LOSC: LiDAR Open-voc Segmentation Consolidator

Nermin Samet[1], Gilles Puy[1], Renaud Marlet[1,2]

[1]Valeo.ai, Paris, France [2]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France

Abstract

We study the use of image-based Vision-Language Models (VLMs) for open-vocabulary segmentation of lidar scans in driving settings. Classically, image semantics can be back-projected onto 3D point clouds. Yet, resulting point labels are noisy and sparse. We consolidate these labels to enforce both spatio-temporal consistency and robustness to image-level augmentations. We then train a 3D network based on these refined labels. This simple method, called LOSC, outperforms the SOTA of zero-shot open-vocabulary semantic and panoptic segmentation on both nuScenes and SemanticKITTI, with significant margins. Code is available at https://github.com/valeoai/LOSC.

1. Introduction

Vision-Language Models (VLMs) are flourishing, offering unprecedented capabilities [17, 30, 34, 38]: employed as general Vision Foundation Models (VFMs), they are usable off-the-shelf for many open-vocabulary tasks: image classification, visual reasoning, visual question answering, image retrieval, image captioning, object detection, semantic segmentation, panoptic segmentation, etc. However, for a number of closed-set and domain-specific tasks, including object detection and semantic or instance segmentation, zero-shot performance of VLMs is not up to the level of supervised finetuning with manual labels [10]. Several reasons may explain it.

First, some VLMs are trained with the sole supervision of image-text pairs, that makes dense tasks as segmentation harder to learn [23]. Second, as training a VLM requires lots of data, some models are trained on data scrapped from the Web, which are thus subject to a distribution bias: they contain comparatively few samples from particular domains, e.g., medical images [8, 26]. It concerns both the visual contents (images) and the associated vocabulary (text). Third, "vision" in "VLM" generally only implies images, as 3D data (depth maps, point clouds, etc.) aligned with text is in much shorter supply than 2D data [18].

Therefore, most existing 3D VLMs (relating 3D data and text) actually operate via intermediate 2D data, leveraging a

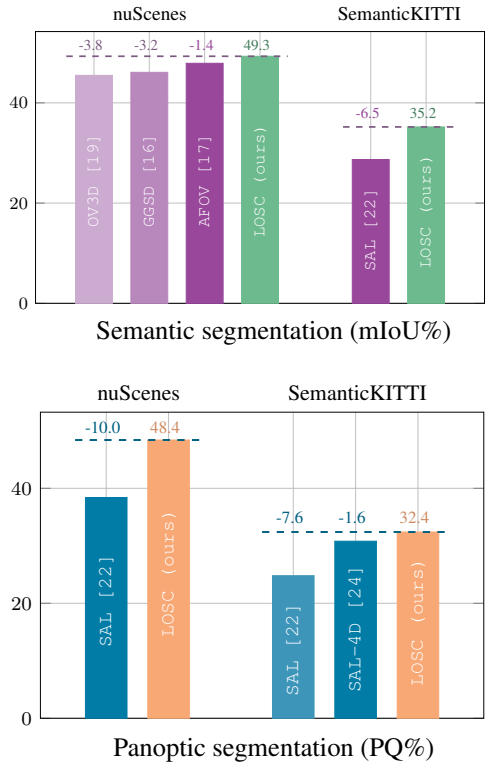

Figure 1. **Performance of LOSC compared to SOTA on zero-shot open-voc segmentation.**

2D VLM already trained on a large quantity of images: the 2D VLM is used to extract 2D information (labels or features), which is then back-projected onto 3D [18]. The 3D labels (or features) can then be directly employed as such, or used as pseudo-labels to train a 3D model, e.g., for the feature-based distillation of a 2D VFM into a 3D VFM [22].

In this paper, we study the use of 2D VLMs for the zero-shot open-vocabulary segmentation of lidar scans in driving settings, transferring 2D knowledge to 3D while specializing it for a specific purpose. Precisely, given (i) *an open-vocabulary 2D VLM*, (ii) *any set of classes*, defined via *arbitrary textual prompts*, and (iii) *an unannotated 3D dataset* of lidar scans with registered images, we create *a*

3D model that efficiently can infer the semantic segmentation of any lidar scan for these specific classes. Although this 3D model is *closed-set*, its specification, based on textual prompts, is *fully open-set*, and the process of creating the 3D model is *annotation-free*. Besides, as we can leverage a state-of-the-art (SOTA) 3D segmenter, which we train from the segments produced by the 2D VLM, we obtain an efficient 3D model.

However, driving scenes suffers from the data shortage mentioned above: for this specific domain, there is relatively little visual content available associated with text [8, 26], and even less involving 3D data [18]. As a result, current VLMs still lag behind full supervision for the semantic and panoptic segmentation of driving scene images, and even more so for lidar scans [1, 10, 31]. Concretely, in driving scenes, VLM-based image segments are often missing or wrongly labeled, while object boundaries are imprecise. This label inaccuracy transfers to 3D point after back-projection. Additionally, some 3D points may get wrong 2D information due to parallax between camera and lidar sensors, and 3D points invisible from cameras (not in their field of view) do not get any information.

We propose three label consolidation mechanisms to address these issues. First, we consider that semantics should be robust to basic image augmentations and discard labels that change according such variations. Second, we assume that 3D labeling should be consistent over time while driving, at least regarding the static part of the scene. Therefore, we ensure that all points falling into the same voxel get the same label over a whole driving sequence. Third, it is known that iterated training has a denoising effect on pseudo-labels. Therefore, starting from a (self-supervised) pretrained 3D model, we run several finetuning iterations, interleaving them with spatio-temporal consolidation.

To get the best performance, we additionally (i) benchmark existing VLMs for semantic segmentation, (ii) experiment with different prompt variants, and (iii) leverage a pretrained 3D backbone. It allows us to improve the state of the art (SOTA) of open-vocabulary 3D semantic segmentation by a significant margin on two classic datasets (see Fig. 1). Our contributions are as follows:

- We propose a way to combine several techniques to improve VML-based 3D segmentation, including augmentation- and time-based consolidations, 3D model pretraining and iterated finetuning.
- We additionally study the specific choice of a basis VLM for the driving setting, for both semantic and panoptic segmentation, using different metrics and prompts.
- Despite the simplicity of our method (no complex training or architecture, no prompt engineering or VLM tweaking, very few and little-sensitive parameters), we largely improve the SOTA of open-vocabulary semantic and panoptic segmentation of lidar data.

2. Related work

Open-vocabulary 2D segmentation. Models such as CLIP [23] and ALIGN [13], which aligns global text and image representations, demonstrate strong capabilities in zero-shot classification. These capabilities have been extended to dense prediction tasks requiring pixel-level information, such as semantic segmentation. Some methods leverage the original CLIP model and find recipes to extract dense pixel representations from it. For example, MaskCLIP [37] extract patch features in the last attention layer. CLIP-DIY [32] aggregates features from several crops. Some other methods finetune or train from scratch a CLIP-like architecture but where text representations are aligned with pixel- or patch-level representations [12, 15, 16, 33, 36, 40]. In this work, we evaluate the following models: OpenSeg [12], SEEM [40], OpenSeed [33] and Grounded-SAM [15].

Open-vocabulary 3D segmentation. The capabilities of open-vocabulary 2D semantic segmentation can be transferred to 3D network by distillation.

OpenScene [20] and CLIP2Scene [7] do so using point-pixel correspondences and training a 3D network to produce point features which align with the corresponding pixel representations extracted from a CLIP-like model. The process was then improved by exploiting, e.g., additional geometric regularization [27, 28], improving text queries [14, 29], or exploiting models such as SAM [15] for denoising [5, 39].

SAL [19] uses SAM [15] to get instance masks from images, and MaskCLIP [9] to get per-mask CLIP features. The masks and features are lifted in 3D with point-pixel correspondences, and a 3D network is trained to predict the pairs of masks and features. SAL therefore enables zero-shot lidar panoptic segmentation. SAL was recently improved by operating on sequences rather than single scans in [35]. In this work, we train our 3D network for semantic segmentation and leverage ALPINE [25] to obtain panoptic labels.

LeAP [11], developed concurrently, is closer to our method. LeAP leverages a VLM to generate 2D soft labels, which are lifted in 3D. The 3D soft labels are refined using a Bayesian update leveraging time consistency. To further improve the quality of these 3D soft labels, a 3D network is trained on the most confident soft labels and the output of this trained 3D network is used to enrich the 3D soft labels originating for images. Unlike LeAP, we exploit one-hot labels instead of soft-labels and a simple majority voting when leveraging time-consistency, instead of a Bayesian update. But more importantly, we provide new insights by (a) comparing the performance of several VLMs, (b) showing how to best combine image augmentations and time-consistency to improve pseudo-label quality, and (c) demonstrating that using a pretrained 3D network gives a significant performance boost.

3. Method

Our goal is the zero-shot open-voc labeling of lidar scans. We assume we are given (i) a closed but unconstrained set of classes $\mathcal{C}$, (ii) a VLM providing semantic segments of images from open-vocabulary textual prompts, (iii) an unlabeled training dataset $\mathcal{D}$ of sequences of synchronized and calibrated images and scans. The overall pipeline of our method, illustrated on Fig. 2, is as follows:

1. Based on prompts for the target set of classes, we use the VLM to produce a semantic segmentation of the dataset of images. Back-projecting these 2D segments onto the associated point clouds then provides initial, basic semantic labels for 3D points.
2. These initial 3D semantic labels are first consolidated using a temporal aggregation of lidar scans in a sequence, and voxel-level majority voting.
3. Considering a set of augmentations applied to images before they go through the VLM, we consolidate 3D point labels by only keeping those that remain consistent.
4. Next, we combine augmentation-based consolidation and time-based consolidation, ensuring that label classes are represented enough to be used for training.
5. Finally, the resulting labels are used to finetune a pretrained 3D network. Iterating a few times temporal voxel consolidation and finetuning further improves segmentation quality.

Each of these steps is detailed in the following subsections.

3.1. Initial Zero-shot Semantic Segmentation of LiDAR data with the VLM

Formally, given an image I and textual prompts $T_{\mathcal{C}}$ for the target classes $\mathcal{C}$, the VLM produces a 2D pixelwise semantic segmentation $S_I = \text{VLM}(I, T_{\mathcal{C}})$. It is back-projected into the 3D points of the associated lidar point cloud P to provide a 3D pointwise semantic segmentation $S_P = \Pi_{I,P}(S_I)$, where $\Pi_{I,P}$ denotes the back-projection. 3D points falling outside of the camera frustrum, or hidden behind other points due to parallax configurations between camera and lidar, get the *ignore* label. When a lidar scan P is captured together with several images $\mathcal{I}(P)$, the images are segmented and back-projected jointly, forming the initial VLM-based point cloud labeling L_{vlm}:

$$L^P_{\text{vlm}} = \cup_{I \in \mathcal{I}(P)} \Pi_{I,P}(\text{VLM}(I, T_{\mathcal{C}})) \tag{1}$$

(In the following, we drop subscripts, superscripts and parameters if they are clear from the context.) These initial labels are both noisy (incorrect VLM labeling, camera-lidar parallax inconsistencies) and sparse (missing 2D VLM labeling, 3D points invisible from cameras) in the sense that a point $p \in P$ is given a label $L_{\text{vlm}}(p) \in \mathcal{C} \cup \{\textit{ignore}\}$, where *ignore* represents the lack of semantics. For a dataset like SemanticKITTI [2], this sparsity is particularly severe (cf. Sec. 4), because the 360° lidar is associated to only one camera, with a 90° field of view. In the following, we introduce three refinement strategies to improve both label quality and quantity.

3.2. Label Refinement with Time-Based Consolidation (TBC)

Due to resolution, faraway objects are hard to identify as they are perceived with only few pixels and points. However, as the ego vehicle drives and comes closer to them, their semantics becomes more apparent. To leverage semantic consistency over time, we aggregate each sequence $\mathcal{P}$ of labeled point clouds into the same coordinate system and enforce label consistency at voxel level by majority voting. Then, for any individual scan $P \in \mathcal{P}$, we assign to each point $p \in P$ the label of the voxel p falls into. It produces scans $L^{\mathcal{P}}_{\text{tim}} = \text{TBC}(L^{\mathcal{P}}_{\text{vlm}})$ with time-consistent labels.

This process may increase label sparsity because, in our implementation, *ignore* is treated as a full-fledged class label. The idea is that a majority of points in a voxel vote for *ignore*, then all points falling into that voxel should be considered as unreliable and labeled as *ignore* too. Besides, even when such a configuration does not arise, the label distribution may also change, with some classes loosing point representatives in favor of other classes.

A variant consists in weighting the label votes with the softmax values of the label prediction. However, empirically, the gain is only marginal and it is simpler just to count class labels in voxels.

3.3. Label Refinement with Augmentation-Based Consolidation (ABC)

A robust network should be little sensitive to small input variations. To filter out "brittle" labels, we probe VLM labeling confidence using image augmentations: a label is considered as untrustful if it changes across a set of image augmentations. This Augmentation-Based Consolidation (ABC) can be seen as a form of Test-Time Augmentation (TTA) applied to VLM inference.

Formally, we consider a set of image augmentations $\mathcal{A}$. For any augmentation $A \in \mathcal{A}$, we denote by L^A_{vlm} the initial point cloud labels produced by the VLM from augmented images $A(\mathcal{I}(P))$. If the label of a point $p \in P$ is the same class c whatever the augmentation $A \in \mathcal{A}$, then the augmentation-based consolidated label $L_{\text{aug}}(p)$ of p is this class c; otherwise, it is the *ignore* label:

$$L^A_{\text{vlm}} = \bigcup_{I \in \mathcal{I}} \Pi_I(\text{VLM}(A(I), T_{\mathcal{C}})) \tag{2}$$

$$L_{\text{abc}}(p) = \text{ABC}(L^{\mathcal{A}}_{\text{vlm}})(p) = \begin{cases} c, & \text{if } \forall A \in \mathcal{A},\ L^A_{\text{vlm}}(p) = c \\ \textit{ignore}, & \text{otherwise} \end{cases} \tag{3}$$

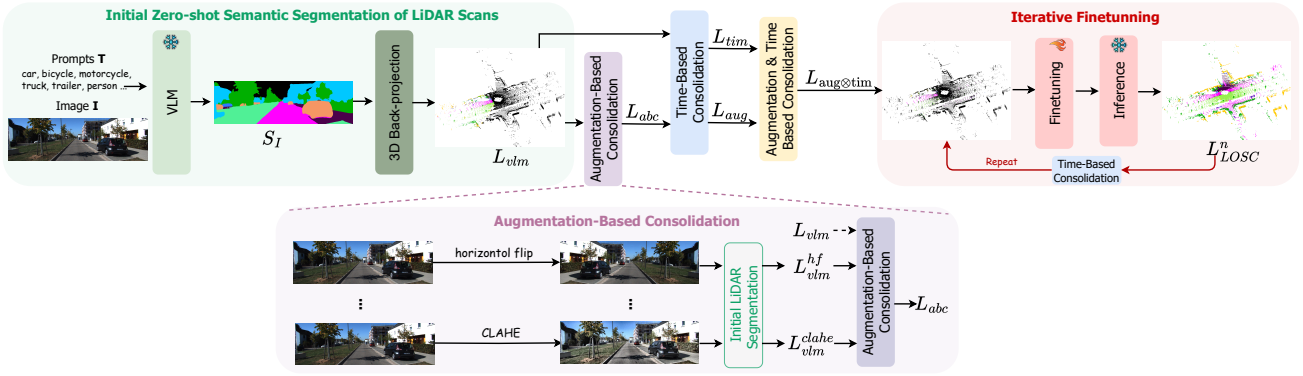

Figure 2. **Overall pipeline of LOSC.** The images are first segmented using an open-vocabulary segmentation model. The semantic 2D labels are then backprojected on the lidar points. These 3D labels are refined using three consolidation steps: the first uses label consistency across several image augmentation, the second uses time consistency, the last combines the best labels from the previous steps. These labels are used to finetune a 3D network. Finally, few steps of self-training are used to improve the results.

We then enforce temporal voxel consistency and build a consolidated labeling $L_{\text{aug}} = \text{TBC}(L_{\text{abc}})$.

3.4. Label Refinement Combining Augmentation- and Time-based Consolidation (ATC)

Even more than time-based consolidation, augmentation-based consolidation tends to increase label quality but also to reduce label quantity (cf. Sec. 4). To make sure points of each class remain both numerous enough and balanced enough, we introduce a way to combine L_{aug} and L_{tim}.

Let $N^c = |\{L(p) = c\}_{p\in P}|$ be the number of points in a labeling L of point cloud P that have been given label c. The set of classes in $\mathcal{C}$ that are robust to augmentation-based consolidation is

$$\mathcal{C}_{\text{robust-aug}} = \{c \in \mathcal{C} \mid N^c_{\text{aug}} \geq N \text{ and } N^c_{\text{aug}}/N^c_{\text{tim}} \geq \tau\} \quad (4)$$

for some minimum number of points N and minimum proportion of preserved points τ. For these classes, it is safe to use the augmentation-based labeling L_{aug}. Otherwise, it is safer to stick to time-based labeling L_{tim}. Therefore, we construct the following combined labeling $L_{\text{aug}\otimes\text{tim}}$:

$$\begin{aligned} L_{\text{aug}\otimes\text{tim}}(p) &= \text{comb}(L_{\text{aug}}, L_{\text{tim}})(p) \\ &= \begin{cases} L_{\text{aug}}(p), & L_{\text{aug}}(p) \in \mathcal{C}_{\text{robust-aug}} \\ L_{\text{tim}}(p), & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

This hybrid strategy helps preserve class balance and label quality, especially for rare classes and for categories that are hard to predict.

3.5. Label Refinement with Iterative FineTuning (FT)

Finally, we use the combined consolidated labels of $L_{\text{aug}\otimes\text{tim}}$, obtained on the whole training set $\mathcal{D}$, as pseudo-labels to train a 3D network. While all previous labelings (L_{vlm}, L_{tim}, L_{aug}, $L_{\text{aug}\otimes\text{tim}}$) are partial, in the sense that some points may be labeled as *ignore*, applying the trained 3D network on any scan yields a fully-labeled point cloud (no *ignore*).

To improve the performance of this 3D segmenter, we do two things. First, we actually start from a pre-trained model M_0 and finetune it (FT) on the pseudo-labels of $L_{\text{aug}\otimes\text{tim}}(\mathcal{D})$, resulting in a model $M_1 = \text{FT}_{L_{\text{aug}\otimes\text{tim}}(\mathcal{D})}(M_0)$. Second, we iterate training, as often done in semi-supervised settings. More precisely, given a trained model M_n, we create new pseudo-labels $M_n(\mathcal{D})$ for the training set, apply time-based consolidation, and use these refined pseudo-labels to finetune M_n, resulting a new model $M_{n+1} = \text{FT}_{\text{TBC}(M_n(\mathcal{D}))}(M_n)$. Please note that augmentation-based consolidation (ABC) is performed only once and used via $L_{\text{aug}\otimes\text{tim}}$ to finetune M_1. For all subsequent iterations, only time-based consolidation is used, to refine pseudo-labels of the training set before the next finetuning. The resulting model, after a few iterations, is our final model, which we refer to as LOSC.

4. Experiments

In this section, we analyze our proposed method and compare it against the current SOTA.

Implementation details. On the image side, for augmentation-based consolidation, we use the Albumentations library [3], applying 10 distinct image transformations: horizontal-flip, hue-saturation, blur, color-jitter, auto-contrast, sharpen, chromatic-aberration, emboss, fancy-pca, clahe. The appendix details the rationale of this choice. This set of augmentations is consistent in the sense that although a single disagreement is enough to discard the label of a point (cf. Eq. (3)), empirical results show the robustness of pseudo-labeling (cf. appendix), leading to SOTA results (cf. Tab. 6).

Method	Segmentation		Prompt		**Label cov.%**	**mAcc %**	**mIoU %**
	Sem.	Pan.	Min.	Rich			
GSAM [24]	✓	✗	✓	✗	68.0	32.8	21.1
GSAM 2 [24]	✓	✗	✓	✗	75.3	46.6	24.3
OpenSeeD [33]	✓	✗	✓	✗	74.9	43.4	33.4
	✓	✗	✗	✓	74.8	42.8	33.8
	✗	✓	✓	✗	63.9	39.5	31.9
	✗	✓	✗	✓	66.9	39.9	32.4
SEEM [40]	✓	✗	✓	✗	76.2	47.7	29.6
	✓	✗	✗	✓	76.3	44.7	25.7
	✗	✓	✓	✗	39.9	25.2	21.1
	✗	✓	✗	✓	23.3	13.1	8.3
OpenSeg [12]	✓	✗	✓	✗	72.8	46.5	26.7
	✓	✗	✗	✓	76.3	43.7	24.0

Table 1. **Semantic segmentation benchmark of 2D open-vocabulary VLMs** on the nuScenes validation set. Performance is evaluated using minimal or rich prompts, with semantic or (merged) panoptic segmentation results where applicable. *Label cov.* is label coverage.

Regarding 3D model pretraining, we leverage a *self-supervised* backbone to benefit from class-agnostic representations for our open-set segmentation approach. As 3D network, we thus chose WaffleIron [21] (as did LeAP [11]), more precisely WI-48-768, for which self-supervised pretrained weights with ScaLR [22] are publicly available. This pretrained model is currently among SOTA self-supervised 3D backbones for lidars. When finetuning this network, we use the protocol of [22] but reduce the number of epochs to 10. For trainings without ScaLR initialization, we train the model from scratch as described in [21].

For spatio-temporal consolidation, we set the voxel size to 10 cm. For the combination of consolidations, we set $N = 200k$ and $\tau = 1/3$. Last, we limit ourselves to 3 finetuning iterations, which defines our method, LOSC. We experiment on both nuScenes [4] and SemanticKITTI [2] with identical parameters.

Benchmarking 2D VLMs. We evaluate the quality of 3D semantic labels L_{vlm} obtained from several established 2D open-vocabulary VLMs after back-projection: Grounded SAM [24], SEEM [40], OpenSeeD [33] and OpenSeg [12]. We use two types of prompts in our experiments: minimal and rich prompts (which are detailed in the Appendix). Some VLMs can output either semantic or panoptic masks. We tested both alternatives: panoptic masks were turned into semantic masks by simply merging all instance-masks from each class. The quality of L_{vlm} labels is evaluated on the validation set on nuScenes. We compute the label coverage (percentage of annotated points after back-projection of annotated pixels) as well as the mAcc and mIoU, where non-annotated points after back-projection are counted as errors. The results are presented in Table 1.

First, we notice that about 65 to 75% of the points receive a label in most configurations, with the exception of the panoptic variant of SEEM where less than 40% of the points get a label. The differences in label coverage is explained by the fact that the VLMs do not provide labels for some pixels. Second, our results show moderate performance differences between the minimal and rich prompts, especially for OpenSeeD, but except for SEEM. For the Grounded SAM family of models, we report results only with minimal prompts as the runtime increases substantially with more elaborate prompts. Third, we notice that the semantic variant of VLMs tends to provide labels of better quality than the (merged) panoptic variant. Finally, we notice that OpenSeeD in the best performing model regarding mIoU. Based on these results, we generate L_{vlm} for the *training sets* of SemanticKITTI and nuScenes by back-projecting 2D labels obtained using the semantic variant of OpenSeeD with minimal prompts, i.e., the fastest, simplest and nearly best discovered configuration.

3D model pretraining and label refinement. Table 2 shows the benefits of leveraging a pretrained 3D network and of using our label refinement strategies.

First, we observe that training from ScaLR pretrained weights, compared to random weights, significantly boosts performance by 3.1 and 16.1 mIoU points on nuScenes and SemanticKITTI, respectively, when starting from L_{vlm}. Starting from pretrained weights continues to improve performance even when starting from our most refined pseudo-labels $L_{\text{aug}\otimes\text{tim}}$.

Second, applying our time-based consolidation strategy (TBC) directly on L_{vlm}, producing L_{tim}, further improves performance by about 6 and 10 points on nuScenes and SemanticKITTI, respectively.

Third, our augmentation-based consolidation (ABC), directly applied on L_{vlm} to produce L_{abc}, brings little gain on SemanticKITTI over just training from L_{vlm}, and even slightly degrades performance on nuScenes. Besides, refining L_{abc} with TBC to produce L_{aug} leads to similar results than with just L_{tim}. But ABC is actually not meant to operate on it own. The benefit of augmentation-based consolidation appears in fact when combining L_{aug} and L_{tim}, where we obtain an extra boost of performance of 1.6 and 2.1 mIoU points on nuScenes and SemanticKITTI, respectively, compared to using L_{tim} alone.

Iterative finetuning. The results in Table 3 show that the quality of labels keeps improving after each round of finetuning, whether starting from L_{tim} labels or $L_{\text{aug}\otimes\text{tim}}$. The best results are obtained when starting from $L_{\text{aug}\otimes\text{tim}}$, which shows that our iterative finetuning strategy is able to maintain the initial advantage provided by our most-refined labels. While a single iteration may provide a gain up to +2.4 mIoU pts, further iterating provides diminishing returns. What thus makes sense practically is to only iterate a few times. In LOSC, we perform 3 finetuning iterations.

Labeling	Time consistency	Augmentation consistency	ScaLR pretraining	nuScenes mIoU%	SemanticKITTI mIoU%
$L_{\text{vlm}} = \Pi(\text{VLM}(\mathcal{I}, T_{\mathcal{C}}))$	✗	✗	✗	37.4	6.5
	✗	✗	✓	40.5	22.6
$L_{\text{tim}} = \text{TBC}(L_{\text{vlm}})$	✓	✗	✓	46.4	32.1
$L_{\text{abc}} = \text{ABC}(L_{\text{vlm}})$	✗	✓	✓	39.2	23.0
$L_{\text{aug}} = \text{TBC}(\text{ABC}(L_{\text{vlm}}))$	✓	✓	✓	46.5	31.1
$L_{\text{aug}\otimes\text{tim}} = \text{comb}(L_{\text{aug}}, L_{\text{tim}})$	✓	✓	✗	45.7	31.0
	✓	✓	✓	48.0	34.2

Table 2. **Effect of our label consolidations** on nuScenes and SemanticKITTI validation sets.

	nuScenes (mIoU%)			SemanticKITTI (mIoU%)		
Labeling	1st iter.	2nd iter.	3rd iter.	1st iter.	2nd iter.	3rd iter.
$L_{\text{tim}} = \text{TBC}(L_{\text{vlm}})$	46.4	47.3	47.7	32.1	33.0	33.2
$L_{\text{aug}\otimes\text{tim}} = \text{comb}(L_{\text{aug}}, L_{\text{tim}})$	48.0	48.9	49.3	34.2	35.0	35.2

Table 3. **Effect of iterative finetuning** on nuScenes and SemanticKITTI validation sets.

		nuScenes		SemanticKITTI	
Labeling	Iter.	Label coverage %	mIoU %	Label coverage %	mIoU %
$L_{\text{vlm}} = \Pi(\text{VLM}(\mathcal{I}, T_{\mathcal{C}}))$	0	75.8	39.3	15.4	25.5
$L_{\text{tim}} = \text{TBC}(L_{\text{vlm}})$	0	74.1	39.9	6.7	26.3
$L_{\text{abc}} = \text{ABC}(L_{\text{vlm}})$	0	63.7	45.1	11.9	28.1
$L_{\text{aug}} = \text{TBC}(\text{ABC}(L_{\text{vlm}}))$	0	61.6	46.1	4.9	29.4
$L_{\text{aug}\otimes\text{tim}} = \text{comb}(L_{\text{aug}}, L_{\text{tim}})$	0	61.9	46.6	5.0	31.4
$L_{\text{aug}\otimes\text{tim}} = \text{comb}(L_{\text{aug}}, L_{\text{tim}})$	1	100.0	46.8	100.0	33.6
$L_{\text{aug}\otimes\text{tim}} = \text{comb}(L_{\text{aug}}, L_{\text{tim}})$	2	100.0	47.5	100.0	34.4

Table 4. **Quantity and quality of training labels** computed on nuScenes and SemanticKITTI training sets. Note that some lidar points are not visible from the cameras and that the VLM does not assign labels to all pixels. Moreover, while the 6 nuScenes cameras observe the scene at 360°, the single camera of SemanticKITTI only covers about 1/4 of the lidar field of view. This explains the label coverage.

Pseudo-label analysis. In this section, we analyze both the quantity and quality the data annotations used *for training*. The results are presented in Table 4 for nuScenes and SemanticKITTI. We observe that the VLM struggles more (lower mIoU) on SemanticKITTI than on nuScenes. Additionally, with each level of label refinement, the number of labeled points decreases, but the quality of the labels improves. Thanks to our iterative finetuning framework, we are able to annotate the entire datasets after the first iteration while maintaining a high annotation quality.

Comparison to SOTA zero-shot semantic segmentation. In Table 6, we compare LOSC against SOTA annotation-free methods. For SAL [19], no code is available and the only reported metric is the mIoU on the panoptic segmentation benchmarks, which happens to slightly differ from the mIoU on the semantic segmentation benchmark in the case of nuScenes (because of slightly different ground truths). We thus report both mIoU_{sem} and mIoU_{pan}, for the nuScenes semantic and panoptic benchmarks, respectively.

LOSC outperforms the SOTA by +3.2 mIoU pts on nuScenes. It even outperforms methods that additionally use images at inference time by +1.4 mIoU point. The gain over the SOTA on SemanticKITTI is +6.5 mIoU points (but only a couple of other methods evaluate on this dataset).

Note that we could not compare to LeAP [11], that evaluates with a specific protocol that is not fully described. In particular, the class mapping from the original and classical 19 classes of SemanticKITTI to their 11 classes is ambiguous and not provided in [11], while the code is not available.

Compared to some other approaches, we obtain those SOTA results without resorting to any prompt engineering, VLM tweaking, intricate architecture, complex training, or use of images for inference. Besides, we only have very few and robust parameters, which are the same for all datasets.

Method		nuScenes PQ	RQ	SQ	PQ^{Th}	PQ^{St}	$mIoU_{pan}$	SemanticKITTI PQ	RQ	SQ	PQ^{Th}	PQ^{St}	mIoU
SAL	[19]	38.4	47.8	**77.2**	**47.5**	29.2	33.9	24.8	32.3	66.8	17.4	30.2	28.7
SAL-4D	[35]	-	-	-	-	-	-	30.8	-	**76.9**	25.5	**34.6**	-
LOSC	(ours)	**48.4**	**58.1**	71.1	46.7	**51.3**	**49.8**	**32.4**	**41.4**	51.8	**36.1**	29.7	**35.2**

Table 5. **Panoptic segmentation results on nuScenes and SemanticKITTI validation sets.**

Method		nuScenes	SemanticKITTI
WaffleIron (full sup.)	[21]	78.7	63.4
CLIP2Scene	[7]	20.8	-
Towards VFMs	[6]	26.8	-
AutoVoc3D w. LAVE	[29]	30.6	-
AdaCo	[39]	31.2	25.7
SAL	[19]	33.9 †	28.7
OV3D w/ OpenScene-3D	[14]	45.5	-
GGSD	[28]	46.1	-
LOSC	(ours)	**49.3**	**35.2**
Method additionally using images at inference time			
OpenScene - OpenSeg	[20]	42.1	-
AFOV	[27]	47.9	-

Table 6. **Semantic segmentation results on nuScenes and SemanticKITTI validation sets.** † indicates the panoptic mIoU, which, in the case of nuScenes, slightly differs (< 1 pt in general) from the classical semantic segmentation mIoU otherwise reported here (see appendix).

For instance, we set τ to 1/3 but any value between 1/4 and 1/2 would have given equal or similar results for both nuScenes and SemanticKITTI. Beyond these bounds, performance would start dropping a bit.

We present in Figure 3 qualitative results on the nuScenes and SemanticKITTI validation sets.

Comparison to SOTA zero-shot panoptic segmentation. We further evaluate our method on the panoptic segmentation task, comparing it against previous annotation-free approaches in Table 5 for both nuScenes and SemanticKITTI.

To obtain our panoptic segments, we apply ALPINE [25] on top of our semantic segmentation predictions. Given the semantic segmentation maps, ALPINE projects points into the BEV space, performs clustering by building a kNN graph and extracts connected components. The method is learning-free. Our method achieves a significant improvement on the main panoptic metric, outperforming SAL by +10 PQ points or more on both datasets. LOSC also outperforms SAL-4D on SemanticKITTI by nearly 2 PQ points, with particularly significant improvement observed in instance segmentation (PQ^{Th}).

Moving objects. The spatio-temporal consolidation (TBC) assumes that voxel semantics remains constant over time. This is true for the static part of the scene, which represents most of the points. However, it may not be true for moving objects. Yet, there is no issue when an empty area is traversed by a moving object: in corresponding voxels, there are no points before the traversal, some points during the traversal, and again no points afterwards. Therefore, voxel-based voting still makes sense.

An actual issue occurs only when two different objects traverse the same empty space, and then only in one particular case: if the objects belong to different classes (e.g., car and bicycle) and if the object speed and frame rate are such that points from different objects are actually scanned within the same voxel. Given the distribution of moving object classes (a vast majority of cars), the low frame rate (2 Hz for nuScenes, 10 Hz for SemanticKITTI), and the small voxel size (10 cm), this happens rarely.

In other words, while dynamic environments can *in theory* break the voxel semantic consistency, they rarely do so *in practice*. The fact is, LOSC outperforms all other methods while the evaluation datasets do include moving objects. Moreover, if needed, time consistency for moving objects can be improved by cutting sequences into smaller sections.

5. Conclusion

Our work shows it is possible to transfer the image segmentation capability of an off-the-shelf open-vocabulary 2D-based VLM, used in a black box manner, in order to segment 3D lidar scans and largely outperform the SOTA. Besides, compared to the bells and whistles of existing approaches, our method is relatively simple and relies on very few and little-sensitive parameters.

Our approach is not bound to 2D VLMs. It could be also applied to any strong 2D semantic segmenter trained with full-supervision, alleviating the need for 3D annotation.

Acknowledgments. We acknowledge EuroHPC Joint Undertaking for awarding the project ID EHPC-REG-2024R02-234 access to Karolina, Czech Republic. This work was granted access to the HPC resources of IDRIS under the allocations AD011014946R1 made by GENCI. This work was also supported by the European Union's Horizon Europe research and innovation programme under grant agreement No 101214398 (ELLIOT).

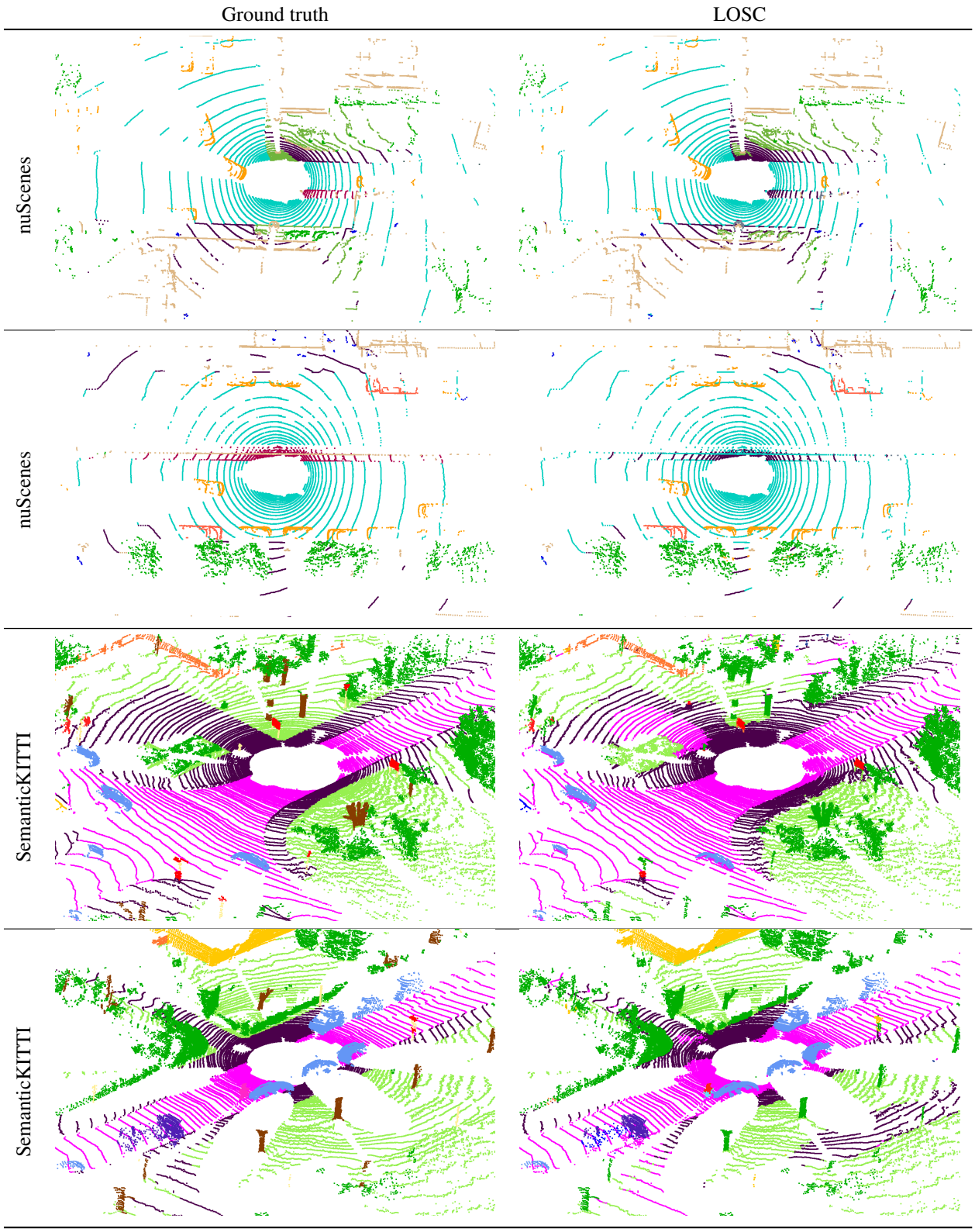

Figure 3. Qualitative results of semantic segmentation from the validation sets of nuScenes and SemanticKITTI. The color code used to represent each class is provided in Supplementary Material. A typical error with LOSC on both datasets is a confusion between different types of flat surfaces such as *road/driveable surface*, *sidewalk* and *terrain*. In SemanticKITTI, *trunks* are also systematically included in *vegetation* rather than considered as a separate class. These observations are consistent with the quantitative class-wise results provided in Supplementary Material.

References

[1] Sanjeda Akter, Ibne Farabi Shihab, and Anuj Sharma. Image segmentation with large language models: A survey with perspectives for intelligent transportation systems, 2025. arXiv preprint 2506.14096. 2

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, pages 9297–9307, 2019. 3, 5

[3] A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin. Albumentations: fast and flexible image augmentations. *ArXiv e-prints*, 2018. 4

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 5

[5] Runnan Chen, Youquan Liu, Lingdong Kong, Nenglun Chen, ZHU Xinge, Yuexin Ma, Tongliang Liu, and Wenping Wang. Towards label-free scene understanding by vision foundation models. In *NeurIPS*, 2023. 2

[6] Runnan Chen, Youquan Liu, Lingdong Kong, Nenglun Chen, Xinge ZHU, Yuexin Ma, Tongliang Liu, and Wenping Wang. Towards label-free scene understanding by vision foundation models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 7

[7] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *CVPR*, 2023. 2, 7

[8] Ziliang Chen, Xin Huang, Xiaoxuan Fan, Keze Wang, Yuyu Zhou, Quanlong Guan, and Liang Lin. Reproducible vision-language models meet concepts out of pre-training. In *CVPR*, 2025. 1, 2

[9] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary universal image segmentation with MaskCLIP. In *ICML*, 2023. 2

[10] Yongchao Feng, Yajie Liu, Shuai Yang, Wenrui Cai, Jinqing Zhang, Qiqi Zhan, Ziyue Huang, Hongxi Yan, Qiao Wan, Chenguang Liu, Junzhe Wang, Jiahui Lv, Ziqi Liu, Tengyuan Shi, Qingjie Liu, and Yunhong Wang. Vision-language model for object detection and segmentation: A review and evaluation, 2025. arXiv preprint 2504.09480. 1, 2

[11] Simon Gebraad, Andras Palffy, and Holger Caesar. Leap: Consistent multi-domain 3d labeling using foundation models. *arXiv preprint arXiv:2502.03901*, 2025. 2, 5, 6

[12] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 2, 5

[13] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 2

[14] Li Jiang, Shaoshuai Shi, and Bernt Schiele. Open-vocabulary 3D semantic segmentation with foundation models. In *CVPR*, 2024. 2, 7

[15] 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. In *ICCV*, 2023. 2

[16] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2

[17] Zongxia Li, Xiyang Wu, Hongyang Du, Huy Nghiem, and Guangyao Shi. A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges. *arXiv:2501.02189*, 2025. 1

[18] Xianzheng Ma, Yash Bhalgat, Brandon Smart, Shuai Chen, Xinghui Li, Jian Ding, Jindong Gu, Dave Zhenyu Chen, Songyou Peng, Jia-Wang Bian, Philip H Torr, Marc Pollefeys, Matthias Nießner, Ian D Reid, Angel X. Chang, Iro Laina, and Victor Adrian Prisacariu. When LLMs step into the 3d world: A survey and meta-analysis of 3D tasks via multi-modal large language models, 2024. arXiv prepreint 2405.10255. 1, 2

[19] Aljoša Ošep, Tim Meinhardt, Francesco Ferroni, Neehar Peri, Deva Ramanan, and Laura Leal-Taixé. Better call sal: Towards learning to segment anything in lidar. In *European Conference on Computer Vision*, pages 71–90. Springer, 2024. 2, 6, 7

[20] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023. 2, 7

[21] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Using a waffle iron for automotive point cloud semantic segmentation. In *ICCV*, 2023. 5, 7

[22] Gilles Puy, Spyros Gidaris, Alexandre Boulch, Oriane Siméoni, Corentin Sautier, Patrick Pérez, Andrei Bursuc, and Renaud Marlet. Three pillars improving vision foundation model distillation for lidar. In *CVPR*, 2024. 1, 5

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sas-

try, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2

[24] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 5

[25] Corentin Sautier, Gilles Puy, Alexandre Boulch, Renaud Marlet, and Vincent Lepetit. Clustering is back: Reaching state-of-the-art LiDAR instance segmentation without training. *arxiv*, 2025. 2, 7

[26] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 1, 2

[27] Boyi Sun, Yuhang Liu, Xingxia Wang, Bin Tian, Long Chen, and Fei-Yue Wang. 3d annotation-free learning by distilling 2d open-vocabulary segmentation models for autonomous driving. In *AAAI*, 2025. 2, 7

[28] Pengfei Wang, Yuxi Wang, Shuai Li, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Open vocabulary 3d scene understanding via geometry guided self-distillation. In *ECCV*, 2024. 2, 7

[29] Weijie Wei, Osman Ülger, Fatemeh Karimi Nejadasl, Theo Gevers, and Martin R Oswald. Auto-vocabulary segmentation for lidar points. *arXiv preprint arXiv:2406.09126*, 2024. 2, 7

[30] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, and Dacheng Tao. Towards open vocabulary learning: A survey. *TPAMI*, 46(7):5092–5113, 2024. 1

[31] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024. 2

[32] Monika Wysoczanska, Michael Ramamonjisoa, Tomasz Trzcinski, and Oriane Siméoni. Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free. In *WACV*, 2024. 2

[33] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *ICCV*, 2023. 2, 5

[34] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *TPAMI*, 46(8):5625–5644, 2024. 1

[35] Yushan Zhang, Aljoša Ošep, Laura Leal-Taixé, and Tim Meinhardt. Zero-shot 4d lidar panoptic segmentation. *arXiv preprint arXiv:2504.00848*, 2025. 2, 7

[36] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. RegionCLIP: Region-based language-image pretraining. In *CVPR*, 2022. 2

[37] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 2

[38] Tianfei Zhou, Wang Xia, Fei Zhang, Boyu Chang, Wenguan Wang, Ye Yuan, Ender Konukoglu, and Daniel Cremers. Image segmentation in foundation model era: A survey. *arXiv:2408.12957*, 2024. 1

[39] Pufan Zou, Shijia Zhao, Weijie Huang, Qiming Xia, Chenglu Wen, Wei Li, and Cheng Wang. AdaCo: Overcoming visual foundation model noise in 3D semantic segmentation via adaptive label correction. In *AAAI*, 2025. 2, 7

[40] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in neural information processing systems*, 36:19769–19782, 2023. 2, 5