Supplementary Material:
TASeg: Temporal Aggregation Network for LiDAR Semantic Segmentation

Here we provide more experimental results, discussions and details to validate the effectiveness of our method. An overview of this supplementary material is as follows:

A. Additional Ablations

A.1. Performance on Different Distances

Here we investigate the performance of our approach on different distances, as summarized in Table 1. Results show that our TASeg consistently outperforms the strong baseline on different distances, demonstrating the effectiveness of our approach. The table also suggests that TASeg achieves more improvement on distant areas, which confirms the importance of temporal and multi-modal information.

Distance(m)	$0 \sim 10$	$10 \sim 20$	$20 \sim 30$	$30 \sim 40$	$40 \sim 50$	$50 \sim \infty$
Baseline	69.6	68.1	64.0	52.8	40.5	20.1
TASeg	73.9	70.4	67.0	58.8	47.5	24.6
Δ	**+4.3**	**+2.3**	**+3.0**	**+6.0**	**+7.0**	**+4.5**

Table 1. Comparison between the baseline and TASeg on different distances on SemanticKITTI *val* set.

A.2. Class-wise IoU Scores

To verify the performance of our TASeg on different classes, we provide class-wise IoU scores comparison, as shown in Table 3. The table shows that our approach can achieve consistent improvement on different classes. In particular, TASeg achieves the most improvement on small or difficult classes, such as bicycles, motorcycles, parking and fences. This can be credited to the sufficient temporal information aggregated by our TASeg.

A.3. Ablation on Mask Distillation

In Table 2, we provide different distillation strategies for Mask Distillation. *Feature Distill* and *Logits Distill* represent that we conduct distillation on feature maps and logit maps, respectively. Results show that distilling on feature maps can achieve higher performance than distilling on logit maps. Distilling on both feature and logit maps can not bring more improvement. Hence, our Mask Distillation only utilizes feature map distillation.

Distillation Strategy	mIoU
-	71.2
Feature Distill	71.8
Logit Distill	71.6
Feature Distill & Logit Distill	71.7

Table 2. Comparison of different distillation strategies.

A.4. Ablation on Feature Gathering Strategy

In our TIAF, we need to gather voxel-wise image features with temporal LiDAR points for temporal multi-modal fusion. Table 4 provides an ablation on different feature-gathering strategies. For *Hard Indexing*, we only gather the image features whose discrete coordinates are exactly the same as temporal LiDAR points. For *KNN*, we select nearest-27 voxel-wise image features around a temporal LiDAR point and feed them to a PointNet to extract image features for the temporal LiDAR point. For *Trilinear Interpolation*, we choose the neighbors whose discrete coordinates are in a Manhattan distance of 2 with the temporal LiDAR point. Then, we utilize trilinear interpolation to aggregate image features for the LiDAR point. Experiments suggest that *Trilinear Interpolation* can bring more improvement than other approaches.

Method	mIoU	car	bicy	moto	truck	o.veh	ped	b.cyc	m.cyc	road	park	walk	o.gro	build	fence	veg	trunk	terr	pole	sign
MinkUNet (baseline)	68.9	97.8	52.4	81.4	89.7	83.8	78.1	90.6	0.0	94.2	55.8	81.7	0.2	91.1	61.9	89.0	69.7	76.5	65.6	50.9
TASeg (Ours)	72.7	98.0	63.1	90.1	95.2	91.6	82.7	92.6	0.1	94.8	64.3	83.3	0.2	93.0	71.7	89.4	72.2	77.5	67.8	54.3
improvement	**3.8**	**0.2**	**10.7**	**8.7**	**5.5**	**7.8**	**4.6**	**2.0**	**0.1**	**0.6**	**8.5**	**1.6**	**0.0**	**1.9**	**9.8**	**0.4**	**2.5**	**1.0**	**2.2**	**3.4**

Table 3. Class-wise IoU scores of our TASeg and MinkUNet (baseline) on *SemanticKITTI* val set.

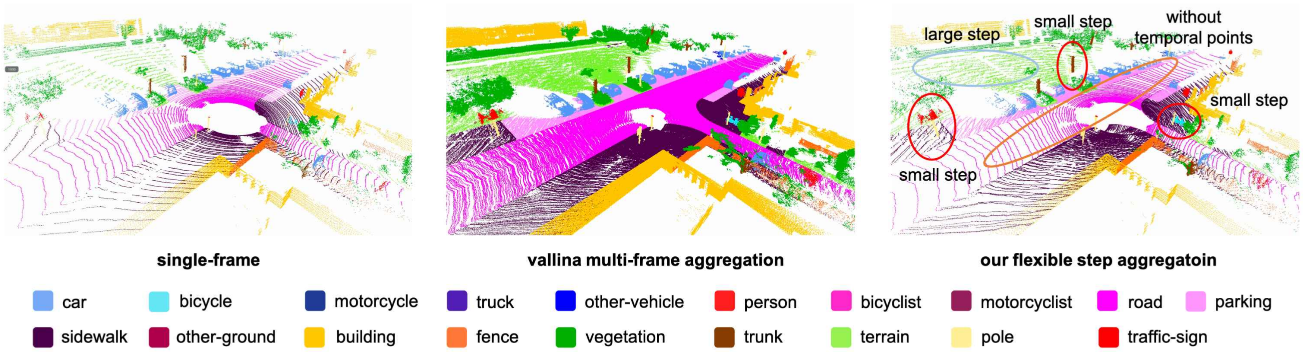


Figure 1. Visualization of single-frame, vanilla multi-frame aggregation (directly concatenating) and our flexible step aggregation. Our approach utilizes different steps to aggregate temporal point clouds for different classes.

Feature Gathering Strategy	mIoU
Hard Indexing	72.4
KNN	72.5
Trilinear Interpolation	72.7

Table 4. Comparison of different feature gathering strategies.

A.5. Ablation on Fusion Strategy

After gathering temporal image features, we need to fuse the gathered temporal image features with temporal LiDAR point features. In Table 5, we investigate several different fusion strategies, *i.e.*, *MaxPool*, *AvgPool*, *Add* and *Concatenation*. Results show that our TASeg is robust to different fusion strategies. Considering that *Add* and *Concatenation* can achieve slightly higher performance, each of them can be used for the final temporal multi-modal fusion.

Fusion Strategy	mIoU
MaxPool	72.6
AvgPool	72.6
Add	72.7
Concatenation	72.7

Table 5. Comparison of different fusion strategies.

A.6. Ablation on SMSA

We provide an ablation study to verify the efficacy of SMSA when the image input is removed, as shown in Table 6. The results show that our method can surpass other multi-scan methods even without images.

Method	SVQNet	2DPASS	TASeg wo/image	TASeg w/image
mIoU	60.5	62.4	64.6	65.7

Table 6. Ablation on *SemanticKITTI multi-scan* test set.

B. Additional Discussions

B.1. Visualization for FSA

Our Flexible Step Aggregation (FSA) decomposes temporal point clouds into several class groups and assigns a specific step for the temporal aggregation of each group. As shown in Figure 1, for easy classes, such as terrains and roads, we assign them a large aggregation step; for difficult classes, such as traffic signs and bicycles, we assign them a small step. This approach can save much memory and computation from easy classes while providing sufficient temporal points for difficult classes.

B.2. Visualization for TIAF

In Figure 2, we provide the visualization of the aggregated point-wise temporal images by TIAF. Comparing the left and right of Figure 2, we can find that the overlap region between the LiDAR and camera is very limited when only using the present image. However, after aggregating temporal images, the colorized area is enlarged greatly, and it can cover most LiDAR point clouds, as shown in the center of Figure 2. This confirms the rationality of our idea of leveraging temporal images to expand the camera FOV and complement present image features.

B.3. Imbalance of Static and Moving Samples

Our Static-Moving Switch Augmentation (SMSA) can alleviate the imbalance between static and moving classes.

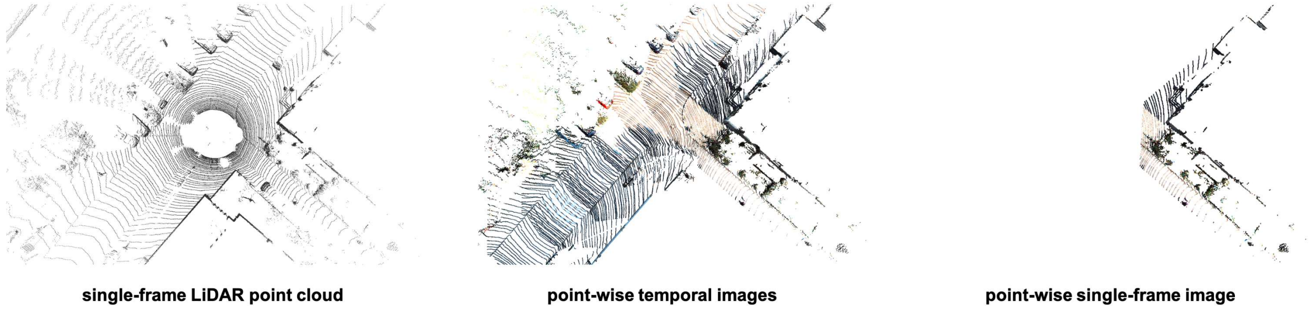

single-frame LiDAR point cloud
point-wise temporal images
point-wise single-frame image

Figure 2. Visualization of single-frame LiDAR point cloud, point-wise temporal images and point-wise single-frame image.

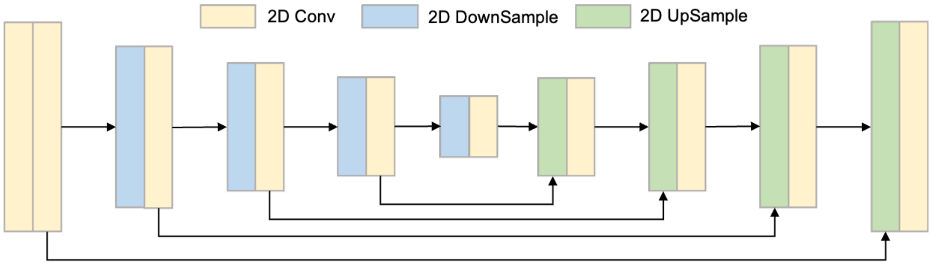


Figure 3. Architecture of 2D backbone in TIAF for pixel-wise image feature extraction.

Class	Car	Truck	Other-vehicle	Person	Bicyclist	Motorcyclist
Static	95302518	4347360	5352883	440239	5	13
Moving	4128968	238730	103005	376574	298599	87766

Table 7. Point number of different movable classes on *SemanticKITTI* training set of the multi-scan benchmark.

Table 7 provides detailed statistics of static and moving classes. It shows that there are few training samples for static bicyclists and motorcyclists, making it difficult to produce accurate predictions for the two classes. For moving trucks and moving other-vehicles, they are more than ten times fewer than the static counterparts, *i.e.*, static trucks and static other-vehicles. The class imbalance issue significantly limits the multi-scan perception ability of the model. Our SMSA enables movable objects to switch their motion states freely. Thus we can switch the motion states of objects in classes holding many samples, such as static trucks and moving bicyclists, to augment the classes with opposite motion states, *i.e.*, moving trucks and static bicyclists.

C. Additional Details

C.1. Static-to-Moving in SMSA

Because static objects often park on the side of the road, which is crowded, when we change their motion states, the shifted temporal parts of them may overlap with other objects. To alleviate this, we define a set of anchor points (4×4) around the center of a static object, as shown in the left of Figure 4. Then, we define a coverage area $(2m \times 8m)$ for each anchor point, and we choose the anchor point whose coverage area contains the fewest LiDAR points, as shown in the right of Figure 4.

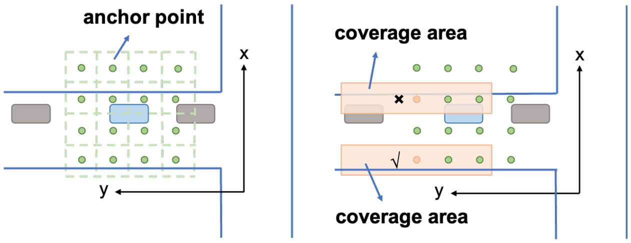


Figure 4. Illustration of Static-to-Moving in SMSA.

C.2. Architecture of Image Branch

Given pixel-wise temporal images, we leverage a 2D image backbone to extract pixel-wise image features. As shown in Figure 3, our 2D backbone is a UNet architecture with four levels of feature map. The 2D backbone is isomorphic to the 3D backbone of the LiDAR branch. The difference is that the 3D backbone utilizes 3D sparse convolutions instead of 2D convolutions. After projecting pixel-wise image features to 3D space, we get point-wise image features. They are first transformed and aggregated to the present coordinate. Then we voxelize them and feed them to a 3D subnetwork that consists of several 3D convolutions, to further fuse them, as depicted in Figure 5.

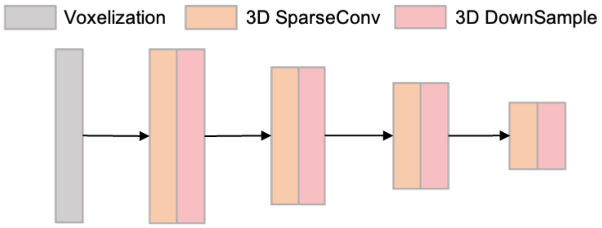


Figure 5. Architecture of 3D subnetwork in TIAF for point-wise image feature fusion.

D. Qualitative Results

We show the qualitative results (error maps) of our TASeg and MinkUNet (the baseline) on different datasets as shown in Figure 6, Figure 7 and Figure 8. To highlight the differences, we paint the correct and incorrect predictions with black and red, respectively. Our method brings visible improvement to the baseline, especially in sparse and distant areas. For better visualization on the multi-scan benchmark, we highlight static and moving objects with red circles. It should be noted that the MinkUNet [3] we use is a high-performance version re-implemented by PCSeg [4]. The notable improvement on the strong baseline further confirms the effectiveness of our approach.

E. Leaderboard Screenshot

We also validate our method by submitting our results to SemanticKITTI [1] and nuScenes [2] test server. As shown in Figure 9, Figure 10 and Figure 11, our TASeg ranks **1st** on leaderboards of three tracks of the two benchmarks, *i.e.*, SemanticKITTI single-scan track, multi-scan track and nuScenes LiDAR semantic segmentation track. The appealing results verify the superiority of our approach over existing LiDAR segmentation algorithms.

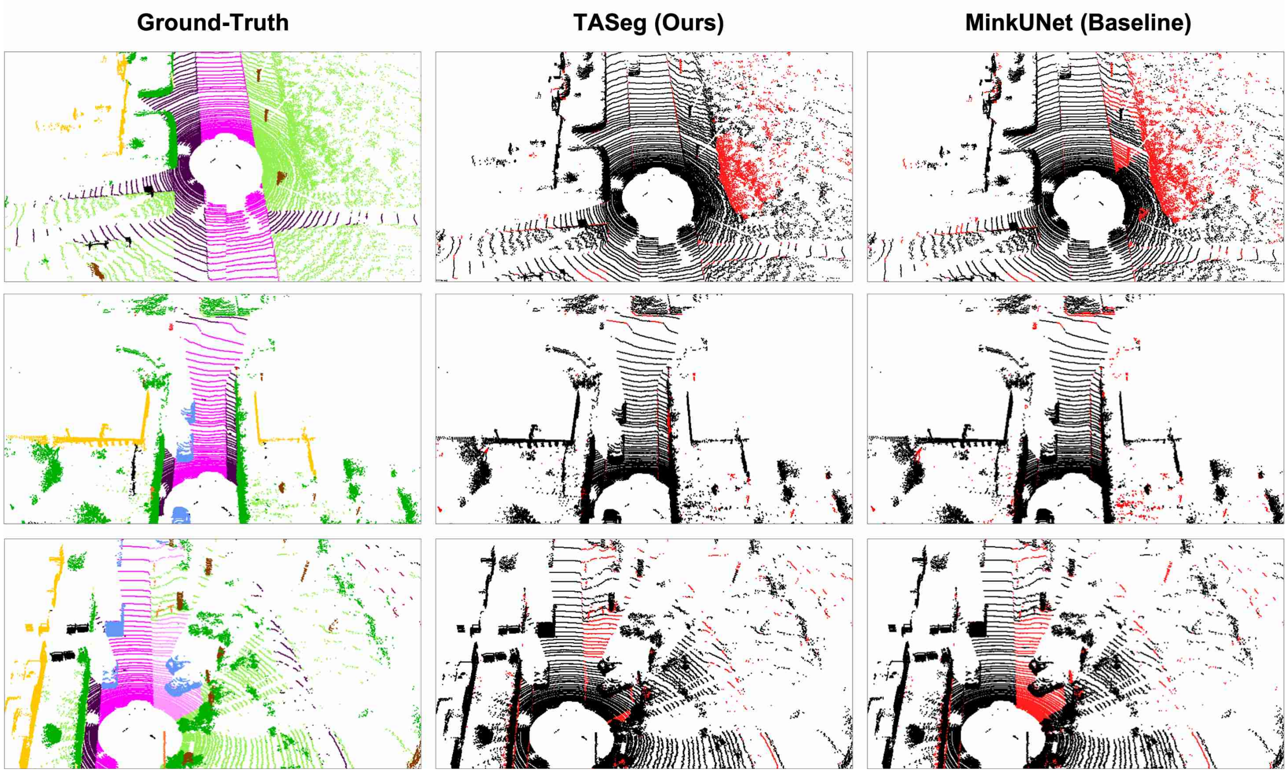


Figure 6. Qualitative results (error maps) of our TASeg and MinkUNet (baseline model) on *SemanticKITTI* single-scan dataset.

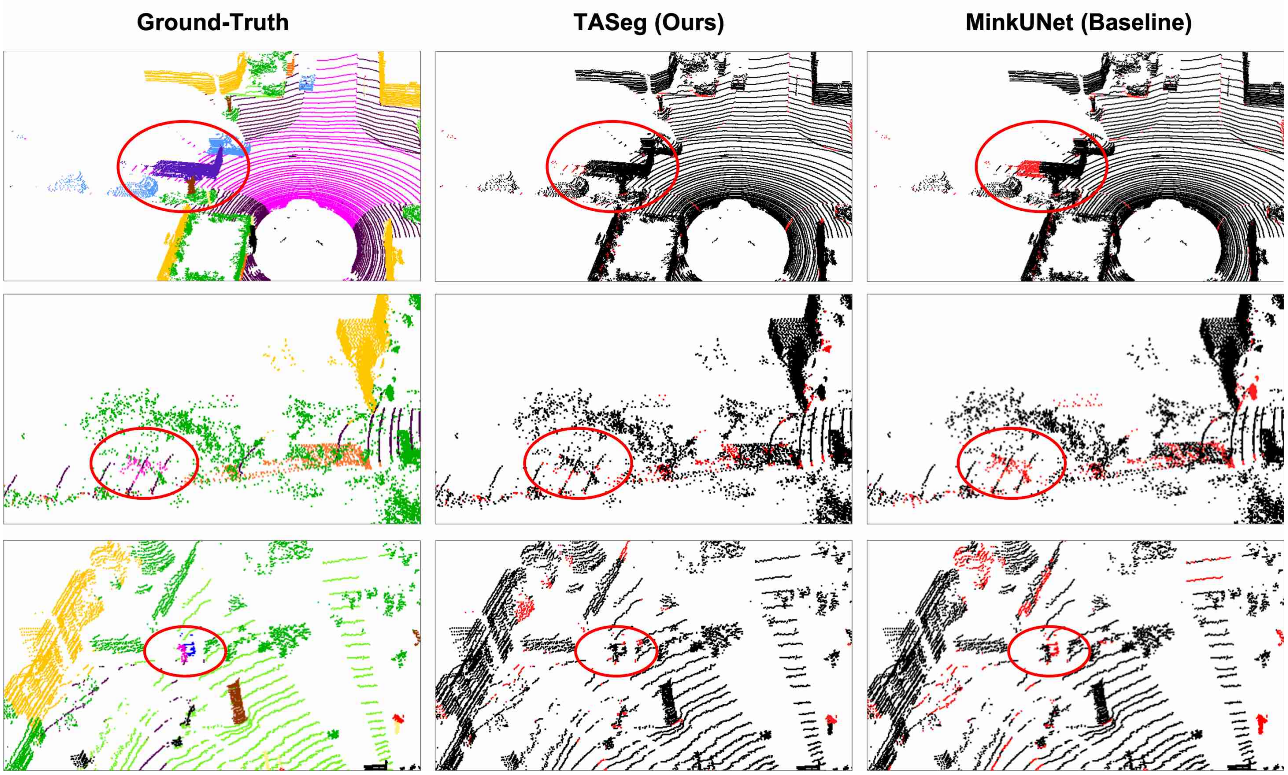


Figure 7. Qualitative results (error maps) of our TASeg and MinkUNet (baseline model) on *SemanticKITTI* multi-scan dataset.

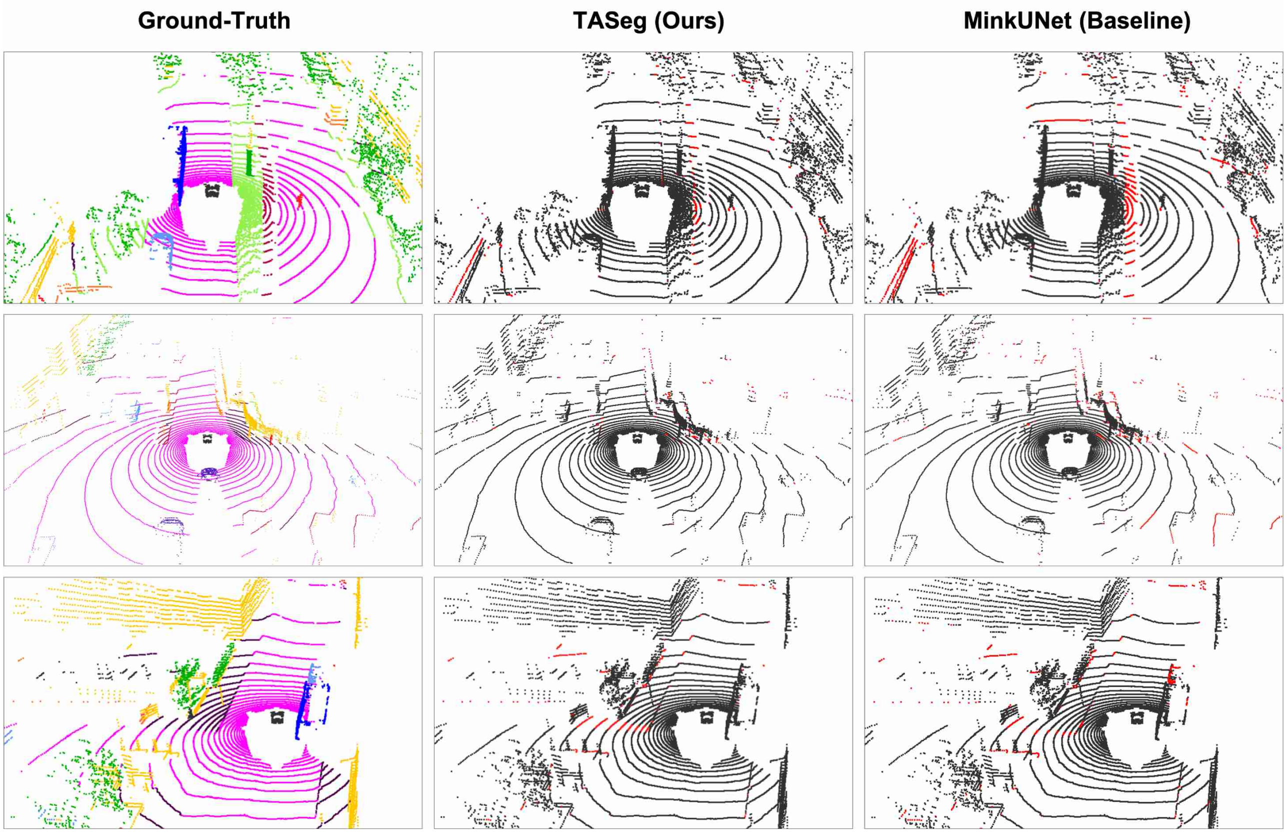
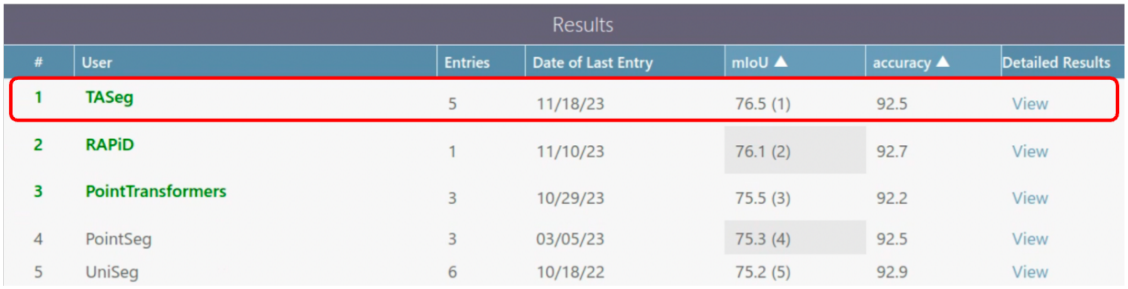


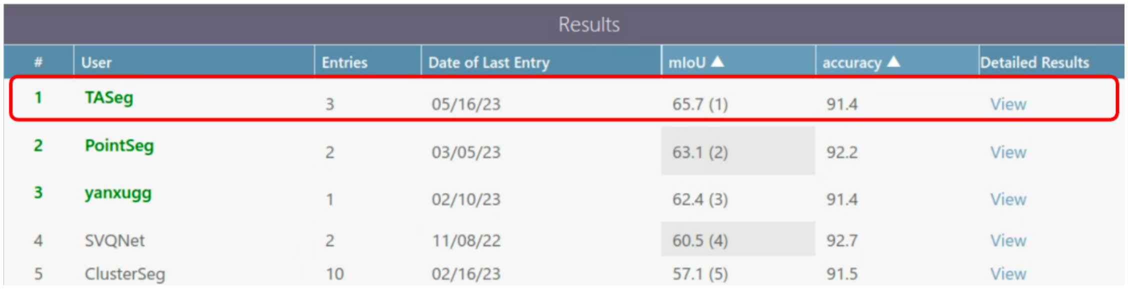
Figure 8. Qualitative results (error maps) of our TASeg and MinkUNet (baseline model) on *nuScenes* val dataset.



Results

#	User	Entries	Date of Last Entry	mIoU ▲	accuracy ▲	Detailed Results
1	TASeg	5	11/18/23	76.5 (1)	92.5	View
2	RAPiD	1	11/10/23	76.1 (2)	92.7	View
3	PointTransformers	3	10/29/23	75.5 (3)	92.2	View
4	PointSeg	3	03/05/23	75.3 (4)	92.5	View
5	UniSeg	6	10/18/22	75.2 (5)	92.9	View

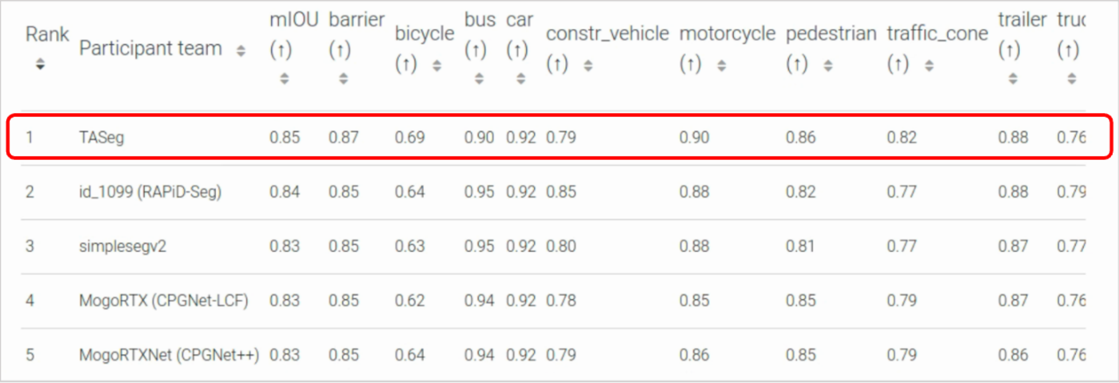
Figure 9. Screenshot of SemanticKITTI single-scan leaderboard on the date of CVPR deadline, *i.e.*, 2023-11-18 07:59 AM UTC.



Results

#	User	Entries	Date of Last Entry	mIoU ▲	accuracy ▲	Detailed Results
1	TASeg	3	05/16/23	65.7 (1)	91.4	View
2	PointSeg	2	03/05/23	63.1 (2)	92.2	View
3	yanxugg	1	02/10/23	62.4 (3)	91.4	View
4	SVQNet	2	11/08/22	60.5 (4)	92.7	View
5	ClusterSeg	10	02/16/23	57.1 (5)	91.5	View

Figure 10. Screenshot of SemanticKITTI multi-scan leaderboard on the date of CVPR deadline, *i.e.*, 2023-11-18 07:59 AM UTC.



Rank ⬍	Participant team ⬍	mIOU (↑) ⬍	barrier (↑) ⬍	bicycle (↑) ⬍	bus (↑) ⬍	car (↑) ⬍	constr_vehicle (↑) ⬍	motorcycle (↑) ⬍	pedestrian (↑) ⬍	traffic_cone (↑) ⬍	trailer (↑) ⬍	truc (↑) ⬍
1	TASeg	0.85	0.87	0.69	0.90	0.92	0.79	0.90	0.86	0.82	0.88	0.76
2	id_1099 (RAPiD-Seg)	0.84	0.85	0.64	0.95	0.92	0.85	0.88	0.82	0.77	0.88	0.79
3	simplesegv2	0.83	0.85	0.63	0.95	0.92	0.80	0.88	0.81	0.77	0.87	0.77
4	MogoRTX (CPGNet-LCF)	0.83	0.85	0.62	0.94	0.92	0.78	0.85	0.85	0.79	0.87	0.76
5	MogoRTXNet (CPGNet++)	0.83	0.85	0.64	0.94	0.92	0.79	0.86	0.85	0.79	0.86	0.76

Figure 11. Screenshot of nuScenes LiDAR segmentation leaderboard on the date of CVPR deadline, *i.e.*, 2023-11-18 07:59 AM UTC.

References

[1] 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 *IEEE International Conference on Computer Vision*, pages 9297–9307, 2019. 4

[2] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 4

[3] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 4

[4] Youquan Liu, Yeqi Bai, Lingdong Kong, Runnan Chen, Yuenan Hou, Botian Shi, and Yikang Li. Pcseg: An open source point cloud segmentation codebase. https://github.com/PJLab-ADG/PCSeg, 2023. 4