

# Supplementary Material: Vision-Language Guidance for LiDAR-based Unsupervised 3D Object Detection

Christian Fruhwirth-Reisinger<sup>1,2</sup>  
reisinger@tugraz.at

Wei Lin<sup>3</sup>  
lin@ml.jku.at

Dušan Malić<sup>1,2</sup>  
dusan.malic@tugraz.at

Horst Bischof<sup>1</sup>  
bischof@tugraz.at

Horst Possegger<sup>1,2</sup>  
possegger@tugraz.at

<sup>1</sup> Christian Doppler Laboratory for  
Embedded Machine Learning

<sup>2</sup> Institute of Computer Graphics  
and Vision  
Graz University of Technology

<sup>3</sup> Institute for Machine Learning  
Johannes Kepler University Linz

## 1 Implementation Details

**Object discovery.** For object segmentation with HDBSCAN [1] we use  $min\_cluster\_size = 15$ ,  $min\_samples = 15$  and  $cluster\_selection\_epsilon = 0.15$ . To avoid uncertain objects, *i.e.* very small objects that lead to ambiguous 2D depth map projections and flying objects, which are mostly likely no objects of interest caused by occlusion, we apply filters after clustering: We remove segments with less than 10 points, exceeding the distance of 1m to the ground plane and segments with a height below 0.5m resulting in segments  $S_i^t$ .

A track is considered *static* when a percentile  $\alpha = 20\%$  of PP-Scores per segment  $S_i^t$  is above the threshold  $\delta = 0.7$  for all its segments, all the boxes of a track overlap with the largest box of the track, or the track has no consistent motion behavior (no smooth linear motion). We limit the greedy assignment between track predictions and detections by a 1m radius w.r.t. Euclidean distance. If no assignment can be found, we relax the radius to 5m and assign a matched detection if the number of points between the track prediction and detection cluster segment differs less than 30%. This relaxed assignment recovers very fast-moving objects, such as vehicles on the highway, and mitigates false assignments between temporally occluded or over-segmented objects.

**Object classification.** We use CLIP<sup>1</sup> [2] with the ViT-B/16 visual encoder [3] for the classification of the projected depth maps. Therefore, we generate  $K = 4$  different views, containing the basic view without rotation, rotations about the z-axis (yaw) of  $\pm 18^\circ$  and about

the y-axis (pitch) of  $6^\circ$ . Compared to synthetic CAD data [9], objects in outdoor LiDAR scans suffer from self-occlusion (recall Fig. 1 in the main manuscript) and thus are only sufficiently visible from small variations of the original viewpoint. Therefore, we only apply small rotations. A tilt in the negative y-direction is also not useful because the ground plane prevents a valid shape at the bottom (in contrast to, for example, viewing the roof of a car from a slightly elevated viewpoint). In Table 1, we provide the refined object categories we use for CLIP classification. The class predictions of the refined categories are later mapped back to the original object classes.

Class	Refined categories
Vehicle	car, truck, bus, van, minivan, pickup truck, school bus, fire truck, ambulance
Pedestrian	pedestrian, human body, human
Cyclist	cyclist, rider, bicycle, bike
Background	traffic light, traffic sign, fence, pole, clutter, tree, house, wall

Table 1: **Category text refinement.** We use the listed refined categories for predicting class labels with CLIP and map the result back to the original class space of the dataset.

**Temporally-coherent class label refinement.** After classification, we assign the class label with the highest score to all objects in the track as long as the maximum class label score exceeds the threshold of 0.5 for *vehicles* and 0.3 for *pedestrians*, *cyclists* and *background*. Additionally, this class label must match at least 60% of the tracks’ predicted classes. This propagation of labels throughout track is done for *static* and *moving* objects. We keep the CLIP label prediction for *static* objects not fulfilling the proposed conditions.

However, assuming that all objects in motion are of interest and the class label space in the automotive domain contains *vehicles*, *pedestrians* and *cyclists*, we added a default classification scheme based on object size priors for all remaining objects. Therefore, we define for *moving* objects:

$$y_i = \begin{cases} \textit{pedestrian}, & \text{if } 0.2 < b_w < 1.0 \text{ and } 0.2 < b_l < 1.0 \text{ and } 0.8 < b_h < 2.2, \\ \textit{cyclist}, & \text{if } 0.2 < b_w < 1.0 \text{ and } 1.0 < b_l < 2.5 \text{ and } 1.4 < b_h < 2.0, \\ \textit{vehicle}, & \text{if } 0.5 < b_w < 3.0 \text{ and } 0.5 < b_l < 8.0 \text{ and } 1.0 < b_h < 3.0, \\ \textit{background}, & \text{otherwise.} \end{cases}$$

The class label for object  $i$  is denoted  $y_i$ , and the bounding box dimensions *width*, *length*, and *height* are denoted  $b_w$ ,  $b_l$ , and  $b_h$ , respectively.

**Temporally-coherent bounding box refinement.** After propagating median box sizes, we filter those static tracks whose corrected box dimensions deviate significantly from the dimensions of the object categories involved. Therefore, we define the bounding box size thresholds for *width*, *length* and *height* as  $0.2 < b_w < 3.5$ ,  $0.2 < b_l < 20.0$  and  $0.5 < b_h < 4.0$  respectively. Finally, to reduce annotation bias, we inflate bounding boxes similar to [9] for each dimension by 0.3m.

## 2 Additional Results

**Spatio-temporal clustering.** In order to fully exploit the inherent temporal information contained in sequential LiDAR scans, we perform spatio-temporal clustering on multiple LiDAR scans, transformed into the same reference coordinate system. In Table 2, we show the advantage of the proposed spatio-temporal clustering compared to simple frame-by-frame spatial clustering with only spatial input features ( $x, y, z$ ).

Clustering	AP (L2)	AP (L2)	APH (L2)	APH (L2)
	BEV	3D	BEV	3D
Spatial	35.1	30.6	25.0	21.2
Spatio-temporal	36.3	32.3	26.0	22.5

Table 2: **Comparison of spatial and spatio-temporal clustering** following the protocols of [10, 9] on the WOD [10] (*i.e.* AP for BEV and 3D, difficulty level L2, IoU 0.4). We additionally report APH which includes the heading angle precision.

**Size prior baseline comparison.** Since no comparable approach performs unsupervised class-aware object detection, we implement a baseline classifying objects based on simple size priors. Therefore, we adopt the default classification scheme for moving objects of our temporally-coherent class label refinement (recall Section 1) for all objects independent of their motion state. Hence, we replace CLIP classification within our approach with simple object size prior thresholds and keep all other parts as is. In Table 3, we show the baseline results compared to our vision-language-guided approach. We can show that the CLIP model’s rich knowledge adds significant value to classifying objects in 3D LiDAR point clouds.

Classification method	Movable		Vehicle	Pedestrian	Cyclist
	BEV	3D	BEV	BEV	BEV
Baseline (size prior)	12.7	10.9	14.1	6.4	2.0
ViLGOD	36.3	32.3	49.0	16.8	7.6

Table 3: **Size prior baseline comparison** following the protocols of [10, 9] on the WOD [10] (*i.e.* AP for BEV and 3D, difficulty level L2, IoU 0.4). We report class-aware detection results for BEV. ViLGOD significantly outperforms the baseline which classifies objects solely on pre-defined object size thresholds.

**Range evaluation.** For the sake of completeness and to gain even more insight, we show the full range evaluation for the Waymo Open Dataset (WOD) [10]. We provide a detailed range analysis for the zero-shot detection of ViLGOD and the pseudo-label trained Centerpoint [9] (ViLGOD-CP) in Table 4. Following [10, 9], we report AP at difficulty level L2 with an intersection over union (IoU) threshold of 0.4 for BEV.

We observe that the detection with ViLGOD and ViLGOD-CP works best in the near range for all object classes. The simple self-supervision with pseudo-labels reinforces the learning for these objects but also improves *pedestrians* and *vehicles* in the medium range by a large

Method	Class	Overall	[0m, 30m)	[30m, 50m)	[50m, +inf)
		AP / APH	AP / APH	AP / APH	AP / APH
ViLGOD	Vehicle	27.2 / 17.0	56.2 / 34.5	20.0 / 13.0	6.0 / 4.0
ViLGOD-CP		29.5 / 21.4	68.8 / 49.2	21.3 / 20.5	1.1 / 0.9
ViLGOD	Pedestrian	12.3 / 11.3	20.0 / 18.0	9.0 / 8.5	6.0 / 5.8
ViLGOD-CP		23.9 / 19.0	42.0 / 33.6	20.5 / 16.6	1.8 / 1.5
ViLGOD	Cyclist	4.7 / 4.7	8.0 / 7.9	2.5 / 2.5	1.9 / 1.9
ViLGOD-CP		4.6 / 4.4	10.9 / 10.5	0.2 / 0.2	0.0 / 0.0

Table 4: **Range evaluation** following the protocols of [10, 11] on the WOD [12] (*i.e.* AP for BEV and 3D, difficulty level L2, IoU 0.4). We extend the range to 160x160 around the ego-vehicle. Even in far ranges (+50m), ViLGOD detects some objects correctly.

Text prompt template	Movable		Vehicle	Pedestrian	Cyclist
	BEV	3D	BEV	BEV	BEV
<i>a point representation of a &lt;class&gt;</i>	36.3	32.3	49.0	16.8	7.6
<i>a silhouette of a &lt;class&gt;</i>	25.8	22.3	29.4	19.0	7.5
<i>a depth map of &lt;class&gt;</i>	29.5	26.0	39.0	16.5	7.5

Table 5: **Text prompt template evaluation** on WOD [12] (following the protocols of [10, 11], *i.e.* AP for BEV and 3D, difficulty level L2, IoU 0.4). We show detection results with different text input templates.

margin. Only *cyclists*, which are underrepresented in the dataset in the first place and additionally not well detected by ViLGOD, degenerate in the middle to far distances. Additional augmentations, such as a pseudo-ground truth database or a more complex training routine, could alleviate this negative effect.

### 3 Impact of Text Prompts

**Text prompt templates.** To bridge the modality gap between 3D point clouds and 2D images, we generate depth maps from 3D point segments with varying densities (depending on the distance to the ego-vehicle). Although not specifically trained for depth images, CLIP [13] can still classify many of these projections correctly. An important design decision is the text prompt we provide CLIP to get the best feature representation matching the image features. In Table 5, we show two additional template variants, *i.e.* *a depth map of <class>* and *a silhouette of <class>*, describing the projected image. It can be observed that *a point representation of <class>* leads to the best results. However, *a silhouette of <class>* seems to be preferable for *pedestrians* but performs worse overall.

## References

- [1] Stefan Baur, Frank Moosmann, and Andreas Geiger. LISO: Lidar-only Self-Supervised 3D Object Detection. *arXiv CoRR*, abs/2403.07071, 2024.

- [2] Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. In *Proc. PAKDD*, 2013.
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. ICLR*, 2021.
- [4] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R. Qi, Xinchun Yan, Scott Ettinger, and Dragomir Anguelov. Motion Inspired Unsupervised Perception and Prediction in Autonomous Driving. In *Proc. CVPR*, 2022.
- [5] 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 *Proc. ICML*, 2021.
- [6] Jenny Seidenschwarz, Aljoša Ošep, Francesco Ferroni, Simon Lucey, and Laura Leal-Taixé. SeMoLi: What Moves Together Belongs Together. In *Proc. CVPR*, 2024.
- [7] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proc. CVPR*, 2020.
- [8] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D Object Detection and Tracking. In *Proc. CVPR*, 2021.
- [9] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyao Zeng, Shanghang Zhang, and Peng Gao. PointCLIP V2: Prompting CLIP and GPT for Powerful 3D Open-world Learning. In *Proc. CVPR*, 2023.