

A Point-Based Approach to Efficient LiDAR Multi-Task Perception

Christopher Lang^{1,2}, Alexander Braun¹, Lars Schillingmann¹, Abhinav Valada²

Abstract—Multi-task perception networks hold great potential as they can improve performance and computational efficiency compared to their single-task counterparts, facilitating online deployment. However, current multi-task architectures in point cloud perception combine multiple task-specific point cloud representations, each requiring a separate feature encoder, making the network significantly large and slow. In this work, we propose PAttFormer, an efficient multi-task learning architecture for joint semantic segmentation and object detection in point clouds, only relying on a point-based representation. The network builds on transformer-based feature encoders using neighborhood attention and grid-pooling, complemented with a query-based detection decoder using a novel 3D deformable-attention detection head topology. Unlike other LiDAR-based multi-task architectures, our proposed PAttFormer does not require separate feature encoders for multiple task-specific point cloud representations, resulting in a network that is $3\times$ smaller and $1.4\times$ faster while achieving competitive performance on the nuScenes and KITTI benchmarks for autonomous driving perception. We perform extensive evaluations that show substantial improvement from multi-task learning, achieving $+1.7\%$ in mIoU for LiDAR semantic segmentation and $+1.7\%$ in mAP for 3D object detection on the nuScenes benchmark compared to the single-task models.

I. INTRODUCTION

Predicting object-level bounding boxes and point-level semantic labels from LiDAR-based point clouds is a vital part of many autonomous driving systems, enabling them to navigate safely around objects and determine drivable areas. Combining these two perception tasks in a single multi-task network architecture promises substantial benefits by sharing and jointly training parameters, including generalized latent representations that are less prone to overfitting and reduced latencies in the perception pipeline. However, learning the dense segmentation and sparse detection tasks in a multi-task setting can also lead to competition for network parameters, resulting in unbalanced performances on single tasks. While the research on point cloud perception has consistently shown performance gains from multi-task learning [1]–[3], the aforementioned challenges seem more dominant in the vision domain, where single-task networks often outperform their multi-task counterparts [4]–[7]. However, it remains unclear which characteristic of point cloud perception contributes to the performance improvement observed in multi-task training.

Many multi-task models in LiDAR perception feature a separate point cloud representation for each task decoder [1]–[3], e.g., the birds-eye view projection for 3D object detection or voxelization for semantic segmentation. This results in multi-task architectures that only share a small voxel encoder

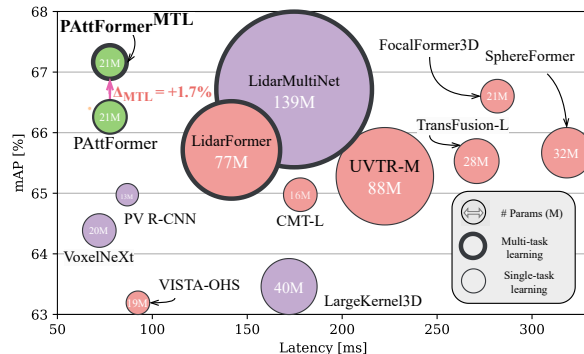


Fig. 1. Performance-latency trade-off compared to related LiDAR object detection models on the nuScenes validation set. Model sizes are encoded in the blob diameter.

and feature heavy task-specific branches dedicated to the complementary point cloud representations, making these architectures parameter heavy and slow (refer to Fig. 1 for a latency comparison). We are interested in whether architectures that share larger portions of parameters can attain similar performance improvement through multi-task learning or if they are prone to performance trade-offs in single tasks, as observed in the vision domain.

To investigate this hypothesis, we propose a simple and efficient point-based multi-task architecture that enables hard parameter-sharing in large parts of the model. We build the network using transformer layers based on neighborhood attention to adapt to the local geometry of the 3D data and design a lightweight detection decoder head based on 3D deformable attention that can be trained end-to-end with a multi-task objective for 3D object detection and semantic segmentation. We perform extensive evaluations on the nuScenes [8] and KITTI [9] datasets, where our proposed multi-task architecture achieves competitive accuracy to state-of-the-art multi-task perception architectures while being $3\times$ smaller and $1.4\times$ faster. During data efficiency experiments using limited annotated training data, we observed consistent performance improvement from multi-task learning for semantic segmentation and object detection compared to single-task training.

Our main contribution can be summarized as follows:

- A novel multi-task architecture that learns raw point cloud features without task-specific projections.
- A lightweight query-based detection head based on 3D deformable attention on point-based representations.
- Extensive evaluations of our method on the nuScenes and KITTI benchmarks for semantic segmentation and 3D object detection.

¹Robert Bosch GmbH, Stuttgart, Germany

²Department of Computer Science, University of Freiburg, Germany

- Evaluations of data efficiency for both single-task and multi-task training.

II. RELATED WORK

In this section, we review LiDAR-based semantic segmentation and object detection methods and embed our work in the landscape of multi-task architectures.

3D Semantic Segmentation: Early works adapted point-based methods [10], [11] from indoor domains to outdoor scenes. However, the point aggregation imposes high computational efforts due to large-scale pointsets from LiDAR scanners due to the neighbor search and decreasing point density with range. Voxel-based methods build on a structured representation of the point cloud that defines a neighborhood relationship, which allows adapting sparse 3D-convolutions [12], [13]. They increase efficiency by trading off between the granularity of the voxel grid and memory requirements. However, the connectivity of feature representations in distant regions of the point cloud is challenging due to the small kernel size, which was approached by cylindrical voxel shapes [13], [14] or improved kernel designs [15]. In recent years, transformer-based modules have been proposed for enabling long-range connections in the feature extractor, either as plug-in modules to sparse convolution networks [16] or as standalone architectures [17], [18].

3D Object Detection: Point-based 3D detectors [19], [20] take unstructured point clouds and process them using set-abstraction modules that aggregate point features on a set of candidate points during the downsampling process. Although high accuracy is achieved in detecting single classes, voxel-based methods dominate multi-class detection benchmarks. Such methods build on a pillar- or voxel-based feature extractor combined with a keypoint [16], [21] or anchor-based detection head [22]. Recently, transformer-based operators have drawn research attention due to their flexibility in capturing contextual features. For attention-based feature extractors, SWFormer [23] uses attention windows of constant volume and proposes to batch windows with a similar number of tokens to deal with varying point density, while other methods [17], [24] explore the possibility of performing self-attention on windows of equal neighbor size but varying volumes. In the detection head, attention-based methods employ features of candidates as detection query embeddings, which are refined in consecutive transformer decoder layers. Those candidates are extracted from center points in bird’s-eye-view (BEV)-based methods [25] or encoded 3D foreground points [26].

Multi-Task LiDAR Perception: LiDAR-based multi-task methods in the literature limit hard-parameter sharing to a pillar-based [3] or voxel-based [1], [2], [27] feature encoder. Except for the pillar-based design in DWSNet [3], which concentrates on detection performance, these multi-task architectures employ multiple decoder heads with task-specific point cloud representation and feature encoding. While [3], [27] limit the inter-dependency of tasks to the encoder, LidarMultiNet [1] proposes to re-project the encoded BEV

detection features back into the global voxel grid for a low-level feature exchange, which they observed to benefit the segmentation task. LidarFormer [2] further proposes a query-based exchange of task-specific prediction features in the task decoder, yielding a performance boost to the detection accuracy. Contrary to [1] and [2], we propose a multi-task architecture based on a point-based representation across all task heads. This eliminates the task-specific feature encoding and explicit feature-exchange mechanisms and allows for a more parameter and computationally efficient architecture at comparable benchmark performances.

III. TECHNICAL APPROACH

In this section, we describe our proposed multi-task architecture in Sec. III-B, followed by the design of the point attention operator in Sec. III-B.1, the detection decoder head in Sec. III-B.3, and finally our multi-task optimization approach in Sec. III-C.

A. Problem Definition

Our model takes as input a point cloud in the form of a bag of unordered points \mathbf{P} , representing a three-dimensional scene captured by range sensors such as a LiDAR scanner. Each point $\mathbf{p}_i \in \mathbf{P}$ in the point cloud is characterized by its three-dimensional coordinates (x, y, z) and additional attributes such as intensity or color. The model predicts a per-point semantic class label s_i , indicating the semantic class of the surface that reflected the laser scan, as well as a set of 3D-oriented bounding boxes \mathbf{b}_m and a class label c_m for all identified objects in a scene. The method is trained with a dataset that contains semantic class and bounding box annotations for each frame.

B. Multi-Task LiDAR PAttFormer Architecture

In a point cloud, the density and other attributes of points may not be uniform across different locations — in 3D scanning, such density variability might arise from perspective effects, radial density variations, motion, etc. Therefore, many recent architectures use task-specific point cloud projection to regain the density of neighborhoods and use convolutional modules to extract features. To become task agnostic in our network design, we use a point-based representation, i.e., we only operate on the non-empty part of the point cloud described by its coordinates and point feature vector. Our core novelty is a point-based multi-task perception model for simultaneous semantic segmentation and object detection. This model avoids task-specific projections and allows hard-parameter sharing in the feature encoding and decoding stages. The model builds on transformer-based feature extraction and prediction modules that operate on point-wise representations, using the PAtt-layers described in Sec. III-B.1.

We follow a multi-scale approach with a U-Net architecture depicted in Fig. 2. The consecutive encoder stages downsample the point density by $8\times$ to extract high-level point cloud features, which are consecutively upsampled and fused with skipped encoder features to preserve fine-grained properties. For downsampling, we use an efficient grid pooling

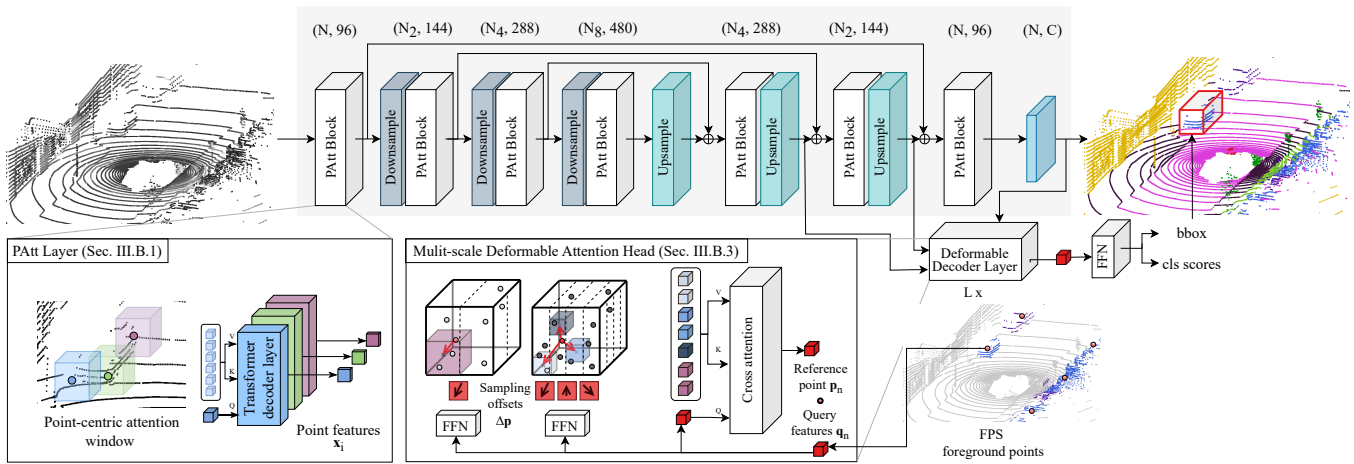


Fig. 2. System architecture for a bottom-up set-up using the proposed PAtt modules that takes a raw point cloud as input and predicts a semantic class label for each scan point. In this setup, we start feature extraction on a subsampled point cloud with a larger attention window volume. We enrich the point cloud with previously masked points in the following stages while decreasing the attention window volume to balance computational complexity.

method [17] that maintains a distributed representation of the points by structuring the point cloud into a regular grid and max-pooling the point features while mean-pooling the point coordinates within each grid cell. The grid size increases with each stage, where N_i denotes the number of non-empty grid cells at stage i . This avoids the expensive neighborhood search and guarantees an upper limit to the point density. During upsampling, point features are unmapped to all points in the grid cell and fused with the point features of the corresponding encoder stage. Dense semantic segmentation labels are predicted by a lightweight task-head described in Sec. III-B.2. Sparse detection predictions are generated by a query-based detection head using deformable attention on the foreground point features described in Sec. III-B.3.

1) *Point-Based Attention (PAtt)*: Instead of the common PointNet architectures, we employ attention-based feature operators that embed positional information in a contextual relative encoding for point-based feature extraction. The design is similar to [17], except that we replace the grouped vector attention with scaled-dot product attention for its lower latency. We compute point-neighborhood windows \mathcal{N}_i for each point separately using a ball-query approach, which selects up to M neighboring points within a specified volume around a query point \mathbf{p}_i . The ball query takes the form of a breadth-first search algorithm on voxel coordinates of each point to exploit the neighbor-aware representation in a voxel grid, similar to the voxel query proposed in [28]. Our ablation study shows that this approach can compete with the expensive nearest-neighbor search on the unordered point cloud used in [17].

We employ a contextual relative position encoding dependent on the query point feature \mathbf{x}_i as a bias to the attention weights given by

$$b_{i,j} = (\mathbf{x}_i \mathbf{W}_r) \mathbf{r}_{i,j}^T \quad \forall j \in \mathcal{N}_i, \quad (1)$$

where $\mathbf{r}_{i,j} = \phi_z(\mathbf{p}_i - \mathbf{p}_j) \in \mathcal{R}^{d_z}$ is the relative position encoding between query point \mathbf{p}_i and neighbor point \mathbf{p}_j by a two-layer MLP ϕ_z and $\mathbf{W}_r \in \mathcal{R}^{d_q \times d_z}$ is a learnable

parameter matrix. The neighborhood attention is then given by

$$\mathbf{y}_i = \sum_{j \in \mathcal{N}_i} \sigma((\mathbf{W}_q \mathbf{x}_i)(\mathbf{W}_k \mathbf{x}_j)^T + b_{i,j}) \mathbf{W}_v(\mathbf{x}_j), \quad (2)$$

where $\mathbf{W}_{\{q,k,v\}}$ are the query, key, and value projections and σ denotes the softmax normalization function.

We integrate the point attention in a point-attention (PAtt)-layer following a standard transformer layer with batch normalization. A PAtt block consists of L consecutive PAtt layers with the same point-attention windows.

2) *Segmentation Head*: The segmentation head consists of an additional PAtt block operating on the full-resolution point cloud and a final classification layer that maps each point feature to the classification scores separately.

3) *Detection Head*: We design a query-based transformer decoder based on a 3D multi-scale formulation of deformable attention as depicted in Fig. 2. First, we extract $Q = 200$ detection queries using the furthest point sampling method from the set of foreground points. The set of foreground points includes all points with a predicted semantic class score exceeding a threshold $\delta_{fg} \geq 0.2$ for any of the thing class types. In the detection-only setting, we train the semantic head as a foreground classifier, for which we consider all points inside a ground truth bounding box as foreground.

The query point features are refined by a deformable attention operation as follows: The query point features \mathbf{q}_k are mapped onto offset vectors $\Delta \mathbf{r}_k = \phi_{off}(\mathbf{q}_k)$ to learn context-dependent sampling points relative to the reference points \mathbf{r}_k . Those sampling points are the center of a voxel query to accumulate point features that build the deformable attention window. We combine the point features of two scale levels; each scale level learns a separate mapping of query features to offset vectors. Furthermore, we use a multi-head deformable attention setting, where each head computes a separate set of sampling offsets. The query features are refined for L decoder layers and finally fed into the classifier head FFN , which regresses a bounding box and class scores, including a background class.

TABLE I
TRAINING SETTINGS. ALL ANGLES ARE IN DEGREES.

Config	nuScenes value	KITTI value
Optimizer	AdamW	AdamW
Learning rate	$1e - 4$	$1e - 4$
LR schedule	CosineAnnealingLR	CosineAnnealingLR
Weight decay	$1e - 2$	$1e - 2$
Epochs	36	50
Pointcloud range	min=[-50, -50, -5] max=[50, 50, 3]	min=[0, -40, -3] max=[70.4, 40, 1]
Random scale	[0.95, 1.05]	[0.95, 1.05]
Random rotate	[-45, 45]	[-45, 45]
Random flip	p=0.5	p=0.5
Laser Mix	pitch angles=[-25, 3]	pitch angles=[-25, 3]
Polar Mix	swap ratio=0.5 rotate paste ratio=1.0	swap ratio=0.5 rotate paste ratio=1.0

C. Multi-Task Training

We use point-wise cross-entropy classification loss $L_{cls,s}$ and a Lovasz Loss $L_{lov,s}$ for the segmentation task. As detection losses, we use a binary focal loss $L_{obj,d}$ for objectness prediction, box classification with a cross-entropy loss $L_{cls,d}$, center offset prediction with a smooth-L1 loss $L_{center,d}$, as well as size prediction with a smooth-L1 loss $L_{size,d}$. In the single-task setting, we compensate for the semantic segmentation losses with a point-wise focal loss for foreground classification. We use uncertainty weighting [29] to balance optimizing the multi-task loss terms.

IV. EXPERIMENTAL EVALUATION

In this section, we evaluate our approach on the nuScenes and KITTI datasets and compare the results against related state-of-the-art methods in Sec. IV-C for object detection and in Sec. IV-D for semantic segmentation. We then analyze the effects of multi-task learning in Sec. IV-E and various architecture design choices Sec. IV-F and report the results of our data efficiency experiments in Sec. IV-G. Finally, we apply the PAttFormer to odometry estimation in Sec. IV-H and present qualitative predictions in Sec. IV-I.

A. Datasets

We evaluate our proposed method on driving domains on the large-scale KITTI and nuScenes datasets that provide the recordings of a single, roof-mounted, LiDAR scanner’s positional and radiometric measurements.

a) nuScenes: The nuScenes [8] dataset consists of 1000 driving sequences of approximately 20 seconds with a LiDAR frequency of $20Hz$. The sequences are split into sets of 700, 150, and 150 for training, validation, and testing and are annotated for 10 classes with a long-tail distribution. The samples were recorded with a Velodyne HDL-32E rotating laser scanner, capturing approximately 30k points per frame [30]. For object detection, annotations are available at $2Hz$, and the point clouds are commonly densified by accumulating over the previous and subsequent sweeps. This dataset suffers from class imbalance, which makes the detection task more challenging.

b) SemanticKITTI: The SemanticKITTI [31] dataset contains 23k training samples and 20k testing samples in autonomous driving scenes with panoptic segmentation annotations for 8 *thing* and 12 *stuff* classes. The samples were recorded with a Velodyne HDL-64E rotating laser scanner, capturing approximately 100k points per cycle [31]. We generate enclosing bounding boxes for all *thing* class types for multi-task learning and follow the common practice to validate the performance on *sequence 08*. We additionally provide detection performance evaluation on the KITTI [9] dataset for benchmark metrics of AP_{3D} at 40 recall points for the car class using an IoU threshold of 0.7 for matching with ground truth objects.

B. Implementation Details

Our implementation is primarily built on the MMDetection3D [32] codebase. The training parameters are detailed in Tab. I. We perform our experiments on a dual-GPU setup using RTX 3090. The reported frame rates are measured on a single A100 for comparability with [1], [2] using the MMDetection3D implementations wherever available. We report the performance using mean results across all classes on the nuScenes and SemanticKITTI datasets. The PAttFormer model parameters are initialized with self-supervised pre-training on the respective train set using the TARK method [33]. We also report results for an efficient multi-task baseline method, which we call MinkUCP, using a MinkowskiUNet [12] segmentation network with a CenterPoint detection head [21] with the same loss terms as our proposed PAttFormer model. We use single sweeps for evaluation and train on accumulated point clouds of three scans for multi-task learning by ignoring unlabeled points in the segmentation losses. For detection training, we follow [25], [34] to initialize additional detection queries from noisy ground truth boxes for faster convergence. We further use class-balanced sampling [35] augmentation during the first 10 training epochs.

C. 3D Object Detection Results

Tab. II summarizes the performance of our proposed architecture on the semi-dense perception task of 3D object detection on the KITTI [9] val and nuScenes [8] test benchmarks. During the post-processing stage, we select bounding boxes with confidence scores larger than 0.2, and we utilize non-maximum suppression (NMS) with an IoU threshold of 0.4. We selected related multi-task architectures (marked with ^{MTL}) and alternative point-based, voxel-based, and center-based (BEV) methods with comparable model sizes or frame rates as baselines. Our proposed PAttFormer model achieves 67.0% in mAP on the nuScenes test set, outperforming all other methods with frame rates $> 10Hz$. PAttFormer even achieves comparable detection accuracy with related multi-task networks [1], [2] at less than one-third of the parameter count. On the KITTI val benchmark, our model outperforms other point-based and two-stage methods by +1.6% and +0.6% in mAP on the *car* class for moderate difficulty and achieves a frame rate of 8.4Hz for a point cloud with 64 horizontal scanlines.

TABLE II

DETECTION RESULTS ON THE nuSCENES AND KITTI BENCHMARKS. MODELS MARKED WITH (^{MTL}) WERE TRAINED IN A MULTI-TASK SETTING. KITTI AP IS REPORTED FOR CLASS "CAR" AT 40 RECALL POINTS. FRAME RATES MARKED WITH (*) WERE TAKEN FROM THE CITED REFERENCES.

Methods	Representation	nuScenes [val]		nuScenes [test]		Params	FPS	KITTI [val] AP _{3D}			Params	FPS
		mAP	NDS	mAP	NDS			Easy	Mod.	Hard		
PointRCNN [20]	P	-	-	-	-	-	-	88.9	78.6	77.4	4.0M	8.7
3D-SSD [19]	P	56.4	42.7	-	-	2.5	18.9	89.7	79.4	<u>78.7</u>	2.5M	18.6
PV-RCNN [22]	V+P	65.0	53.6	-	-	13.2	10.5	<u>90.3</u>	<u>81.4</u>	<u>76.8</u>	13.1M	13.1
CenterPoint [21]	V+BEV	57.1	65.4	58.0	65.5	8.9	17.8	-	-	-	-	-
Li3DDeTr [26]	V+BEV	-	-	61.3	67.6	-	-	87.6	76.8	73.9	-	-
SphereFormer [16]	V+BEV	-	-	65.5	70.7	-	-	-	-	-	-	-
LiDARMultiNet ^{MTL} [1]	V+BEV	63.8	69.5	67.0	71.6	139.0M	7.9*	-	-	-	-	-
LiDARFormer ^{MTL} [2]	V+BEV	<u>66.6</u>	<u>70.8</u>	68.9	<u>72.4</u>	77.0M	6.1*	-	-	-	-	-
FocalFormer3D [25]	V+BEV	66.5	71.1	<u>68.7</u>	72.6	21.3M	3.6	-	-	-	-	-
PAttFormer	P	64.3	68.7	65.3	69.2	21.5M	11.0	88.8	78.5	77.3	21.5M	8.4
PAttFormer ^{MTL}	P	66.7	69.5	67.0	69.3	21.5M	11.0	91.0	82.0	79.1	21.5M	8.4

TABLE III

SEMANTIC SEGMENTATION RESULTS ON nuSCENES AND SEMANTICKITTI BENCHMARKS. MODELS MARKED WITH (^{MTL}) WERE TRAINED IN A MULTI-TASK SETTING. FRAME RATES MARKED WITH (*) WERE TAKEN FROM THE CITED REFERENCES.

Method	Representation	nuScenes		Params	FPS	SemanticKITTI		Params	FPS
		mIoU [val]	mIoU [test]			mIoU[val]	mIoU [test]		
Convolutional methods									
PolarNet [14]	V	71.0	69.8	14.0M	14.9	53.6	54.3	14.0M	12.1
MinkowskiNet [12]	V	70.3	-	37.8	21.7	61.1	63.1	21.7	15.3
Cylinder3D [13]	V	76.1	77.2	55.8M	8.9	64.3	61.8	55M.8	8.9
LiDARMultiNet ^{MTL} [1]	V	<u>82.0</u>	81.4	139.0M	7.9*	-	-	-	-
LiDARFormer ^{MTL} [2]	V	82.7	<u>81.0</u>	77.0M	6.1*	-	-	-	-
Transformer-based methods									
RangeViT [36]	RV	75.2	-	27.1M	-	-	64.0	27.1M	-
SphereFormer [16]	V	78.4	81.9	32.3M	5.4	67.8	74.8	32.2M	2.6
PAttFormer	P	78.0	76.9	16.9M	14.8	<u>68.4</u>	67.3	16.9M	9.5
PAttFormer ^{MTL}	P	79.8	78.7	16.9M	14.8	69.8	<u>69.3</u>	16.9M	9.5

TABLE IV

PER-CLASS SEMANTIC SEGMENTATION RESULTS ON nuSCENES VAL SET.

Method	mIoU	Class														
		barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic cone	trailer	truck	driveable	other flat	sidewalk	terrain	manmade
PolarNet [14]	71.0	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.1	74.7	74.0	87.3
Cylinder3D [13]	76.1	76.4	40.3	91.2	<u>93.8</u>	31.3	78.0	78.9	64.9	62.1	84.4	96.8	71.6	76.4	75.4	90.5
PVKD [37]	76.0	76.2	40.0	90.2	94.0	50.9	77.4	78.8	64.7	62.0	84.1	96.6	71.4	76.4	76.3	90.3
RPVNet [38]	77.6	78.2	43.4	92.7	93.2	49.0	85.7	80.5	<u>66.0</u>	66.9	84.0	<u>96.9</u>	73.5	75.9	76.0	<u>90.6</u>
SphereFormer [16]	78.4	<u>77.7</u>	43.8	94.5	93.1	52.4	<u>86.9</u>	<u>81.2</u>	65.4	85.3	97.0	<u>73.4</u>	75.4	75.0	91.0	-
PAttFormer	78.0	76.1	<u>48.5</u>	<u>95.2</u>	91.2	<u>58.2</u>	86.7	80.0	63.0	72.3	<u>87.1</u>	96.3	73.0	74.3	74.0	88.1
PAttFormer ^{MTL}	79.8	<u>77.8</u>	50.8	95.5	92.6	60.3	88.3	82.5	67.0	75.5	88.8	96.7	74.9	75.7	75.1	88.8

D. Semantic Segmentation Results

We evaluate the semantic segmentation performance on the nuScenes and SemanticKITTI in Tab. III benchmarks. We report the latency with the detection head excluded. The proposed PAttFormer^{MTL} model achieves 78.7% in mIoU on the nuScenes test set, outperforming other voxel-based transformer methods and in close range to the much larger multi-task networks. The per-class evaluation in Tab. IV suggests that the PAttFormer model exhibits especially strong performance on *thing* classes, with improvement of +2.5% in IoU for pedestrians from multi-task training alone. on the SemanticKITTI test set, the PAttFormer^{MTL} method achieves

69.3% in mIoU without using test-time augmentations. We observe a performance improvement through multi-task training of +1.8% and +2.0% in mIoU on the nuScenes and SemanticKITTI test sets, respectively.

E. Multi-Task Learning Evaluation

In this section, we evaluate the benefits of multi-task training for benchmark performance on the nuScenes dataset in Tab. VI. Apart from models from the literature, we report results for a runtime-efficient baseline consisting of a MinkUNet [12] voxel-encoder and a CenterPoint [21] detection head, which we call MinkUCP. This baseline model was trained with comparable training settings to our

TABLE V

ABLATION EXPERIMENTS ON nuSCENES VALIDATION SET. DETECTION IS ALSO SPLIT INTO CLOSE RANGE ($\leq 10m$), MEDIUM RANGE ($10m < x \leq 30m$), AND FAR RANGE ($30m < x$) TO THE GT BBOX CENTER.

Windowing Search	Sampling Size	Loss balancing	Det. head	Seq mIoU	mAP	Det AP_c	AP_m	AP_f	FPS	
VQ	32	Grid	Uncertainty	DDETR3D	<u>79.8</u>	66.8	<u>79.1</u>	71.1	37.0	11.0
VQ	16	Grid	Uncertainty	DDETR3D	74.3	52.6	73.4	58.0	24.9	10.7
VQ	64	Grid	Uncertainty	DDETR3D	79.5	64.5	76.6	66.4	32.1	9.2
kNN	32	Grid	Uncertainty	DDETR3D	79.9	<u>66.3</u>	<u>79.1</u>	<u>70.7</u>	<u>36.5</u>	8.3
VQ	32	None	Uncertainty	DDETR3D	75.8	48.7	70.7	53.7	22.0	7.0
VQ	32	Grid	Uncertainty	CenterPoint	76.5	59.6	75.9	64.4	30.9	12.5
VQ	32	Grid	GradNorm	DDETR3D	79.3	64.8	79.3	67.2	30.6	11.0

TABLE VI

MULTI-TASK EVALUATION ON nuSCENES VALIDATION SET. FRAME RATES MARKED WITH * WERE TAKEN FROM THE CITED REFERENCES.

	MTL	Segmentation mIoU	Δ_{MTL}	Detection mAP	Δ_{MTL}	FPS	Params
LidarMultiNet [1]	✓	82.0	-	67.0	-	7.9*	139.0M
LidarFormer [2]	✗	<u>82.1</u>	-	65.9	-	-	77.0M
	✓	82.6	+0.5	66.0	+0.1	6.1*	77.0M
MinkUCP	✗	74.5	-	57.1	-	13.7	44.1M
	✓	75.0	+0.5	56.9	-0.2	13.7	44.1M
PAttFormer	✗	78.0	-	64.3	-	11.0	21.5M
	✓	79.8	+1.8	<u>66.7</u>	+2.4	11.0	21.5M

PAttFormer model, adapting the learning to $lr = 0.01$. PAttFormer runs at $1.4\times$ faster frame rate compared to LiDARMultiNet [1] at slightly inferior performance by $-0.3\%/ -2.2\%$ for the detection and semantic segmentation, respectively. Our proposed point-based PAttFormer model outperforms the efficient MinkUCP baseline in segmentation and detection but with a slightly lower frame rate. Note that the PAttFormer model achieves the highest improvement from multi-task learning among all architecture designs, with a $+2.4\%$ increase in mAP for the detection task.

F. Ablation Study

The proposed feature extractor depends on a multitude of hyperparameters. In Tab. V, we summarize the effect of each hyperparameter on perception performance for models trained in a multi-task learning setting.

Attention Window Size First, we are interested in how the size of the attention window affects the model’s ability to understand the scene. We trained three models with attention windows of 16, 32, and 64 neighbor points, respectively, and achieved the best performance for a window size of 32. Interestingly, the larger window size 64 has lower detection performance, especially for distant objects, with the frame rate dropping by $-2Hz$ compared to computing attention over 32 neighbor points. We assume that the larger window size, especially in downsampled representations, includes points of multiple objects or background points that hinder learning discriminative features. Next, we replaced the ball-query neighbor-search with a k-nearest neighbor algorithm, which had a minor effect on the model performance but caused the frame rate to drop by $-2.7Hz$.

TABLE VII

DATA EFFICIENCY RESULTS ON nuSCENES VALIDATION SET.

Model	Train seq.	Seg mIoU [%]			Det mAP [%]		
		25%	50%	100%	25%	50%	100%
MinkowskiNet		61.9	70.0	70.3	-	-	-
CenterPoint		-	-	-	41.2	52.8	57.1
MinkUCP ^{MTL}		62.5	70.5	70.8	42.7	52.5	56.9
PAttFormer		<u>65.0</u>	<u>74.8</u>	<u>78.0</u>	<u>48.1</u>	<u>59.1</u>	<u>64.3</u>
PAttFormer ^{MTL}		70.9	76.3	79.8	52.6	63.2	66.7

Multi-Scale Hierarchy When disabling the point cloud downsampling in the encoder blocks, we observe a performance decrease by -11.1% in detection mAP. We assume the downsampling facilitates the connectivity of the point cloud, as it increases the field of view of each point with a fixed neighborhood size, allowing the model to learn more discriminative features in the encoder blocks.

Loss Weighting We compared two loss balancing strategies to trade-off performance across all tasks, *i.e.* uncertainty weighting [29] and gradient normalization. The uncertainty weighting strategy outperformed gradient normalization in both detection and segmentation tasks, whereby the performance drops most noticeably for the detection task by -2.0% in mAP.

Detector Head Lastly, we replace the detection head with a BEV projection and a CenterPoint decoder. Our results indicate that switching to a BEV-based detection head impairs detection (-7.2% in mAP) and segmentation (-3.3% in mIoU) performance alike. We suspect that the BEV projection and separate feature encoding required by the CenterPoint [21] method results in the decoupling of the perception tasks that impede multi-task learning. As a result, the optimizer struggles with tradingoff performance between both tasks such that the segmentation score even falls below the single-task performance by -1.5% in mIoU (compared to Tab. III).

G. Data Efficiency Analysis

We assess the effect of multi-task training on overfitting by a data efficiency analysis. Therefore, we train the models with multiple fractions of nuScenes *train* set sequences and compare their performance on the *val* set in Tab. VII. We compare validation set performance for training on 25%, 50%, and 100% of training sequences, whereby we adopt the number of training epochs to 144, 72, and 36,

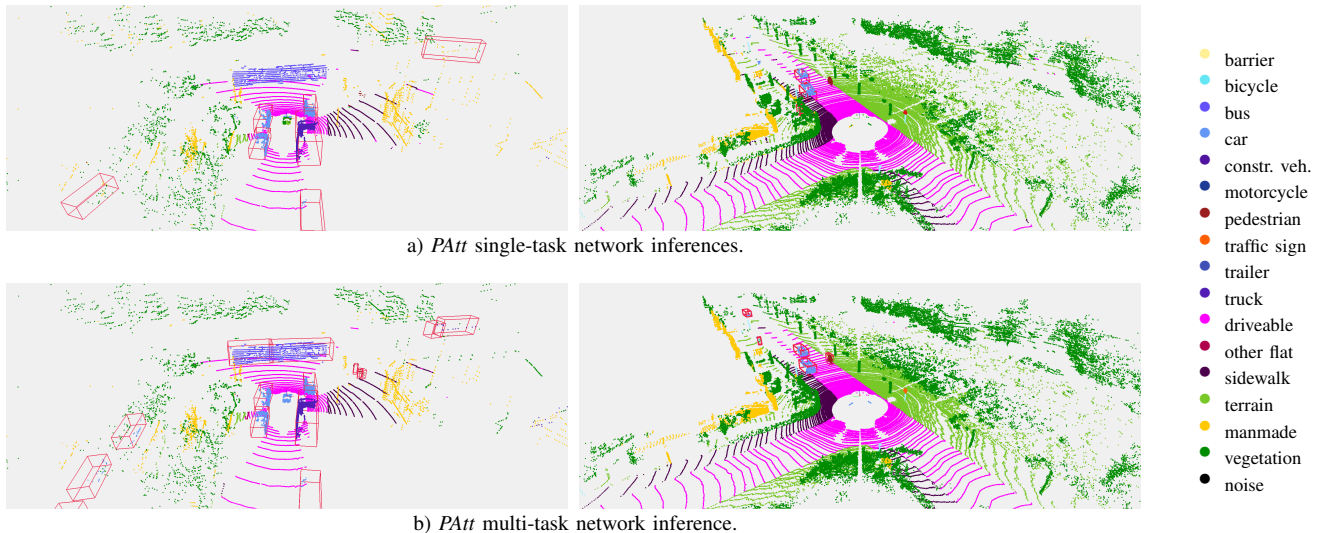


Fig. 3. Comparison of predicted bounding boxes and segmentation labels on the 32-line scan from nuScenes (left) and 64-line scan from SemanticKITTI (right) val sets.

TABLE VIII

ABSOLUTE TRAJECTORY ERROR (ATE) AND ABSOLUTE ROTATIONAL ERROR (ARE) AVERAGED OVER ALL SEQUENCES FOR LiDAR ODOMETRY.

	SemanticKITTI [Seq 8]		nuScenes [val]	
	ATE [m]	ARE [rad]	ATE [m]	ARE [rad]
Raw pointcloud	5.97	<u>0.026</u>	1.80	0.018
MinkowskiUNet	6.47	0.028	<u>1.56</u>	<u>0.013</u>
PAttFormer	5.87	<u>0.026</u>	1.61	0.014
PAttFormer ^{MTL}	<u>5.88</u>	<u>0.026</u>	1.42	0.012

respectively. We also provide results for the MinkUCP multi-task baseline and its task-specific components. The PAttFormer^{MTL} model trained in a multi-task setting achieves good performance even with reducing training data variety to approximately 25%. Using only half of the training annotations, the PAttFormer^{MTL} achieves 76.3% in mIoU and 63.2% in mAP, which is only -1.7% / -1.1% less than the PAttFormer model trained on single-tasks with all available training annotations. We did not observe such a trend with the MinkUCP baseline.

H. Odometry Analysis

We evaluated the advantages of using semantic segmentation to prefilter point clouds for LiDAR odometry methods. Our study focused on the KISS-ICP [39] method, which uses iterative closest point matching methods on point clouds to correct the pose predictions of a constant velocity model. As this method builds on a static scene assumption, we excluded points labeled as members of a vehicle- or human-based category by different semantic segmentation models. In Tab. VIII, we compare the performance of different semantic segmentation models by the absolute translational and rotational errors across all sequences in the SemanticKITTI and nuScenes validation sets. We use the default hyperparameter settings reported in [39].

Our results indicated that excluding points with a semantic label of a vehicle- or human-related category generally resulted in lower overall translation and rotational errors,

except for the MinkowskiUNet model on the SemanticKITTI dataset. Among the evaluated semantic segmentation methods, filtering the point cloud based on semantic labels predicted by the PAttFormer^{MTL} model produced the lowest translational and rotational errors. This was especially evident in the nuScenes datasets, which have a high density of dynamic objects per scan. We believe this is due to the model’s improved accuracy on *thing* class types, which was also observed in the class-wise semantic segmentation performance in Tab. IV.

I. Qualitative Evaluations

To assess the quality of our semantic segmentation results, we compared them with the predictions of the baseline multi-task network in Fig. 3. The first row depicts the predictions of two PAttFormer models, one trained for predicting object bounding boxes and one trained to classify semantic labels. The second row depicts the predictions of a single PAttFormer model trained in a multi-task learning setting.

The PAttFormer model in the bottom row, trained with a multi-task setting, provides more consistent semantic labels with the predicted detection bounding boxes, highlighting the mutual benefits of the perception tasks. Upon visual inspection, the PAttFormer model predictions in the bottom row, trained with a multi-task setting, correctly detect the bendy bus and pedestrians in the left scene and the bicyclist in the right image. Furthermore, multi-task learning appears to result in more consistent semantic labels inside the predicted detection bounding boxes, highlighting the mutual benefits of the perception tasks.

V. CONCLUSION

We proposed a novel point-based multi-task architecture for joint semantic segmentation and object detection in point clouds. Our design allows hard-parameter sharing for large parts of the model, resulting in an efficient architecture that can achieve an 11Hz frame rate on a single A100 GPU. The effectiveness of our architecture was evaluated on two

benchmarks, KITTI and nuScenes, where it achieved an mAP of 67.2% on detection and a mIoU of 79.8% on semantic segmentation. We observed consistent performance improvement from multi-task learning, resulting in +2.4% in detection mAP and +1.8% in segmentation mIoU compared to the single-task training. These improvements observed for only point-based architecture are even higher than observed for related multi-task architectures that use multiple point cloud representations with fewer network parts appropriate for hard parameter-sharing. Our data efficiency experiments further indicate that multi-task learning is particularly beneficial for smaller training datasets, where the performance improvement compared to single-task training is even more significant. This finding can have a broader impact on robotics applications with little available training data. Future work can evaluate whether similar improvements can be observed from training with automated labels generated by vision foundation models, as well as the robustness of the learned representations under out-of-distribution scenarios.

REFERENCES

- [1] D. Ye, Z. Zhou, W. Chen, Y. Xie, Y. Wang, P. Wang, and H. Foroosh, "Lidarmultinet: Towards a unified multi-task network for lidar perception," in *AAAI*, vol. 37, no. 3, 2023, pp. 3231–3240.
- [2] Z. Zhou, D. Ye, W. Chen, Y. Xie, Y. Wang, P. Wang, and H. Foroosh, "Lidarformer: A unified transformer-based multi-task network for lidar perception," *arXiv preprint arXiv:2303.12194*, 2023.
- [3] X. Zhang, C. Min, Y. Jia, L. Chen, J. Zhang, and H. Sun, "Boosting lidar 3d object detection with point cloud semantic segmentation," in *Int. Conf. on Intelligent Robots and Systems*, 2023, pp. 7614–7621.
- [4] N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "Continual slam: Beyond lifelong simultaneous localization and mapping through continual learning," in *The International Symposium of Robotics Research*, 2022, pp. 19–35.
- [5] B. Bešić and A. Valada, "Dynamic object removal and spatio-temporal rgb-d inpainting via geometry-aware adversarial learning," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 170–185, 2022.
- [6] N. Vödisch, K. Petek, W. Burgard, and A. Valada, "Codeps: Online continual learning for depth estimation and panoptic segmentation," *Robotics: Science and Systems*, 2023.
- [7] R. Mohan and A. Valada, "Perceiving the invisible: Proposal-free amodal panoptic segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9302–9309, 2022.
- [8] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 11 621–11 631.
- [9] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.
- [10] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Adv. Neural Inform. Process. Syst.*, 2017.
- [11] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.
- [12] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.
- [13] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
- [14] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, "Polarnet: An improved grid representation for online lidar point clouds semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 9601–9610.
- [15] T. Lu, X. Ding, H. Liu, G. Wu, and L. Wang, "Link: Linear kernel for lidar-based 3d perception," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 1105–1115.
- [16] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, "Spherical transformer for lidar-based 3d recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 17 545–17 555.
- [17] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," in *Adv. Neural Inform. Process. Syst.*, 2022, pp. 33 330–33 342.
- [18] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 5240–5250.
- [19] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 11 040–11 048.
- [20] S. Shi, X. Wang, and H. Li, "Pointcnn: 3d object proposal generation and detection from point cloud," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.
- [21] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
- [22] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 10 529–10 538.
- [23] P. Sun, M. Tan, W. Wang, C. Liu, F. Xia, Z. Leng, and D. Anguelov, "Swformer: Sparse window transformer for 3d object detection in point clouds," in *Eur. Conf. Comput. Vis.*, 2022, pp. 426–442.
- [24] Z. Liu, X. Yang, H. Tang, S. Yang, and S. Han, "Flatformer: Flattened window attention for efficient point cloud transformer," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 1200–1211.
- [25] Y. Chen, Z. Yu, Y. Chen, S. Lan, A. Anandkumar, J. Jia, and J. M. Alvarez, "Focalformer3d: focusing on hard instance for 3d object detection," in *Int. Conf. Comput. Vis.*, 2023, pp. 8394–8405.
- [26] G. K. Erabati and H. Araujo, "Li3detr: A lidar based 3d detection transformer," in *IEEE/CVF Winter Conf. on Applications of Comput. Vis.*, 2023, pp. 4250–4259.
- [27] D. Feng, Y. Zhou, C. Xu, M. Tomizuka, and W. Zhan, "A simple and efficient multi-task network for 3d object detection and road understanding," in *Int. Conf. on Intelligent Robots and Systems*, 2021.
- [28] C. He, R. Li, S. Li, and L. Zhang, "Voxel set transformer: A set-to-set approach to 3d object detection from point clouds," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 8417–8427.
- [29] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7482–7491.
- [30] W. K. Fong, R. Mohan, J. V. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada, "Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3795–3802, 2022.
- [31] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.
- [32] M. Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," <https://github.com/open-mmlab/mmdetection3d>, 2020.
- [33] L. Nunes, L. Wiesmann, R. Marcuzzi, X. Chen, J. Behley, and C. Stachniss, "Temporal Consistent 3D LiDAR Representation Learning for Semantic Perception in Autonomous Driving," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.
- [34] B. Zhu, Z. Wang, S. Shi, H. Xu, L. Hong, and H. Li, "Conquer: Query contrast voxel-detr for 3d object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 9296–9305.
- [35] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.
- [36] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu, "Rethinking range view representation for lidar segmentation," *arXiv preprint arXiv:2303.05367*, 2023.
- [37] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 8479–8488.
- [38] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," in *Int. Conf. Comput. Vis.*, 2021, pp. 16 024–16 033.
- [39] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp—simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.