4D Panoptic LiDAR Segmentation

Mehmet Aygün[1]*   Aljoša Ošep[1]*   Mark Weber[1]   Maxim Maximov[1]

Cyrill Stachniss[2]   Jens Behley[2]   Laura Leal-Taixé[1]

[1]Technical University of Munich, Germany   [2]University of Bonn, Germany

[1]{mehmet.ayguen, aljosa.osep, leal.taixe, mark-cs.weber, maxim.maximov}@tum.de

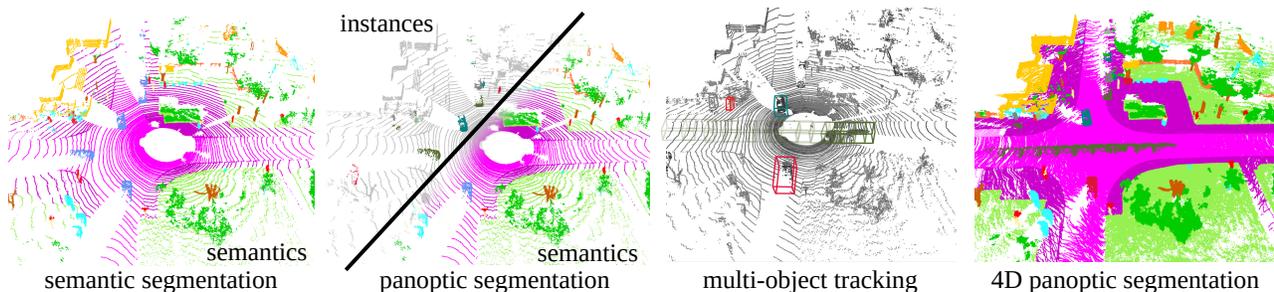[2]{firstname.lastname}@igg.uni-bonn.de

Figure 1: Types of LiDAR-based scene understanding. Semantic and panoptic segmentation assign semantic classes and determine instances in 3D space. Multi-object tracking encompasses 3D object detection in space, followed by association over time. 4D panoptic LiDAR segmentation jointly tackles semantic and instance segmentation in 3D space over time.

Abstract

Temporal semantic scene understanding is critical for self-driving cars or robots operating in dynamic environments. In this paper, we propose 4D panoptic LiDAR segmentation to assign a semantic class and a temporally-consistent instance ID to a sequence of 3D points. To this end, we present an approach and a point-centric evaluation metric. Our approach determines a semantic class for every point while modeling object instances as probability distributions in the 4D spatio-temporal domain. We process multiple point clouds in parallel and resolve point-to-instance associations, effectively alleviating the need for explicit temporal data association. Inspired by recent advances in benchmarking of multi-object tracking, we propose to adopt a new evaluation metric that separates the semantic and point-to-instance association aspects of the task. With this work, we aim at paving the road for future developments of temporal LiDAR panoptic perception.

1. Introduction

Spatio-temporal interpretation of raw sensory data is important for autonomous vehicles to understand how to interact with the environment and perceive how trajectories of moving agents evolve in 3D space and time.

In the past, different aspects of dynamic scene understanding such as semantic segmentation [22, 18, 48, 72, 85, 69], object detection [23, 62, 40, 65, 64, 66], instance segmentation [28], and multi-object tracking [43, 8, 53, 10, 76, 61, 58] have been tackled independently. The developments in these fields were largely fueled by the rapid progress in deep learning-based image [37] and point-set representation learning [59, 60, 72], together with contributions of large-scale datasets, benchmarks, and unified evaluation metrics [44, 22, 25, 19, 17, 74, 26, 5, 18, 13, 68]. In the pursuit of image-based holistic scene understanding, recent community efforts have been moving towards convergence of tasks, such as multi-object tracking (MOT) and segmentation [74, 83], and semantic and instance segmentation, *i.e.*, panoptic segmentation [35]. Recently, panoptic segmentation was extended to the video domain [34]. Here, the dataset, task formalization, and evaluation metrics focused on interpreting short and sparsely labeled video snippets in 3D (2D image+time) in an offline setting. Autonomous vehicles, however, need to continuously interpret sensory data and localize objects in a 4D continuum.

Tackling sequence-level LiDAR panoptic segmentation is a challenging problem, since state-of-the-art methods [72] usually need to downsample even single-scan point clouds to satisfy the memory constraints. Therefore, the common approach in (3D) multi-object tracking is detect-

ing objects in individual scans, followed by temporal association [24, 76, 77], often guided by a hand-crafted motion model. In this paper, we take a substantially different approach, inspired by the unified space-time treatment philosophy. We form overlapping 4D volumes of scans (see Fig. 1) and, in parallel, assign to 4D points a semantic interpretation while grouping object instances jointly in 4D space-time.

Importantly, these 4D volumes can be processed in a single network pass, and the temporal association is resolved implicitly via clustering. This way, we retain inference efficiency while resolving long-term association between overlapping volumes based on the point overlap, alleviating the need for explicit data association.

For the evaluation, we introduce a point-centric higher-order tracking metric, inspired by recent metrics for multi-object tracking [45] and concurrent work on video panoptic segmentation [75] which differ from the available metrics [35, 9] that overemphasize the recognition part of the tasks. Our metric consist of two intuitive terms, one measuring the semantic aspect and second the spatio-temporal association of the task. Together with the recently proposed SemanticKITTI [5, 6] dataset, this gives us a test bed to analyze our method and compare it with existing LiDAR semantic/instance segmentation [40, 72, 76, 47] approaches, adapted to the sequence-level domain.

In summary, our **contributions** are: (i) we propose a unified space-time perspective to the task of 4D LiDAR panoptic segmentation, and pose detection/segmentation/tracking jointly as point clustering which can effectively leverage the sequential nature of the data and process several LiDAR scans while maintaining memory efficiency; (ii) we adopt a point-centric evaluation protocol that fairly weights semantic and association aspects of this task and summarizes the final performance with a single number; (iii) we establish a test bed for this task, which we use to thoroughly analyze our model's performance and the existing LiDAR panoptic segmentation methods used in conjunction with a tracking-by-detection mechanism. Our code, experimental data[1] and benchmark[2] are publicly available.

2. Related Work

Our work is related to tasks covering different aspects of scene perception, such as semantic segmentation, object detection/segmentation, and tracking. In the following, we review related methods and tasks.

Datasets and Metrics. The growing interest in autonomous vehicles has sparked interest in scene perception using LiDAR sensors. Here the progress has been fueled by advances in deep learning on point sets [59, 60, 31, 36, 38,

79, 80, 70, 72, 47] and datasets with standardized benchmarks for 3D semantic/instance segmentation [25, 5] and 3D object detection and multi-object tracking [13, 68]. This confirms the importance of advancing both spatial and temporal aspects of mobile robot perception. Our proposed task formulation and evaluation metric is the first that unifies both aspects to the best of our knowledge.

Recent community efforts in the field of image-based perception have been moving towards the convergence of different tasks. For instance, Kirillov *et al.* [35] proposed to unify semantic and instance segmentation, which they termed panoptic segmentation, together with an evaluation metric, the panoptic quality (PQ). Others proposed to tackle multi-object tracking and instance segmentation (MOTS) in videos jointly [74, 83]. Moreover, [34] recently extended panoptic segmentation to videos – however, the dataset and the evaluation metrics focus on interpreting short and sparse video snippets offline. This is reflected in the evaluation metric, which is essentially PQ evaluated based on the 3D IoU [83] and averaged over temporal windows of varying sizes to compensate that the difficulty of the task depends on the sequence length. This setting is not suitable for autonomous vehicles that need to interpret raw sensor data continuously. Hurtado *et al.* [32] proposes to combine ideas from MOTA [9] and PQ [35] by adding a penalty related to ID switches to the PQ. Nonetheless, both PQ and MOTA were criticized [57, 45], and the proposed evaluation inherits all of their well-known issues.

In this paper, we propose a different approach and bring ideas recently introduced in the context of benchmarking vision-based multi-object tracking [45] to the domain of sequential LiDAR semantic and instance segmentation. Together with the metric, we also propose an approach that operates directly on spatio-temporal point clouds providing object instances in space and time.

Point Cloud Segmentation. Semantic segmentation or point-wise classification of point clouds is a well-known research topic [2]. Traditionally, it was solved using feature extractors in combination with traditional classifiers [1] and conditional random fields to enforce label consistency of neighboring points [73, 50, 81]. Availability of large-scale datasets, such as S3DIS [3], Semantic3D [27], and recently SemanticKITTI [5], made it possible to also investigate end-to-end pipelines [39, 48, 72, 30, 85, 63, 60, 59, 47]. Similar to recent trends in RGB-D [33, 20] and LiDAR segmentation [78], our method performs bottom-up point grouping in a data-driven fashion. However, different from the aforementioned, we perform grouping in 3D space and time. We use the backbone by [72] that applies deformable point convolutions directly on the point clouds. In our case, this empirically performed better compared to backbones, specifically designed for point sequences [15, 63].

---

[1] https://github.com/mehmetaygun/4d-pls
[2] http://bit.ly/4d-panoptic-benchmark

Multi-Object Tracking and Segmentation. The majority of vision-based MOT methods follow tracking-by-detection [52]. Here the idea is to first run a pre-trained object detector independently in each video frame and then associate detections across time. In the past, there was a strong focus on developing robust and, preferably, globally optimal methods for data association [84, 42, 56, 11, 12]. Recent data-driven trends mainly focus on learning to associate detections [41, 67] or to regress targets [8], often in combination with end-to-end learning [16, 82, 10].

In the realm of robot vision, it is critical to localize object trajectories in 3D space and time. Early methods localized monocular detections in 3D *e.g.*, using stereo [43, 53, 21], or performed tracking in a category-agnostic manner by first performing bottom-up segmentation based on spatial proximity followed by point-segment association [71, 29]. Recently, LiDAR-based MOT became popular, thanks to the emergence of reliable 3D object detectors [65, 40] and LiDAR-centric datasets [13, 68]. Weng *et al.* [76] demonstrated that simple methods based on linear assignment and constant-velocity motion models can perform surprisingly well when 3D detections are localized reliably. Our method departs from 3D object detection in the spatial domain, followed by the detection association in the temporal domain. Instead, we follow recent advances in image [51, 14] and video instance segmentation [4]. We localize possible object instance centers within a 4D volume and associate points to estimated centers in a bottom-up manner, while a semantic branch assigns semantic classes to points.

3. Method

In this paper, we propose a method and a metric for 4D Panoptic LiDAR Segmentation task that tackles LiDAR semantic segmentation and instance segmentation jointly in the spatial and temporal domain. Given a sequence of LiDAR scans, the goal of this task is to predict for each 3D point (i) a semantic label for both *stuff* and *thing* classes, and (ii) a unique, identity-preserving object instance ID that should persist over the whole sequence.

3.1. 4D Panoptic LiDAR Segmentation: 4D-PLS

In this work, we take a different path compared to the tracking-by-detection paradigm to video-instance and video-panoptic segmentation [74, 83, 34, 32]. We pose 4D panoptic segmentation as two joint processes. The first one is responsible for point grouping in the 4D continuum using clustering, while the second assigns a semantic interpretation to each point.

We provide an overview of our method in Fig. 2. In a nutshell, we first form 4D point clouds from several consecutive LiDAR scans. In parallel, within a single network pass, we localize the most likely object centers (inspired by point-based tracking methods by [86, 14]) in the sequence
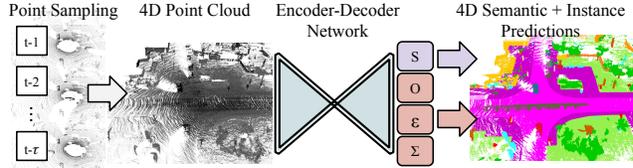

Figure 2: Visualization of our method. We sample points from past scans to form a 4D point cloud. Our encoder-decoder network estimates point objectness map (O), point variance map (Σ), and point embeddings (ε). We use these maps to assign points to their respective instances via density-based clustering in a 4D continuum. We obtain semantic interpretation from the semantic decoder (S).

(objectness map O), assign semantic classes to points (semantic map S), and compute per-point embeddings (embedding map ε) and variances (variance map Σ).

The clustering can be performed efficiently by evaluating the probability of each 4D point belonging to a certain "seed" point, which is similarly performed in the context of images and video segmentation [51, 4]. Finally, to associate 4D sub-volumes, we examine point intersections between overlapping point volumes.

4D Volume Formation. During inference and training, we form overlapping 4D point cloud volumes in an online setting. In particular, for scan t and temporal window size τ, we align together point clouds within temporal window $\{\max(0, t - \tau), ..., t\}$ using ego-motion estimates provided by a SLAM approach [7]. Our experiments in Sec. 4.1 reveal that processing multiple point clouds significantly improves spatial and temporal point association performance. However, due to the linear growth in memory requirements, stacking point clouds along the temporal dimension is prohibitively expensive. To overcome this issue, we build on the intuition that *thing* classes are most critical for a stable temporal association, since these classes correspond to potentially moving objects. As we operate in an online setting, where the past scans have already been processed, we can sample points that belong to *thing* classes or lie near to an object centers from earlier scans.

Density-based Clustering. We model object instances via Gaussian probability distributions. Given an estimate of the object center, *i.e.*, clustering "seed" point, we can assign points to their respective instance by evaluating each point under the Gaussian pdf based on the point's embedding vectors. The estimated centers do not need to correspond to exact object centers but are merely used to initiate the clustering. Thus, our approach is in practice fairly robust to occlusions and cross-time view changes. We note that the Gaussian assumption is only valid for shorter temporal windows. In particular, given a point $\mathbf{p}_i$ representing the instance center and its embedding vector e_i, and a query point p_j with its embedding vector e_j, we can evaluate the

probability of point p_j belonging to its center "seed" point p_i as:

$$\hat{p}_{ij} = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(e_i - e_j)^\top \Sigma_i^{-1}(e_i - e_j)\right) \quad (1)$$

where Σ_i is a diagonal matrix constructed using variance prediction σ_i of point p_i. We concatenate coordinate values (x, y, z, t) with the learned point embedding vectors to combine spatial and temporal coordinates with learned embeddings. We account for these additional dimensions during the training of the variance map.

Network and Training. To perform such clustering, we need to identify most likely instance centers, *i.e.*, "seed" points, in a 4D point cloud. We also need variance predictions for each point to evaluate probability scores during clustering, and a posterior over all semantic classes.

We estimate all these quantities using an encoder-decoder architecture that operates directly on the 4D point cloud $P \in \mathbb{R}^{N \times 4}$. The encoder network is based on the KP-Conv [72] backbone that uses deformable point convolutions. The decoder predicts point-wise feature embeddings $\varepsilon \in \mathbb{R}^{N \times D}$ using consecutive point convolutions. On top of the encoder, we add an object centerness decoder in $\mathbb{R}^{N \times 1}$, point variance decoder in $\mathbb{R}^{N \times D}$, and semantic decoder in $\mathbb{R}^{N \times C}$. We train our network in an end-to-end manner and in online fashion.

To train the semantic decoder, we use cross-entropy classification loss L_{class}. As the semantic classes are highly imbalanced, we sample points to ensure that the probability of sampling a point from a certain class is roughly uniform.

To learn the point centerness and point variance, we use three different losses. First, we impose the mean squared error (MSE) loss to train the object centerness decoder. However, there will generally be no points near the actual object centers, unlike the image and video domain [51, 4]. Therefore, instead of predicting per-point centerness, we predict the proximity of the point to its instance center. We compute for each point $\mathbf{p}_i$ its objectness o_i as Euclidean distance between the point and its instance center, *i.e.*, mean point of all instance points, normalized to $[0, 1]$. This objectness o_i is then compared to the regressed objectness $\hat{o}_i$:

$$L_{\text{obj}} = \sum_{i=1}^{N} (\hat{o}_i - o_i)^2, \quad \hat{o}_i, o_i \in [0, 1] \quad (2)$$

Since we want the embeddings of instances to form clusters in the spatio-temporal domain, we introduce our instance loss. Given a 4D point cloud of N points and K instances, it is defined as:

$$L_{\text{ins}} = \sum_{j=1}^{K} \sum_{i=1}^{N} (\hat{p}_{ij} - p_{ij})^2, p_{ij} = \begin{cases} 1, & \text{if } \mathbf{p}_i \in I_j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $\hat{p}_{ij}$ is evaluated under the Gaussian pdf (Eq. 1) with points embedding e_i as well as instance embedding and variance e_j and σ_j. In addition, we employ variance smoothness loss L_{var}, similar to [4, 51] for training the variance decoder. In summary, we use four different losses to train our network in end-to-end manner: $L = L_{\text{class}} + L_{\text{obj}} + L_{\text{ins}} + L_{\text{var}}$.

Inference. We resolve point-to-instance associations in two stages, first within a processed 4D volume, and then across volumes. First, based on the point cloud centerness map, we select the point $\mathbf{p}_i$, which has the highest objectness score. Then, we evaluate probabilities p_{ij} for all candidate points and assign them to the cluster in case $p_{ij} > 0.5$. The assigned points are then removed from the candidate pool. We repeat these steps until the next highest objectness score is below a certain threshold. To transfer identities across processed 4D volumes, we perform cross-volume association greedily based on the overlap score, taking all scans into account. When the overlap is below a threshold, we assign a new id.

3.2. Measuring Performance

The central question when proposing a novel task and benchmark is how to evaluate and compare different methods. Preferably, we would like to summarize performance with a single number to rank the methods while retaining the capability of looking at different aspects of the task.

3.2.1 Existing Evaluation Measures

To motivate our approach to evaluation, we first briefly discuss established metrics for image-based panoptic segmentation (PQ [35]) and multi-object tracking and segmentation (MOTSA/MOTSP [9, 74]). Then, we discuss two recently proposed extensions of PQ to the temporal domain and argue why we do not promote their adaptation for the task of 4D LiDAR panoptic segmentation.

Segment-centric Evaluation. PQ and MOTSA/MOTSP are instance-centric evaluation metrics. Both first determine a unique matching between sets of ground-truth objects and model predictions for each frame individually to determine true positives (TPs), false positives (FPs), and false negatives (FNs). Both metrics provide measures for the segmentation and recognition aspects of the task. The segmentation quality (SQ) term of PQ and MOTSP integrates IoUs over the set TPs and normalizes it by the size of the TP set. The recognition quality (RQ) term of PQ is expressed as the F_1 score. Similarly, MOTSA combines detection errors (FNs and FPs) with ID switch (IDSW) penalty in a single term. IDSW occurs when a track is lost, and the tracker assigns a new identity to a tracked object. This is the only term that takes the temporal aspect of the task into account.

A criticism of PQ is that it over-emphasizes the importance of very small segments and *stuff* classes can be difficult to match [57]. MOTSA overemphasizes the detection compared to the association aspect and it is nonintuitive, since the score can be negative and is unbounded, as can be seen in Sec. 4. Furthermore, the influence of the ID switches on the final score depends on the frame rate, and MOTSA does not reward trackers that recover from incorrect associations. Importantly, both metrics are sensitive to the selection of the matching threshold. Thus, instances that slightly miss this threshold will cause both a FN and a FP. This is not the case for pixel or point centric metrics used for evaluating semantic segmentation. The standard mean IoU (mIoU) metric [22] computes sets of TPs, FPs and FNs pixel (or point) basis, effectively bypassing the segment matching.

PQ Extensions. Recent work [34] proposes video panoptic quality, a variant of PQ for the sequential domain. Different from PQ, gt-to-prediction mapping is established based on the sequential IoU matching criterion [83]. As objects are not present throughout the clip and the difficulty of the task critically depends on the length of the temporal window, the final metric is averaged over varying temporal window sizes. This is suitable for the setting defined in [34], where the task is to evaluate short, sparsely labeled video snippets. However, this approach does not scale to real-world sequences of arbitrary length. Another extension to PQ, panoptic tracking quality (PTQ) [32] combines MOTA and PQ by adding an ID penalty to the PQ measure. This approach inherits issues from both PQ and MOTSA metrics.

3.2.2 LiDAR Segmentation and Tracking Quality

In the following, we assume a sequence of 3D point clouds of length l, sampled at discrete time-steps: $\Omega = \{(\mathbf{p}, n) \in \mathbb{R}^3 \times \mathbb{N} | n < l\}$. We define the ground-truth assignment function as $gt(\mathbf{p}, n) \rightarrow (c, id)$ and a prediction function as $pr(\mathbf{p}, n) \rightarrow (c, id)$, that map each 4D tuple, consisting of a point $\mathbf{p}$ and a timestamp n, to a certain class c and identity id. In the following, we devise an evaluation metric that, for each pair $(\mathbf{p}, n)$, evaluates whether (i) it was assigned to a correct class, and (ii) for the *thing* classes, whether it was assigned to the correct object instance. Inspired by the recently introduced Higher Order Tracking Accuracy (HOTA) [45], proposed in the context of MOT, and concurrent work on video panoptic segmentation proposing the Segmentation and Tracking Quality (STQ) [75], our *LSTQ* (LiDAR Segmentation and Tracking Quality) consists of two terms, the classification score S_{cls} and the association score S_{assoc}.

We adopt a fundamentally different evaluation philosophy compared to other metrics [45, 35, 34, 32]. In particular, we drop the concept of the frame-level "detection" and do not match segments between ground-truth and predic-

tion. Our association score measures point-to-instance association quality in a unified way – in space *and* time at point level, which is more natural for segmentation tasks.

Classification Score. For the classification score, we first define instance-agnostic ground-truth and predictions sets:

$$gt_{\text{agn}}(c) = \{(\mathbf{p}, n) \mid gt(\mathbf{p}, n) = (c, *)\},$$
$$pr_{\text{agn}}(c) = \{(\mathbf{p}, n) \mid pr(\mathbf{p}, n) = (c, *)\},$$

representing the ground truth and predicted points that belong to class c irrespective of their assigned ids. Then, the TP, FP, FN sets are computed as in semantic segmentation evaluation with respect to gt class c and predicted class c':

$$\text{TP}_c = |pr_{\text{agn}}(c) \cap gt_{\text{agn}}(c)|,$$
$$\text{FP}_c = |pr_{\text{agn}}(c) - pr_{\text{agn}}(c) \cap gt_{\text{agn}}(c)|,$$
$$\text{FN}_c = |gt_{\text{agn}}(c) - pr_{\text{agn}}(c) \cap gt_{\text{agn}}(c)|.$$

The classification score then simply boils down to intersection-over-union (IoU) over these sets, which is the standard approach for evaluating semantic segmentation (however, this is different from segment-centric PQ, where points contribute to the TP_c term only if the segment that they belong is matched). We follow the standard procedure and average over the classes:

$$S_{cls} = \frac{1}{|\mathcal{C}|} \sum_{c=1}^{\mathcal{C}} \frac{|\text{TP}_c|}{|\text{TP}_c| + |\text{FN}_c| + |\text{FP}_c|} = \frac{1}{C} \sum_{c=1}^{\mathcal{C}} IoU(c).$$

Association Score. To evaluate the association score, we introduce the following class-agnostic predictions and ground-truth for the *thing* classes:

$$gt_{id}(id) = \{(\mathbf{p}, n) \mid gt(\mathbf{p}, n) = (c, id), c \in things\},$$
$$pr_{id}(id) = \{(\mathbf{p}, n) \mid pr(\mathbf{p}, n) = (c, id), c \in things\}.$$

We define the true positive association (*TPA*) set between a ground-truth object t with identity id and prediction s, that was assigned identity id'. This gives us a set of points with mutually consistent identities id and id':

$$TPA(id, id') = |pr_{id}(id') \cap gt_{id}(id)|. \quad (4)$$

Similarly, we define the set of false positive associations:

$$FPA(id, id') = |pr_{id}(id') - pr_{id}(id') \cap gt_{id}(id)|. \quad (5)$$

Intuitively, this set contains predicted point assignments with identity id', that were assigned a different ground-truth identity ($\neq id$), or were not assigned to a valid object instance. Finally, the set of false negative assignments:

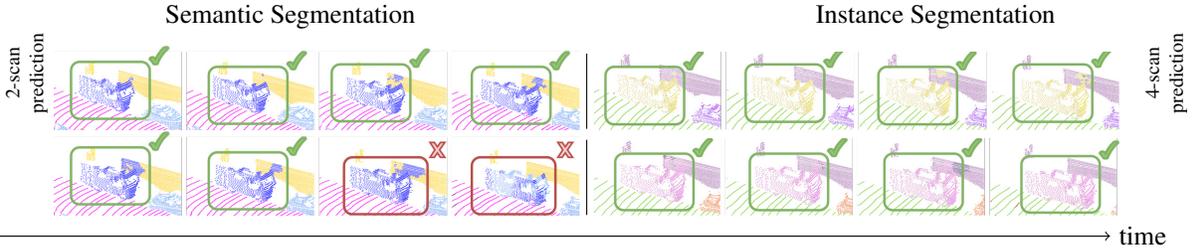$$FNA(id, id') = |gt_{id}(id) - pr_{id}(id') \cap gt_{id}(id)| \quad (6)$$

Figure 3: Predictions from 2 and 4 scan versions, MOTSA=$1.0/0.0$, S_{assoc}=$0.96/0.93$, S_{cls}=$0.86/0.75$. While both models track the instance correctly, due to slight difference in semantic segmentation predictions, MOTSA scores differ drastically.

contains ground-truth points with identity id that were assigned an identity, different to id', or were missed. We note that the concept of TPA, FPA and FNA was first introduced in the context of MOT evaluation for measuring the quality of temporal detection association. Therefore, to establish these sets, a bijective mapping between gt and pred needs to be established (as in the case of [9]). However, in *LSTQ*, these sets are established with respect to each 4D point, treating association in space and time in a unified manner.

Once we have quantified these sets, we can evaluate how well a predicted segment *s agrees* with ground-truth segment t. Because a ground truth segment t may be explained by multiple different predictions, we sum contributions of all pairs with non-zero overlap:

$$S_{assoc} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{|gt_{id}(t)|} \sum_{\substack{s \in S \\ s \cap t \neq 0}} TPA(s,t) IoU(s,t), \quad (7)$$

where the IoU term is evaluated using the TPA, FNA and FPA sets (Eq. 4, 6, 5). In practice, we do not need to perform any point segment association, and even a prediction with a single common point will contribute to this term. We normalize these contributions by the tube volume, and we weigh each contribution by the volume of the TPA set. This weighting term ensures that instances with larger temporal spans have a higher contribution to the final score. Finally, our metric is computed as a geometric mean of the two terms: $LSTQ = \sqrt{S_{cls} \times S_{assoc}}$. The advantage over the arithmetic mean is that the final score will become zero if any of the two terms approach zero. This reflects our intuition that failing at either of two aspects of the task should yield a very low final score.

LSTQ tolerates different semantic predictions within a spatio-temporal segment by design (the IoU term in Eq. 7 is evaluated in a class-agnostic manner). Following STQ [75], we *decouple semantic and association errors*, otherwise, *e.g.*, a truck mistaken for a bus would be harshly penalized by the association term, even though it was tracked correctly. This behavior that dis-entangles association and classification errors is different from MOTSA/PTQ/VPQ where semantics and temporal association are entangled.

4. Experimental Evaluation

In this section, we first evaluate different strategies for forming 4D point cloud volumes, assess the impact of processing multiple scans on the final performance, and discuss several possibilities for the embeddings used for point grouping. We compare our method to baselines for single-scan LiDAR panoptic segmentation [6] and 4D panoptic LiDAR segmentation by extending existing methods.

We use the SemanticKITTI [5] LiDAR dataset to conduct our experiments. It contains 22 sequences from KITTI odometry dataset [25] and provides point-wise semantic and temporally consistent instance annotations [6]. We use the training/validation/test split by SemanticKITTI [5, 6].

4.1. Ablation studies

We perform all ablations on the validation set and interpret results through the lens of *LSTQ* metric (Sec. 3.2.2).

Point Propagation. As discussed in Sec. 3, we cannot simply stack point clouds temporally due to the memory constraints. We build on the intuition that we can subsample a set of points from the past scans that are most beneficial for the end-task performance. As we are operating in an online setting, and the past scans have been already processed, we can leverage past predictions.

In this experiment, we discuss different temporal point sampling strategies for varying temporal window sizes of $\tau = 2, 4, 6$. In the *thing-propagation* strategy, we exclusively sample points which are assigned only to a thing class, as they represent only a small number of all point. In the *importance sampling* strategy, we sample 10% of points with a probability proportional to the objectness. This way, we focus on points likely to represent *thing* classes, while still allowing to propagate points belonging to *stuff* classes, which can aid the semantic segmentation of the task. Similarly, the *temporal decay* sampling uses the objectness score as a deciding factor, but we decay the number of sampled points based on the proximity to the current scan. Finally, the *strided sampling* samples points with a stride of 2 along the temporal dimension.

As can be seen in Tab. 1, the *importance sampling* strategy yields higher performance compared to sampling only

Strategy	# fr.	LSTQ	S_{assoc}	S_{cls}	IoU^{St}	IoU^{Th}	Mem.
Base	1	51.92	45.16	59.69	64.60	60.40	1x
Thing-prop.	2	59.20	58.71	59.69	64.04	61.16	1.05x
	4	60.94	63.95	57.84	63.95	56.67	1.15x
	6	58.88	61.13	56.71	63.86	53.97	1.25x
Importance samp.	2	59.86	58.79	60.95	64.96	63.06	1.1x
	4	**62.74**	**65.11**	60.46	65.36	61.26	1.3x
	6	61.52	64.28	58.88	65.32	57.38	1.5x
Temporal decay	2	59.86	58.79	60.95	64.96	63.06	1.1x
	4	62.14	64.05	60.30	65.33	60.91	1.3x
	6	61.03	63.34	58.81	65.38	57.11	1.5x
Temporal stride	2	62.29	63.52	**61.08**	65.10	**63.18**	1.1x
	4	61.63	62.89	60.39	**65.45**	60.39	1.3x
	6	59.33	59.95	58.72	65.39	56.88	1.5x

Table 1: Ablation study on point sampling strategies for building 4D point cloud volumes with respect to different temporal window sizes.

Mixing	# Sc.	LSTQ	S_{assoc}	S_{cls}	IoU^{St}	IoU^{Th}
xyz	2	51.65	43.77	60.95	64.96	63.06
$xyzt$	2	51.95	44.30	60.93	64.80	63.15
Emb.	2	48.58	43.63	54.10	59.54	50.41
Emb. + xyz	2	54.15	56.63	55.11	61.14	53.71
Emb. + $xyzt$	2	59.86	58.79	60.95	64.96	63.06
xyz	4	54.29	48.77	60.44	65.29	61.32
$xyzt$	4	54.46	49.55	59.87	64.80	61.15
Emb.	4	56.77	60.63	53.17	58.00	52.25
Emb. + xyz	4	58.43	63.90	54.68	61.12	52.67
Emb. + $xyzt$	4	62.74	65.11	60.46	65.36	61.26

Table 2: Embedding design ablation.

Method	PQ	$PQ^{\dagger}$	SQ	RQ	mIoU
RangeNet++ [48] + PointPillars [40]	37.1	45.9	75.9	47.0	52.4
KPConv [72] + PointPillars [40]	44.5	52.5	80.0	54.4	58.8
Panoptic RangeNet [47]	38.0	47.0	76.5	48.2	50.9
Our method (single scan)	**50.3**	**57.8**	**81.6**	**61.0**	**61.3**

Table 3: Single scan Panoptic Segmentation (test set).

Method	LSTQ	S_{assoc}	S_{cls}	IoU^{St}	IoU^{Th}
RangeNet++[48] + PP + MOT	35.52	24.06	52.43	64.52	35.82
KPConv [72] + PP + MOT	38.01	25.86	55.86	66.90	47.66
RangeNet++[48] + PP + SFP	34.91	23.25	52.43	64.52	35.82
KPConv [72] + PP + SFP	38.53	26.58	55.86	66.90	47.66
Our (single scan) + MOT	40.18	28.07	57.51	66.95	51.50
Our (single scan) + SFP	43.88	33.48	57.51	66.95	51.50
Ours (multi scan)	56.89	56.36	57.43	66.86	51.64

Table 4: 4D Panoptic (test set). MOT – *tracking-by-detection* by [76]; SFP – *tracking-by-detection* via scene flow based propagation [49]; PP – PointPillars [40].

Next, we investigate the performance of the coordinate mixing formulation that combines learned embeddings with 3D spatial and 4D spatio-temporal coordinates. As can be seen, the variant in which we combine both yields the best results, not only in terms of S_{assoc}, but also S_{cls}. This shows that a well-designed embedding branch has a positive effect on learning the backbone features. Note that for the baseline that uses only spatio-temporal coordinates, we still use our fully trained network.

4.2. Benchmark Results

Single-scan Prediction. First, we evaluate our method using single-scan LiDAR panoptic segmentation [5, 6] to demonstrate the effectiveness of our network solely in the spatial domain. We use points from single scans during training and testing. We follow the standard evaluation protocol and compare to published and peer-reviewed methods.

As can be seen from Tab. 3, our method achieves state-of-the-art results on all metrics for semantic and panoptic segmentation [35, 5, 6]. The first two entries use two different networks for object detection and semantic segmentation, followed by fusion of the results. We use a single network to obtain both semantic and instance segmentation of the point cloud in a single network pass. We note that the recently proposed Panoptic RangeNet [47] and RangeNet++ [48] combined with PointPillars [40] detector operate on the range image and not the point cloud, and therefore, use a different backbone. However, KPConv with PointPillars uses the same backbone as our method.

4D Panoptic Segmentation. For evaluation in the multi-scan setting for the 4D panoptic segmentation task, we extend all single-scan methods reported in Tab. 3, except for the Panoptic RangeNet [47]. We adapt them to the sequential domain using two strategies. AB3DMOT [76] uses a constant-velocity motion model to obtain track predictions associated with object detection based on a 3D bounding

thing classes, at a slightly increased memory cost. As expected, this approach improves association quality and aids semantics as it also propagates points representing *stuff*.

Interestingly, even a temporal window of size 2 drastically improves the performance compared to a single scan baseline, at negligible memory consumption ($1.1\times$). We observe the largest gains when the scans are temporally close: our 4-scan multi-scan baseline improves performance $51.92 \rightarrow 62.74$ in terms of *LSTQ*. The association term benefits more from processing multiple scans compared to the segmentation term. This confirms that our model learns to exploits temporal cues very well. While temporal decaying does aid semantic or temporal aspect, introducing a temporal stride of 2 yields the highest performance gains for semantic point classification. However, denser sampling in the temporal domain benefits association. Hence, we focus on the importance sampling strategy with $\tau = 4$. In the supplementary, we highlight the performance with temporal window size $\tau = 1, 2, 3, 4, 6, 8$. As can be seen, the association accuracy is increasing up to $\tau = 4$ and then saturates, while classification accuracy saturates a $\tau = 2$; however, it only decreases marginally.

Embedding Design. In this experiment, we study different point embeddings for clustering and show our findings in Tab. 2. We investigate the base performance of using only 3D spatial (xyz) and 4D spatio-temporal point coordinates ($xyzt$), and using only learned embeddings (Emb.).

Method	$LSTQ$	S_{assoc}	S_{cls}	IoU^{St}	IoU^{Th}	sPTQ	PTQ	sMOTSA	MOTSA
RangeNet++[48] + PP + MOT	43.76	36.28	52.78	60.49	42.17	34.58	33.83	-7.88	-4.57
KPConv [72] + PP + MOT	46.27	37.58	56.97	64.21	54.13	39.13	38.11	-6.16	-2.41
RangeNet++[48] + PP + SFP	43.38	35.66	52.78	60.49	42.17	35.83	35.46	-3.13	-0.01
KPConv [72] + PP + SFP	45.95	37.07	56.97	64.21	54.13	41.44	41.05	2.83	6.1
MOPT [32]	24.80	11.73	52.41	62.37	45.27	41.82	42.39	12.88	17.07
Our (single scan) + MOT	51.92	45.16	59.69	64.60	60.40	48.36	47.84	6.65	12.69
Our (single scan) + SFP	45.45	34.61	59.69	64.60	60.40	48.24	47.72	3.01	7.93
Ours (2 scans)	59.86	58.79	**60.95**	64.96	**63.06**	51.14	50.67	**29.04**	**33.2**
Ours (4 scans)	**62.74**	**65.11**	60.46	**65.36**	61.26	**51.50**	**51.20**	0.34	4.8

Table 5: 4D Panoptic (validation set). MOT – *tracking-by-detection* method by [76]; SFP – *tracking-by-detection* via scene flow based segment propagation [49]; PP – PointPillars [40] detector.

Category	# Instances	% Instances	# Scans	TP	FP	FN	IDS	Precision	Recall	MOTSA	S_{assoc}	S_{cls}
Motorcycle	255	0.01	2	209	151	46	31	0.58	0.82	0.11	0.56	0.88
			4	231	747	24	9	0.24	0.91	-2.06	0.81	0.74
Other-vehicle	2138	0.06	2	778	362	1360	162	0.68	0.36	0.12	0.17	0.56
			4	1022	1131	1116	99	0.47	0.48	-0.10	0.38	0.55

Table 6: Per-class evaluation on SemanticKITTI validation set (2 and 4 scan versions).

box overlap. The second strategy, Scene Flow Propagation (SFP) is inspired by standard baselines that perform optical flow warping of points, followed by mask-IoU based association. This approach is commonly used in vision-based video object segmentation [46], video instance segmentation [83], and multi-object tracking and segmentation [54, 55, 74]. Instead of optical flow, we use state-of-the-art LiDAR scene flow [49]. We outline our results, obtained on the test set in Tab. 4. As can be seen, the baseline that uses KPConv [72] to obtain per-pixel classification, PointPillars (PP) detector [40] and a network for point cloud propagation (SFP [49]) performs slightly better in terms of association accuracy than the standard 3D MOT baseline. Our method that unifies all three aspects in a single network outperforms all tracking-by-detection baselines by a large margin, including our single-scan baseline. This confirms the importance of tackling all three aspects of these tasks in a unified manner. An important contribution of our paper is the finding that even when processing smaller overlapping sub-sequences with our network (and resolving intra-window associations with a simple overlap-based approach), we perform significantly better compared to single-scan baselines that use more elaborate association techniques (*e.g.*, Kalman filter), as can be confirmed in Tab. 4.

Metric Insights. In this section, we analyze the performance on the validation split (Tab. 5) through the lens of several evaluation metrics and analyze per-class performance (Tab. 6). Our method outperforms all baselines with respect to all metrics. However, while our 4-scan variant performs better than the 2-scan variant in terms of *LSTQ*, we observe a significant drop in the MOTSA score. Our analysis shows that this is caused by negative MOTSA scores on some classes due to a decrease in precision while having fewer ID switches (see Tab. 6, and supplementary). We visualize such case in Fig. 3. As can be seen, the difference is due to the semantic interpretation of the points and not due to the segmentation and tracking quality at the instance level. This confirms the nonintuitive behavior of MOTSA, while our metric provides insights on both semantic interpretation and instance segmentation and tracking. For a more details, we refer to the supplementary.

5. Conclusion

In this paper, we extended LiDAR panoptic segmentation to the temporal domain resulting in the 4D Panoptic Segmentation task. We presented an evaluation metric suitable for analyzing this task's performance and proposed a new model. Importantly, we have shown that a single model tackling semantic segmentation and point-to-instance association jointly in space and time substantially outperforms methods that independently tackle these aspects. We hope that our unified view and model, accompanied by a public benchmark, will pave the road to future developments.

References

[1] Anuraag Agrawal, Atsushi Nakazawa, and Haruo Takemura. MMM-classification of 3D Range Data. In *ICRA*, 2009. 2

[2] Dragomir Anguelov, Ben Taskar, Vassil Chatalbashev, Daphne Koller, Dinkar Gupta, Geremy Heitz, and Andrew Ng. Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. In *CVPR*, 2005. 2

[3] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *CVPR*, 2016. 2

[4] Ali Athar, Sabarinath Mahadevan, Aljoša Ošep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020. 3, 4

[5] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, 2019. 1, 2, 6, 7

[6] Jens Behley, Andres Milioto, and Cyrill Stachniss. A benchmark for LiDAR-based panoptic segmentation based on KITTI. In *ICRA*, 2021. 2, 6, 7

[7] Jens Behley and Cyrill Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *RSS*, 2018. 3

[8] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *ICCV*, 2019. 1, 3

[9] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *JIVP*, 2008:1:1–1:10, 2008. 2, 4, 6

[10] Guillem Braso and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, June 2020. 1, 3

[11] William Brendel, Mohamed R. Amer, and Sinisa Todorovic. Multi object tracking as maximum weight independent set. *CVPR*, 2011. 3

[12] Asad A. Butt and Robert T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *CVPR*, June 2013. 3

[13] 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. 1, 2, 3

[14] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020. 3

[15] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 2

[16] Peng Chu and Haibin Ling. FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *ICCV*, 2019. 3

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1

[18] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 1

[19] Patrick Dendorfer, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, and Stefan Roth Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *IJCV*, pages 1–37, 2020. 1

[20] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi Proposal Aggregation for 3D Semantic Instance Segmentation. In *CVPR*, 2020. 2

[21] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. Robust multiperson tracking from a mobile platform. *PAMI*, 31(10):1831–1846, 2009. 3

[22] Mark Everingham, Luc van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, 2010. 1, 5

[23] Pedro Felzenszwalb, David Mcallester, and Deva Ramanan. A Discriminatively Trained, Multiscale, Deformable Part Model. In *CVPR*, 2008. 1

[24] Davi Frossard and Raquel Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. *ICRA*, 2018. 2

[25] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 1, 2, 6

[26] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1

[27] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D. Wegner, Konrad Schindler, and Marc Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS*, 2017. 2

[28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1

[29] David Held, Jesse Levinson, Sebastian Thrun, and Silvio Savarese. Combining 3d shape, color, and motion for robust anytime tracking. In *RSS*, 2014. 3

[30] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, 2020. 2

[31] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *CVPR*, 2018. 2

[32] Juana Valeria Hurtado, Rohit Mohan, and Abhinav Valada. MOPT: Multi-object panoptic tracking. In *CVPR Workshops*, 2020. 2, 3, 5, 8

[33] 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. 2

[34] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *CVPR*, 2020. 1, 2, 3, 5

[35] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 1, 2, 4, 5, 7

[36] Artem Komarichev, Zichun Zhong, and Jing Hua. A-cnn: Annularly convolutional neural networks on point clouds. In *CVPR*, 2019. 2

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[38] Shiyi Lan, Ruichi Yu, Gang Yu, and Larry S Davis. Modeling local geometric structure of 3d point clouds using geo-cnn. In *CVPR*, 2019. 2

[39] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018. 2

[40] 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. 1, 2, 3, 7, 8

[41] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. *CVPR Workshops*, 2016. 3

[42] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *ICCV Workshops*, 2011. 3

[43] Bastian Leibe, Konrad Schindler, Nico Cornelis, and Luc Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *PAMI*, 30(10):1683–1698, 2008. 1, 3

[44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[45] Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *IJCV*, 2020. 2, 5

[46] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018. 8

[47] Andres Milioto, Jens Behley, Chris McCool, and Cyrill Stachniss. Lidar panoptic segmentation for autonomous driving. In *IROS*, 2020. 2, 7

[48] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *IROS*, 2019. 1, 2, 7, 8

[49] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. In *CVPR*, 2020. 7, 8

[50] Daniel Munoz, J. Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual Classification with Functional Max-Margin Markov Networks. In *CVPR*, 2009. 2

[51] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *CVPR*, 2019. 3, 4

[52] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004. 3

[53] Aljoša Ošep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. Combined image- and world-space tracking in traffic scenes. In *ICRA*, 2017. 1, 3

[54] Aljoša Ošep, Wolfgang Mehner, Paul Voigtlaender, and Bastian Leibe. Track, then decide: Category-agnostic vision-based multi-object tracking. *ICRA*, 2018. 8

[55] Aljoša Ošep, Paul Voigtlaender, Mark Weber, Jonathon Luiten, and Bastian Leibe. 4d generic video object proposals. *ICRA*, 2020. 8

[56] Hamed Pirsiavash, Deva Ramanan, and Charles C.Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 3

[57] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *CVPR*, 2019. 2, 5

[58] Johannes Pöschmann, Tim Pfeifer, and Peter Protzel. Factor graph based 3d multi-object tracking in point clouds. *arXiv preprint arXiv:2008.05309*, 2020. 1

[59] 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. 1, 2

[60] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 1, 2

[61] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2b: Point-to-box network for 3d object tracking in point clouds. In *CVPR*, 2020. 1

[62] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1

[63] Hanyu Shi, Guosheng Lin, Hao Wang, Tzu-Yi Hung, and Zhenhua Wang. Spsequencenet: Semantic segmentation network on 4d point clouds. In *CVPR*, 2020. 2

[64] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 2020. 1

[65] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointR-CNN: 3D Object Proposal Generation and Detection From Point Cloud. In *CVPR*, 2019. 1, 3

[66] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *PAMI*, 2020. 1

[67] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *CVPR*, 2017. 3

[68] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1, 2, 3

[69] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Ar-

chitectures with Sparse Point-Voxel Convolution. In *ECCV*, 2020. 1

[70] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *CVPR*, 2018. 2

[71] Alex Teichman and Sebastian Thrun. Tracking-based semi-supervised learning. *IJRR*, 31(7):804–818, 2012. 3

[72] 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 *ICCV*, 2019. 1, 2, 4, 7, 8

[73] Rudolph Triebel, Krisitian Kersting, and Wolfram Burgard. Robust 3D Scan Point Classification using Associative Markov Networks. In *ICRA*, 2006. 2

[74] Paul Voigtlaender, Michael Krause, Aljoša Ošep, Jonathon Luiten, B.B.G Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *CVPR*, 2019. 1, 2, 3, 4, 8

[75] Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, Aljoša Ošep, Laura Leal-Taixé, and Liang-Chieh Chen. STEP: Segmenting and Tracking Every Pixel. *arXiv preprint arXiv:2102.11859*, 2021. 2, 5, 6

[76] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. *IROS*, 2020. 1, 2, 3, 7, 8

[77] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris Kitani. GNN3DMOT: Graph neural network for 3d multi-object tracking with multi-feature learning. In *CVPR*, 2020. 2

[78] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *CoRL*, 2020. 2

[79] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, 2018. 2

[80] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, 2019. 2

[81] Xuehan Xiong, Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. 3-D Scene Analysis via Sequenced Predictions over Points and Regions. In *ICRA*, 2011. 2

[82] Yihong Xu, Aljoša Ošep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *CVPR*, 2020. 3

[83] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1, 2, 3, 5, 8

[84] Li Zhang, Li Yuan, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 3

[85] 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. 1, 2

[86] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, 2020. 3