DiSCO-3D : Discovering and segmenting Sub-Concepts from Open-vocabulary queries in NeRF

Supplementary Material

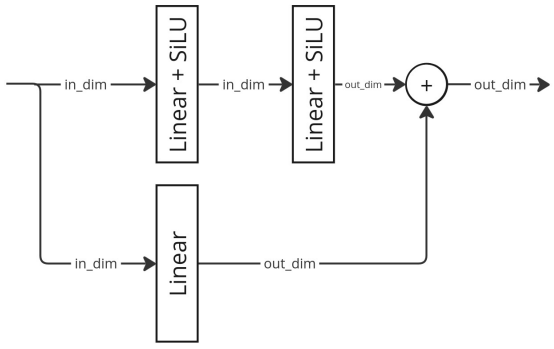

Figure 5. **Projector Architecture.**

6. DiSCO-3D

6.1. Additional Architecture Details

Some architecture details and minor contributions have been overlooked in the main paper that we want to cover here.

Projector Architecture. Although some USS methods implement a simple linear MLP projector [9], we follow SmooSeg [18] and decide to use a slightly more complex architecture depicted in Figure 5. It combines a non-linear MLP (with SiLU activations) with a linear layer serving as a residual connection. It is however to be noted that the impact (in both quality and time) is minimal, as evaluated in the following ablative experiments of the supplementary material.

Filtering Uncertain Samples. The prototypes of DiSCO-3D are updated each epoch via an EMA with a two-fold weighted average on both the density weights of the batch's samples and the prediction confidence (corresponding to the prediction probability of the class). In practise, we decide to further regularize this EMA by filtering out of the update process samples with very low weights (both density and confidence weights). For each sample k classified as i, if $D_{k,i} < 0.2$ or $w_k < 0.2$, the related feature f_k^{proj} will not participate in the update process.

6.2. LeRF Multi-scale CLIP Pyramid

Because CLIP outputs an embedding per image, rather than pixel-wise embeddings, it is not trivial to encode a scene as a CLIP feature field. While some methods work on adapting CLIP to pixel-wise embeddings [8, 19, 36, 38] (for instance, OpenSeg, which is a feature field used in our paper proposes a CLIP model adapted for dense tasks such as segmentation), LeRF proposes to pass image patches of different sizes into CLIP to produce a multi-scale pyramid used as supervision material. Regarding the LeRF model in itself, a scale parameter is added as input to the feature decoder and the training is done by randomly sampling scales across the pyramid for each sampled ray and retrieving the associated CLIP feature. During inference, the relevancy related to a query is computed for a pre-defined number of different scales and we display the relevancy heatmap of the scale resulting in maximum global relevancy, as done in Figure 7, Figure 8 and Figure 16.

In order to accommodate DiSCO-3D to this multi-scale pyramid when plugging into LeRF, several small modifications are made on the CLIP branch (no changes on the DINO branch because DINO produces pixel-wise embeddings). For each sample at each epoch, we decipher the associated CLIP embedding to be used for the computation of $\mathcal{L}_{irr}$ and $\mathcal{P}^{CLIP}$ by choosing the scale which outputs the maximum similarity to the user's query. This computation is performed by evaluating the similarity on a discrete number of scales, as done in LeRF inference (except we use the per-sample maximum similarity rather than per-image).

Note that when we use an empty query (i.e. when doing unsupervised semantic segmentation), CLIP prototypes can be computed using random scales for each samples. Multi-scales prototypes (which stores an average CLIP embedding per scale) has been tested, with a minimal increase of performance for an important increase of compute duration.

6.3. Notion of Confidence in DiSCO-3D

Although DiSCO-3D performs open-vocabulary segmentation (with hard class assignment rather than relevancy computation as in LeRF and OpenNeRF), confidence scores can be obtained by using the probability distributions D after the softmax operation. These scores define how similar each sample's post-projection feature is to its associated prototype compared to the other prototypes. Figure 6 illustrates a confidence heatmap for the query "door". The predictions are globally confident, which is normal as DiSCO-3D encourages high confidence by design (especially with the β scheduling defined in subsection 3.3). However, we can notice less confidence on the door edges, and especially on the narrow window at the left of the door, which is rather coherent as it could arguably not be considered as part of the door.

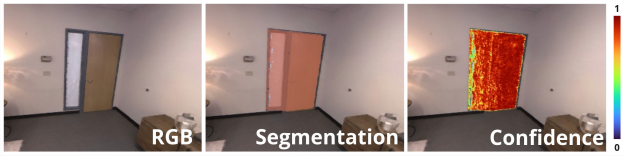

Figure 6. **Segmentation Confidence of DiSCO-3D.** The query is "door".

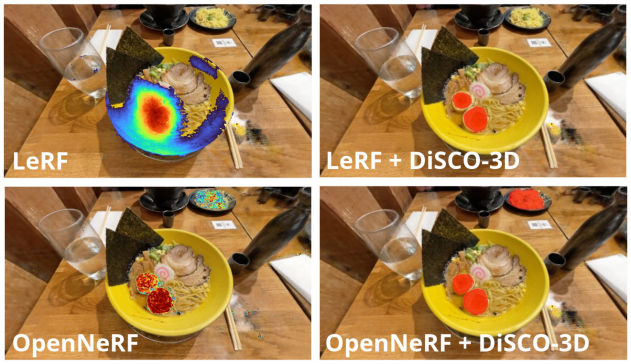

Figure 7. **Limitation** #1**.** By querying "Eggs", the LeRF and OpenNeRF baselines makes different prediction, both regarding the responding objects and their precision. While DiSCO-3D can "repair" segmentation imprecision via the DINO features, it is dependent of the open-vocabulary expressivity making the OpenNeRF+DiSCO also segment the background dish even though it does not seem to contain eggs.

6.4. Limitations and Failure Cases

Here, we discuss and illustrate a number of limitations inherent to our method.

Feature Field Quality Dependent. First of all, we mentioned the dependency of our method to the pre-trained feature field performance. Since this field provides input features for both segmentation and open-vocabulary queries, inaccuracies can negatively impact the results, as illustrated in Figure 7 and Figure 8. Regarding the open-vocabulary field, errors are common due to the limited quality of 2D open-vocabulary models and inaccuracies in NeRF's 3D projection, often caused by imprecise camera poses. These errors can lead to unexpected query results—either an excessive number of objects being labeled as relevant (e.g., the "Eggs" example in Figure 7) or a failure to correctly interpret some queries, especially when they regard abstract concepts, preventing DiSCO-3D from segmenting the intended sub-concepts. An example of the latter issue is shown in Figure 8, where the query "Art" fails to recognize the painting while incorrectly identifying seemingly random parts of the scene. This confusion propagates through the model, leading to incorrect segmentations. The projector feature field (e.g. DINO) can also suffer some issues which can have an impact on DiSCO's performances. Depending on

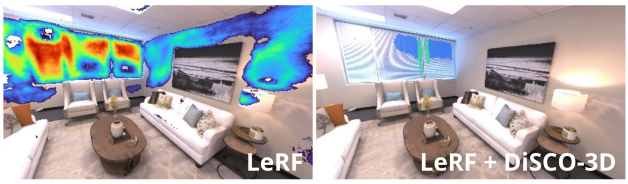

Figure 8. **Limitation** #2**.** We query "Art", which is incorrectly detected in LeRF, resulting in an irrelevant segmentation of parts of the windows rather than the painting.

the used encoder, some models like DINO tend to produce features which describes the scene at object parts-level rather than object-level. This can lead sometimes to over-segmentation of sub-concepts. Although this may be useful in certain applications (eg. object decomposition), this phenomenom is not wanted in OV-SD and this is why we proposed the $\mathcal{L}_{proto}$ to reduce this over-segmentation.

Although these issues originate from the input feature fields not introduced by our method, we can derive a few perspectives to improve the performances, which can be ordered in two classes. First, we can simply improve the quality of the feature fields, notably by using newer better image encoders, as discussed in the next subsection. On the other hand, we can work on the robustness of DiSCO-3D to mitigate the described issues. Although major failures caused by the input feature fields are hardly solvable, architecture improvements could be studied to incorporate more 3D geometry coherency in the segmentation process.

Query-Specific Optimization. Contrary to similarity-based open-vocabulary NeRF methods which only rely on a forward pass of their model to process a user query, DiSCO needs an optimization process of both the projector and the prototypes for each query to perform segmentation. However, we insist that the optimization is very fast, necessitating only very few and fast epochs to converge. Indeed, we typically achieve convergence in less than 100 epochs of approximately 20ms each, averaging a standard training of 2s. In Figure 9, we display the evolution of the segmentation during the optimization process. While this per-query optimization limits for now true real-time processing, we believe the optimization to be fast enough for the method to be truly useful and applicable in real-life scenario.

6.5. Extension to other feature fields

We introduced in subsection 3.5 the possibility to use different feature fields, as long as we have a queriable feature field to serve as the query latent space and a spatially precise one to serve as input to the projector. In the main paper, the query feature space has been tested only with a multi-scale CLIP and the dense OpenSeg while the input to the projector has been respectively a DINO and

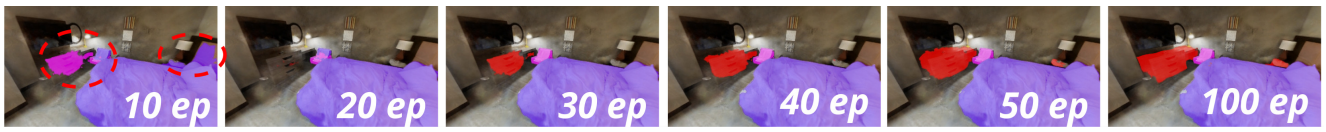

Figure 9. **Optimization Timelapse.** In average, one epoch takes **22ms**, resulting in a training of 200 epochs in $\sim$ 4**s**. The query is "furniture".

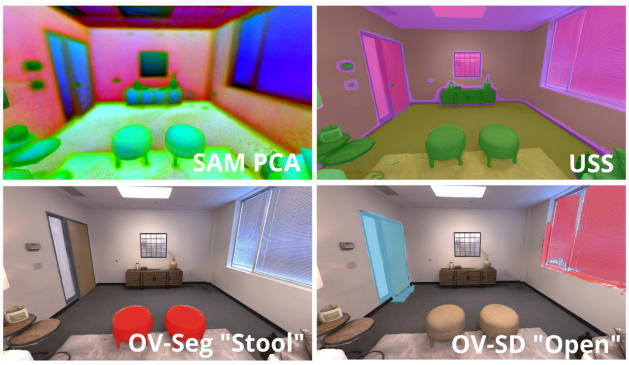

Figure 10. **SAM Feature Field.** We replace the DINO feature field in LeRF by a SAM feature field and demonstrate its capacity to perform USS, OV-Seg and OV-SD.

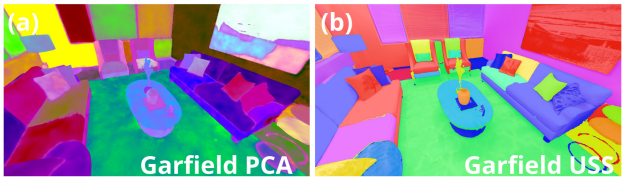

Figure 11. **Garfield Feature Field.** We use Garfield (SAM Masks outputs) as the segmentation field and perform USS. Note that Garfield being an instance feature field, it cannot be used as a replacement for DINO to perform OVSeg and OV-SD. USS also cannot be entirely considered as semantic segmentation.

OpenSeg feature field.

Regarding the segmentation feature field, there exists a large range of precise image encoders that can be injected into a feature field. For instance, Figure 10 shows an example of DiSCO-3D applied on a modified LeRF where the DINO is replaced by the image encoder of SAM (without the decoder). Because SAM is also quite spatially precise (as shown in the PCA), it can successfully be used to perform any of the 3 proposed tasks (OV-SD, OVSeg and USS). In Figure 11, we display another example of segmentation feature field by using a Garfield feature field. Garfield is a method producing a multi-scale feature field using SAM segmentation masks and contrastive learning. However, for better understanding, we limit here the Garfield feature field to mono-scale segmentation. This results in extremely precise (but over-segmented) scene decomposition as shown in the PCA which can be used to perform unsupervised segmentation. However, it is important to note that SAM produces instance segmentation masks that are unaware of the semantics. Hence, they cannot be used to perform true USS, nor OV-Seg and OV-SD, but rather instance segmentation. Future works could focus on combining Garfield with previously introduced semantic fields to perform both semantic and instance segmentation at once.

Regarding the query feature field, we evaluated in the main paper two adaptations of the open-vocabulary model CLIP (LeRF and OpenNeRF). However, we could broaden the range of feature fields used for the query, using other open-vocabulary models for instance or change the modality of the query with other feature spaces (e.g. image queries with DINO encodings or user clicks with any feature, as shown in Figure 3 with CLIP).

7. Experiments

7.1. Hyperparameters

In this section, we list the used hyperparameters for our different experiments (both quantitative and qualitative) of the article.

Base Nerfacto Model Configuration. We use most of the default Nerfstudio setup including with 16 hash grids and a dictionary size of 2^{19}. For quantitative experiments with Replica's synthetic scenes, we use a feature size of 2 and bump it to 8 for more complex real scenes used in qualitative experiments. We also disable the camera optimizer and appearance embedding on Replica, as they overcomplexify the models for no real gain in segmentation performance. Finally, for all indoor scenes, we reduce the far plane to the scene's maximum dimension.

Pre-trained Feature Fields. The configurations for the feature fields follow standard setups defined by LeRF. We use a set of hashgrids disjoint from the Nerfacto grids of 24 levels (2^{19} dict size) with 8 feature size and resolutions ranging from 16 to 512. Following both LeRF and OpenNeRF, we use respectively an OpenCLIP base (ViT-B/16) and a CLIP large (ViT-L/14). The DINO used for LeRF is a ViT-S/8.

DiSCO-3D Hyperparameters. Unlike 2D USS methods that cluster DINO features, which are notoriously sensi-

tive to hyperparameter tuning and prone to failures on diverse datasets, DiSCO-3D benefits from NeRF's scene-specificity, making it more robust. However, while DiSCO-3D has relatively few hyperparameters, certain parameters still require careful adjustments.

- **Number of Prototypes.** We showed in subsection 3.4 that the chosen number of relevant prototypes is not crucial as long as there are enough to describe each sub-concept. Regarding the number of irrelevant prototypes, we use three irrelevant prototypes in all experiments. However, this is not a sensitive hyperparameter, only requiring sufficient expressivity to encompass diverse irrelevant objects.
- **Projector.** The projector follows the introduced architecture and uses linear layers which both have as hidden dimension and output dimension the input dimension (ie. the feature dim). A dropout of probability $p = 0.2$ is also applied on it.
- β **Scheduling.** The β hyperparameter and its linear scheduling configuration, which affect the sharpness of the probability distributions, also exhibit minimal impact across scenes as long as we keep a sound configuration. In the experiments, we use an initial value of 0.5, linearly decreasing to 0.1 over the training.
- **Thresholds.** The threshold for $\mathcal{L}_{proj}$ has little impact and is fixed at 0.5, but $\mathcal{L}_{irr}$'s threshold is more crucial and depends on the feature field. Indeed, OpenNeRF with its OpenSeg encoder generally outputs higher relevancy scores than LeRF with CLIP. To accommodate this difference, we use distinct thresholds: thresholds are set at 0.55 for OpenNeRF and 0.5 for LeRF.
- **Loss Weights.** We balance the three proposed losses to optimize the trainings and obtain $\mathcal{L} = w_{proj}\mathcal{L}_{proj} + w_{irr}\mathcal{L}_{irr} + w_{proto}\mathcal{L}_{proto}$ with $w_{proj} = 20$, $w_{irr} = 1$ and $w_{proto} = 0.5$.

Finally, for all experiments, the model is trained for solely 200 epochs with an EMA decay factor $\alpha = 0.998$ and an Adam optimizer of learning rate exponentially decreasing from $1e-2$ to $1e-4$ across the optimization.

7.2. Ablative Experiments on USS (subsubsection 4.3.2)

In the main paper, in order to propose a solution for the OV-SD problem adapted to Neural Fields, we began by proposing a novel USS NeRF-based method as, to the best of our knowledge, no existing USS method exist for the NeRF representation. In this section, we perform ablative experiments to evaluate the contributions of the different modules of our USS branch and show the results in Table 5. We evaluate the full method on USS (i.e. with no user query) and then either modify the projector (we test a simple linear MLP) or disable separately various components: the linear scheduling of β and the two different ponderations of the prototypes update EMA process.

We note that the selected architecture used in DiSCO-3D indeed presents the best results amongst the different versions. Each of the other versions outputs diminished results, ranging from minimal loss of performances when changing the projector to maximal degradation when foregoing both ponderations in the EMA process.

7.3. Open-Vocabulary Sub-concepts Discovery

Replica Sub-Concepts Dataset. The complete list of groupings of our extended Replica dataset can be found in Table 8.

Naive Baselines Visualization. In subsection 3.4, we quantitatively compared DiSCO-3D with two naive baselines designed for the OV-SD problem. These baselines use the same architecture and configuration as DiSCO-3D but differ fundamentally in their segmentation process. Instead of jointly performing OV-Seg and USS as in our method, they execute the two tasks sequentially, each following a specific order.

We refer to these baselines as "naive" because a straightforward approach to solving OV-SD might be to apply OV-Seg and USS successively. However, as demonstrated in the quantitative evaluation presented in the main paper, this approach has notable shortcomings. To complement these results, Figure Figure 14 provides a visual comparison using the query "light," which should correspond to the *window*, the *bed-side lamps* and the *ceiling lamps*.

For the OVSeg-to-USS baseline, segmentation performance is significantly reduced due to the spatial imprecision of open-vocabulary relevancy. This leads to two key issues: (1) irrelevant objects may be partially segmented due to relevancy spilling (e.g., a large part of the wall above the bed), and (2) relevant objects, such as the window, may be incompletely segmented because the computed relevancy does not fully encompass the object. In contrast, DiSCO-3D mitigates these issues by leveraging DINO features as input to the projector, allowing it to refine spatial precision and avoid these relevancy errors.

For the USS-to-OVSeg baseline, while the segmentation aligns better with the scene's geometry, the main issue lies in classification. Since USS is performed without query information, the resulting clusters are not structured according to the query. As a consequence, after OV-Seg filtering, objects that should be distinguished with respect to the query remain grouped together based on their overall similarity in the scene, leading to incorrect decomposition. Here, the ceiling with its lamps are segmented together with the window. Because the average CLIP embedding answers to the query, this grouping is considered a sub-concept in this naive baseline.

Additional Results and Analysis. We display in Table 10 and Table 9 additional metrics on the experiments of the

Decreasing β	Projector	Ponderation by D_k	Ponderation by w_k	mIoU ↑	mAcc ↑
✓	Full	✓	✓	**27.47**	**51.99**
✓	2 Linear Layers	✓	✓	27.12	50.88
✗	Full	✓	✓	26.52	50.41
✓	Full	✗	✓	26.17	50.59
✓	Full	✓	✗	26.02	50.09
✓	Full	✗	✗	16.77	43.68

Table 5. **DiSCO-3D Ablative Experiments for USS.**

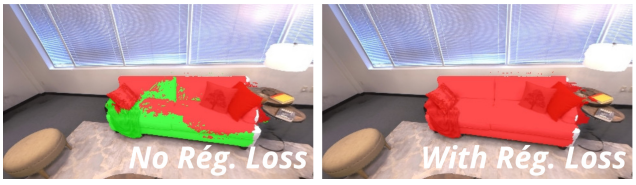

Figure 12. **Effect of the Regularization Loss.** Adding the regularization loss reduces over-segmenting (ie. describing single objects with more than one prototype). The query is "furniture".

$\mathcal{L}_{proto}$	N_{add}	0	2	5	10	20	$N = 10$
✓	Used N_{add}	-0.12	1.08	1.52	1.91	1.96	1.80
	PQ ↑	8.53	9.52	10.06	10.15	10.12	**10.19**
	mIoU ↑	8.81	10.45	12.38	12.70	12.59	**12.77**
	mAcc ↑	36.72	39.60	42.81	43.63	43.47	**44.29**
✗	Used N_{add}	-0.07	1.33	1.98	2.62	3.02	2.60
	PQ ↑	8.56	9.49	9.72	9.71	9.55	**9.77**
	mIoU ↑	8.77	10.27	12.13	**12.42**	12.30	12.35
	mAcc ↑	35.82	39.06	42.52	43.14	42.64	**43.36**

Table 6. **Additional metrics on the ablative on # of Prototypes** ($N = N_{GT} + N_{add}$).

main paper. We complete the given metrics by giving also the segmentation quality (SQ) and recognition quality (RQ) metrics, considered as sub-metrics of PQ such that :

$$PQ = \underbrace{\frac{\sum_{(p,g)\in TP} \text{IoU}_{pg}}{|TP|}}_{SQ} \underbrace{\frac{|TP|}{|TP| + 0.5|FP| + 0.5|FN|}}_{RQ} \tag{6}$$

Finally, while we chose to display in the main paper the mIoU and mAcc metrics computed on the relevant classes, we complete the evaluation here by augmenting those metrics' computations with the irrelevant class. Note that because the background class presents better quantitative metrics in average due to the sheer size of the irrelevant class against the relevant sub-concepts, its metrics are much higher and thus might bias the global results towards the background class.

Table 6 gives additional metrics for the ablative experiment on the number of prototypes (Table 2 in the main paper) and Figure 12 shows a visual examples on how adding the regularization loss reduces over-segmenting of objects.

Another example of using the CLIP Prototypes. In figure Figure 4 of the main article, we show results of *a posteriori* linking of the automatic sub-concepts with class names using the corresponding CLIP prototypes. Here, we dive deeper and provide another example in Figure 13 where we give the corresponding probability attributions of the top-10 semantic classes (amongst the 51) of each sub-concept. We query the scene for "furniture" and compute for each CLIP prototype the distances to each CLIP embedding of the 51 semantic classes. The probability distribution is then obtained by performing a softmax operation on the inverse of the distances (multiplied by a factor 100 to accentuate the sharpness of the distribution, as the distances between an image CLIP embedding and a text CLIP embedding are all rather close). Although 2 out of the 6 sub-concepts are not linked to the correct semantic classes, the 4 other correct classes are predicted with high confidence (up to 90.25% for the "stool" sub-concept class), showing the confidence of our model with unambiguous concepts. Regarding the incorrect predictions, we can first notice that the cushion class is the second most probable prediction with only 0.72% of differences in confidence. This result reflects that the associated CLIP prototype refers to an intermediate concept corresponding to a sofa-cushion, a cushion in itself not being a furniture while a cushion as part of a sofa can be considered as one. Similarly, the CLIP prototype corresponding to the armchair matches with the sofa at 52.57% and with a chair at 22.96%. This is consistent with the definition of an armchair: an intermediate concept between a sofa and a chair. The other incorrect sub-concept corresponding to the lamp is the less correct prediction, as once again the second probable prediction but with more differences in confidence. However, this error can be partly explained by the difficulty of the prediction as the lamp object in itself has a peculiar form less discriminative than the form of a sofa.

7.4. Open-Vocabulary Segmentation

We display some additional qualitative results in Figure 15, both with singular queries (composed of precise classes and concepts) and multiple queries at once.

Relevancy Holes and Spilling. We stated in subsubsection 4.3.1 that DiSCO-3D is able to mitigate common open-vocabulary segmentation issues, namely relevancy spillings

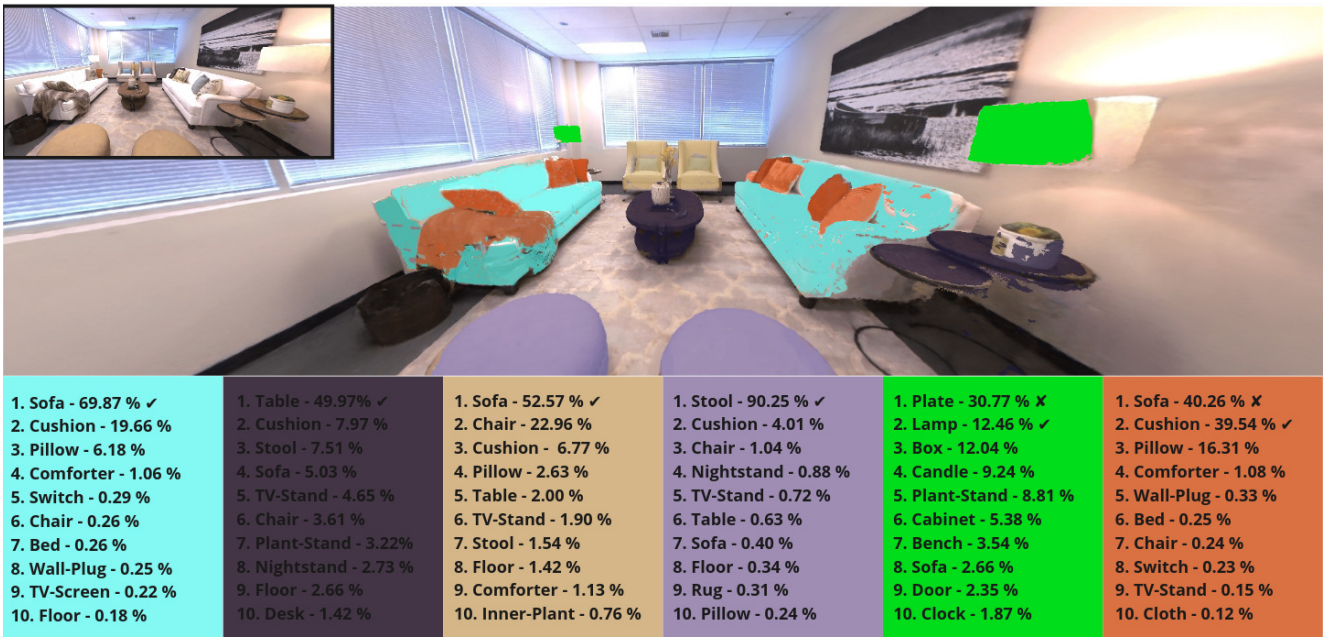

Figure 13. **Top-10 Class Labels Linking for every Sub-Concepts.** The query is "furniture".

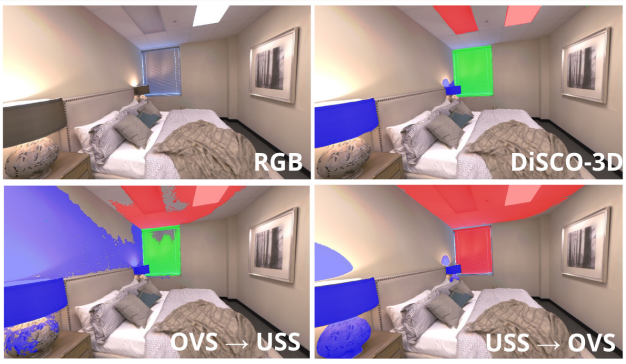

Figure 14. **OV-SD Example Naive Baselines vs DiSCO-3D.** The query is "light".

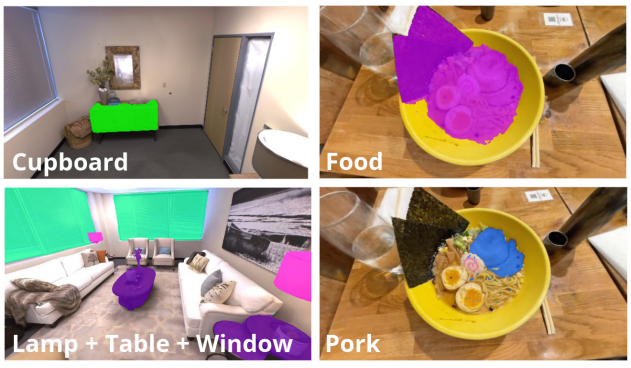

Figure 15. **Additional OVSeg Results.**

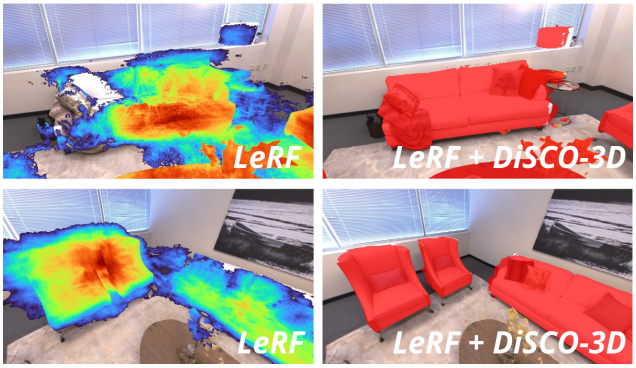

Figure 16. **Display of relevancy holes and spillings.** The queries of the fist and second lines are respectively "Furniture" and "Seatings" in the OV-Seg setting.

and relevancy holes. We illustrate this affirmation in Figure 16. Relevancy holes, displayed on the first lines, define areas where only parts of an object relevant to the query in theory responds well in practise. DiSCO-3D succeeds in completing the segmentation to encompass the whole object in the segmentation. Relevancy spilling rather relates to the opposite phenomenom, where irrelevant areas around a relevant object can be detected by the open-vocabulary models due to spatial imprecision. This is illustrated in the second line of the figure. DiSCO-3D also reduces this issue by focusing only on highly relevant areas and completing them.

Mono-Label Paradigm. Additionally to what we call

Method	*Mono-Label*	
	mIoU ↑	mAcc ↑
LeRF [12]	10.49	22.02
LeRF + DiSCO-3D	**13.43**	**28.37**
OpenNeRF [6]	19.08	**31.96**
OpenNeRF + DiSCO-3D	**20.76**	30.19

Table 7. **DiSCO-3D Quantitative Evaluation for OV-Seg in the *mono-label* paradigm.**

the *multi-label* paradigm (ie. each 3D point can be assigned zero or multiple labels based on independent query predictions), some methods such as [6] evaluate themselves on the *mono-label* paradigm where each point receives a single label corresponding to the most probable class amongst all queries. Regarding DiSCO, this translates into training 1 models with N_q simultaneous queries, meaning that this paradigm is a way to evaluate DiSCO's ability to handle multiple queries at once. Note that because this paradigm needs the labels to be non-overlapping and needs to cover the whole scene, it cannot be evaluated on the grouping dataset which has overlapping queries (eg. "furnitures" and "seating" have common sub-concepts). The *multi-label* setup is considered more challenging as it requires segmenting each class independently without relying on other class names as priors.

We report quantitative results of this paradigm in Table 7. Regarding LeRF, we notice improvements for every metrics and paradigms when adding DiSCO (+2.94 mIoUs and +6.35 mAccs respectively). This is because applying DiSCO to LeRF greatly improves the segmentation by reducing the relevancy spilling (as illustrated in Figure 7): it directly improves mIoU and also increases mAcc because reducing spilling in a paradigm where every point is labeled increases correct classification. Regarding OpenNeRF, whose segmentation performances are already much better than LeRF, integrating DiSCO slightly improves the mIoUs (resp. +1.68) at the expense of mAcc. This trade-off arises because DiSCO segments directly from features rather than relying on similarity maps like OpenNeRF. As a result, DiSCO provides better boundary refinement by leveraging additional information but introduces minor misclassification, particularly for small less frequent classes.

7.5. Unsupervised Semantic Segmentation

We display here two figures of 3D USS. Figure 17 shows an example on real 2D data (in particular the "Waldo Kitchen" scene from LeRF). On this latter figure, the "SmooSeg" baseline refers to the 2D method being trained on the multi-view images without injecting 3D inside the segmentation, while the "SmooSeg + NeRF" image is a semantic render from a NeRF model trained using the segmentation maps of the SmooSeg. As explained in subsubsection 4.3.2, 2D USS methods such as SmooSeg do not have multi-view consistency. This makes the training of a Semantic-NeRF hardly consistent, resulting in very noisy segmentations. Although multi-view inconsistent, SmooSeg actually performs well when doing per-image segmentation as illustrated in the figure. While some noise subsist, the results are semantically and spatially coherent. However, DiSCO-3D still produces better segmentation as it profits from multi-view information for more precise DINO features, thus better spatial precision of the segmentation (e.g. the bottles on the top left of the image). Note that no GrowSP results can be obtained as there is no available point cloud for these hand-captured images of real data. Similarly, figure Figure 18 displays 3D point cloud segmentation results (used for quantitative evaluation) on Replica. The obtained renders are consistent with previous observations, as SmooSeg lacks multi-view consistency once again. Although GrowSP gives better segmentation with actually more precise details (e.g. the background shelves) but there are several areas with unexpected spillings which degrades the segmentation.

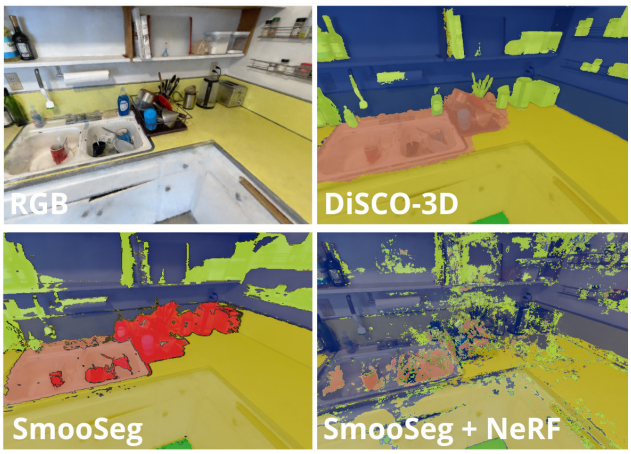

Figure 17. **Example of USS on real data.**

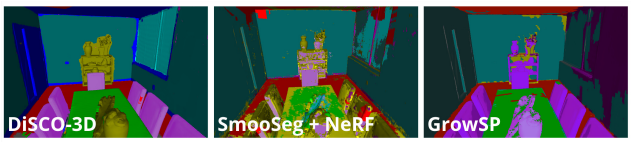

Figure 18. **USS on the 3D Point Cloud of Replica.**

ID	Concept	Associated Sub-Concepts
1	Furniture	chair, sofa, bench, stool, table, desk, cabinet, nightstand, shelf
2	Seating	chair, sofa, bench, stool, cushion, pillow
3	Sleeping	bed, comforter, blanket, pillow
4	Storage	cabinet, shelf, basket, box, desk-organizer
5	Walls	wall, panel
6	Floors	floor, rug
7	Ceilings	ceiling, vent
8	Entrances	door, window, blinds
9	Screens	tv-screen, monitor, tablet
10	Light	lamp, candle
11	Plants	indoor-plant, plant-stand
12	Art	picture, sculpture
13	Time	clock
14	Trash	bin
15	Soft	pillow, cushion, comforter, blanket, bed, cloth
16	Decor	sculpture, vase, candle
17	Organize	desk-organizer, box, basket
18	Airflow	vent
19	Work	desk, monitor, lamp
20	Eat	table, plate, bowl
21	Reflect	monitor, tv-screen
22	Warm	blanket, cloth
23	Watch	tv-screen, monitor, tablet
24	Tidy	desk-organizer, basket
25	Walk	floor, rug
26	Container	pot, bottle
27	Press	switch
28	Cushion	cushion, pillow
29	Displays	tv-screen, monitor, tablet
30	Rest	sofa, bed, pillow
31	Relax	sofa, chair, bed, cushion, pillow, blanket
32	Electronics	monitor, tablet, tv-screen, clock, camera
33	Lounge	sofa, bench, pillow, cushion
34	Dining	table, plate, bowl, bottle
35	Ventilation	vent, window
36	Opening	door, window, blinds
37	Comfort	pillow, cushion, blanket, bed, sofa
38	Portable	basket, box, tablet
39	Fragile	vase, sculpture, monitor, tv-screen
40	Heavy	table, cabinet, sofa, bed, sculpture

Table 8. **Replica Sub-Concepts Dataset.**

FF	Method	$\mathcal{P}_{CLIP}$						
		PQ ↑	RQ ↑	SQ ↑	mIoU$_{rel}$ ↑	mAcc$_{rel}$ ↑	mIoU$_{all}$ ↑	mAcc$_{all}$
LeRF	USS → OVS	4.76	32.48	11.62	6.52	22.54	29.89	44.12
	OVS → USS	5.99	30.47	13.41	8.71	21.44	39.82	49.38
	DiSCO-3D	**8.13**	**45.45**	**15.39**	**10.79**	**33.39**	**40.64**	**58.58**
OpenNeRF	USS → OVS	4.97	25.01	13.02	6.08	13.98	30.44	39.71
	OVS → USS	5.47	24.11	13.40	8.94	13.56	38.66	41.99
	DiSCO-3D	**8.65**	**39.36**	**17.84**	**10.82**	**19.24**	**40.57**	**49.88**

Table 9. **DiSCO-3D Quantitative Evaluation for OV-SD using $\mathcal{P}_{CLIP}$ matching.**

FF	Method	*Hungarian*						
		PQ ↑	RQ ↑	SQ ↑	mIoU$_{rel}$ ↑	mAcc$_{rel}$ ↑	mIoU$_{all}$ ↑	mAcc$_{all}$ ↑
LeRF	USS → OVS	6.94	53.96	11.72	10.92	35.57	34.70	55.60
	OVS → USS	7.48	44.09	13.24	10.90	27.11	41.50	54.74
	DiSCO-3D	**10.19**	**57.54**	**14.64**	**12.77**	**44.29**	**42.61**	**63.49**
OpenNeRF	USS → OVS	6.53	38.29	12.77	8.67	23.85	38.52	52.54
	OVS → USS	6.73	34.84	13.31	10.58	22.00	41.72	51.86
	DiSCO-3D	**10.49**	**52.42**	**16.65**	**12.69**	**29.06**	**42.23**	**55.82**

Table 10. **DiSCO-3D Quantitative Evaluation for OV-SD using Hungarian Matching.**