3D-AVS: LiDAR-based 3D Auto-Vocabulary Segmentation

Supplementary Material

S1. Abstract

This supplementary material provides additional details and analysis of our method. Implementation details and ablation studies are presented in Sec. S2 and Sec. S3. We discuss the fusion of outputs from different captioners and provide qualitative comparisons under challenging lighting conditions in Sec. S4. The impact of the vocabulary mapper on segmentation evaluation is analyzed in Sec. S5. Additional qualitative results are shown in Sec. S6. Finally, the nomenclature used throughout the paper is summarized in Sec. S7.

S2. Implementation Details

Image Captioner. We generate the image-based vocabulary with the xGen-MM (BLIP-3) [56] model using a temperature of 0.05, number of beams set to 5 and top-p set to the default value, 1. Our prompt is "*Briefly describe all objects in the <image>. Be concise. Only name the object names.*", where <image> refers to the image token.

Caption Module in Point Captioner. We follow LidarCLIP [17] to use the pre-trained caption model from ClipCap [34] as our captioning decoder. It decodes a CLIP feature vector to one caption.

Segmenter. We exploit OpenSeg [15] model and CLIP text encoder [39] as our image encoder h_{im}^{hr} and text encoder h_{tx}, respectively. We employ as our point encoder OpenScene [38] with its released OpenSeg pre-trained weights on nuScenes [4] and ScanNet [14], respectively. We also follow the inference phase of OpenScene where dot production is used as similarity metric SIM.

SMAP. We employ mean square error (MSE) as our loss function. The number of views J for SMAP is varying. During training, it is set the same as the number of images per point cloud, which is six for nuScenes [4] and variable for ScanNet [14]. During inference, it is set to 12 for nuScenes, indicating each point subset occupies a sector of 30 degrees. For ScanNet, each point cloud is divided into $0.5m \times 0.5m$ squares according to their x and y coordinates and then each square is treated as a view.

Training. To train SMAP, we use Adam [29] as the optimizer with an initial learning rate of $1e-5$. The learning rate is decreased following the polynomial learning rate policy [33] with a decay of 0.9. The SMAP has trained 20 and 10 epochs for nuScenes [4] and ScanNet [14].

Table S1. **Ablation study of Image Captioner on nuScenes [4] and ScanNet [14].** CN (Compound Nouns) means allowing to use continuous two or more words as a query, *e.g.* asphalt road.

Ablation Target	**Setting**	**nuScenes [4]**		**ScanNet [14]**	
		TPSS	**mIoU**	**TPSS**	**mIoU**
VLM	BLIP [26]	8.53	27.24	3.27	37.17
	RAM [65]	8.70	34.14	3.30	38.59
	xGen-MM [56]	8.72	33.75	3.37	40.27
CN	xGen-MM [56] + CN	**8.78**	**34.56**	**3.49**	**44.38**

Table S2. **Ablation study of Point Captioner on nuScenes [4].** T is a hyperparameter indicating the number of point cloud areas. LidarCLIP [17] employs a 2D global positional encoding to generate a single global caption, whereas our method utilizes a 3D local positional encoding combined with SMAP, allowing flexible control over the number of point cloud areas to caption.

Method	T in SMAP	Positional Encoding	TPSS	mIoU
LidarCLIP	-	2D global	6.25	20.58
Ours w/o. PE	12	✗	8.61	30.89
Ours	1	3D local	6.32	17.94
	6		8.66	29.45
	12		**8.80**	**33.42**
	24		8.77	32.96

Table S3. **Ablation study of Point Captioner on ScanNet [14].**

	Pillar Size (m^2)	Num. of Pillars per Scene	TPSS	mIoU
Ours	0.5×0.5	87.3	**3.71**	**29.25**
	1×1	27.6	3.53	22.58

S3. Ablation Study

Ablation studies are conducted to validate our design choices and hyperparameters.

Image Captioner. Table S1 presents the performance of 3D-AVS-Image using different image captioners. We begin with BLIP [26], but observe that it often generates low-quality nouns that are not semantically meaningful entities, such as *side*, *front*, or *night*. To address this limitation, we replace it with RAM [65] and xGen-MM [56], both of which produce more precise nouns and lead to improved segmentation performance. Moreover, RAM outputs both single and compound nouns (*e.g.* car and asphalt road), which inspires us to enhance BLIP3 with a compound noun extraction technique that identifies consecutive nouns within captions and treats them as an individual query. This modification yields the best overall performance.

Table S4. **IoU comparison on nuScenes [4].** For a quantitative comparison, we employ LAVE [70] to map auto-classes from an Unknown Vocabulary (**UV**) to the nuScenes categories. Overall, 3D-AVS demonstrates a significant improvement over OpenScene [38], achieving higher IoU scores on most individual labels, particularly for ambiguous classes such as *drivable surface*, *terrain*, and *man-made*.

Method	**Label Set**	**UV**	**mIoU**	*barrier*	*bicycle*	*bus*	*car*	*constr. vehicle*	*motorcycle*	*person*	*traffic cone*	*trailer*	*truck*	*drivable surface*	*other flat*	*sidewalk*	*terrain*	*man-made*	*vegetation*
OpenScene [38]	Official	✗	30.1	9.2	16.3	67.2	70.4	16.4	62.6	**47.6**	4.0	**5.3**	52.0	39.3	0.0	18.1	0.2	17.5	56.2
3D-AVS (Ours)	Image	✓	34.6	**13.1**	**20.1**	**67.6**	65.5	25.0	58.9	2.5	5.5	2.8	**61.6**	52.7	**0.3**	16.8	40.2	**55.4**	**65.0**
	LiDAR	✓	33.4	8.6	0.1	64.2	**72.3**	21.2	57.6	44.4	4.6	2.4	55.2	63.1	0.1	12.7	22.8	52.1	53.3
	I+L	✓	**36.2**	12.3	5.9	65.1	72.2	**25.5**	**64.2**	18.0	**7.1**	4.8	56.7	**68.2**	0.2	**20.0**	**41.4**	53.5	64.4

Point Captioner. Table S2 shows the ablation studies of the point captioner on the nuScenes [4] dataset. LidarCLIP generates a single caption per scene, typically covering only 2–4 common categories (*e.g*. car and road). In contrast, our optimal performance is achieved at $T = 12$ using a 3D relative positional encoding adapted from [50], which we adopt as our final configuration. Both LidarCLIP and our polar masking are tailored for rotating LiDAR scanners and sparse outdoor data, making them unsuitable for indoor datasets like ScanNet [14] and ScanNet200 [43]. We instead divide the scene using vertical pillars and caption each pillar. We find that a pillar size of 0.25 m^2 yields better performance (see Tab. S3). However, memory usage increases exponentially as pillar size decreases, so we set $0.5m \times 0.5m$ as the final resolution to avoid memory issues.

S4. Impact of Captioner Fusion.

Quantitative Analysis. We report the segmentation results of 3D-AVS and its variants—using only the image captioner (3D-AVS-Image) or the point captioner (3D-AVS-Point)—on nuScenes and ScanNet in Tables S4 and S5. Table S4 presents class-wise IoU on nuScenes. The improvements are particularly notable for ambiguous categories such as *drivable surface*, *terrain*, and *man-made*, with mIoU scores of 68.2, 41.4, and 55.4, respectively—substantially outperforming OpenScene [38]. Table S5 presents the segmentation results of 3D-AVS and its variants on ScanNet. Due to the wide variety of objects in ScanNet, 3D-AVS-LiDAR exhibits a performance drop, indicating its limited capacity for recognizing diverse object categories.

Impact of Captioner Fusion on Challenging Scenes. In Sec. 5.2 and Fig. 5, we explored the text-point similarity of generated vocabularies across various subsets of the nuScenes [4] dataset. Our analysis indicates that in challenging conditions, such as night and rainy scenes, the point captioner outperforms the image captioner. When the two captioners are combined, referred to as 3D-AVS, the resulting vocabularies show the strongest alignment with the data. To illustrate this qualitatively, we present three difficult examples from the night and rainy subsets in Fig. S1. These examples clearly show that even in these demanding scenarios, fusing the image and point captioners leads to more effective vocabulary generation, successfully identifying relevant objects in the scene. Furthermore, our method discovers additional object categories that were not originally annotated in the dataset.

Table S5. **IoU comparison on ScanNet [14] validation set.**

Method	**Unknown Vocabulary**	**Label Set**	**mIoU**
3D-AVS (Ours)	✓	Image	44.38
	✓	LiDAR	29.25
	✓	I+L	40.51

S5. Segmenation Performance in Relation to Vocabulary Mapper.

In Sec. 5.3 and Tab. S5, we evaluated segmentation performance using LAVE [70] on the nuScenes [4] and ScanNet [14] datasets. To investigate the impact of different vocabulary mappers on segmentation performance, we compare three automated mappers and a manually crafted mapper on a subset of the ScanNet dataset in this section.

Automated Mapper. We evaluate segmentation performance with three automated mappers:

- SentenceBERT [41], which maps generated categories to target categories by measuring the similarity between two text prompts.
- LAVE-Llama [70], a LLM-based Auto-Vocabulary Evaluation (LAVE) using Llama [1] as the core. This method

Image Captioner	person, building, car, road, streetlight, street light, tree , …
Point Captioner	building, car, cement wall, fire hydrant, forest, front, gate, parking meter, road, street, traffic light, van, window, …
3D-AVS	person, parking meter, gate, van, window, street light, streetlight, fire hydrant, cement wall, road, traffic light, front, tree, forest, building, street, car, …
Human-annotated	pedestrian, barrier, traffic cone, car, construction vehicle, driveable surface, other flat, sidewalk, terrain, manmade, vegetation

Image Captioner	person, bush, car, fence, mailbox, headlight, road, trash, vehicle, streetlight, street light, tree, …
Point Captioner	building, car, fence, house, man, road, sign, street, traffic light, truck, …
3D-AVS	house, person, vehicle, bush, man, street light, sign, streetlight, fence, trash, headlight, road, traffic light, mailbox, tree, building, truck, street, car, …
Human-annotated	traffic cone, car, drivable surface, sidewalk, terrain, manmade, vegetation

Image Captioner	traffic light, sculpture, building, parking lot, trash can, pedestrian, truck, car, parking sign, street sign, bicycle lane marking, lamp post, streetlight, vehicle, tree, …
Point Captioner	car, city street, lot, person, sidewalk, street scene, street sign, traffic light, …
3D-AVS	person, lot, vehicle, lamp post, sidewalk, parking sign, streetlight, pedestrian, traffic light, tree, building, street sign, truck, sculpture, trash can, car, parking lot, …
Human-annotated	pedestrian, car, trailer, truck, driveable surface, sidewalk, terrain, manmade, vegetation

Figure S1. **Examples of captioners under challenging conditions.** Even in challenging weather conditions, our method is capable of generating useful descriptions of the scene, combining the strengths of both the image (when visual information is present) and the point captioner (when geometric information is present). Green classes correspond to categories that overlap with human-annotated categories provided in the dataset. Purple classes are additionally recognized by 3D-AVS which we deem plausible and useful.

Table S6. **Comparison of automated mappers on ScanNet [14] dataset.**

Method	Label Set	LAVE [70]		Sentence
		GPT-4o [37]	Llama [1]	BERT [41]
3D-AVS (Ours)	Image	**44.38**	37.32	42.60
	LiDAR	**29.25**	25.24	23.21
	Image+LiDAR	**40.51**	34.54	39.01

queries Llama interactively to identify the most similar target category for a given generated category.

- LAVE-GPT-4o, which extends LAVE by employing the more powerful GPT-4o [37] as the core language model.

Experimental results demonstrate that GPT-4o achieves the best mapping performance. Therefore, we report results using LAVE-GPT-4o as the mapper in the main text.

Manual Mapper. Given the impracticality of manual mapping for large-scale datasets, we manually mapped the automatically generated classes-125 in total-from a subset of 10 scenes in ScanNet [14] to the 20 original categories. The recalculated mIoU scores, detailed in Tab. S8, reveal that the GPT-4o has demonstrated performance that is very close to human level on this specific task.

Table S7. **Comparison of automated mappers on nuScenes [4] dataset.**

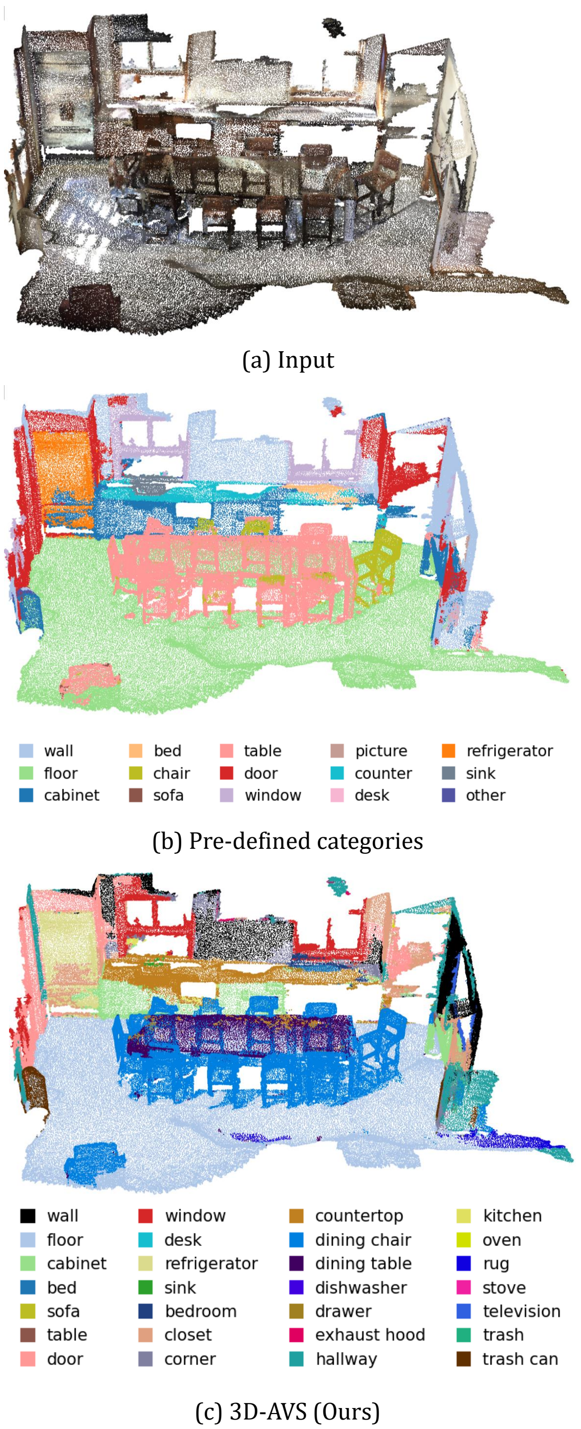
Method	Label Set	LAVE [70]		Sentence
		GPT-4o [37]	Llama [1]	BERT [41]
3D-AVS (Ours)	Image	**34.56**	33.17	26.68
	LiDAR	**33.42**	28.92	26.72
	Image+LiDAR	**36.22**	33.68	28.67

Table S8. **Mapper comparison on 10 ScanNet [14] validation samples.** The results indicated by subscripts for LAVE (L) mapper with GPT-4o demonstrate performance comparable to manual mapping (M).

Method	Modality	mIoU$_L$	mIoU$_M$
3D-AVS (Ours)	Image	30.07	29.81
	LiDAR	24.59	24.56
	Image+LiDAR	31.93	31.45

S6. Qualitative Resutls

Figure 6 shows the qualitative results on the ScanNet [14] dataset. The complex contextual input of indoor scenarios leads to a much richer vocabulary. Notably, the *chairs* around the table in Fig. 6 is misclassified as *table*, while 3D-AVS successfully segments them as *dining chair*.



(a) Input

(b) Pre-defined categories

(c) 3D-AVS (Ours)

Figure S2. **Qualitative comparison between pre-defined categories and 3D-AVS on ScanNet Dataset [14].** 3D-AVS generates much more categories than pre-defined in ScanNet. With pre-defined categories, the *chair* and *table* in the middle of the scene are messed up while 3D-AVS outputs a better result with generated *dining chair* and *dining table*.

S7. Nomenclature

Method variables

h_{tx}	CLIP text encoder
h_{im}	CLIP image encoder
$h_{\text{im}}^{\text{hr}}$	CLIP image encoder (High resolution)
h_{pt}	CLIP point encoder
g_{tx}	Text encoder (used in TPSS calculation)
g_{pt}	Point encoder (used in TPSS calculation)
$\mathbf{P}$	Point cloud, $\mathbf{P} \in \mathbb{R}^{N\times 3}$
p_n	n-th point
$\mathbb{S}$	Semantic space
$\mathbf{I}$	A group of images, $\mathbf{I} \in \mathbb{R}^{K\times H\times W\times 3}$
$\boldsymbol{d}_{\text{im}}$	Captions from images
$\boldsymbol{d}_{\text{pt}}$	Captions from points
$\mathbf{L}$	Label set, $\mathbf{L} \in \mathbb{R}^{M}$
l_m	m-th label
$\hat{l}_n$	Predicted label for n-th point
$\mathbf{E}_{\text{tx}}$	Text embeddings, $\mathbf{E} \in \mathbb{R}^{M\times C}$
e_m	m-th text embedding
$\mathbf{F}_{\text{im}}$	Image feature embeddings, $\mathbf{F}_{\text{im}} \in \mathbb{R}^{K\times H\times W\times C}$
f_k	k-th image feature embedding
$\mathbf{F}_{\text{pt}}$	Point feature embeddings, $\mathbf{F}_{\text{pt}} \in \mathbb{R}^{N\times C}$
f_n	n-th point feature embedding
f_n^{im}	n-th point feature lifted from image
x_n, y_n, z_n	Cartesian coordinates of the n-th point
ρ_n	Radius of the n-th point in a polar coordinate system
φ_n	Polar angle of the n-th point in a polar coordinate system
$\mathcal{B}$	Binary masks for a point cloud, $\mathcal{B} \in \mathbb{R}^{N\times T}$ or $\mathcal{B} \in \mathbb{R}^{N\times K}$
$\mathcal{M}$	The same as the transpose of $\mathcal{B}$. Appears in figures, $\mathcal{M} \in \mathbb{R}^{J\times N}$
b_n^t	t-th binary mask for n-th point

$\mathcal{C}$	Coordinates of a point cloud, $\mathcal{C} \in \mathbb{R}^{N \times 3}$
$\mathcal{F}$	Features of a point cloud, $\mathcal{F} = \mathbf{F}_{\text{pt}} \in \mathbb{R}^{N \times C}$
$\mathcal{Q}$	Query for MHA
$\mathcal{K}$	Key for MHA
$\mathcal{V}$	Value for MHA
$\mathcal{F}''$	Output feature of SMAP, $\mathcal{F}'' \in \mathbb{R}^{J \times C}$
SIM	Similarity metric
S_n	Similarity score for n-th point

Scalars

N	Number of points
n	Index of a point
K	Number of images
k	Index of a image
M	Number of labels
m	Index of a label
C	Number of channels
T	Number of point cloud area
t	Index of an area
J	Number of binary masks
j	Index of a mask