Is clustering enough for LiDAR instance segmentation? A state-of-the-art training-free baseline

Corentin Sautier[1,2] Gilles Puy[2] Alexandre Boulch[2] Renaud Marlet[1,2] Vincent Lepetit[1]

[1]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, France [2]Valeo.ai, France

Abstract

Panoptic segmentation of LiDAR point clouds is fundamental to outdoor scene understanding, with autonomous driving being a primary application. While state-of-the-art approaches typically rely on end-to-end deep learning architectures and extensive manual annotations of instances, the significant cost and time investment required for labeling large-scale point cloud datasets remains a major bottleneck in this field. In this work, we demonstrate that competitive panoptic segmentation can be achieved using only semantic labels, with instances predicted without any training or annotations. Our method outperforms most state-of-the-art supervised methods on standard benchmarks including SemanticKITTI and nuScenes, and outperforms every publicly available method on SemanticKITTI as a drop-in instance head replacement, while running in real-time on a single-threaded CPU and requiring no instance labels. It is fully explainable, and requires no learning or parameter tuning. ALPINE combined with state-of-the-art semantic segmentation ranks first on the official panoptic segmentation leaderboard of SemanticKITTI. ps

1. Introduction

To move autonomously in an outdoor environment, an agent must understand and segment its surroundings into categories. With a LiDAR, it involves recognizing the semantics of points and identifying individual instances of "things" (e.g., 'cars' or 'pedestrians'). This capability, known as LiDAR panoptic segmentation, is crucial for tasks such as object avoidance, trajectory forecasting, and path planning.

State-of-the-art (SOTA) approaches typically combine semantic segmentation and instance segmentation by training end-to-end networks using query-based mechanisms [19, 36, 71, 77] or regressing instance centers [30, 31, 38]. Their architectures usually involve instance prediction heads, which require panoptic labels for training.

Code is available at https://github.com/valeoai/Alpine/

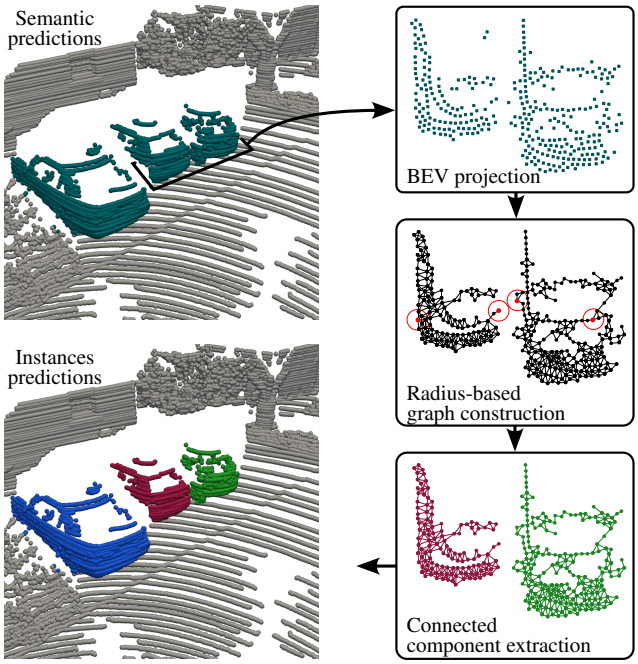
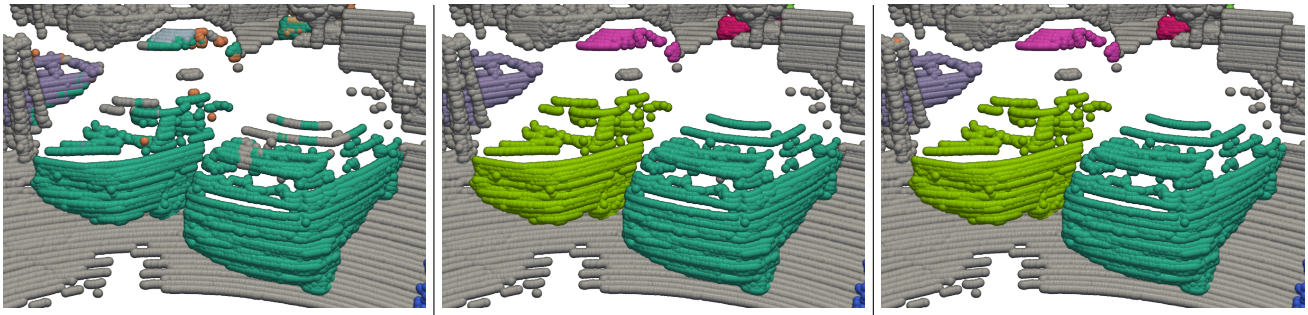


Figure 1. **ALPINE clustering.** For a given semantic class a) we project the points in the BEV space (subsampled on the figure for visualization purpose), b) we build a kNN graph and filter by edge length and c) we extract the connected components.

Instead, early methods [7, 24, 45, 80, 85] proposed unsupervised, learning-free, clustering algorithm to extract instances, sometimes completed by semantic segmentation networks to obtain panoptic segmentation [85]. These methods appear to perform poorly in panoptic segmentation benchmark. One is easily tempted to disregard them and prefer to use end-to-end panoptic segmentation networks requiring costly panoptic segmentation labels.

This paper demonstrates that achieving SOTA panoptic segmentation does not require end-to-end networks or panoptic labels. The early pipeline combining semantic segmentation and unsupervised clustering suffices, as validated by extensive comparisons with current LiDAR panoptic methods. We achieve this result thanks to the use of mod-



D&M [85] w. MinkUNet | ALPINE w. MinkUNet | Ground Truth

Figure 2. **Examples of Instance predictions on SemanticKITTI.** We present the results obtained with D&M [85] and ALPINE restricted to the *car* class. Both methods are training-free clusterings, and use the same MinkUNet to obtain pointwise semantic predictions. When compared to the Ground Truth, we notice that D&M does not satisfactorily separate the cars while ALPINE segment them correctly.

ern, high-performing semantic segmentation networks and to a clustering algorithm inspired by [33] with key changes to boost its performance including working per semantic class and in Bird's eye view rather than 3D.

This work has several practical consequences: (a) Thanks to the performance of modern semantic segmentation networks, it is possible to compete with SOTA methods without using instance labels. (b) The architecture is simple since it does not require a trainable instance head. (c) The method can be directly applied on top of any semantic prediction, without adaptation or access to underlying features. (d) As the clustering algorithm is extremely fast, the panoptic segmentation can be produced at high frequency, as required for embedding on an autonomous vehicle, while using very little compute resources.

This clustering, which we call ALPINE for "A Light Panoptic INstance Extractor", works by constructing a nearest-neighbor graph on semantically consistent points, followed by some refinement using a box splitting mechanism. ALPINE builds on top of any off-the-shelf semantic segmentation network. It is a learning-free method that does not require any knowledge of instance labels relying instead on a rough estimate of per-class object size, which is set using publicly available information found on the web. This makes ALPINE a tuning free method, easy to use on any LiDAR dataset given a semantic segmentation backbone. ALPINE is thus a strong baseline that can be used as reference before training with any instance labels in order to measure the benefits of such additional annotations.

2. Related Work

2.1. Semantic segmentation

Semantic segmentation permits us to separate, at the point level, the main types of objects in a scene. Several methods and backbones have been designed for LiDAR semantic segmentation. They are usually classified under four main categories: point-based [13, 46–49] with the most recent ones leveraging a transformer architecture [68, 69, 83], projection-based [3, 4, 25, 39, 72, 81], sparse convolution-based [12, 14, 29, 88], fusion-based when leveraging different point cloud representations [17, 50, 62, 73].

We present a method that requires semantic predictions as input, regardless of how they are obtained. We test our method using different architectures for semantic segmentation: MinkUNet [14], WaffleIron [46], PTv3 [69]. We chose these methods for their competitive performance and diverse architectures: voxel-based using sparse convolutions, point-based using successive projections on 2D planes, point-based using attention layers.

2.2. Panoptic segmentation

Panoptic segmentation is an extension of semantic segmentation where one should also segments all instances of objects in "thing" classes.

Detection-based methods. The first methods for panoptic segmentation were built on top of networks for semantic segmentation and object detection. The object detection branch puts boxes around each object instance, and the semantic segmentation branch is used to obtain the semantic class of each point. Baseline methods are constructed using off-the-shelf detector [26, 58, 76] and segmentor [12, 14, 62, 65]. More efficient methods often train the two tasks jointly [2, 22, 59, 75, 78], with additional losses specifically designed for panoptic segmentation.

Clustering-based methods. These type of methods also appeared quite early. Our method actually falls in the category. These methods works in two successive steps [84]. A semantic segmentation backbone is first used to assign a class to each point. The points labeled as "thing" are then clustered to find instances.

One of the earliest method, the Euclidean Cluster [54, 55, 84] consists in sequentially visiting each point, assign-

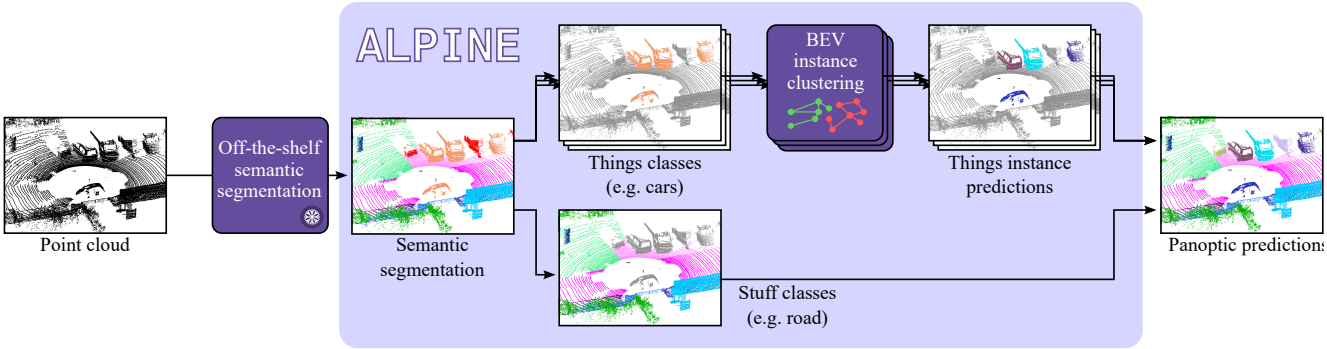


Figure 3. **Overview.** In ALPINE we take the output of a semantic segmentation model and apply our clustering algorithm on each *things* classes to obtain instance masks and form panoptic predictions.

ing to this point a new instance index if it is not already labeled, and labeling all its unlabeled neighbors within a certain radius with the same instance index. Another proposition consists in adapting SLIC [1] from images to point clouds, using voxel representations [45, 84]. More advanced methods [80, 85] design a clustering method tailored for LiDAR point cloud, by taking into consideration the scan-line nature of the sensor's capture. The hyperparameters of these methods are typically tuned according to expected object sizes and/or sensor properties.

More recent methods [16, 20, 21, 27, 30–32, 37, 40, 70, 74, 82, 82, 87] leverage panoptic annotations and replace some hand-crafted steps of the clustering process by learnable steps. For example, GP-S3Net [52] creates a graph for points of thing classes using learned point features to obtain edge features, then perform edge classification.

Among clustering-based methods, a common approach consists in regressing for each point the position of the corresponding instance centroid, allowing an easier clustering done on the predicted cluster centers [20, 21, 28, 31, 38], while having an easy to define training objective. This approach is refined in Panoptic-PHNet where centroid-offset prediction and clustering are performed in the Bird's-Eye-View (BEV) to mitigate vertical over-segmentation.

Query-based methods. These methods [18, 19, 36, 71, 77] build upon MaskFormer [11], an image segmentation method. To propose an end-to-end unified framework for panoptic segmentation, they create a set of (usually learnable) queries fed into a transformer decoder in charge of (softly) assigning points and queries. Each query encodes a different instance. Points responding maximally to the same query are part of the same instance. In these architectures, the point features given to the transformer decoder can be obtained from different backbones working on range-view [18], voxels [36], BEV [77], or using both cartesian and polar space [71] representation of the point cloud.

Use of annotations. Some datasets offer annotations for multiple tasks such as semantic & panoptic segmentation and object detection [8]. Some methods leverage all the available annotation and train a single model on all tasks jointly, benefiting from the additional supervision [10, 78]. At the other end of the spectrum, some methods [42, 43, 57] segment objects without using any annotation. However, these method are unable to predict semantics. In this work, we take inspiration from early clustering-based methods using only semantics annotations.

3. A strong baseline for panoptic segmentation

Our approach is designed around three main components: a fast clustering algorithm having its roots in RBNN [24] and LESS [33], an annotation-free and training-free parameter selection process, and a refinement mechanism based on box splitting.

3.1. ALPINE clustering

The overview of our instance clustering process is depicted in Fig. 1. It is inspired by the clustering done in [33] and works in four main steps.

One semantic class at a time. After semantic prediction, we consider each "thing" class c independently. We denote by P_c the original point cloud P restricted to the points with predicted label c. We denote n_c the size of P_c. Each point cloud P_c is processed separately.

Projection in BEV. Inspired by previous works [26, 30, 56] leveraging the fact that objects of "thing" classes are rarely stacked vertically, we project the 3D points in the Bird's Eye View (BEV) representation (orthogonal projection onto the (x, y) plane). In the following, we assume that points clouds P_c are therefore two-dimensional.

k-nearest neighbors graph. For each point in P_c, we obtain the list of its k closest neighbors (typically $k = 32$), using a 2D-Tree [6]. The distances between the points are computed using only the point coordinates in the BEV representation, and therefore do not involve any learned features. We then construct a directed graph $\mathcal{G}$ where the nodes

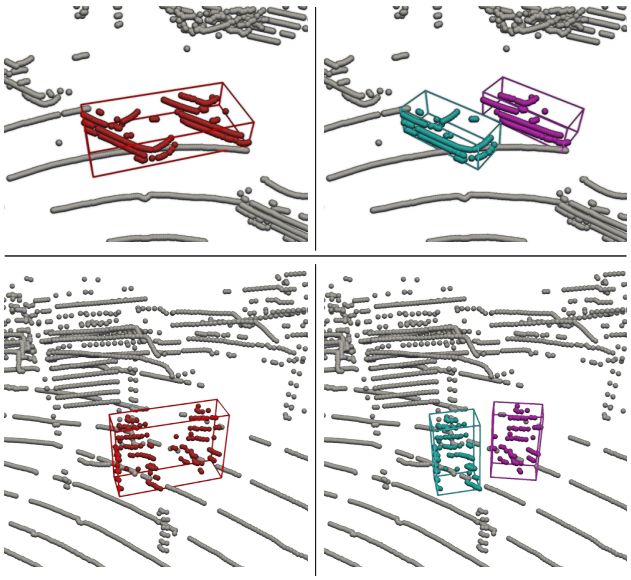


Figure 4. **Example of bounding box splittings.** In the top examples, two cars are parked close to each other. While merged by the clustering, they do not fit in a car's box and the cluster is then split. In the bottom examples, the same mechanism is applied to two pedestrians in a bus shelter. Boxes are shown in 3D for illustrative purposes but the mechanism is purely 2-dimensional.

are the points in P_c and each point is connected by an edge to its k nearest neighbors, at most. We remove connections between points if the distance in BEV is larger than a threshold (see Sec. 3.2). Finally, the adjacency matrix of this directed graph is made symmetric by adding missing half-edges to obtain an undirected graph. The choice of k is not very critical for the method; it needs to be large enough not to miss too many connections with isolated points, while keeping it small enough to benefit from fast runtime. This is verified experimentally in the supplementary material. The complexity of the construction of the 2D-Tree is $\mathcal{O}(n_c \log n_c)$, while the search for the k nearest neighbors for all points is $\mathcal{O}(n_c\, k \log n_c)$.

Connected components. The clusters themselves are obtained by identifying the connected components in the constructed graph. The complexity of this algorithm is $\mathcal{O}(n_c k)$ in the worst case. We associate to each component a different instance index.

Key differences with LESS and RBNN. ALPINE builds upon the strong graph-based clustering foundations laid by RBNN [24] and LESS [33]; however applying a few key changes making it both stronger and more practical to use. (1) Unlike in both other methods, the adjacency of each point is limited to at most k reducing the complexity of the algorithm from $\mathcal{O}(n_c^2)$ to $\mathcal{O}(n_c\, k \log n_c)$, scaling better to large clusters; (2) it is also applied class-wise, (3) and in BEV, with a per-class constant threshold. Besides, while conceptually similar, RBNN did not use connected components to find clusters, relying instead on a custom cluster merging scheme. Contrary to ours, LESS' clustering was proposed as a pre-annotation tool only.

3.2. Choice of distances and thresholds

In order to decide whether two points should be connected in $\mathcal{G}$, and thus should belong to the same instance, we use a threshold t_c on the distance between two points in BEV.

Class-wise thresholds. When considering whether two points should be connected, an important quantity to consider is the typical size of the objects to segment: points that are 3 meters apart are never going to belong to the same pedestrian, but might belong to the same truck. We therefore use separate thresholds t_c for each semantic class c. The only information that we require to set t_c is the an estimate of the average size of the objects in each class.

This information can be obtained *without any ground-truth annotation, and even without any access to the dataset*. For example, for cars, one can retrieve from the Internet the average size of vehicles where the system is to be deployed. For each class, we can thus get an estimate of the typical size of bounding boxes around objects in this class. We set t_c to the smallest side of this reference bounding box. More considerations as to how we obtained such approximate average boxes are given in Sec. 4.2.

This parametrization addresses a known domain shift issue related to sizes when training and testing in different countries [66]. It enables a simple form of domain adaptation where only a few scalars t_c have to be set from one country to another, besides any adaptation or generalization regarding the underlying semantics.

Range vs threshold. The distance between two points in a LiDAR point clouds naturally increases with their range (distance between a point and the sensor): LiDAR point clouds are less dense far away from the sensor. For this reason, to similarly create object clusters and decide if an edge between two points should be removed or not, LESS [33] used a threshold that varies with the range. We tested such a strategy in Sec. 4.6 but found no improvement in doing so. We thus simply use a constant class-wise threshold.

3.3. Box splitting

To refine our segmentation further, we use a box-splitting strategy. When a cluster of a given semantic class c does not fit within the reference bounding box of this class (as defined in Sec. 3.2), enlarged by a constant (not class-wise) proportional margin (of 30%) to account for bigger-than-average objects, it likely actually contains multiple instances of this class. In this case, our box-splitting method performs a binary search on the threshold parameter t to recursively find the largest clusters that all fit into the reference bounding box. The pseudo-code for this algorithm is

given in Alg. 1, and an illustrative example is provided in Fig. 4. The box-fitting procedure that is used internally in the algorithm is borrowed from the literature [79] and discussed in the appendix.

Algorithm 1: Box splitting algorithm. This algorithm splits clusters that do not fit in B into sub-clusters that all fit in B. **box_fitting** returns the smallest bounding box encompassing P and **clusterize** is the clustering algorithm described in Sec. 3.1.

```
function split_cluster(P, B, t)
  input: Points P, Average box B, Threshold t
  output: List of cluster points
  if box_fitting(P) fits in B then
    return [P] // Points already fit in B
  else
    t ← t/2
    dt ← t
    while true do
      dt ← dt/2
      C ← clusterize(P, t)
      if len(C) = 1 then
        t ← t − dt// Decrease threshold
      else if len(C) > 2 then
        t ← t + dt// Increase threshold
      else
        return split_cluster(C[0], B, t) +
         split_cluster(C[1], B, t)
```

4. Experiments

4.1. Datasets and metrics

nuScenes [8] is a dataset captured in Boston and Singapore, with a 32-beam LiDAR sensor. It contains 10 things classes and 6 stuff classes. The instance labels were automatically generated by combining object detection and semantic segmentation labels, and were not manually curated. The results are reported on the official validation set.

SemanticKITTI [5] is a dataset captured around Karlsruhe in Germany with a 64-beam LiDAR sensor. It contains 8 things classes and 11 stuff classes. The instance labels were manually annotated. The results are reported on Sequence 08, commonly used as the validation sequence.

SemanticPOSS [44] is a smaller dataset captured in the surroundings of the Peking University with a 40-beam LiDAR, focusing on having many dynamic (thing) objects. It contains only 3 things classes, namely person, rider and car, as well as 10 stuff classes. The results are reported on Sequence 02, used as a validation sequence as done in [52].

We use standard metrics, which are defined in Appendix.

Method		Inst. lbs.	TTA	Ens.	PQ	PQ†	RQ	SQ	mIoU
LCPS (lidar)	[82]	✓	✗	✗	55.7	65.2	65.8	74.0	61.1
D&M (Cyl.3D)	[85]	✗	✗	✗	57.2	-	-	-	-
Panop.-PolarNet	[87]	✓	✗	✗	59.1	64.1	70.2	78.3	64.5
MaskPLS	[36]	✓	✗	✗	59.8	-	69.0	76.3	61.9
DS-Net (SPVCNN)	[21]	✓	?	?	61.4	65.2	72.7	79.0	69.6
Panop.-PHNet	[30]	✓	✗	✗	61.7	-	-	-	65.7
PANet	[37]	✓	?	?	61.7	66.6	71.8	79.3	68.1
D&M (Mink.)	[85]	✗	(✓)	✗	61.8	66.2	72.8	79.6	71.4
CenterLPS	[38]	✓	?	?	62.1	67.0	72.0	80.7	68.1
ALPINE (PTv3)		✗	(✓)	✗	62.4	66.2	72.0	76.7	67.3
P3Former	[71]	✓	✗	✗	62.6	66.2	72.4	76.2	-
CFNet	[31]	✓	✗	✗	62.7	67.5	-	-	67.4
LPST	[2]	✓	✗	✗	63.1	70.8	73.1	79.2	69.7
GP-S3Net	[52]	✓	?	?	63.3	71.5	75.9	81.4	73.0
DQFormer	[77]	✓	✗	✗	63.5	67.2	73.1	81.7	-
ALPINE (WI)		✗*	(✓)	✗	64.2	69.0	74.1	79.7	70.3
ALPINE (Mink.)		✗	✗	✗	64.2	68.9	74.1	84.4	70.7
PUPS (w/o ens.)	[61]	✓	✗	✗	64.4	68.6	74.1	81.5	-
IEQLPS	[19]	✓	?	?	64.7	68.1	74.7	81.3	-
ALPINE (Mink.)		✗	(✓)	✗	65.9	70.2	75.5	81.4	72.2
PUPS	[61]	✓	✓	✓	66.3	70.2	75.6	82.5	-
ALPINE (Mink.+PTv3)		✗	(✓)	(✓)	66.6	70.8	76.1	82.6	72.0

Mink.: MinkUNet [14] WI: WaffleIron-256 [46]

*: the publicly available model for WI-256 [46] used instance annotations in its data augmentation pipeline

Table 1. Panoptic segmentation results on the validation set of SemanticKITTI. 'TTA' and 'Ens.' stand for Test-Time Augmentation and Ensembling. (✓) denotes that TTA/ensemble was used on the semantic head only. The main metric is the PQ. A more detailed version of this table is available in the supplementary material.

4.2. Size parameters setting

The parameters needed in ALPINE, as described in Secs. 3.2 and 3.3, are set after estimating the average size of object bounding boxes in each class.

We consider two kinds of size parameter settings: the web-based setting obtains sizes from a few queries on the Internet; the dataset-based setting exploits ground-truth knowledge coming with datasets.

For web-based box sizes on nuScenes, we used the publicly available average size of cars in the United States in 2018 (15.6×6.3 ft) [15]; we made the simplifying assumption that buses, trailers, trucks and construction vehicles have size 10×3 m; we used standard bike and motorcycle sizes found online [9, 64] ; we made the the approximation that pedestrians fit in box of half their arm span, estimates using [51, 67]; and we made an educated size guess of 2×0.5 m for barriers and 40 cm for cones.

Similar web-based information was exploited to set the sizes for SemanticKITTI. We used the average car size in Europe of 4.4×1.8 m [41]. The same found sizes for bi-

Method	Inst. lbs.	TTA	Ens.	PQ	$PQ^{\dagger}$	RQ	SQ	mIoU
DS-Net(SPVCNN)[21]	✓	?	?	64.7	67.6	76.1	83.5	76.3
Panop.-PolarNet [87]	✓	✗	✗	67.7	71.0	78.1	86.0	69.3
PANet [37]	✓	?	?	69.2	72.9	80.7	85.0	72.6
CPSeg [28]	✓	?	?	71.1	75.6	82.5	85.5	73.2
LCPS (lidar) [82]	✓	✗	✗	72.9	77.6	82.0	88.4	75.1
PUPS [61]	✓	?	?	74.7	77.3	83.3	89.4	-
Panop.-PHNet [30]	✓	✗	✗	74.7	77.7	84.2	88.2	79.7
CFNet [31]	✓	✗	✗	75.1	78.0	84.6	88.8	79.3
P3Former [71]	✓	✗	✗	75.9	78.9	84.7	89.7	-
CenterLPS [38]	✓	?	?	76.4	79.2	86.2	88.0	77.1
ALPINE (WI)	✗	✗	✗	76.9	79.9	85.7	89.3	80.3
IEQLPS [19]	✓	?	?	77.1	79.1	86.5	88.2	-
LPST [2]	✓	✗	✗	77.1	79.9	86.5	88.6	80.3
DQFormer [77]	✓	✗	✗	77.7	79.5	86.8	89.2	-
ALPINE (WI)	✗	(✓)	✗	77.9	80.7	86.5	89.6	81.4
ALPINE (PTv3)	✗	(✓)	✗	78.9	81.3	87.0	90.4	81.5
ALPINE (WI+PTv3)	✗	(✓)	(✓)	79.5	81.9	87.6	90.5	82.7
LidarMultiNet [78]	✓	✓	✓	81.8	-	89.7	90.8	83.6

WI: WaffleIron-768 [46]

Table 2. Panoptic segmentation results on the validation set of nuScenes. 'TTA' and 'Ens.' stand for Test-Time Augmentation and Ensembling. (✓) denotes that TTA/ensemble was used on the semantic head only. The main metric is the PQ. A more detailed version of this table is available in the supplementary material.

Method	Instance labels	PQ	$PQ^{\dagger}$	RQ	SQ	mIoU
LPSAD (in [52])[40]	✓	22.5	32.7	34.0	53.5	35.5
TORNADO-Net [17]	✓	33.7	43.3	46.0	68.4	44.5
DS-Net [20]	✓	35.6	45.9	49.2	68.6	54.5
GP-S3Net [52]	✓	48.7	60.3	63.7	61.3	61.8
ALPINE (PTv3)	✗	51.4	57.7	67.2	74.3	58.3

Table 3. Panoptic segmentation results on Sequence 02 of SemanticPOSS as validation set. The main metric is the PQ.

cycle and motorcycle as in nuScenes [9, 64] as well as for bicyclist and motorcyclist. The class person was treated as in nuScenes. For truck and other-vehicle, we made the same assumption as in nuScenes of a 10×3 m size.

For the 3 classes of SemanticPOSS, we used the value we obtained for the equivalent class in SemanticKITTI, i.e. pedestrians, bicyclists and cars.

For dataset-based box sizes, we use the average size of the bounding boxes used in [63, 76]. On nuScenes, these average size is provided for all 10 thing classes. On SemanticKITTI, these sizes are only available for cars, pedestrians and bicyclists; we use the value computed on nuScenes for the 5 remaining classes. On SemanticPOSS, we just re-use the sizes obtained on SemanticKITTI.

In the end, we did not witness a significant change of performance between both sets of parameters in the experiments that follow and use by default, in all results, the annotation-free web-based sizes.

Rank/ Method	PQ	$PQ^{\dagger}$	RQ	SQ	PQ_{Th}	mIoU
5/ P3Former	65.3	67.8	74.9	86.6	67.4	66.1
4/ PUPS	65.7	70.3	75.8	85.7	68.1	68.5
3/ P3Former & ALPINE	66.1	68.7	75.9	86.7	69.4	66.1
2/ UniSeg	67.2	72.1	78.1	85.5	67.5	73.8*
1/ UniSeg & ALPINE	**70.2**	**75.1**	**80.5**	**86.7**	**74.3**	**75.1***

Table 4. **SemanticKITTI test set results** (official leaderboard, June 2025). We used P3Former's and UniSeg's semantic predictions and used ALPINE for instance segmentation. *(See Sec. 4.4)

Panop. seg.	sem. mIoU	Original head PQ	$PQ^{\dagger}$	PQ_{Th}	ALPINE PQ	$PQ^{\dagger}$	PQ_{Th}	FPS (Hz)
DS-Net	63.5	57.7	63.4	61.8	**58.5**	**64.1**	**63.6**	0.7
Panoptic-P.Net	64.4	58.9	63.9	65.2	**59.5**	**64.5**	**66.7**	1.6
MaskPLS	61.9	59.8	63.6	63.7	**60.2**	**64.0**	**64.7**	N/A
LCPS (lidar)	64.3	61.3	**65.6**	70.0	**61.4**	**65.6**	**70.1**	1.8
PANet	67.9	61.2	66.1	66.7	**61.3**	**66.3**	**66.9**	7.1
P3Former	66.8	62.6	66.2	69.4	**62.9**	**66.5**	**70.1**	N/A

Table 5. **Comparison between ALPINE and the instance segmentation capabilities of other methods**, on the validation set of SemanticKITTI. For each panoptic segmentation method with available code and checkpoint, we isolated the plain semantic predictions and report the mIoU ('sem.'), and constructed on top of them our instances ('ALPINE'), which we compare to the instances obtained by the original methods ('Original head'). An indicative instance prediction frequency (FPS) is given after subtracting the time of the semantic forward pass to the total runtime.

4.3. Comparison to the state-of-the-art

On all datasets, we apply ALPINE combined with different backbones trained for semantic segmentation. We re-used the provided checkpoints when publicly available or retrain them ourselves with the provided code.

SemanticKITTI. Tab. 1 presents our results on the validation set of SemanticKITTI. On this dataset, we consider 3 different semantic segmentation backbones: PTv3 [69], WI-256 [46], and MinkUnet [14] from OpenPCSeg [34].

ALPINE is able to reach SOTA performance (PQ), outperforming, e.g., the end-to-end method PUPS [61]. Moreover, when comparing to other methods at equivalent semantic segmentation mIoU, ALPINE is comparable or outperforms these methods: ALPINE w. PTv3 vs CFNet [31], PANet [37], CenterLPS [38]; or ALPINE w. MinkUNet vs GP-S3Net [52]. It shows that ALPINE benefits easily from better semantic segmentation backbones.

nuScenes. Tab. 2 presents our results on the validation set of nuScenes. On this dataset, we consider two different semantic segmentation backbones: PTv3 [69] and WI-768 [46]. We are only outperformed by LidarMultiNet [78],

ALPINE †		DBSCAN		HDBSCAN		D&M	
PQ	FPS	PQ	FPS	PQ	FPS	PQ	FPS
65.5	**14.4 Hz**	56.7	3.2 Hz	55.1	4.5 Hz	61.8	0.5 Hz

Table 6. **Comparing PQ and computation times of clustering methods** on the validation set of SemanticKITTI, applied to the same MinkUNet semantics. †No box splitting.

Method	PQ	RQ	SQ	mIoU
3DUIS	60.2	70.7	79.7	72.2
UNIT	61.1	72.0	79.0	72.2
ALPINE	65.9	75.5	81.4	72.2

Table 7. **Comparison to semantic-free instance segmentations** evaluated on the SemanticKITTI validation set.

Sem. Seg	Inst. Seg	PQ	RQ	SQ	mIoU
Oracle	ALPINE	96.2	98.4	97.8	100.0
WI-768	*Oracle*	81.7	88.1	94.1	81.4
PTv3	*Oracle*	83.2	87.6	92.8	81.5
Oracle	*Oracle*	100.0	100.0	100.0	100.0

Table 8. **Oracle results** evaluated on the nuScenes validation set.

which was trained using both semantics and object boxes, using ensembling and test-time augmentations (TTA) on both tasks, and benefits from a better semantic prediction.

SemanticPOSS. Tab. 3 presents results obtained on Sequence 02 of SemanticPOSS. We notice that ALPINE combined with PTv3 [69] reaches state-of-the-art results.

4.4. Study of ALPINE's components

Comparison to other instance heads. We present results obtained with ALPINE on the test set of SemanticKITTI. First, we use the public model of P3Former for semantic prediction and extract object instances using ALPINE. The PQ which is at 65.3 with P3Former improves to 66.1 thanks to ALPINE, the third best entry on the leaderboard. Second, we repeat the same procedure with the semantic labels of UniSeg [35] (labels kindly provided by the authors). UniSeg with ALPINE's instance labels ranks first on the leaderboard of SemanticKITTI. These results highlight the high quality of the instances extracted by our method. We note nevertheless that these results are not strictly obtained without instance labels as both UniSeg and P3Former did use them during training. We also remark that while the semantic mIoU of P3Former and P3Former&ALPINE are identical, UniSeg&ALPINE's semantic mIoU is above UniSeg's: 75.1 vs 73.8. We did not modify UniSeg's semantic labels (75.1 is in line with the score of UniSeg on the *semantic* segmentation leaderboard). A misformatting in UniSeg's semantic labels reduces their score from 75.1 to 73.8, which we could verify on the semantic benchmark.

We conduct a more complete study on the validation set of SemanticKITTI. Tab. 5 shows how all panoptic methods for which we found code and checkpoints on SemanticKITTI compare against ALPINE *at equal semantic prediction*: we use the semantic prediction of the original method but replace the instance prediction mechanism with ALPINE. We show that ALPINE outperforms every single method, while using no annotation and requiring no GPU. This further indicates that current instance prediction heads do not perform as well as expected and can be outmatched by a simple clustering scheme without training nor instance annotation. An indicative "instance runtime" of all methods have been obtained by subtracting the segmentation forward pass to the total runtime, when this was possible. This approximate measurement is comparable (while not directly equivalent) to the ALPINE FPS reported in Tab. 6.

Comparison to other clustering methods. Tab. 6 shows that ALPINE is both faster and better performing on our task than other clustering methods, namely the standard DBSCAN and HDBSCAN, as well as D&M [85], which was specifically designed for this task too. We are also within the limit of real-time processing, as SemanticKITTI is captured at 10 Hz, while we use only limited computing resources (single-core CPU usage). In these experiments, all clustering methods use the same semantics, predicted by MinkUNet [14]. The box splitting was not used on any method in order to compare clustering capacity only. DBSCAN and HDBSCAN were performed on BEV, with fixed parameters of $\epsilon = 1$ for DBSCAN and parameters gathered from 3DUIS [42] and ϵ proportional to t_c for HDBSCAN; with the proportionality coefficient obtained with a fine parameter search. Those settings were those that we found worked best. We see that our clustering is outperforming D&M by a comfortable margin: $+3.7$ PQ pts. With box-splitting, ALPINE would gain $+0.4$ further PQ pts.

Comparison to semantic-free instance segmentation. 3DUIS [42] and UNIT [57] are two unsupervised instance segmentation methods. They propose to find individual instances of objects in a totally unsupervised manner, regardless of semantics. While their task is more involved, as they cannot rely on predicted semantics mask, we can still compare their instance predictions with that of ALPINE. In practice, we use the intersection of their instance mask and each semantic mask as panoptic predictions. As can be noted in Tab. 7, ALPINE significantly outperforms those methods. It can still be noted that both unsupervised method work quite impressively, almost reaching the level of performance of D&M while not using any semantic information.

BEV	Box splitting	Threshold	PQ	PQ†	mIoU
✗	✗	const.	76.3	78.7	81.5
✓	✗	const.	76.9	79.3	81.5
✗	✓	const.	77.1	79.5	81.5
✓	✓	const.	78.9	81.3	81.5
✗	✗	$\propto \lvert d \rvert$ [33]	75.9	78.3	81.5

Table 9. **Ablation study** of our different clustering components on the nuScenes validation set. $|d|$ is the distance to the sensor.

Method	sem. mIoU	statistics-based PQ	statistics-based PQ†	statistics-based PQ$_{\text{Th}}$	web-based PQ	web-based PQ†	web-based PQ$_{\text{Th}}$
SemanticKITTI							
MinkUNet w/o TTA	70.7	64.1	68.8	72.5	64.2	68.9	72.6
PTv3	67.3	62.4	66.1	65.9	62.4	66.2	66.1
WI-256	70.3	64.0	68.9	69.0	64.2	69.0	69.3
MinkUNet	72.2	65.8	70.1	73.7	65.8	70.1	73.7
MinkUNet & PTv3	72.0	66.5	70.7	73.2	66.6	70.8	73.4
nuScenes							
WI-768 w/o TTA	80.3	77.2	80.1	78.3	76.9	89.9	77.9
WI-768	81.4	78.2	81.0	79.0	77.9	80.7	78.6
PTv3	81.5	79.0	81.4	79.6	78.9	81.3	79.3
WI-768 & PTv3	82.7	79.8	82.2	80.5	79.5	81.9	80.2
SemanticPOSS							
PTv3	58.3	51.1	57.4	63.6	51.4	57.7	64.8

Table 10. **Comparison between statistics-based and web-based parameters settings.** Run details can be found in Tabs. 1 and 2.

4.5. Evaluation of the clustering upper bound

To evaluate what remains to be gained by developing better semantics or instance heads, we devise an oracle head for both subtasks: a "semantic oracle" using the ground-truth semantic segmentation as semantic predictions, and an "instance oracle" splitting the semantic predictions of a given method using the ground-truth instance boundaries, without affecting the semantic prediction. Results, in Tab. 8, shows how much scores are expected to improve with both oracles. When using the "semantic oracle", the PQ on nuScenes raises to 96.2, (and 99.0 on SemanticKITTI), while using PTv3 predictions and the instance oracle only raises the PQ by +4.3 to 83.2, showing that instance extraction is much more saturated than semantic segmentation.

4.6. Ablation and sensitivity study

Influence of each ALPINE modules. We present our ablation study in Tab. 9, conducted on the validation set of nuScenes, using PTv3 as semantic backbone. We see that constructing the graph in BEV or post-processing the result with our box splitting strategy improve the results: respectively $+0.6$ point and $+0.8$ point in PQ over the baseline. The combination of both improve the PQ by 2.6 points.

Instead of using a constant threshold to keep or remove the edge between two points when constructing our graph, we also tested a threshold varying linearly with the range, as used in LESS [33]. The linear coefficient is the ALPINE threshold t_c multiplied by a constant optimized at dataset scale to maximize performance. Despite this optimization, we find that, in our case, using a constant threshold depending on the typical size of the objects leads to better results.

Dataset-based vs. web-based results. Tab. 10 gives the difference in scores for the different sets of parameters described in Sec. 4.2. Both choices of parameters give very similar results, from which we conclude ALPINE parameters are both easy to setup, have a sensible meaning, and do not require annotations.

5. Limitations and scope of this article

Since our method has a single fixed threshold per class, failure cases can be crafted by making two objects closer than the chosen threshold. It would thus be tempting to build intricate training-based solutions, for example by learning the neighboring condition [52, 53].

However, ALPINE proves that old techniques can reach state-of-the-art results, with no need for any object label, tuning, learning or extremely complex designs. ALPINE is a very strong baseline that future methods will have to beat to prove that their design actually benefits the task.

Furthermore, this object resolution limitation naturally reduces as the point density (or equivalently the LiDAR angular resolution) increases and intra-object average points distances reduces, leading to believe this limitation could mostly solve itself with new LiDAR technologies.

6. Conclusion

ALPINE is reaching state-of-the-art results in LiDAR panoptic segmentation with no instance labels, no training, and no heavy computation requiring a GPU. Furthermore, ALPINE can be taken off-the-shelf and applied as is on top of any semantic predictor. Our study of ALPINE in replacement of prior instance predictions' heads proves that instance extraction is largely saturated.

This work shows that (1) instance labels are not fully exploited by current LiDAR panoptic segmentation methods and (2) simple clustering as ALPINE should always be tested as a baseline in order to assess the gain obtained with trainable panoptic heads. We also release our code for integration into any semantic segmentation backbone for panoptic segmentation.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 2012. 3

[2] Abhinav Agarwalla, Xuhua Huang, Jason Ziglar, Francesco Ferroni, Laura Leal-Taixé, James Hays, Aljoša Ošep, and Deva Ramanan. LiDAR panoptic segmentation and tracking without bells and whistles. In *IROS*, 2023. 2, 5, 6, 16, 18

[3] Eren Erdal Aksoy, Saimir Baci, and Selcuk Cavdar. SalsaNet: Fast Road and Vehicle Segmentation in LiDAR Point Clouds for Autonomous Driving. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020. 2

[4] Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. RangeViT: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *CVPR*, 2023. 2

[5] 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*, 2019. 5

[6] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. 3

[7] Igor Bogoslavskyi and Cyrill Stachniss. Fast range image-based segmentation of sparse 3d laser scans for online operation. In *IROS*, 2016. 1

[8] 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. 3, 5

[9] carparkjourney.wordpress.com. What is the average size of a motorbike. https://carparkjourney.wordpress.com/2013/07/16/what-is-the-average-size-of-a-motorbike/, 2013. 5, 6

[10] Qi Chen, Sourabh Vora, and Oscar Beijbom. Polarstream: Streaming object detection and segmentation with polar pillars. *NeurIPS*, 34, 2021. 3

[11] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 3

[12] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (AF)2-S3Net: Attentive Feature Fusion With Adaptive Feature Selection for Sparse Semantic Segmentation Network. In *CVPR*, 2021. 2

[13] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. PointMixer: MLP-Mixer for Point Cloud Understanding. In *ECCV*, 2022. 2

[14] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019. 2, 5, 6, 7

[15] FINN. Supersized: A decade-long growth of U.S. cars reveals a bigger picture. https://www.finn.com/en-DE/campaign/supersized, 2024. 5

[16] Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari. Panoster: End-to-end panoptic segmentation of LiDAR point clouds. *IEEE Robotics and Automation Letters*, 6(2):3216–3223, 2021. 3, 16

[17] Martin Gerdzhev, Ryan Razani, Ehsan Taghavi, and Liu Bingbing. Tornado-net: multiview total variation semantic segmentation with diamond inception module. In *ICRA*, 2021. 2, 6, 16, 18

[18] Yi Gu, Yuming Huang, Chengzhong Xu, and Hui Kong. Maskrange: A mask-classification model for range-view based LiDAR segmentation. *arXiv preprint arXiv:2206.12073*, 2022. 3

[19] Ngoc-Quan Ha-Phan, Hung Viet Vuong, and Myungsik Yoo. Instance embedding as queries for detr-based LiDAR panoptic segmentation. *IEEE Transactions on Intelligent Vehicles*, 2024. 1, 3, 5, 6, 16, 18

[20] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. LiDAR-based panoptic segmentation via dynamic shifting network. In *CVPR*, 2021. 3, 6, 18

[21] Fangzhou Hong, Lingdong Kong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Unified 3d and 4d panoptic segmentation via dynamic shifting networks. *PAMI*, 2024. 3, 5, 6, 16, 18

[22] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *arXiv preprint arXiv:2004.08189*, 2020. 2, 16, 18

[23] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, 2020. 16

[24] Klaas Klasing, Dirk Wollherr, and Martin Buss. A clustering method for efficient segmentation of 3d laser data. In *ICRA*, 2008. 1, 3, 4

[25] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *ICCV*, 2023. 2

[26] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 2, 3

[27] Enxu Li, Ryan Razani, Yixuan Xu, and Bingbing Liu. SMAC-Seg: LiDAR panoptic segmentation via sparse multi-directional attention clustering. In *ICRA*, 2022. 3, 18

[28] Enxu Li, Ryan Razani, Yixuan Xu, and Bingbing Liu. CPSeg: Cluster-free panoptic segmentation of 3d LiDAR point clouds. In *ICRA*, 2023. 3, 6, 18

[29] Jiale Li, Hang Dai, and Yong Ding. Self-distillation for robust lidar semantic segmentation in autonomous driving. In *ECCV*, 2022. 2

[30] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. Panoptic-PHNet: Towards real-time and high-precision LiDAR panoptic segmentation via clustering pseudo heatmap. In *CVPR*, 2022. 1, 3, 5, 6, 16, 18

[31] Xiaoyan Li, Gang Zhang, Boyue Wang, Yongli Hu, and Baocai Yin. Center focusing network for real-time LiDAR panoptic segmentation. In *CVPR*, 2023. 1, 3, 5, 6, 16, 18

[32] Minzhe Liu, Qiang Zhou, Hengshuang Zhao, Jianing Li, Yuan Du, Kurt Keutzer, Li Du, and Shanghang Zhang. Prototype-voxel contrastive learning for LiDAR point cloud panoptic segmentation. In *ICRA*, 2022. 3, 18

[33] Minghua Liu, Yin Zhou, Charles R Qi, Boqing Gong, Hao Su, and Dragomir Anguelov. LESS: Label-efficient semantic segmentation for LiDAR point clouds. In *ECCV*, 2022. 2, 3, 4, 8

[34] Youquan Liu, Yeqi Bai, Lingdong Kong, Runnan Chen, Yuenan Hou, Botian Shi, and Yikang Li. OpenPCSeg: An open source point cloud segmentation codebase. https://github.com/PJLab-ADG/PCSeg, 2023. 6

[35] Youquan Liu, Runnan Chen, Xin Li, Lingdong Kong, Yuchen Yang, Zhaoyang Xia, Yeqi Bai, Xinge Zhu, Yuexin Ma, Yikang Li, Yu Qiao, and Yuenan Hou. UniSeg: A unified multi-modal lidar segmentation network and the openpcseg codebase. In *ICCV*, 2023. 7

[36] Rodrigo Marcuzzi, Lucas Nunes, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Mask-based panoptic LiDAR segmentation for autonomous driving. *IEEE Robotics and Automation Letters*, 8(2):1141–1148, 2023. 1, 3, 5, 16, 18

[37] Jianbiao Mei, Yu Yang, Mengmeng Wang, Xiaojun Hou, Laijian Li, and Yong Liu. PANet: LiDAR panoptic segmentation with sparse instance proposal and aggregation. In *IROS*, 2023. 3, 5, 6, 16, 18

[38] Jianbiao Mei, Yu Yang, Mengmeng Wang, Zizhang Li, Xiaojun Hou, Jongwon Ra, Laijian Li, and Yong Liu. CenterLPS: Segment instances by centers for LiDAR panoptic segmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1884–1894, 2023. 1, 3, 5, 6, 16, 18

[39] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. In *IROS*, 2019. 2

[40] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss. LiDAR panoptic segmentation for autonomous driving. In *IROS*, 2020. 3, 6, 16, 18

[41] Motor1. Average vehicle size in the US and europe is larger than ever. https://www.motor1.com/news/707996/vehicles-larger-than-ever-usa-europe/, 2024. 5

[42] Lucas Nunes, Xieyuanli Chen, Rodrigo Marcuzzi, Aljosa Osep, Laura Leal-Taixé, Cyrill Stachniss, and Jens Behley. Unsupervised Class-Agnostic Instance Segmentation of 3D LiDAR Data for Autonomous Vehicles. *IEEE Robotics and Automation Letters (RA-L)*, 2022. 3, 7

[43] Lucas Nunes, Louis Wiesmann, Rodrigo Marcuzzi, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Temporal Consistent 3D LiDAR Representation Learning for Semantic Perception in Autonomous Driving. In *CVPR*, 2023. 3

[44] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. SemanticPOSS: A point cloud dataset with large quantity of dynamic instances. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020. 5

[45] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *CVPR*, 2013. 1, 3

[46] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Using a waffle iron for automotive point cloud semantic segmentation. In *ICCV*, 2023. 2, 5, 6, 16

[47] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 2017.

[48] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 2017.

[49] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In *NeurIPS*, 2022. 2

[50] Haibo Qiu, Baosheng Yu, and Dacheng Tao. GFNet: Geometric Flow Network for 3D Point Cloud Semantic Segmentation. *Transactions on Machine Learning Research*, 2022. 2

[51] Philip H Quanjer, André Capderou, Mumtaz M Mazicioglu, Aggarwal Ashutosh N, Banik Sudip Datta, Popovic Stevo, Tayie Francis A, and Zelter Marc Golshan Mohammad, Ip Mary S M. All-age relationship between arm span and height in different ethnic groups. In *European Respiratory Journal*, 2014. 5

[52] Ryan Razani, Ran Cheng, Enxu Li, Ehsan Taghavi, Yuan Ren, and Liu Bingbing. GP-S3Net: Graph-based panoptic sparse semantic segmentation network. In *ICCV*, 2021. 3, 5, 6, 8, 16, 18

[53] Damien Robert, Hugo Raguet, and Loic Landrieu. Scalable 3d panoptic segmentation as superpoint graph clustering. In *3DV*, 2024. 8

[54] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24:345–348, 2010. 2

[55] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, 2011. 2

[56] Corentin Sautier, Gilles Puy, Alexandre Boulch, Renaud Marlet, and Vincent Lepetit. BEVContrast: Self-supervision in bev space for automotive lidar point clouds. In *3DV*, 2024. 3

[57] Corentin Sautier, Gilles Puy, Alexandre Boulch, Renaud Marlet, and Vincent Lepetit. UNIT: Unsupervised online instance segmentation through time. In *3DV*, 2025. 3, 7

[58] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 2

[59] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. EfficientLPS: Efficient LiDAR panoptic segmentation. *IEEE Transactions on Robotics*, 38 (3):1894–1914, 2021. 2, 16, 18

[60] Gianni Spear. Algorithm to find the minimum-area-rectangle for given points in order to compute the major and minor axis length. https://stackoverflow.com/questions/13542855/algorithm-to-find-the-minimum-area-rectangle-for-given-points-in-order-to-comput, 2012. 12

[61] Shihao Su, Jianyun Xu, Huanyu Wang, Zhenwei Miao, Xin Zhan, Dayang Hao, and Xi Li. PUPS: Point cloud unified panoptic segmentation. In *AAAI*, 2023. 5, 6, 16, 18

[62] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020. 2

[63] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 6

[64] thebestbikelock.com. One bike average size. https://thebestbikelock.com/wp-content/uploads/2020/01/one-bike-average-size.gif, 2020. 5, 6

[65] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. In *CVPR*, 2019. 2

[66] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in Germany, test in the USA: Making 3D object detectors generalize. In *CVPR*, 2020. 4

[67] Wikipedia. Average human height by country. https://en.wikipedia.org/wiki/Average_human_height_by_country, 2024. 5

[68] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. 2

[69] 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, 6, 7

[70] Guozeng Xian, Changyun Ji, Lin Zhou, Guang Chen, Junping Zhang, Bin Li, Xiangyang Xue, and Jian Pu. Location-guided LiDAR-based panoptic segmentation for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 8(2): 1473–1483, 2022. 3, 16

[71] Zeqi Xiao, Wenwei Zhang, Tai Wang, Chen Change Loy, Dahua Lin, and Jiangmiao Pang. Position-guided point cloud panoptic segmentation transformer. *International Journal of Computer Vision*, pages 1–16, 2024. 1, 3, 5, 6, 16, 18

[72] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation. In *ECCV*, 2020. 2

[73] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation. In *ICCV*, 2021. 2

[74] Shuangjie Xu, Rui Wan, Maosheng Ye, Xiaoyi Zou, and Tongyi Cao. Sparse cross-scale attention network for efficient LiDAR panoptic segmentation. In *AAAI*, pages 2920–2928, 2022. 3, 18

[75] Yixuan Xu, Hamidreza Fazlali, Yuan Ren, and Bingbing Liu. AOP-Net: All-in-one perception network for LiDAR-based joint 3d object detection and panoptic segmentation. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023. 2

[76] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely embedded convolutional detection. *Sensors*, 18, 2018. 2, 6

[77] Hao Yang, Shuyuan Lin, Runqing Jiang, Yang Lu, and Hanzi Wang. Dqformer: Dynamic query transformer for lane detection. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. 1, 3, 5, 6, 16, 18

[78] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for LiDAR perception. In *AAAI*, 2023. 2, 3, 6, 18

[79] Yurong You, Katie Z Luo, Cheng Perng Phoo, Wei-Lun Chao, Wen Sun, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Learning to detect mobile objects from lidar scans without labels. In *CVPR*, 2022. 5, 12

[80] Dimitris Zermas, Izzat Izzat, and Nikolaos Papanikolopoulos. Fast segmentation of 3d point clouds: A paradigm on LiDAR data for autonomous vehicle applications. In *ICRA*, 2017. 1, 3

[81] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online LiDAR point clouds semantic segmentation. In *CVPR*, 2020. 2

[82] Zhiwei Zhang, Zhizhong Zhang, Qian Yu, Ran Yi, Yuan Xie, and Lizhuang Ma. LiDAR-camera panoptic segmentation via geometry-consistent and semantic-aware alignment. In *ICCV*, 2023. 3, 5, 6, 16, 18

[83] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point Transformer. In *ICCV*, 2021. 2

[84] Yiming Zhao, Xiao Zhang, and Xinming Huang. A technical survey and evaluation of traditional point cloud clustering methods for LiDAR panoptic segmentation. In *ICCV*, 2021. 2, 3

[85] Yiming Zhao, Xiao Zhang, and Xinming Huang. A divide-and-merge point cloud clustering algorithm for LiDAR panoptic segmentation. In *ICRA*, 2022. 1, 2, 3, 5, 7, 16, 17

[86] Yuanxin Zhong, Minghan Zhu, and Huei Peng. VIN: Voxel-based implicit network for joint 3d object detection and segmentation for lidars. In *BMVC*, 2021. 18

[87] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free LiDAR point cloud panoptic segmentation. In *CVPR*, 2021. 3, 5, 6, 16, 18

[88] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for LiDAR segmentation. In *CVPR*, 2021. 2