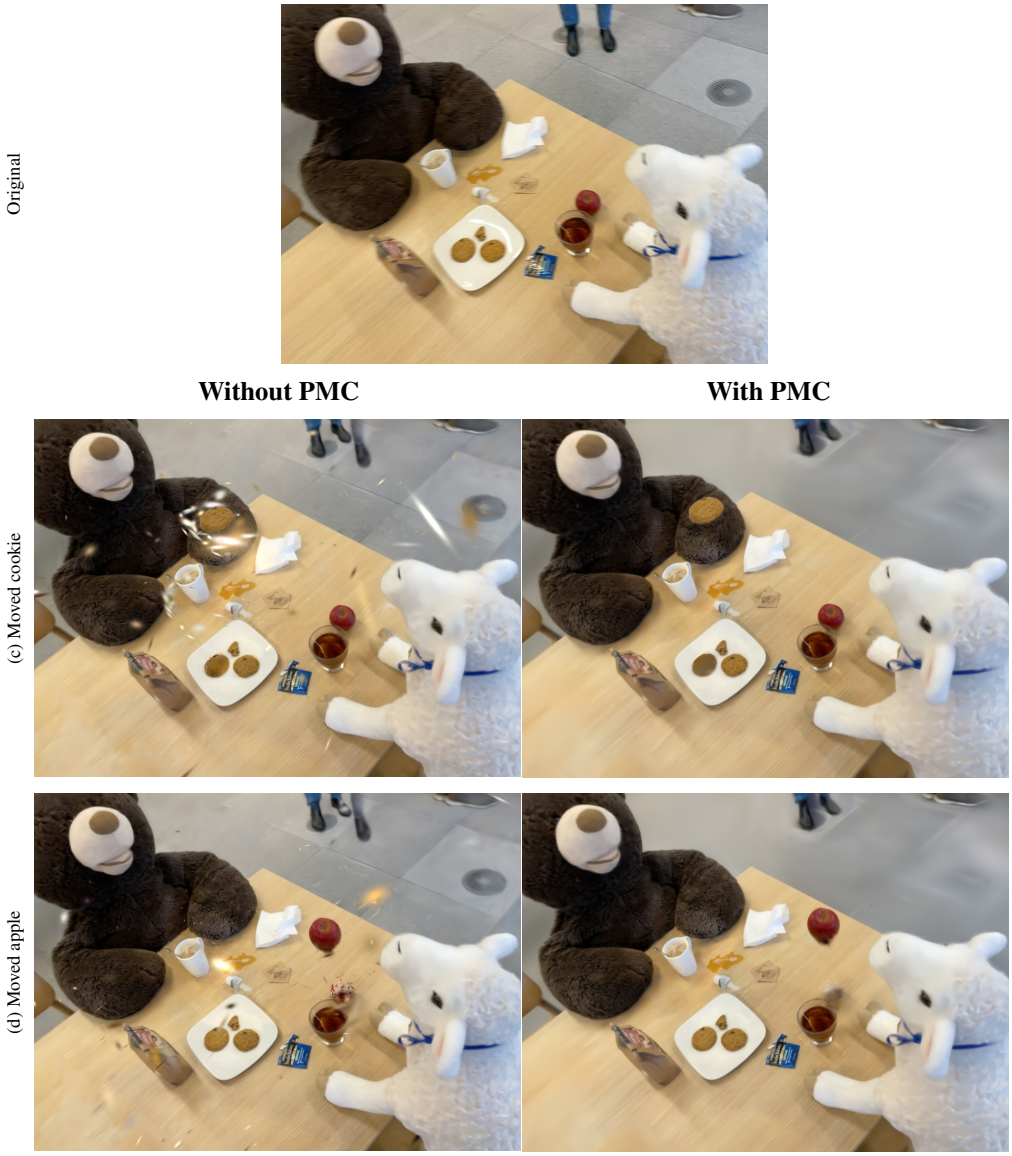
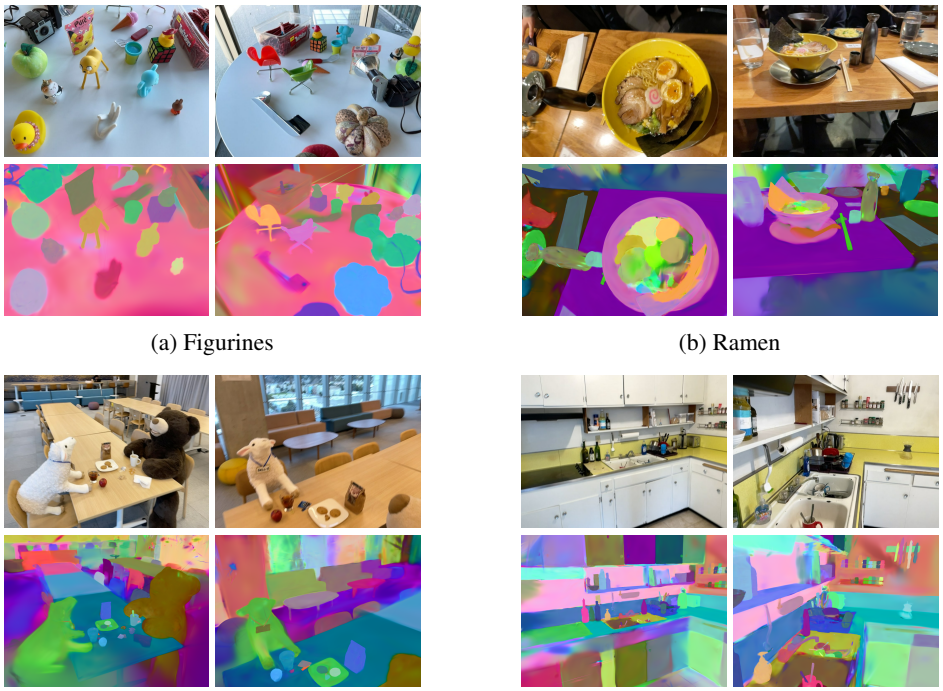


Figure E.2: 3D interactive scene editing on the *Teatime* scene. Top: original scene. Each row shows an edit without PMC (left) and with PMC (right).



Figure E.3: 3D interactive scene editing on the *Waldo Kitchen* scene. Top: original scene. Each row shows an edit without PMC (left) and with PMC (right).



(a) Figurines (b) Ramen

(c) Teatime (d) Kitchen

Figure E.4: PCA visualization of rendered instance features on the *Figurines*, *Ramen*, *Teatime*, and *Waldo Kitchen* scenes. Object boundaries are sharp and consistent across all scenes, and even small items, such as the corn in *Ramen* and the teabag tag in *Teatime*, remain distinctly separated from their surroundings.

E.2 Robot Manipulation

We build our manipulation demo on TiPToP (Shen et al., 2026), a modular open-vocabulary manipulation system that pairs off-the-shelf perception models with a GPU-parallelized task-and-motion planner. TiPToP's original perception pipeline operates on a single image from the wrist camera at the arm's initial pose, so every task-relevant object must be at least partially visible from that viewpoint. We replace this pipeline with 3DISCO: using an I2RT YAM arm, we capture 180 views of the scene with the wrist-mounted Intel RealSense D405, train 3DISCO on these views, and query the resulting Gaussians with the language prompt to obtain a segmented 3D point cloud of the target object, which is passed to TiPToP's planner in place of its original segmentation. To showcase this, we place a stapler behind a box so that it is hidden from the arm's initial camera pose (Figure 4). The unmodified TiPToP cannot ground the query "pick the stapler" because the stapler never appears in its single observation, whereas with 3DISCO the stapler is reconstructed from the additional views and segmented directly in 3D, allowing the arm to successfully pick it up.