

4D-StOP: Panoptic Segmentation of 4D LiDAR using Spatio-temporal Object Proposal Generation and Aggregation

Lars Kreuzberg¹, Idil Esen Zulfikar¹, Sabarinath Mahadevan¹,
Francis Engelmann², and Bastian Leibe¹

¹RWTH Aachen University, Germany ²ETH Zurich, AI Center, Switzerland
lars.kreuzberg@rwth-aachen.de
{zulfikar,mahadevan,leibe}@vision.rwth-aachen.de
francis.engelmann@ai.ethz.ch

Abstract. In this work, we present a new paradigm, called 4D-StOP, to tackle the task of 4D Panoptic LiDAR Segmentation. 4D-StOP first generates spatio-temporal proposals using voting-based center predictions, where each point in the 4D volume votes for a corresponding center. These tracklet proposals are further aggregated using learned geometric features. The tracklet aggregation method effectively generates a video-level 4D scene representation over the entire space-time volume. This is in contrast to existing end-to-end trainable state-of-the-art approaches which use spatio-temporal embeddings that are represented by Gaussian probability distributions. Our voting-based tracklet generation method followed by geometric feature-based aggregation generates significantly improved panoptic LiDAR segmentation quality when compared to modeling the entire 4D volume using Gaussian probability distributions. 4D-StOP achieves a new state-of-the-art when applied to the SemanticKITTI test dataset with a score of 63.9 LSTQ, which is a large (+7%) improvement compared to current best-performing end-to-end trainable methods. The code and pre-trained models are available at: <https://github.com/LarsKreuzberg/4D-StOP>.

1 Introduction

In recent years, we have made impressive progress in the field of 3D perception, which is mainly due to the remarkable developments in the field of deep learning. With the availability of RGB-D sensors, we have made progress on tasks like [16, 17, 27, 40], 3D instance segmentation [11, 14, 15, 22, 26, 54] and 3D object detection [35, 41, 59] which work on indoor scenes. Similarly, the accessibility to modern LiDAR sensors has made it possible to work with outdoor 3D scenes, where again recent works have targeted the tasks of 3D object detection [43, 44, 53, 55], semantic segmentation [33, 47, 57, 61], panoptic segmentation [5, 21, 34, 60] and multi-object tracking [3, 12, 50, 56]. Perceiving outdoor 3D environments from LiDAR data is particularly relevant for robotics and autonomous driving applications, and hence has gained significant traction in the recent past.

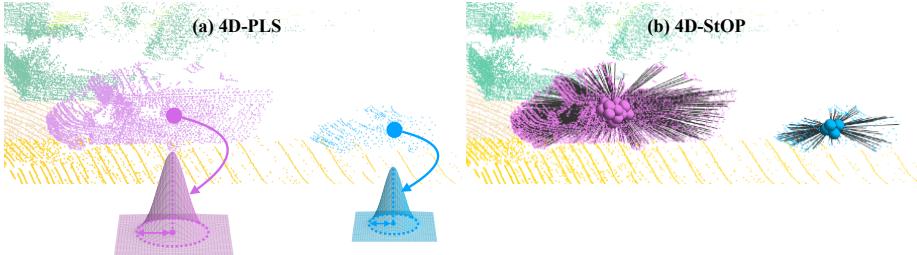


Fig. 1: Comparing 4D-PLS to 4D-StOP. **(a)** The current end-to-end trainable state-of-the-art method models each tracklet as a Gaussian probability distribution in the 4D space-time volume. **(b)** Our work, 4D Spatial-temporal Object Proposal Generation and Aggregation (4D-StOP), represents each tracklet with multiple proposals generated using voting-based center predictions in the 4D space-time volume, which are then aggregated to obtain the final tracklets.

In the wake of these advancements, Aygün *et al.* [2] proposed the 4D Panoptic LiDAR Segmentation task, where the goal is to perform panoptic segmentation of LiDAR scans across space-time volumes of point cloud data. Hence, each LiDAR point is assigned a semantic class label and a temporally consistent instance ID in a given sequence of 3D LiDAR scans. This task is more challenging than its predecessor 3D Panoptic LiDAR Segmentation [5] since temporally consistent instance IDs and semantic class labels have to be preserved over the time continuum. It is also critical for domains such as autonomous driving, where autonomous agents have to be able to continuously interact with 4D environments and be robust to rapid changes in the space-time volume.

For this task, the current research can be categorized into two different groups. In the first group, the existing methods follow a tracking-by-detection paradigm where an off-the-shelf 3D panoptic segmentation network [33, 47, 61] is first employed to obtain the instances in each scan along with the corresponding semantic class labels. The instances from different scans are then further assigned a consistent ID with the help of various data association methods [28, 30, 36, 50] in a second step. Since an off-the-shelf network is used, these methods are not end-to-end trainable, and often utilize multiple different networks.

The second group of methods [2, 23] uses a single end-to-end trainable network to directly generate the semantic class label for each point and a temporally consistent instance ID for the corresponding objects. The current top performing method [2] takes as input a 4D space-time volume which is generated by combining multiple LiDAR scans. A semantic segmentation network assigns a semantic class to each of the input points, while the object instances are modeled as Gaussian probability distributions over the foreground points. In addition to the Gaussian parameters, the network also predicts object centers and the tracklets are generated by clustering the points around these centers in the features space, by evaluating a Gaussian probability function as seen in Fig. 1(a). While such clustering techniques generally work well on small 4D volumes, the method

generates only one cluster center and therefore generates only one proposal for each object which limits the representation quality.

In this work, instead of representing tracklets as Gaussian distribution, we follow a new paradigm for forming tracklets in 4D space-time volume. Inspired by [15], we represent tracklets as spatio-temporal object proposals, then aggregate these proposals to obtain the final tracklets as shown in Fig. 1(b). Object proposals have been used successfully for object detection and instance segmentation in both 2D [20, 42] and 3D [15, 41] domains. The success of these methods relies on producing a large number of proposals, thereby producing reliable outputs. In our paper, we investigate whether the success of using object proposals can also be observed in a 4D space-time volume, and empirically prove that they in fact do. For this, each feature point in our method votes for the closest center point to first generate object proposals, which are then aggregated using high-level learned geometric features in the 4D volume. For the first step, we utilize Hough voting [41] in the 4D volume and show that it works well in this large and dynamic space. An alternative strategy to generate multiple proposals would be the adaptation of the Gaussian probability-based clustering, successfully used in both 2D [1, 37] and 3D [2]. However, we show that the performance of such an adaptation is inferior to Hough voting used in this work (see Sec 4.2).

Our method, 4D Spatio-temporal Object Proposal Generation and Aggregation (4D-StOP), generates better tracklets associations over an input 4D space-time volume, and correspondingly outperforms the state-of-the-art methods on the 4D Panoptic LiDAR Segmentation task. In summary, we make **the following contributions**: (i) we tackle the task of 4D Panoptic LiDAR Segmentation in an end-to-end trainable fashion using a new strategy that first generates tracklet proposals and then further aggregates them; (ii) our method uses a center based voting technique to first generate tracklet proposals, which are then aggregated using our novel learned geometric features; (iii) we achieve the state-of-the-art results among all methods on SemanticKITTI test set, where we outperform the best performing end-to-end trainable method by a large margin.

2 Related Work

Point Cloud Segmentation. With the availability of multiple large-scale outdoor datasets [4, 18], deep learning models are leading the field of LiDAR semantic segmentation [13, 33, 46, 47, 52, 57, 61]. These end-to-end trainable methods can be categorized into two groups: In the first group, the methods [13, 33, 52, 57] convert 3D point clouds to 2D grids and make use of 2D convolution networks. PolarNet [57] projects 3D points to a bird’s-eye-view and works on a polar coordinate system. RangeNet++ [33] relies on a spherical projection mechanism and exploits range images. The second group of methods [46, 47, 61] directly works on 3D data to preserve the 3D geometry by using 3D convolution networks. Cylinder3D [61] uses cylindrical partitioned coordinate systems and asymmetrical 3D convolution networks. Similar to [2], we rely on KPConv [47] as feature backbone which directly applies deformable convolutions on point clouds.

Multi-Object Tracking. In 2D multi-object tracking (MOT), earlier methods [6, 7, 29, 32] follow the tracking-by-detection paradigm where off-the-shelf 2D object detection networks [20, 42] are utilized to obtain objects in the frames, then the objects across frames are associated with different data association methods. Later methods [31, 39, 49, 58] shifted to joint detection and association in a single framework. Recently, 3D LiDAR MOT has become popular in the vision community, due to the impressive developments in 3D object detectors [28, 42] and newly emerging datasets [8, 45]. The methods tackle 3D MOT either with tracking-by-detection paradigm [25, 51] or with joint-detection-and-tracking paradigm [3, 56], similar to the methods in 2D MOT. All of these methods work on single scans, *i.e.*, on the spatial domain, to track objects over time. In contrast, our method works on a unified space and time domain, *i.e.*, 4D volume formed by combining multi-scans, to localize and associate objects.

Panoptic LiDAR Segmentation and Tracking. 4D Panoptic LiDAR Segmentation [2, 23, 30] is an emerging research topic that unifies semantic segmentation, instance segmentation and tracking of LiDAR point clouds into a single framework. The initial work proposes a unified architecture consisting of three heads that tackle semantic segmentation, instance segmentation and tracking tasks individually. The recent work [30] obtains semantic labels and 3D object detections from an off-the-self panoptic segmentation network and adapts contrastive learning [9, 10, 38] into tracking to associate the 3D detections. The state-of-the-art method [2] in the end-to-end trainable frameworks is a bottom-up approach that jointly assigns semantic labels and associates instances across time. It works on a space-time volume where tracklets are generated by evaluating Gaussian distributions. We direct future research towards end-to-end trainable models, our work aims to improve tracklet representation in space-time volume, and propose to tackle tracklets representation as multi-proposal generation and aggregation in 4D volume.

3 Method

The overall architecture of our method is depicted in Fig. 2. We pose the task of 4D Panoptic LiDAR Segmentation by splitting it into the two subtasks, semantic segmentation as well as instance segmentation, and solve them in parallel. The semantic segmentation branch performs point-wise classification for each point in the 4D volume. For the instance segmentation task, we apply an object-centric voting approach in the 4D space-time volume. Based on the votes, multiple spatio-temporal proposals are generated. The features of these spatio-temporal proposals are learned to aggregate the proposals and generate the final tracklets. In the end, the results of the semantic segmentation and the instance segmentation branches are combined. To ensure that all points in an instance have the same semantic class label, we exploit majority voting where the most frequently occurring semantic class label among the instance points determines the semantic labels for all points within the instance.

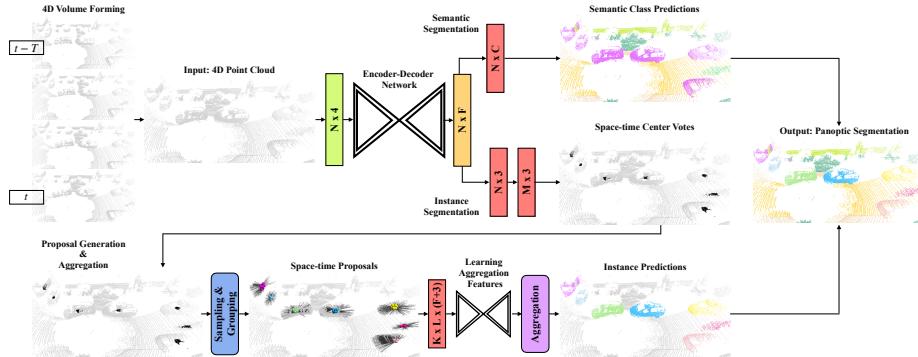


Fig. 2: 4D-StOP Overview. Given a 4D point cloud, our method splits the task of 4D Panoptic LiDAR Segmentation into two subtasks semantic segmentation and instance segmentation. The semantic segmentation branch performs point-wise classification in 4D volume. The instance segmentation branch applies object-centric voting in 4D space-time to generate proposals followed by proposal aggregation based on learned features to obtain the final instances. L is the number of points associated to a proposal.

3.1 4D Volume Formation

To form overlapping 4D point clouds, multiple consecutive scans are stacked together. For a scan at time-step t and a temporal window size T , the point clouds within the temporal window $\{\max(0, t-T), \dots, t\}$ are merged. However, stacking all the points from the scans in the temporal window T is not possible due to memory constraints. To overcome this problem, we follow the *importance sampling* strategy from [2]. Here, 10% of the points with a probability proportional to the per-point objectness score are sampled from the previous scans at time-steps $t-T, \dots, t-1$ while processing the scan at time-step t , which enables to focus on points from *thing* classes while preserving the points from *stuff* classes.

3.2 4D-StOP

Semantic Segmentation and Voting for 4D Volume. Given a 4D point cloud with N points as input, we first learn strong point features with a backbone that encodes the semantics and the geometry of the underlying multi-scan scene. We use an encoder-decoder architecture as our backbone [47] to generate per-point features $f_i \in \mathbb{R}^F$, where $i \in \{1, \dots, N\}$. On top of the backbone, we add a semantic head module and a voting module; and perform these tasks jointly. The semantic head predicts a semantic class label for each point and encodes the semantic context into the point features. To learn the semantic labels, a standard cross-entropy classification loss L_{sem} is used. After the semantic head predicts a semantic class label for each point in the 4D volume, we distinguish between the points assigned to foreground and background classes and run the voting module only on the foreground points.

In the voting module, we adapt the deep Hough voting used in [41] to work on 4D space-time volumes. Every foreground point in the 4D volume votes for the corresponding space-time object center. In particular, this is realized by learning per-point relative offsets $\Delta x_i \in \mathbb{R}^3$ that point from the point positions $x_i \in \mathbb{R}^3$ to the corresponding ground truth object centers $c_i^* \in \mathbb{R}^3$ in the 4D space-time volume. The loss function for the voting module can be formulated as:

$$L_{vot} = \frac{1}{M} \|x_i + \Delta x_i - c_i^*\|_H \cdot \mathbf{1}(x_i), \quad (1)$$

where $\|\cdot\|_H$ is the Huber-Loss, $\mathbf{1}(\cdot)$ is a binary function indicating whether a point belongs to foreground, and M is the number of points in foreground. In a multi-scan scenario, the ground truth bounding-box centers of the 3D objects, *i.e.*, the ground truth bounding-box centers on a single 3D scan, cannot be simply regressed in 4D volume. Since the objects move constantly in consecutive scans, it would lead to multiple regressed centers that would eventually yield insufficient results. To avoid that, we recompute the ground truth bounding-box centers for the objects in the 4D space-time volumes such that the ground-truth centers c_i^* in Eq. 1 are the ground-truth bounding-box centers of the objects in the merged 4D volume.

Proposal Generation. The proposals are generated by uniformly sampling and grouping the points based on the spatial-temporal proximity of the votes within a 4D volume. For each foreground point, there exists a vote to the corresponding space-time object centers. From these center votes, K samples are chosen using the Farthest Point Sampling (FPS) selection technique used in [41] to serve as proposal centers $y_j = x_j + \Delta x_j, \forall j \in \{1, \dots, K\}$. As suggested in [15], we also tested Random Sampling technique, which yields results on the same level as FPS. Although both sampling methods give similar results, we continue with the FPS technique as the probability of missing an object is much lower. To complete the proposal generation, all points are assigned to proposal j if a point votes for a center within a radius $r (= 0.6m)$ of the sampled proposal position y_j . In the end, each proposal contains the proposal positions y_j near the space-time object centers and a set of points s_i associated with each proposal j .

Proposal Aggregation. A simple way to aggregate multiple proposals is to use the proposal center positions, *i.e.* use relatively simple features. However, proposal aggregation can be enhanced by learning more advanced geometric features. We have K object proposal in 4D space-time volume consisting of the proposal positions y_j and a set of points s_i associated with each proposal j . Also, each of the points associated with a proposal has a per-point feature $f_i \in \mathbb{R}^F$. We get the proposal features $g_i \in \mathbb{R}^D$ which describes the local geometry of their associated objects by using a PointNet-like module [40] applied to the per-point features of the associated points. In particular, the per-point features in combination with the per-point positions that belong to a proposal are processed by a shared MLP with output sizes (128, 128, 128), then channel-wise max-pooling is applied to predict a single feature vector $g_i \in \mathbb{R}^D$. Each proposal is processed independently. The proposal features are further processed by another MLP with

output sizes $(128, 128, E)$ to learn the E dimensional geometric aggregation features. Overall, we use three geometric features, where two of them, a refined proposal center point prediction $y_i + \Delta y_i$ and an object radius estimation r_i , are adapted from [15]. The third novel geometric feature that we use for aggregation is the predicted bounding box size $bb_i = (l_i, h_i, w_i) \in \mathbb{R}^3$. This additional geometric feature is helpful in better aggregating the proposal as shown in our experiments (see Sec. 4.2). The aggregation-loss is defined as:

$$L_{agg} = \|y_i + \Delta y_i - c_i^*\|_H + \|r_i - r_i^*\|_H + \|bb_i - bb_i^*\|_H \quad (2)$$

where c_i^* is the center, r_i^* is the radius of the closest ground truth bounding sphere and bb_i^* are the length, height and width of closest ground truth bounding-box. Finally, we aggregate the proposals by applying DBScan clustering to the learned geometric features. The proposals whose geometric features end up in the same cluster are combined. The union of the points in the aggregated proposals builds the final space-time object predictions, that is the final tracklets.

3.3 Tracking

Within an input 4D volume, temporal association is resolved implicitly as object instances are grouped jointly in 4D space-time. Due to memory constraints, there is a limitation to the size of the 4D volume that can be processed at once. As a result, we split the entire 4D point cloud into multiple overlapping volumes of smaller sizes. Following the strategy used in [2], final tracklets across the entire space-time volume are obtained by associating the individual fragmented tracklets based on the best IoU overlap scores. For IoU scores below a particular threshold, we discard the corresponding associations and start a new tracklet.

3.4 Training and Implementation Details

The model is trained end-to-end using the multi-task loss $L = \alpha \cdot L_{sem} + \beta \cdot L_{vot} + \gamma \cdot L_{agg}$ for 550K iterations in total. For the first 400K iterations, we set $\alpha = 1$, $\beta = 1$ and $\gamma = 0$ in order to predict the semantic classes and to obtain the spatio-temporal proposals in a sufficient quality. In the remaining 150K iterations, we set $\alpha = 0$, $\beta = 0$ and $\gamma = 1$ and freeze the semantic segmentation and the voting modules, then learn high-quality geometric features for aggregating the spatio-temporal proposals. Our model is implemented in PyTorch. The training is performed on a single NVIDIA A40 (48GB) GPU and the inference on a single NVIDIA TitanX (12GB) GPU. We apply random rotation, translation and scaling of the scene as data augmentations. Moreover, we randomly drop some points of the scene and randomly add noise.

4 Experiments

4.1 Comparing with State-of-the-Art Methods

Dataset and Metric. We evaluate our method on the SemanticKITTI LiDAR dataset [4]. It consists of 22 sequences from KITTI odometry dataset [19]. There

	Method	<i>LSTQ</i>	S_{assoc}	S_{cls}	IoU^{St}	IoU^{Th}
Not-End-to-End	RangeNet++ [33] + PP + MOT	35.5	24.1	52.4	64.5	35.8
	KPConv [47] + PP + MOT	38.0	25.9	55.9	66.9	47.7
	RangeNet++ [33] + PP + SFP	34.9	23.3	52.4	64.5	35.8
	KPConv [33] + PP + SFP	38.5	26.6	55.9	66.9	47.7
	CIA [30]	63.1	65.7	60.6	66.9	52.0
End-to-End	4D-PLS (4 scan) [2]	56.9	56.4	57.4	66.9	51.6
	4D-StOP(Ours)(2 scan)	62.9	67.3	58.8	68.3	53.3
	4D-StOP(Ours)(4 scan)	63.9	69.5	58.8	67.7	53.8

Table 1: **Scores on SemanticKITTI 4D Panoptic Segmentation test set.** MOT - *tracking-by-detection* by [50], SFP - *tracking-by-detection* via scene flow based propagation [36], PP - PointPillars [28].

exist three splits: training, validation and test split where sequences 00 to 10 are used as training and validation split, and sequences 11 to 21 as test split. Each point is labeled with a semantic class and a temporally consistent instance ID. In total, there are 19 semantic classes among which 8 are *things* classes and the remaining 11 are *stuff* classes. The main evaluation metric is *LSTQ*, which is the geometric mean of the classification score S_{cls} and the association score S_{assoc} , *i.e.* $LSTQ = \sqrt{S_{cls} \times S_{assoc}}$ [2]. The classification score S_{cls} measures the quality of semantic segmentation, while the association score S_{assoc} measures the quality of association in a 4D continuum. Unlike the previous evaluation metrics [23, 24, 48], *LSTQ* performs semantic and instance association measurements independently to avoid penalization due to the entanglement of semantic and association scores. Our strategy of generating proposals and then aggregating them in the 4D volume generates very good object associations while maintaining the semantic classification quality, which is reflected in the association (S_{assoc}) and the classification scores (S_{cls}) in the following experiments.

Results. In Tab. 1, we present the results on the SemanticKITTI 4D Panoptic Segmentation test set. Among the end-to-end trainable methods, 4D-StOP outperforms the previous state-of-the-art method 4D-PLS [2] by a large margin, obtaining a 7% boost in *LSTQ* score, when utilizing the same number of scans (4 scans). Especially, we achieve **13.1%** improvement in S_{assoc} , which shows that 4D-StOP produces better segmented and associated tracklets in a 4D space-time volume compared to 4D-PLS. While we achieve state-of-the-art results when using 4 scans, 4D-StOP also performs well with only 2 scans. The performance drop in the *LSTQ* score is merely 1%.

Comparing 4D-StOP with the non-end-to-end trainable methods, we observe that 4D-StOP also outperforms the current not-end-to-end trainable state-of-the-art method CIA [30] (+0.8 *LSTQ*). We would like to highlight that CIA leverages an off-the-shelf 3D panoptic segmentation network with a stronger backbone than used in 4D-StOP. Thus, it has a better S_{cls} score than 4D-StOP,

	Method	LSTQ	S_{assoc}	S_{cls}	IoU^{St}	IoU^{Th}
Not-End-to-End	RangeNet++ [33] + PP + MOT	43.8	36.3	52.8	60.5	42.2
	KPConv [47] + PP + MOT	46.3	37.6	57.0	64.2	54.1
	RangeNet++ [33] + PP + SFP	43.4	35.7	52.8	60.5	42.2
	KPConv [33] + PP + SFP	46.0	37.1	57.0	64.2	54.1
End-to-End	MOPT [23]	24.8	11.7	52.4	62.4	45.3
	4D-PLS [2] (2 scan)	59.9	58.8	61.0	65.0	63.1
	4D-PLS [2] (4 scan)	62.7	65.1	60.5	65.4	61.3
	4D-StOP(Ours)(2 scan)	66.4	71.8	61.4	64.9	64.1
	4D-StOP(Ours)(4 scan)	67.0	74.4	60.3	65.3	60.9

Table 2: **Scores on SemanticKITTI 4D Panoptic Segmentation validation set.** MOT - *tracking-by-detection* by [50], SFP - *tracking-by-detection* via scene flow based propagation [36], PP - PointPillars [28].

i.e. the increase in S_{cls} score results from the semantic segmentation performance of the backbone network. In contrast to CIA, we do not take advantage of such a stronger backbone but rather preserve the exact same configuration of our baseline 4D-PLS. Despite the worse classification score, the *LSTQ* score for 4D-StOP is still better than for CIA. Reason for this is the better working object association in our end-to-end trainable approach. 4D-StOP is better in S_{assoc} than CIA by a large margin with **3.8%** boosting, where the increase is gained from the method basically.

In Tab. 2, we report the results on the SemanticKITTI 4D Panoptic Segmentation validation set. In the validation set results, we observe a similar pattern as in the test set. 4D-StOP outperforms all the methods by a large margin in both not-end-to-end trainable and end-to-end trainable categories. It produces better scores compared to 4D-PLS by **+4.3 LSTQ** and **+9.3 S_{assoc}** for 4 scans, and by **+6.5 LSTQ** and **+13.0 S_{assoc}** for 2 scans. Although there is a slight performance drop between the two and four scan set-ups, this drop is smaller compared to 4D-PLS (**0.6% LSTQ** drop for 4D-StOP compared to **2.8% LSTQ** drop for 4D-PLS). This shows that 4D-StOP is more robust to the number of input scans. Since processing two scans is faster than processing four scans, 4D-StOP offers a better trade-off between accuracy and speed than 4D-PLS.

4.2 Analysis

We ablate all of our design choices on the validation split of the SemanticKITTI dataset, *i.e.* only on sequence 08 of the dataset. In this section, we provide detailed experimental analysis, and discuss the impact of each of the components in our method, including the geometric features. In addition, we also reformulate 4D-StOP, where the proposals are modeled as Gaussian distributions similar to our baseline 4D-PLS, and compare this formulation with our voting based

	<i>LSTQ</i>	S_{assoc}	S_{cls}
① 4D-PLS [2] (2 scan)	59.9	58.8	61.0
② Our Baseline	63.7	66.7	60.8
③ Aggregating w/ Proposal Positions	65.3	70.0	60.8
④ Aggregating w/ Proposal Positions + Majority Voting	65.5	70.0	61.2
⑤ Aggregating w/ Geometric Features + Majority Voting	66.4	71.8	61.4

Table 3: Analysis of each component in 4D-StOP.

approach. In the experimental analysis, a 4D volume is formed by combining 2 scans unless it is not explicitly indicated it is generated from 4 scans.

Effect of 4D-StOP components. We evaluate the impact of each 4D-StOP component in Tab. 3. The main improvements come from the proposal generation and aggregation steps. To investigate the impact of the aggregation step, we create a baseline in experiment ② by applying the traditional non-maximum-suppression (NMS) on the spatio-temporal proposals. Even this baseline outperforms 4D-PLS by a large margin with **3.8%** boosting in the *LSTQ* score. In order to see the effect of the aggregation step, we first conduct a naive aggregation by combining proposals based on the proposal positions (experiment ③) and observe another **1.6%** increase in the *LSTQ* score. In experiment ④, we ablate adding majority voting into our pipeline which only brings a slight increment (**+0.4** S_{cls}). Lastly, in experiment ⑤, we show the results for adding the learned high-level geometric features as the final component, which boosts the performance in the S_{assoc} and *LSTQ* scores by **1.8%** and **0.9%** respectively. Overall, our method outperforms the prior end-to-end trainable method 4D-PLS [2] significantly irrespective of whether we use our proposal aggregation strategy or a much simpler NMS. This shows that, similar to the observations made in 2D and 3D spatial domains, proposal-based methods surpass other bottom-up approaches in the 4D space-time domain.

4D-StOP with Gaussian Distribution. An alternative way of generating proposals would be to use Gaussian probability distributions. To do this, we adapt 4D-PLS [2] to work with our proposal generation and aggregation paradigm. In all of the following experiments, 4 scan setup is used to be consistent with the best results in 4D-PLS. We generate proposals based on per-point objectness scores, which indicate the probability that a point is close to an object center. We select all the points in the 4D volume with an objectness score above a certain threshold as the proposal center points. Then, we assign points to a proposal by evaluating each point under the Gaussian probability distribution function with the proposal center as cluster seed point. For each point, the Gaussian probability indicates whether the point belongs to the proposal. All the points with a probability above a certain threshold are assigned to the proposal. We generate the proposals in parallel, and thus obtain overlapping Gaussian probabilities for each object, *i.e.* the multiple proposals for each ob-

	PT	R	<i>LSTQ</i>	S _{assoc}	S _{cls}
① 4D-StOP(Ours)	-	-	67.0	74.4	60.3
② 4D-PLS	0.5	-	62.7	65.1	60.5
③ 4D-StOP w/ Gaussian Distribution	0.5	0.0	61.2	62.1	60.3
④ 4D-StOP w/ Gaussian Distribution	0.7	0.0	63.4	66.7	60.3
⑤ 4D-StOP w/ Gaussian Distribution	0.7	0.6	63.1	66.0	60.3

Table 4: **Analysis for studying 4D-StOP with Gaussian Distribution.** PT - *Probability Threshold* and R - *Radius*. All of the experiments are carried out on 4 scans and the objectness score threshold is set to 0.7 as in 4D-PLS.

ject. In contrast, in 4D-PLS each object is represented with one proposal. After the proposals are generated, they are aggregated using the cluster mean points. In Tab. 4, we compare 4D-PLS and 4D-StOP with Gaussian distribution (experiments ② and ③). 4D-StOP with Gaussian distribution yields worse results than 4D-PLS (**-1.5 LSTQ**). We realized that using all proposals with an objectness score above a threshold is prone to produce more FPs than 4D-PLS, so we decided to increase the probability threshold while assigning each point to the proposals. A slight boosting **+0.7 LSTQ** is achieved in this experiment (the experiment ④), but it is still largely under our work 4D-StOP. To eliminate more FPs, we consider another strategy, where only a subset of the proposals is selected for aggregation. After generating the proposals, we start selecting the proposal with the highest objectness score from the proposals candidate pool. The proposals within a certain radius around the center point of the selected proposal are removed from the candidate pool. We repeat it until no proposal is left in the candidate pool and only aggregate the selected proposals in the end. With this strategy, a minor boosting by **+0.4 LSTQ** is observed compared to 4D-PLS in experiment ⑤, but the results are still under 4D-StOP.

In conclusion, we show that the multi-proposal generation and aggregation strategy can be implemented with Gaussian probability-based clustering used in 4D-PLS, but this does not bring a significant improvement to the base method. More importantly, the performance of such an adaptation still falls below 4D-StOP by a large margin. Thereby, with these experiments we show that Gaussian distributions can be used to represent tracklet proposals, although the performance improvement brought by them saturates soon, and are inferior to our center-based voting approach.

Impact of the learned geometric features. Our 4D-StOP method makes use of three different geometric features: the refined proposal centers, the radius estimation and the predicted bounding-box size (Sec. 3.2), to aggregate the space-time proposals. We evaluate the impact of each geometric feature on the results of our method in Tab. 5. Grouping proposals based on the proposals center positions (first row), *i.e.* grouping without utilizing the learned geometric features, serves as our baseline. First, we analyze the performance for two learned geometric features, the refined center positions and the radius estimation, which

Geometric Features				<i>LSTQ</i>	S_{assoc}	S_{cls}
Center Pos.	Ref. Center Pos.	Radius	Bounding-Box			
✓	—	—	—	65.5	70.0	61.2
—	✓	—	—	65.8	70.7	61.3
—	✓	✓	—	65.9	70.9	61.3
—	✓	✓	✓	66.4	71.8	61.4

Table 5: Analysis of the learned geometric features.

we adapt from [15]. When the refined center positions are used individually (second row) or are combined with the radius estimation (third row), a performance increase can be observed. However, we achieve a significant performance boost by introducing the bounding-box size as an additional geometric feature (fourth row). We obtain **+1.8** and **+0.9** increase on the S_{assoc} and *LSTQ* scores, respectively. The improvement is doubled by the additional use of the predicted bounding-box sizes compared to only use the refined proposal center positions and the radius estimation as the geometric features. By that, we prove two aspects. Our idea, introducing new geometric features, brings a notable boosting. Moreover, the quality of the learned geometric features affects the performance.

The impact of the learned geometric features is also visualized in an example scan in Fig. 3. The refined center positions are used as the geometric feature. In the first column, it is possible to observe how the proposal center positions get more precise by the use of the learned refinement. In the second row, the refined predicted proposal centers are closer to the ground truth centers compared to the predicted proposal centers without refinement in the first row. Also, we show how the predictions change with the learned geometric features in the second column. As seen in the blue boxes, we can correctly separate two objects predicted as one single object without the learned geometric features.

4.3 Qualitative Results

In Fig. 4, we present qualitative results for different example scans in the first and second column for 4D-PLS and our 4D-StOP method respectively. We indicate each specific case within the boxes. Compared to 4D-PLS, our method is able to catch and segment the missing object (left example in the first row), complete the partly segmented objects (right example in the first row and the examples in the second row), and correctly separate the two objects predicted as one single object (the middle example in the third row).

Fig. 5 shows two examples where 4D-StOP fails (red boxes). We observe that 4D-StOP can fail when objects and thus their object center votes are very close to each other and objects have a similar size. Then, the aggregation features of the corresponding proposals are quite similar and it is hard for 4D-StOP to distinguish which proposal belongs to which object. As a consequence, 4D-StOP mixes the objects (left) or predicts the objects as one instance (right).

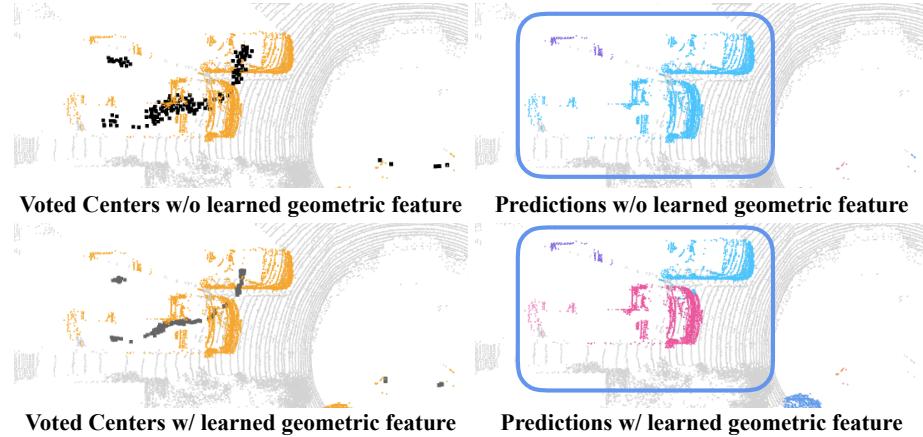


Fig. 3: Learned Geometric Features. On an example scan, we show the impact of a learned geometric feature, the refined space-time proposal center point positions. In the first column, we display how the center positions change in 4D space-time volume with the learned geometric feature. More precise center positions can be observed using the learned geometric features. In the second column, the predictions are depicted with and without the learned geometric feature. Inside the blue boxes, it can be seen how a wrong prediction is corrected by using the learned geometric feature. In order to emphasize the results, the points with the *stuff* class label are colored gray. The example is generated by combining two scans.

5 Conclusion

In this work, we have proposed a new end-to-end trainable method 4D-StOP for 4D Panoptic LiDAR Segmentation task, which models tracklets in a space-time volume by generating multiple proposals, then aggregating them. Our model is based on center voting for generating tracklet proposals and learning high-level geometric features for aggregating tracklet proposals. We have introduced novel geometric features to enhance the proposal aggregation that improves performance notably. Additionally, we have proved that center voting is feasible in the space-time volume. We achieve state-of-the-art results on SemanticKITTI test-set. We hope that our work would inspire new representations of tracklets for future methods tackling 4D Panoptic LiDAR Segmentation task in a unified space-time volume.

Acknowledgments. We thank Sima Yagmur Zulfikar for her help and feedback on the figures, and István Sárándi for his helpful comments on our manuscript. This project was funded by ERC Consolidator Grant DeeVise (ERC-2017-COG-773161). The computing resources for several experiments were granted by RWTH Aachen University under project 'supp0003'. Francis Engelmann is a post-doctoral research fellow at the ETH AI Center. This work is part of the first author's master thesis.

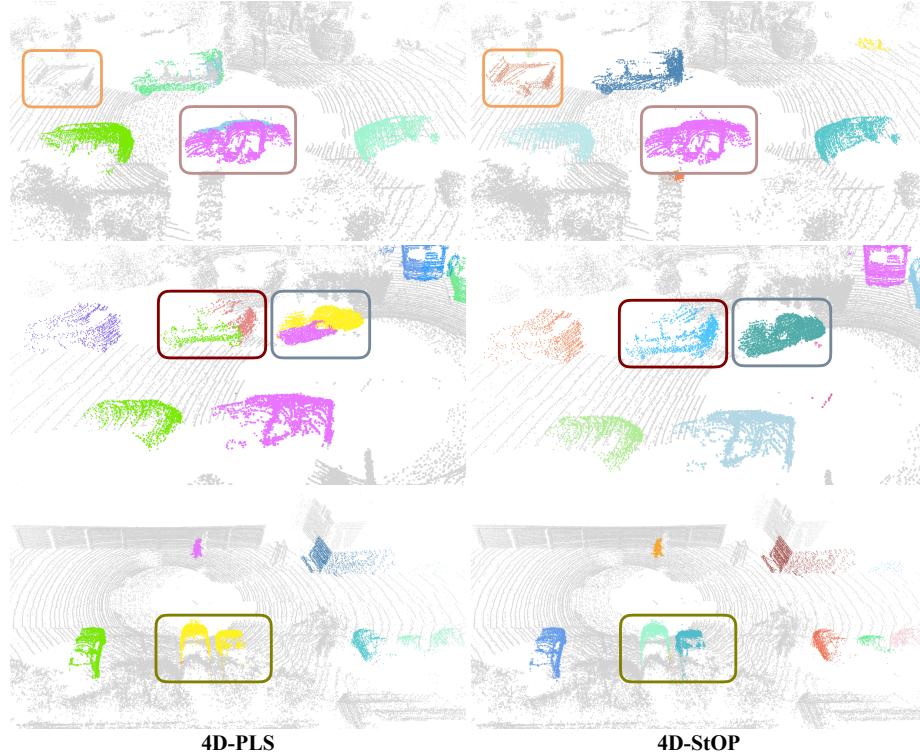


Fig. 4: Qualitative results on SemanticKITTI validation. We show results for 4D-PLS (left) and 4D-StOP (right) of the same scans. Interesting cases are highlighted with boxes. Our method catches missed objects (first row), complete partially segmented objects (second row), and separate two incorrectly predicted objects as one single object (third row). For simplicity, points with *stuff* class label are colored gray. All the scans are 4D volumes over 4 scans.

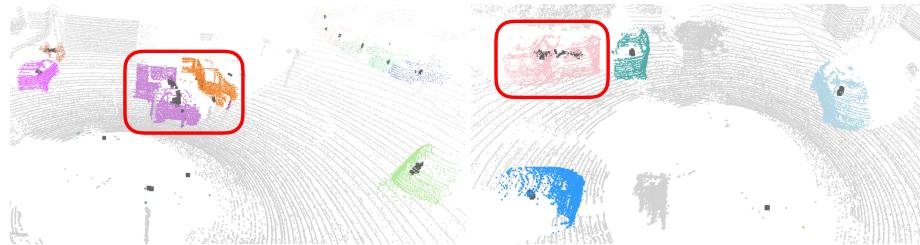


Fig. 5: Failure cases of 4D-StOP on the SemanticKITTI validation split. We show two scans where 4D-StOP fails. Here, 4D-StOP cannot separate two objects and mixes them or predicts them as one single object (see red boxes). The dark gray squares show the refined proposal centers. Again, the points with *stuff* class label are colored gray and all the scans are 4D volumes over 4 scans.

References

1. Athar, A., Mahadevan, S., Osep, A., Leal-Taixé, L., Leibe, B.: STEM-Seg: Spatio-temporal Embeddings for Instance Segmentation in Videos. In: ECCV (2020) [3](#)
2. Aygün, M., Osep, A., Weber, M., Maximov, M., Stachniss, C., Behley, J., Leal-Taixé, L.: 4D Panoptic Segmentation. In: CVPR (2021) [2](#), [3](#), [4](#), [5](#), [7](#), [8](#), [9](#), [10](#)
3. Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., Tai, C.L.: TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers. In: arXiv preprint arXiv:2203.11496 (2022) [1](#), [4](#)
4. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: ICCV (2019) [3](#), [7](#)
5. Behley, J., Milioto, A., Stachniss, C.: A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI. In: ICRA (2021) [1](#), [2](#)
6. Bergmann, P., Meinhardt, T., Leal-Taixé, L.: Tracking without bells and whistles. In: ICCV (2019) [4](#)
7. Braso, G., Leal-Taixé, L.: Learning a Neural Solver for Multiple Object Tracking. In: CVPR (2020) [4](#)
8. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR (2020) [4](#)
9. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In: NIPS (2020) [4](#)
10. Chen, T., Kornblith, S., 0002, M.N., Hinton, G.E.: A Simple Framework for Contrastive Learning of Visual Representations. In: International Conference on Machine Learning (ICML) (2020) [4](#)
11. Chibane, J., Engelmann, F., Tran, T.A., Pons-Moll, G.: Box2Mask: Weakly Supervised 3D Semantic Instance Segmentation Using Bounding Boxes. In: ECCV (2022) [1](#)
12. Chiu, H.K., Prioletti, A., Li, J., Bohg, J.: Probabilistic 3D Multi-Object Tracking for Autonomous Driving. In: arXiv preprint arXiv:2001.05673 (2020) [1](#)
13. Cortinhal, T., Tzelepis, G., Erdal Aksoy, E.: SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds. In: International Symposium on Visual Computing (2020) [3](#)
14. Elich, C., Engelmann, F., Schult, J., Kontogianni, T., Leibe, B.: 3D-BEVIS: Birds-Eye-View Instance Segmentation. In: German Conference on Pattern Recognition (GCPR) (2019) [1](#)
15. Engelmann, F., Bokeloh, M., Fathi, A., Leibe, B., Nießner, M.: 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In: CVPR (2020) [1](#), [3](#), [6](#), [7](#), [12](#)
16. Engelmann, F., Kontogianni, T., Hermans, A., Leibe, B.: Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In: ICCV Workshops (2017) [1](#)
17. Engelmann, F., Kontogianni, T., Schult, J., Leibe, B.: Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In: ECCV Workshops (2018) [1](#)
18. Fong, W.K., Mohan, R., Hurtado, J.V., Zhou, L., Caesar, H., Beijbom, O., Valada, A.: Panoptic nuScenes: A Large-Scale Benchmark for LiDAR Panoptic Segmentation and Tracking. In: arXiv preprint arXiv:2109.03805 (2021) [3](#)

19. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: CVPR (2012) [7](#)
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017) [3](#), [4](#)
21. Hong, F., Zhou, H., Zhu, X., Li, H., Liu, Z.: LiDAR-based Panoptic Segmentation via Dynamic Shifting Network. In: CVPR (2021) [1](#)
22. Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In: CVPR (2019) [1](#)
23. Hurtado, J.V., Mohan, R., Burgard, W., Valada, A.: MOPT: Multi-Object Panoptic Tracking. In: CVPR Workshops (2020) [2](#), [4](#), [8](#), [9](#)
24. Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Video Panoptic Segmentation. In: CVPR (2020) [8](#)
25. Kim, Aleksandr, O.A., Leal-Taixé, L.: EagerMOT: 3D Multi-Object Tracking via Sensor Fusion. In: ICRA (2021) [4](#)
26. Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3D Instance Segmentation via Multi-Task Metric Learning. In: ICCV (2019) [1](#)
27. Landrieu, L., Simonovsky, M.: Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In: CVPR (2018) [1](#)
28. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: PointPillars: Fast Encoders for Object Detection From Point Clouds. In: CVPR (2019) [2](#), [4](#), [8](#), [9](#)
29. Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., Savarese, S.: Learning an image-based motion context for multiple people tracking. In: CVPR (2014) [4](#)
30. Marcuzzi, R., Nunes, L., Wiesmann, L., Vizzo, I., Behley, J., Stachniss, C.: Contrastive Instance Association for 4D Panoptic Segmentation using Sequences of 3D LiDAR Scans. In: IEEE Robotics and Automation Society (2022) [2](#), [4](#), [8](#)
31. Meinhardt, T., Kirillov, A., Leal-Taixé, L., Feichtenhofer, C.: TrackFormer: Multi-Object Tracking with Transformers. In: CVPR (2022) [4](#)
32. Milan, A., Leal-Taixé, L., Schindler, K., Reid, I.D.: Joint Tracking and Segmentation of Multiple Targets. In: CVPR (2015) [4](#)
33. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In: IROS (2019) [1](#), [2](#), [3](#), [8](#), [9](#)
34. Milioto, A., Behley, J., McCool, C., Stachniss, C.: LiDAR Panoptic Segmentation for Autonomous Driving. In: IROS (2020) [1](#)
35. Misra, I., Girdhar, R., Joulin, A.: An End-to-End Transformer Model for 3D Object Detection. In: ICCV (2021) [1](#)
36. Mittal, H., Okorn, B., Held, D.: Just Go With the Flow: Self-Supervised Scene Flow Estimation. In: CVPR (2020) [2](#), [8](#), [9](#)
37. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth. In: CVPR (2019) [3](#)
38. Oord, A.v.d., Li, Y., Vinyals, O.: Representation Learning with Contrastive Predictive Coding. In: arXiv preprint arXiv:1807.03748 (2018) [4](#)
39. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-Dense Similarity Learning for Multiple Object Tracking. In: CVPR (2021) [4](#)
40. Qi, C., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: CVPR (2017) [1](#), [6](#)
41. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep Hough Voting for 3D Object Detection in Point Clouds. In: ICCV (2019) [1](#), [3](#), [6](#)
42. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: NIPS (2015) [3](#), [4](#)

43. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In: CVPR (2020) [1](#)
44. Shi, S., Wang, X., Li, H.: PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In: CVPR (2019) [1](#)
45. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S.M., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In: CVPR (2020) [4](#)
46. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In: ECCV (2020) [3](#)
47. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: KPCConv: Flexible and Deformable Convolution for Point Clouds. In: ICCV (2019) [1, 2, 3, 5, 8, 9](#)
48. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: MOTS: Multi-Object Tracking and Segmentation. In: CVPR (2019) [8](#)
49. Wang, Y., Kitani, K., Weng, X.: Joint Object Detection and Multi-Object Tracking with Graph Neural Networks. In: ICRA (2021) [4](#)
50. Weng, X., Wang, J., Held, D., Kitani, K.: 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. In: IROS (2020) [1, 2, 8, 9](#)
51. Weng, X., Wang, J., Held, D., Kitani, K.: AB3DMOT: A Baseline for 3D Multi-Object Tracking and New Evaluation Metrics. In: ECCV Workshops (2020) [4](#)
52. Wu, B., Wan, A., Yue, X., Keutzer, K.: SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In: ICRA (2018) [3](#)
53. Yan, Y., Mao, Y., Li, B.: SECOND: Sparsely Embedded Convolutional Detection. Sensors (Basel, Switzerland) (2018) [1](#)
54. Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, A.: Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In: arXiv preprint arXiv:1906.01140 (2019) [1](#)
55. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3DSSD: Point-based 3D Single Stage Object Detector. In: CVPR (2020) [1](#)
56. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3D Object Detection and Tracking. In: CVPR (2021) [1, 4](#)
57. Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Foroosh, H.: PolarNet: An Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation. In: CVPR (2020) [1, 3](#)
58. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking Objects as Points. In: ECCV (2020) [4](#)
59. Zhou, Y., Tuzel, O.: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In: CVPR (2018) [1](#)
60. Zhou, Z., Zhang, Y., Foroosh, H.: Panoptic-PolarNet: Proposal-free LiDAR Point Cloud Panoptic Segmentation. In: CVPR (2021) [1](#)
61. Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D.: Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation. In: CVPR (2021) [1, 2, 3](#)

Supplementary Material

A Additional Ablation Studies

In order to provide a better insight, we present additional ablation studies on the validation split of the SemanticKITTI dataset. In this section, we compare run-time performance of our method 4D-StOP with 4D-PLS. We also evaluate how the number of generated proposals affects the results. Moreover, we discuss utilizing the random sampling technique over the Farthest Point Sampling (FPS) technique for selecting K proposal centers. In all experimental analysis, we form a 4D volume by combining 2 scans unless it is not explicitly indicated it is generated from 4 scans.

Run-time Analysis. We show run-time analysis for 4D-PLS and 4D-StOP in Fig. 6. We analyze the results for 2 scan and 4 scan setup separately. Times are recorded as scans per second (sps). We initially compare the run-time performance of 4D-PLS with a simpler version of 4D-StOP where the proposals are combined based on proposal positions, *i.e.* without having the network module for learning aggregation features. For this case, we have an on-par run-time performance with 4D-PLS for both 2 scan and 4 scan setups, the difference is insignificantly small (~ 0.2 sps). We would like to remind that this simpler version of 4D-StOP outperforms 4D-PLS in the $LSTQ$ score by 5.4% in 2 scan setup while having the on-par run-time (see Sec 4.2), *i.e.* this simpler version offers much better accuracy than 4D-PLS without losing any speed. If we add the network module for learning aggregation features, *i.e.* when we have the complete version of our method, 4D-StOP is 1.2 and 1.1 sps slower than 4D-PLS for 2 scan and 4 scan setups, respectively. The run-time performance drops basically due to the network module for learning aggregation features. Note that, all of these run-time analyses are performed on a single Nvidia TitanX (12GB) GPU.

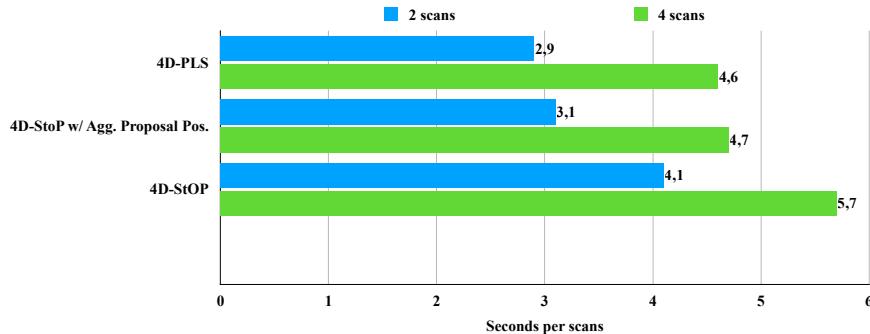


Fig. 6: **Run-time performance comparison of 4D-PLS and 4D-StOP.** The results are depicted for 2 scan and 4 scan versions. Times are scans per second (sps).

# Proposals	$LSTQ$	S_{assoc}	S_{cls}
100	65.6	70.1	61.4
200	66.4	71.7	61.4
300	66.3	71.5	61.4
400	66.3	71.7	61.4
500	66.4	71.8	61.4
600	66.3	71.7	61.4

Table 6: **Ablation study on the impact of the number of proposals in 4D-StOP.** Proposals are aggregated based on the learned geometric features.

Effect of the Number of Proposals. In Tab. 6, we show the impact of the number of generated proposals. As seen, our method already works well with only 200 proposals. However, the number of proposals should not be set too low. With 100 proposals, we lose 1.6% in S_{assoc} and 0.8% in $LSTQ$ compared to the results with 200 proposals. The fewer proposals, the more difficult to cover all points of space-time objects. Also, when we use more than 600 proposals, we encounter memory issues. In all of our experiments, 500 proposals are used due to the highest S_{assoc} score.

Farthest Point Sampling (FPS) vs. Random Sampling. In order to generate proposals, we need to sample K proposal centers from the center votes of all foreground points. We utilize the farthest point sampling (FPS) selection technique where the votes with the farthest point positions from each other are sampled. An alternative selection technique would be random sampling where we simply select a random set of K votes. The results of both selection techniques are compared in Tab. 7. The difference is negligibly small (0.1% $LSTQ$), showing the robustness of our method to the sampling strategies. Although we obtain similar results with both sampling strategies, we make use of the FPS selection technique as the probability of completely missing an object in a scene is much lower.

	$LSTQ$	S_{assoc}	S_{cls}
Farthest-Point-Sampling	66.4	71.8	61.4
Random-Sampling	66.3	71.5	61.4

Table 7: **Ablation study on the impact of the proposal sampling strategy in 4D-StOP.** We aggregate $K = 500$ proposals based on the learned geometric features.

B Aggregation Network Details

In Fig. 7, we show the part of the 4D-StOP network in which the aggregation features are learned in detail. K Proposals are aggregated based on learned aggregation features. Firstly, the features ($F = 256$) and positions of the L points associated with a proposal are passed to a shared MLP. The output features are further max pooled to get a single feature vector for each proposal which indicates the proposal features ($D = 128$). These proposal features are passed to an additional MLP to get the final aggregation features ($E = 7$).

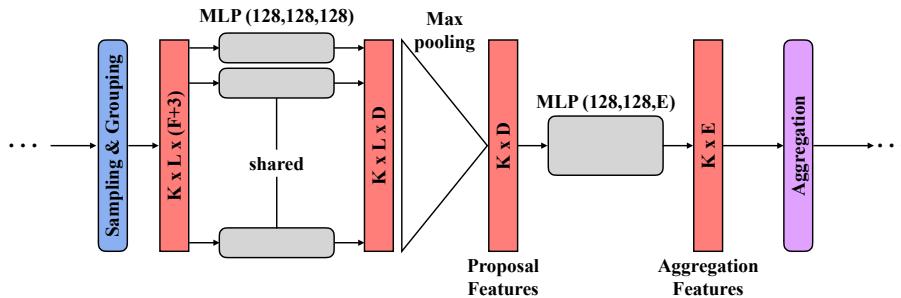


Fig. 7: The detailed architecture of Aggregation Feature Learning in 4D-StOP.

C Per-Category Results

In Tab. 8 and Tab. 9, we provide the per-category results of 4D-StOP on SemanticKITTI 4D Panoptic Segmentation test and validation sets, respectively. The scores are reported for 2 scan and 4 scan versions. The S_{assoc} scores are 0.00 for the objects from the *stuff* classes as they are not evaluated for assigning associated instance IDs in time continuum.

D Additional Qualitative Results

We depict the additional qualitative results on example scans for 4D-PLS and 4D-StOP in the first and second columns of Fig. 8. Each specific case is indicated in the boxes. In Fig. ??, we also show the additional failure cases within the red boxes.

Category	# Scans	S _{assoc}	S _{cls}
Car	2	0.78	0.97
	4	0.79	0.97
Truck	2	0.48	0.53
	4	0.56	0.46
Bicycle	2	0.16	0.44
	4	0.20	0.47
Motorcycle	2	0.35	0.52
	4	0.42	0.57
Other-vehicle	2	0.35	0.55
	4	0.42	0.53
Person	2	0.34	0.57
	4	0.40	0.55
Bicyclist	2	0.62	0.59
	4	0.66	0.62
Motorcyclist	2	0.51	0.09
	4	0.45	0.14
Road	2	0.00	0.89
	4	0.00	0.88
Sidewalk	2	0.00	0.73
	4	0.00	0.72
Parking	2	0.00	0.66
	4	0.00	0.66
Other-ground	2	0.00	0.31
	4	0.00	0.28
Building	2	0.00	0.92
	4	0.00	0.92
Vegetation	2	0.00	0.84
	4	0.00	0.84
Trunk	2	0.00	0.69
	4	0.00	0.69
Terrain	2	0.00	0.68
	4	0.00	0.68
Fence	2	0.00	0.66
	4	0.00	0.68
Pole	2	0.00	0.57
	4	0.00	0.58
Traffic-sign	2	0.00	0.55
	4	0.00	0.54

Table 8: Per-category scores on SemanticKITTI 4D Panoptic Segmentation test set.

Category	# Scans	S _{assoc}	S _{cls}
Car	2	0.87	0.97
	4	0.87	0.97
Truck	2	0.10	0.32
	4	0.17	0.31
Bicycle	2	0.49	0.75
	4	0.62	0.66
Motorcycle	2	0.89	0.85
	4	0.98	0.76
Other-vehicle	2	0.47	0.66
	4	0.56	0.63
Person	2	0.39	0.70
	4	0.47	0.66
Bicyclist	2	0.80	0.89
	4	0.75	0.88
Motorcyclist	2	0.30	0.00
	4	0.76	0.00
Road	2	0.00	0.93
	4	0.00	0.93
Sidewalk	2	0.00	0.45
	4	0.00	0.47
Parking	2	0.00	0.80
	4	0.00	0.80
Other-ground	2	0.00	0.01
	4	0.00	0.00
Building	2	0.00	0.90
	4	0.00	0.90
Vegetation	2	0.00	0.61
	4	0.00	0.60
Trunk	2	0.00	0.88
	4	0.00	0.88
Terrain	2	0.00	0.69
	4	0.00	0.68
Fence	2	0.00	0.74
	4	0.00	0.76
Pole	2	0.00	0.64
	4	0.00	0.64
Traffic-sign	2	0.00	0.49
	4	0.00	0.51

Table 9: Per-category scores on SemanticKITTI 4D Panoptic Segmentation validation set.



Fig. 8: Additional qualitative results on SemanticKITTI validation split.