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

Supplementary Material

A. Margin of improvement with oracle analysis

A.1. Discussion

Results in Sec. 4.5 motivate our conclusion that instance labels are mostly unnecessary or not correctly utilized for outdoor LiDAR panoptic segmentation.

On the other hand, we observe that the only methods competing with ALPINE on SemanticKITTI and nuScenes are end-to-end, query-based methods. This might hint that there is more to benefit from and end-to-end panoptic training than those two-staged oracles would suggest, for which we show in Tab. 8 that there is little gain left to expect with better instance predictions. Indeed end-to-end methods, and in particular query-based ones, benefit from the simultaneous instance and semantic segmentation tasks training. However, based on the tables of the main paper, the gain remains currently small, if positive, compared to our instance-annotation-free approach. This could hint at an existing margin for improvement for end-to-end methods.

B. Metrics

In LiDAR point cloud segmentation, two primary metrics are used for evaluation: mean Intersection over Union (mIoU) and Panoptic Quality (PQ), each serving different segmentation goals.

Mean Intersection over Union (mIoU). The mIoU is widely used in semantic segmentation to measure the overlap between predicted and ground truth masks for each class. For a given class c, the IoU_c is defined as

$$\text{IoU}_c = \frac{|O_c \cap G_c|}{|O_c \cup G_c|},$$

where O_c and G_c represent predicted and ground truth masks for class c, respectively. The mIoU score is then averaged across all classes C:

$$\text{mIoU} = \frac{1}{|C|} \sum_{c \in C} \text{IoU}_c.$$

The mIoU does not differentiate between instances of the same class.

Panoptic Quality (PQ). The PQ combines both semantic and instance segmentation and is computed as

$$\text{PQ}_c = \underbrace{\frac{\sum_{(p,g)\in \text{TP}_c} \text{IoU}(p,g)}{|\text{TP}_c|}}_{\text{Segmentation Quality (SQ)}} \underbrace{\frac{|\text{TP}_c|}{|\text{TP}_c| + \frac{1}{2}|\text{FP}_c| + \frac{1}{2}|\text{FN}_c|}}_{\text{Recognition Quality (RQ)}},$$

where TP_c, FP_c and FN_c are respectively the sets of true positives, false positives and false negatives computed after matching the predicted and ground truth instances in class c. The PQ score is then averaged across all classes:

$$\text{PQ} = \frac{1}{|C|} \sum_{c \in C} \text{PQ}_c.$$

Finally, $\text{PQ}^\dagger$ satisfies

$$\text{PQ}^\dagger = \frac{1}{|\text{things}| + |\text{stuff}|} \left(\sum_{c \in \text{things}} \text{PQ}_c + \sum_{c \in \text{stuff}} \text{IoU}_c \right).$$

C. Box fitting algorithm

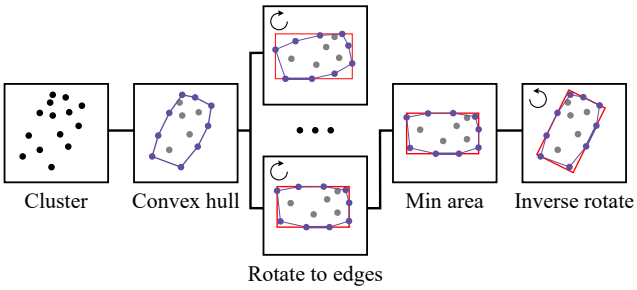

Figure 5. Visual description of the box fitting algorithm

The box fitting algorithm was borrowed from [79] which was inspired from [60]. It is described in Alg. 2 and Fig. 5. It works as follows. First, we compute the convex hull of the set of points on which we need to fit a box. Second, for each edge of the convex hull, we compute its angle with respect to the x-axis, we rotate the point cloud to align this edge with the x-axis, and then fit the axis-aligned bounding box of minimum area that cover the rotated point cloud. Finally, among all bounding boxes computed in the previous step (one for each edge of the convex hull), we keep the bounding box with the smallest area.

D. Number of neighbors k.

We study the variation of the performance of ALPINE with the choice of the number of neighbors considered in the clustering methods k. The results are presented in Fig. 6. As mentioned in Sec. 3.1, the choice of k is not critical above a certain value as edges to distant neighbors will be removed by the thresholding, which we verify experimentally by finding almost no difference in PQ between $k = 32$

Algorithm 2: Box fitting algorithm. This algorithm finds the best fitting box B given a set of $2D$ points P. **convex_hull** returns the vertices and edges of the convex hull of P, **angle** gives the angle of an edge with respect to the x axis, and **rotate** applies a rotation of a given angle.

```
function fit_box(P)
    input: Points P
    output: Best fitting bounding box
    V, E ← convex_hull(P)
    B, S ← [], []
    for e ∈ E do
        θ ← angle(e)
        P' ← rotate(P, θ)
        b^x_min ← min(P', axis=x)
        b^x_max ← max(P', axis=x)
        b^y_min ← min(P', axis=y)
        b^y_max ← max(P', axis=y)
        b ← (b^x_min, b^y_min, b^x_max, b^y_max)
        S.append((b[2] − b[0]) ∗ (b[3] − b[1]))
        B.append(rotate(b, −θ))
    i ← argmin(S)
    return B[i]
```

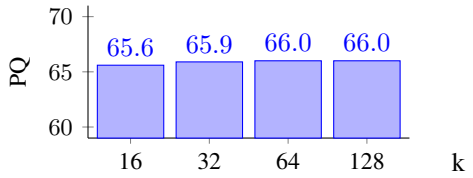

Figure 6. **Influence of the neighbors count** k evaluated on SemanticKITTI's validation set

and 64. We thus keep $k = 32$ for faster runtimes at a very little cost in terms of metrics.

E. Choice of margins in the box splitting

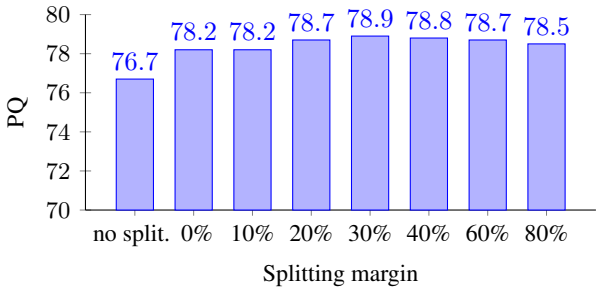

Figure 7. **Influence of the box splitting margin** evaluated on SemanticKITTI's validation set

We present in Fig. 7 an ablation on the choice of the margin parameter in the box splitting algorithm. We recall that the box splitting algorithm splits clusters when they don't fit in the average box of the object class, with some tolerance (margin). The margin is expressed in % of the box size, applied in both dimensions.

First, we notice that the absence of margin hinders performance because of over-segmentation: no cluster bigger than the average box size can exist. Second, we remark that the results are stable for wide range of margins: only 0.2 points of absolute variation in PQ for a margin between 20% and 60%. Therefore, the margin does not need to be heavily finetuned.

The box splitting could reach a very high number of recursions, e.g., in the pathological case where points are to be removed one by one, or a high number of iterations in the dichotomy if the border between two objects is very sensitive. For this reason, we actually put a limit in how small dt can be in Alg. 1. This limit is set to $1e-3$. This accelerates the algorithm by avoiding too long computations for corner cases. We verified in Tab. 11 that this choice does not impact the performance.

ϵ	PQ	PQ†	RQ
0.1	78.1	80.5	86.0
0.01	78.8	81.2	86.8
0.001	78.9	81.3	87.0
0.0001	78.9	81.3	87.0
0.00001	78.9	81.3	87.0

Table 11. **Influence of the box dichotomy precision limit** evaluated on nuScenes' validation set.

F. Reporting and methodology

In the Tabs. 1 to 3 and 15 to 17, we decided to compare only against ALPINE obtained with no instance training in the semantic backbone. We reported for all methods for which we could verify, whether they used TTA or model ensembling. We checked if either the article mentioned that TTA and ensembling were not used, or the code had no mechanism in place to perform either. We put a question mark on those we could not verify.

G. Explainability and limitations

ALPINE's clustering is fully deterministic, and explainable in simple terms: in the absence of box splitting, two points A and B will belong to the same instance if they are predicted to be of the same semantics c, and there exist a path from A to B going only through points predicted to be of class c with distance of at most t_c between each edges. As such, a typical failure case exists when two instances of the same object are closer to each other than the threshold. The box splitting alleviates this issue but failures can still hap-

Method	barrier		bicycle		bus		car		con. veh.		motorcycle		pedestrian		tra. cone	
	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU
ALPINE w. PTv3	61.9	84.4	77.3	54.0	83.0	96.1	93.3	95.2	64.4	49.8	89.9	89.7	93.8	86.6	92.4	75.0
ALPINE w. WI-768	61.0	84.3	77.5	54.8	81.0	94.5	93.2	95.4	61.1	49.4	90.8	90.1	93.6	86.2	92.6	76.2
Sem. Oracle	79.5	100	95.5	100	95.9	100	97.2	100	94.9	100	97.5	100	98.1	100	99.2	100

Method	trailer		truck		driveable		other flat		sidewalk		terrain		manmade		vegetation	
	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU
ALPINE w. PTv3	66.2	78.6	72.8	86.6	57.6	76.2	87.7	89.8	97.0	97.2	63.6	75.4	90.0	91.9	73.1	77.0
ALPINE w. WI-768	64.6	79.4	75.2	87.0	57.3	76.6	85.9	88.8	96.9	97.1	58.8	75.0	89.0	91.1	72.3	76.4
Sem. Oracle	89.1	100	95.0	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 12. Per-class panoptic segmentation results on nuScenes' validation set.

Method	car		bicycle		motorcycle		truck		other-vehicle		person		bicyclist		motorcyclist		road		parking	
	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU
ALPINE w. WI-256	93.0	96.9	68.4	61.9	79.4	84.7	72.2	92.2	61.3	66.4	87.4	83.0	91.7	93.2	00.0	00.7	95.7	95.7	35.6	51.1
ALPINE w. MinkUNet	93.1	98.0	69.5	64.8	81.1	87.0	79.0	93.7	73.6	84.9	87.7	83.3	93.9	94.3	12.6	23.4	94.3	94.7	42.9	53.7
Oracle	97.4	100	94.3	100	95.0	100	97.3	100	97.5	100	98.6	100	99.5	100	99.9	100	100	100	100	100

Method	sidewalk		other-ground		building		fence		vegetation		trunk		terrain		pole		traffic-sign	
	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU	PQ	IoU
ALPINE w. WI-256	81.1	84.4	00.7	06.3	89.7	92.6	29.3	70.0	87.0	88.3	61.8	74.7	59.5	73.7	63.8	67.3	60.2	52.6
ALPINE w. MinkUNet	80.4	83.0	00.0	00.1	89.9	92.3	26.7	68.4	88.1	88.8	55.5	70.1	61.3	75.3	61.8	66.3	59.0	49.9
Oracle	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 13. Per-class panoptic segmentation results on SemanticKITTI' validation set.

pen, e.g., if the distance between points within each instance is greater than the distance between the two instances. This issue vanishes as the angular resolution of the LiDAR is improved, which is experimentally verified by the higher PQ score achieved by a semantic oracle on the dense SemanticKITTI than on the sparser nuScenes.

Those weaknesses are a consequences of the simplicity and explainability of the method, and especially in the choice to use euclidean distance as the criterion of discrimination. ALPINE is a baseline, demonstrating what can be achieved by a training-free method, and reaches state-of-the-art results.

H. Per-class results

For completeness, we present in Tabs. 12 and 13 the per-class results of our method on both nuScenes and SemanticKITTI.

I. Performance of ALPINE and distance to the sensor

We report in Tab. 14 a breakdown of ALPINE's performance with respect to the distance to the sensor. ALPINE improves over P3Former at short range, but slightly deteriorates at long range. This suggests that ALPINE is particularly efficient on dense point clouds (i.e. close to the sensor)

Method	0-15m			15-30m			30m+		
	PQ	PQ†	mIoU	PQ	PQ†	mIoU	PQ	PQ†	mIoU
ALPINE w/ Ens.	68.5	71.0	73.0	68.6	70.7	68.4	65.8	65.9	54.4
P3Former	63.6	65.8	67.8	64.8	66.9	63.4	59.6	60.1	47.2
P3Former & ALPINE	64.0	66.2	67.8	64.8	66.9	63.4	59.4	59.8	47.2

Table 14. **Performance w.r.t sensor distance** on SemanticKITTI's validation set.

but gets lesser performance on sparser point clouds.

J. More visualisations

More visualization of our method can be seen on Fig. 8. We can see that D&M has some boundaries issues, which ALPINE does not have. Furthermore, on those examples, the remaining false predictions are mostly semantic prediction errors from the MinkUNet, rather than errors in our clustering.

Method		Inst. labels	TTA	Ens.	PQ	PQ†	RQ	SQ	PQ_{Th}	RQ_{Th}	SQ_{Th}	PQ_{St}	RQ_{St}	SQ_{St}	mIoU
LPSAD (impl. from [52])	[40]	✓	?	?	37.4	44.2	47.8	66.9	25.3	32.4	65.2	46.2	58.9	68.2	49.4
PanopticTrackNet	[22]	✓	?	?	40.0	-	48.3	73.0	29.9	33.6	76.8	47.4	59.1	70.3	53.8
PointGroup	[23]	✓	✗	✗	46.1	54.0	56.6	74.6	47.7	55.9	73.8	45.0	57.1	75.1	55.7
TORNADO-Net (fusion)	[17]	✓	?	?	50.6	55.9	62.1	74.9	48.1	57.5	72.5	52.4	65.4	76.7	59.2
Panoster	[16]	✓	✗	✗	55.6	-	66.8	79.9	56.6	65.8	-	-	-	-	61.1
LCPS (lidar)	[82]	✓	✗	✗	55.7	65.2	65.8	74.0	-	-	-	-	-	-	61.1
Cylinder3D & D&M	[85]	✗	✗	✗	57.2	-	-	-	-	-	-	-	-	-	-
Location-Guided	[70]	✓	?	?	59.0	63.1	69.4	78.7	65.3	73.5	88.5	53.9	66.4	71.6	61.4
Panoptic-PolarNet	[87]	✓	✗	✗	59.1	64.1	70.2	78.3	65.7	74.7	87.4	54.3	66.9	71.6	64.5
EfficientLPS	[59]	✓	✗	✗	59.2	65.1	69.8	75.0	58.0	68.2	78.0	60.9	71.0	72.8	64.9
MaskPLS	[36]	✓	✗	✗	59.8	-	69.0	76.3	-	-	-	-	-	-	61.9
DS-Net (SPVCNN)	[21]	✓	?	?	61.4	65.2	72.7	79.0	65.2	72.3	79.3	57.9	71.1	79.3	69.6
Panoptic-PHNet	[30]	✓	✗	✗	61.7	-	-	-	69.3	-	-	-	-	-	65.7
PANet	[37]	✓	?	?	61.7	66.6	71.8	79.3	-	-	-	-	-	-	68.1
D&M w. MinkUnet	[85]	✗	(✓)	✗	61.8	66.2	72.8	79.6	64.5	73.7	87.2	59.9	72.1	74.2	71.4
CenterLPS	[38]	✓	?	?	62.1	67.0	72.0	80.7	68.4	75.2	91.0	57.5	69.7	73.2	68.1
ALPINE w. PTv3		✗	(✓)	✗	62.4	66.2	72.0	76.7	66.1	72.1	80.0	59.8	71.9	74.3	67.3
P3Former	[71]	✓	✗	✗	62.6	66.2	72.4	76.2	69.4	-	-	57.7	-	-	-
CFNet	[31]	✓	✗	✗	62.7	67.5	-	-	70.0	-	-	57.3	-	-	67.4
LPST	[2]	✓	✗	✗	63.1	70.8	73.1	79.2	68.7	75.7	86.7	58.9	71.2	73.7	69.7
GP-S3Net	[52]	✓	?	?	63.3	71.5	75.9	81.4	70.2	80.1	86.2	58.3	72.9	77.9	73.0
DQFormer	[77]	✓	✗	✗	63.5	67.2	73.1	81.7	-	-	-	-	-	-	-
ALPINE w. MinkUNet		✗	✗	✗	64.2	68.9	74.1	84.4	72.6	79.3	90.6	58.1	70.4	79.8	70.7
ALPINE w. WI-256		✗*	(✓)	✗	64.2	69.0	74.1	79.7	69.3	75.9	79.8	60.4	72.7	79.7	70.3
PUPS (w/o ens.)	[61]	✓	✗	✗	64.4	68.6	74.1	81.5	73.0	79.3	92.6	58.1	70.4	73.5	-
IEQLPS	[19]	✓	?	?	64.7	68.1	74.7	81.3	73.5	79.1	93.0	58.3	71.4	72.8	-
ALPINE w. MinkUNet		✗	(✓)	✗	65.9	70.2	75.5	81.4	73.9	80.1	91.2	60.0	72.2	74.2	72.2
PUPS	[61]	✓	✓	✓	66.3	70.2	75.6	82.5	74.6	80.3	93.4	60.2	72.2	74.5	-
ALPINE w. MinkUNet & PTv3		✗	(✓)	(✓)	66.6	70.8	76.1	82.6	73.4	79.5	93.2	61.6	73.6	74.9	72.0

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

Table 15. 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.

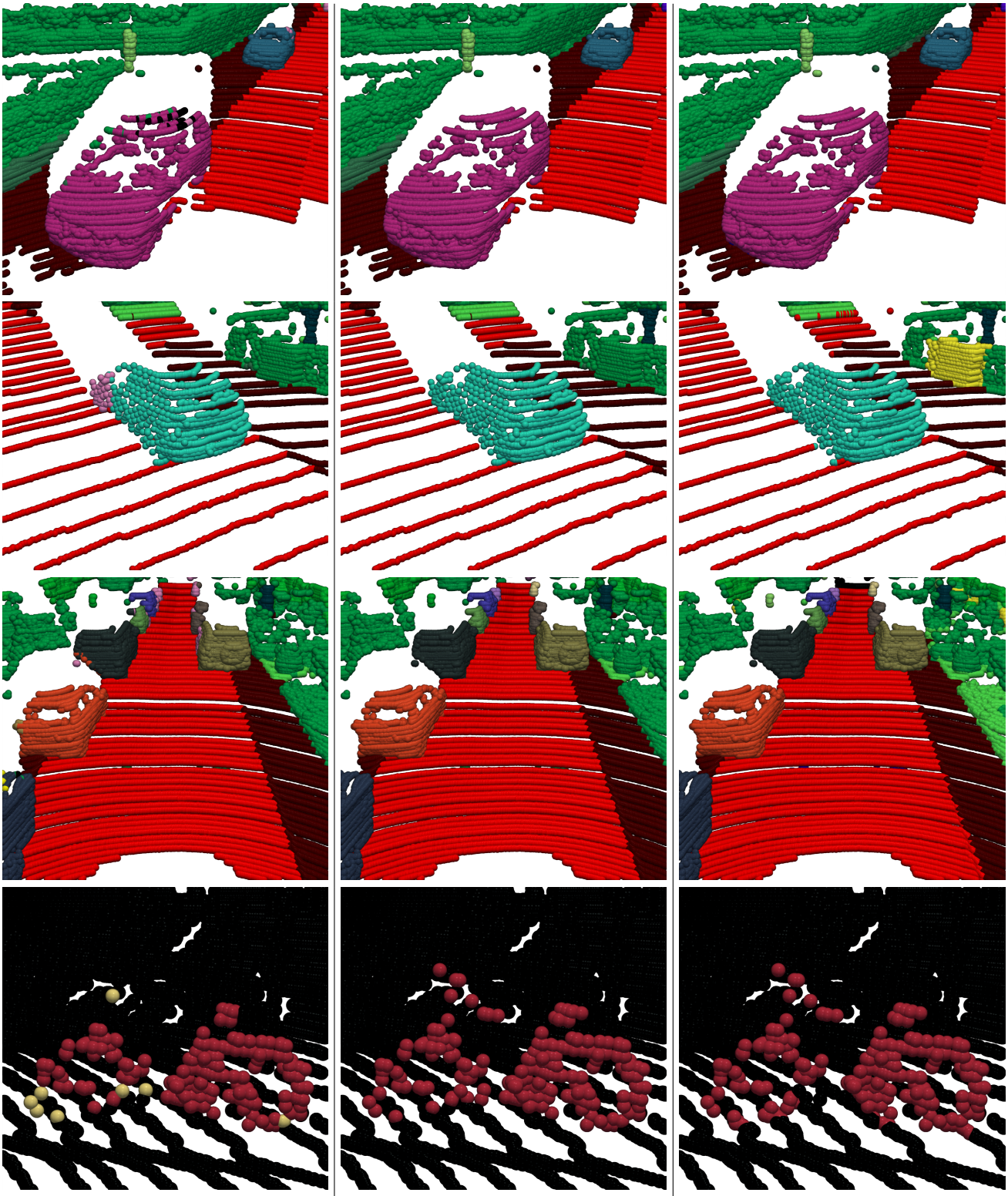

Figure 8. **Examples of panoptic predictions on SemanticKITTI.** We present the results obtained with D&M [85] (left) and ALPINE (middle). Both methods use the same MinkUNet to obtain pointwise semantic predictions. The Ground Truth masks are presented on the rightmost panels. On the last row, only points predicted as "bicycle" are colored for visual clarity.

Method		Inst. labels	TTA	Ens.	PQ	PQ†	RQ	SQ	PQ_{Th}	RQ_{Th}	SQ_{Th}	PQ_{St}	RQ_{St}	SQ_{St}	mIoU
LPSAD (impl. from [52])	[40]	✓	?	?	50.4	57.7	62.4	79.4	43.2	53.2	80.2	57.5	71.7	78.5	62.5
PanopticTrackNet	[22]	✓	?	?	51.4	56.2	63.3	80.2	45.8	55.9	81.4	60.4	75.5	78.3	58.0
VIN	[86]	✓	✗	?	51.7	57.4	61.8	82.6	45.7	53.7	83.6	61.8	75.4	80.9	73.7
TORNADO-Net (fusion)	[17]	✓	?	?	54.0	59.8	65.4	80.9	44.1	53.9	80.1	63.9	76.9	81.8	68.0
MaskPLS	[36]	✓	✗	✗	57.7	60.2	66.0	71.8	64.4	73.3	84.8	52.2	60.7	62.4	62.5
GP-S3Net	[52]	✓	?	?	61.0	67.5	72.0	84.1	56.0	65.2	85.3	66.0	78.7	82.9	75.8
EfficientLPS	[59]	✓	✗	✗	62.0	65.6	73.9	83.4	56.8	68.0	83.2	70.6	83.6	83.8	65.6
DS-Net (SPVCNN)	[21]	✓	?	?	64.7	67.6	76.1	83.5	58.6	64.2	82.8	74.7	86.5	85.5	76.3
PVCL	[32]	✓	?	?	64.9	67.8	77.9	81.6	59.2	72.5	79.7	67.6	79.1	77.3	73.9
SCAN	[74]	✓	?	?	65.1	68.9	75.3	85.7	60.6	70.2	85.7	72.5	83.8	85.7	77.4
Panoptic-PolarNet	[87]	✓	✗	✗	67.7	71.0	78.1	86.0	65.2	74.0	87.2	71.9	84.9	83.9	69.3
SMAC-Seg	[27]	✓	?	?	68.4	73.4	79.7	85.2	68.0	77.2	87.3	68.8	82.1	83.0	71.2
PANet	[37]	✓	✗	✗	69.2	72.9	80.7	85.0	69.5	79.3	86.7	68.7	82.9	82.1	72.6
CPSeg	[28]	✓	?	?	71.1	75.6	82.5	85.5	71.5	81.3	87.3	70.6	83.7	83.6	73.2
LCPS (lidar)	[82]	✓	✗	✗	72.9	77.6	82.0	88.4	72.8	80.5	90.1	73.0	84.5	85.5	75.1
PUPS	[61]	✓	?	?	74.7	77.3	83.3	89.4	75.4	81.9	91.8	73.6	85.6	85.3	-
Panoptic-PHNet	[30]	✓	✗	✗	74.7	77.7	84.2	88.2	74.0	82.5	89.0	75.9	86.9	86.8	79.7
CFNet	[31]	✓	✗	✗	75.1	78.0	84.6	88.8	74.8	82.9	89.8	76.6	87.3	87.1	79.3
P3Former	[71]	✓	✗	✗	75.9	78.9	84.7	89.7	76.9	83.3	92.0	75.4	87.1	86.0	-
CenterLPS	[38]	✓	?	?	76.4	79.2	86.2	88.0	77.5	87.1	88.4	74.6	84.9	87.3	77.1
ALPINE w. WI-768		✗	✗	✗	76.9	79.9	85.7	89.3	77.9	85.3	90.9	75.3	86.3	86.7	80.3
IEQLPS	[19]	✓	?	?	77.1	79.1	86.5	88.2	79.5	86.6	91.7	73.0	86.4	83.9	-
LPST	[2]	✓	✗	✗	77.1	79.9	86.5	88.6	79.3	87.5	90.3	73.6	84.9	85.7	80.3
DQFormer	[77]	✓	✗	✗	77.7	79.5	86.8	89.2	77.8	86.7	89.5	77.5	87.0	88.6	-
ALPINE w. WI-768		✗	(✓)	✗	77.9	80.7	86.5	89.6	78.6	86.0	91.0	76.7	87.3	87.3	81.4
ALPINE w. PTv3		✗	(✓)	✗	78.9	81.3	87.0	90.4	79.3	86.2	91.7	78.2	88.3	88.1	81.5
ALPINE w. WI-768 & PTV3		✗	(✓)	(✓)	79.5	81.9	87.6	90.5	80.1	87.1	91.7	78.6	88.5	88.3	82.7
LidarMultiNet	[78]	✓	✓	✓	81.8	-	89.7	90.8	-	-	-	-	-	-	83.6

Table 16. 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.

Method		Inst. labels	TTA	Ens.	PQ	PQ†	RQ	SQ	PQ_{Th}	RQ_{Th}	SQ_{Th}	PQ_{St}	RQ_{St}	SQ_{St}	mIoU
LPSAD (impl. from [52])	[40]	✓	?	?	22.5	32.7	34.0	53.5	18.7	25.7	70.5	24.0	37.1	47.1	35.5
TORNADO-Net (fusion)	[17]	✓	?	?	33.7	43.3	46.0	68.4	41.2	49.6	83.1	30.9	44.7	62.9	44.5
DS-Net	[20]	✓	?	?	35.6	45.9	49.2	68.6	27.4	33.8	76.8	38.7	55.0	65.5	54.5
GP-S3Net	[52]	✓	?	?	48.7	60.3	63.7	61.3	61.6	71.7	86.4	43.8	60.8	51.8	61.8
ALPINE w. PTv3		✗	✗	✗	51.4	57.7	67.2	74.3	64.8	74.1	87.2	47.4	65.1	70.5	58.3

Table 17. Panoptic segmentation results on Sequence 02 of SemanticPOSS as validation set.