2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

Contrastive Learning for Automotive mmWave Radar Detection Points Based Instance Segmentation

Weiyi Xiong[1*], Jianan Liu[2*], Yuxuan Xia[3], Tao Huang[4], Bing Zhu[1†] and Wei Xiang[5]

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922540

***Abstract*— The automotive mmWave radar plays a key role in advanced driver assistance systems (ADAS) and autonomous driving. Deep learning-based instance segmentation enables real-time object identification from the radar detection points. In the conventional training process, accurate annotation is the key. However, high-quality annotations of radar detection points are challenging to achieve due to their ambiguity and sparsity. To address this issue, we propose a contrastive learning approach for implementing radar detection points-based instance segmentation. We define the positive and negative samples according to the ground-truth label, apply the contrastive loss to train the model first, and then perform fine-tuning for the following downstream task. In addition, these two steps can be merged into one, and pseudo labels can be generated for the unlabeled data to improve the performance further. Thus, there are four different training settings for our method. Experiments show that when the ground-truth information is only available for a small proportion of the training data, our method still achieves a comparable performance to the approach trained in a supervised manner with 100% ground-truth information.**

I. INTRODUCTION

Automative mmWave radars, LiDARs and cameras are all important sensors for autonomous driving. A camera can take images containing rich information such as the colors and edges of objects, and a dual camera system can also determine the distance; the LiDAR emits laser waves and receives them after reflection, generating dense LiDAR points that distribute on the surface of objects so that shapes of objects are obtained. Although cameras and LiDARs provide data easy to understand by both humans and computers, there are some drawbacks of them. Firstly, they cannot work in certain weather conditions, such as heavy rain, snow or fog [1]. In addition, some objects cannot be detected in presence of occlusion. Last but not least, they do not provide velocity information, which is important for autonomous driving. As a result, mmWave radars are indispensable in the Advanced Driver Assistance Systems (ADAS), as they work in all-weather conditions, detect occluded objects, and measure the radial velocity of each object [2].

However, the sparsity of radar detection points makes it challenging for scene understanding. Modern deep learning methods could be employed to solve this problem, but those methods need massive labeled data to achieve high performance [3][4]. Typically, the radar data are manually annotated by human experts [5] or semi-automatically annotated by certain approaches for simplification [6][7]. But labeled data from cameras or LiDARs are usually required in either method to guarantee high accuracy, which is costly. Moreover, the radar detection points are often semantically ambiguous. Thus, it is difficult to annotate them with a semantic or instance label. To solve the aforementioned issues, we propose a contrastive learning-based method that only relies on a few labeled training data to perform instance segmentation on radar detection points.

Contrastive learning is a learning strategy that aims to train the model in a self-supervised fashion by finding a proper representation of unlabeled data. For each datum (also called an anchor datum), data which have high similarity with it are defined as its positive samples, while those dissimilar to it are defined as its negative samples. As the first step of contrastive learning, properly defining positive and negative samples are important to let the model learn to extract corresponding features. This process is important as it can help in performing the downstream task efficiently. In this step, the contrastive loss and the backpropagation process tries to pull apart the features of negative samples learned by the model, and draw the features of positive samples closer at the same time. To solve the downstream task, a fine-tuning step is followed. In this step, a few labeled data are taken as the input; the parameters of the backbone are frozen; and the model is trained in a supervised learning strategy.

This paper follows [3] to apply semantic segmentation-based clustering for radar points instance segmentation, and adopts the contrastive learning to replace the traditional supervised learning to train the semantic segmentation model. Specifically, our contributions are threefold:

- According to the best of authors' knowledge, we are the first to adopt contrastive learning on the radar detection points-based instance segmentation task to tackle the issue of insufficient point-wise annotation of radar detection points. We expect to inspire more investigations on radar-based perception without sufficient labeled radar data through this study.
- We propose an efficient contrastive learning-based strategy for semantic segmentation with radar detection

[1]W. Xiong and B. Zhu are with School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, P.R. China weiyixiong@buaa.edu.cn (W.Xiong); zhubing@buaa.edu.cn (B.Zhu)
[2]J. Liu is with Vitalent Consulting, Gothenburg 41761, Sweden, and Silo AI, Stockholm, Sweden jianan.liu@vitalent.se
[3]Y. Xia is with Department of Electrical Engineering, Chalmers University of Technology, Gothenburg 41296, Sweden yuxuan.xia@chalmers.se
[4]T. Huang is with College of Science and Engineering, James Cook University, Cairns, Australia tao.huang1@jcu.edu.au
[5]W. Xiang is with School of Engineering & Mathematical Sciences, La Trobe University, Melbourne, Australia w.xiang@latrobe.edu.au
*Both authors contribute equally to the work and are co-first authors.
†Corresponding author.

points. By training the model in the proposed contrastive learning strategy, the performance reaches a satisfactory level constrained by limited labeled data. Several settings can be applied to our proposed strategy, i.e., fully-supervised setting/non-joint training, semi-supervised setting/non-joint training, fully-supervised setting/joint training and semi-supervised setting/joint training.
- Experiments show that the model trained with the proposed contrastive learning strategy in all settings outperforms that trained in a supervised learning manner when only a small proportion (5%) of labeled training data are available. The performance of the latter is 75.59% mean coverage (mCov) and 70.86% mean average precision with the IoU threshold of 0.5 ($\text{mAP}_{0.5}$), while the different settings of our method obtain an improvement of about 2%-3.5% without introducing any additional inference time and memory consumption.

The rest of the paper is organized as follows. Section II introduces the related work, including LiDAR and radar based instance segmentation methods and contrastive learning strategies in the field of computer vision. Then our proposed contrastive learning strategy for radar points instance segmentation and its different settings are described in Section III. The information of the dataset, as well as the experimental results are presented in Section IV. Analyses are also made in this section. Finally, Section V summarizes our work and presents the future research direction.

II. RELATED WORK

There are mainly two types of point clouds: one is the dense point cloud such as the LiDAR point cloud, and the other is the sparse point cloud such as the radar point cloud. In this section, we review related studies on instance segmentation with LiDAR point clouds and automotive radar detection points. Some of the methods are based on traditional algorithms, and others apply neural networks in a deep learning driven fashion. However, the former are restricted with their performance, and the latter need a lot of labeled training data. In addition, works on constrative learning are introduced, most of which focus on tasks about images. These works only requires a few labeled data to train the models, and thus inspires us to propose our strategy.

A. Instance Segmentation with LiDAR Point Cloud

There are many researches on LiDAR point cloud-based instance segmentation [8][9][10][11][12]. The concept of LiDAR slices is proposed in [13], and a Slice Growing algorithm is devised to perform instance segmentation [8]. The method first extracts and merges major parts, and grows them by searching their neighbors. After that, semantic labels are predicted through an RNN for labeling move objects and a simple judgement algorithm for labeling static objects. However, this method is mostly based on traditional algorithm. Thus, as the environment of the ego-vehicle becomes more complicated, the performance of the method might be restricted. As a result, recent researches in the community shift the interest in deep learning-based methods.

Some researchers transform the LiDAR point cloud into a high-resolution Bird's Eye View (BEV) image which is then processed by a CNN-based model. Recent work includes [9] which applies a revised stacked hourglass block to predict the semantic label and the center of the object for each foreground grid pixel, getting the predicted instances after grouping points and merging objects whose predicted centers are close enough. Xiong *et al.* predict semantic segmentation results, center offsets and a contour map, and further predict a binary mask for the BEV image [10]. After that, the final instance segmentation results are generated based on the predicted binary mask and the semantic information.

For deep learning methods taking raw LiDAR points as input, Behley *et al.* treat the instance segmentation task as an object detection task, i.e., they generate a 3D bounding box for each instance [11]. Gasperini *et al.* propose a model called Panoster [12], which has a shared encoder and two independent decoders for semantic segmentation and instance segmentation, respectively. It is a completely learning-based method because its instance segmentation branch optimizes the loss based on the soft confusion matrix and do not require the external grouping process.

B. Instance Segmentation with Automotive Radar Detection Points

Due to the sparsity of automotive radar detection points, tasks such as instance segmentation with automotive radar detection points are more challenging than those with LiDAR point clouds. As the most commonly-used method for instance segmentation on radar points is clustering-based classification, efforts are made by enhancing the clustering methods. For instance, DBSCAN algorithm was modified in [14], and a score function is optimized in a traditional supervised machine learning strategy so that the parameters of DBSCAN could be automatically chosen to achieve a higher performance [15]. However, each point cloud in training and testing is composed of all detection points within a certain time range, which may cause overlap, and the clustering parameters are still not related to the class of the instance, which restricts further improvements on performance.

Other work perform instance segmentation using deep learning methods. The work in [3] predicts the semantic label as well as a center shift vector for each detection point using gMLP [16] based PointNet++ [17], and applies DBSCAN [14] on the shifted points to obtain the instance information. The strategy in [4] is similar to the semantic segmentation-based clustering method in [3], and a memory point cloud is introduced to jointly consider the related information among consecutive frames to improve the segmentation performance. However, these deep learning-based methods require a large number of labeled training data, which restricts their application because of the difficulty to obtain enough point-wise labeled radar frames.

C. Contrastive Learning

As a self-supervised learning strategy, contrastive learning which explores a proper representation of unlabeled data, plays a vital role when annotating training data is difficult.

To learn a good representation of the data, positive and negative samples need to be carefully chosen. Most of efforts are made in the field of computer vision, especially in image classification. For example, SimCLR [18] uses data augmentation methods to generate a "copy" of each image. For every image, the "copy" is defined as its positive sample and other images as well as their augmentations are regarded as negative samples.

However, there are less literature about adopting contrastive learning for semantic segmentation, due to the difficulty of choosing positive and negative samples. The method proposed in [19] inputs two different views of an image into two encoders, each with a dense projection head. The distances between each pair of pixels are calculated and the closest ones are selected as positive pairs. Other researchers [20][21] use the pixel-wise label to define positive and negative samples. By doing that, in the case of a few labeled images in the dataset, the pixels in those labeled images are sufficient for supervised contrastive learning. To boost performances, a memory bank is introduced to accumulate negative samples in the used minibatch [20]. Differently, a semi-supervised training setting applies the unlabeled training data to enhance the outcome [21].

III. PROPOSED METHOD

This section introduces the proposed instance segmentation method with the details of different settings in contrastive learning. In particular, our training can be performed in either fully-supervised setting or semi-supervised setting, which is similar to [21]. The former uses the small amount of training data which have been labeled, while the latter takes both labeled and massive unlabeled training data as input.

A. Overview

In this work, we adopt the semantic segmentation-based clustering method for the instance segmentation task, as it has been demonstrated in [3] that this method is feasible in automotive mmWave radar real-time applications. To be specific, the points are input into a semantic segmentation model trained in a contrastive learning strategy, obtaining their predicted semantic label. Then DBSCAN [14] is used to cluster the points belonging to the same class into different clusters, where each cluster represents an instance.

As mentioned in Section I, the process of contrastive learning consists of two steps: representation learning and downstream task fine-tuning. Representation learning is the critical step in the process because in this step, appropriate positive and negative samples are identified for the further training process. In an image classification problem, data augmentation is widely used to help define positive and negative samples [18]. However, data augmentation is challenging for radar detection points based semantic segmentation, and there is no suitable data augmentation method for *radar* point cloud *segmentation* tasks.

Fortunately, many labeled points can be obtained even when the number of labeled frames of radar detection points is small. Thus, defining positive and negative samples according to the ground-truth label is feasible to train the model using a contrastive learning strategy. Furthermore, we can generate pseudo labels to make use of unlabeled training data, and define the positive and negative samples in a way similar to the semi-supervised setting in [21]. More details are in the following subsections.

B. Fully-Supervised Setting

1) Representation Learning: The process of representation learning is illustrated by gray and green parts of Fig. 1. Positive and negative samples are defined according to ground-truth labels in fully-supervised setting due to sufficient labeled points, even if the number of labeled frames is small. In detail, points with the same semantic label are positive samples, and points with different semantic labels are negative samples. To balance the number of points

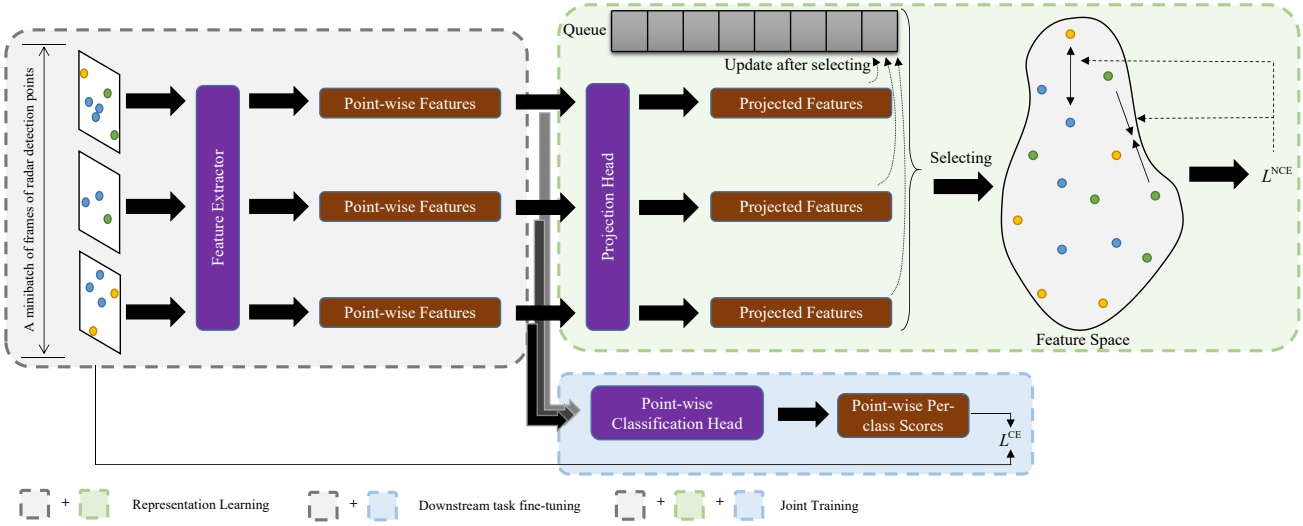

Fig. 1. The training process of our proposed contrastive learning based model. The gray and green parts of the figure show the representation learning process, where features are projected by a projection head, points are selected from the frames in a minibatch as well as the queue, and the contrastive loss is calculated. The gray and blue parts show the downstream task fine-tuning process, during which the feature extractor is frozen, a point-wise classification head rather than the projection head is connected, and the cross entropy loss is calculated. The whole figure illustrates the joint training process. In this training strategy, both heads exist and the total loss is the weighted sum of the two losses.

$$L_i^{\mathrm{NCE}} = \frac{1}{|P_i|} \sum_{i^+ \in P_i} -\log \frac{\exp(f_i \cdot f_{i^+}/\tau)}{\exp(f_i \cdot f_{i^+}/\tau) + \sum_{i^- \in N_i} \exp(f_i \cdot f_{i^-}/\tau)} \tag{1}$$

in each class, the following operations are performed: in each minibatch, n_{point} points are randomly selected from $n_{\mathrm{minibatch}}$ frames, where number of points in each class are the same; as a result, $N = n_{\mathrm{point}}/n_{\mathrm{class}}$ points for each class are selected, where n_{class} denotes the number of classes.

To make sure that there is enough positive and negative samples (i.e., number of points in each class) in every minibatch, a queue is used to store the features of points in the previous minibatchs. That is, if there are not enough points in some classes, the remaining points will be selected from the queue, and after that the queue will be updated. Let $n_{i,j}^{\mathrm{minibatch}}$ denote the number of points belonging to the j-th class in the i-th minibatch, and $n_{i,j}^{\mathrm{queue}}$ denote the number of points belonging to the j-th class in the queue before the i-th minibatch. It should be noted that, while the maximum size of the queue is large enough, the situation that $n_{i,j}^{\mathrm{queue}} + n_{i,j}^{\mathrm{minibatch}} < N$ will not happen as long as the first minibatch satisfies $n_{1,j}^{\mathrm{minibatch}} > N$.

After getting n_{point} selections, the contrastive loss of each point in the minibatch is calculated, as shown in (1), where i is one of the n_{point} selected points and τ denotes the temperature parameter; P_i and N_i are the sets of positive and negative samples of point i, while i^+ and i^- are an element of P_i and N_i, respectively. In our case, $|P_i| = N - 1$ and $|N_i| = (n_{\mathrm{class}} - 1) \times N$. It is worth noting that a projection head is added to the backbone of our method, as shown in the green part of Fig. 1, and f_i in (1) represents the projected feature of point i.

The contrastive loss of the minibatch is the average loss of all selected points, i.e.,

$$L^{\mathrm{NCE}} = \frac{1}{n_{\mathrm{point}}} \sum_{i \in S} L_i^{\mathrm{NCE}}, \tag{2}$$

where S is the set of selected points in the minibatch. It satisfies $|S| = n_{\mathrm{point}}$; and for each point i, $S = \{i\} \cup P_i \cup N_i$. The loss in (1) is known as InfoNCE [22], which is widely used in contrastive learning. By backpropagation, features of the positive samples will be pulled together, and the features of the negative samples will be pushed away. Thus, in the feature space, the features of negative samples can be dispersed, while the features of positive samples are gathered.

2) Downstream Task Fine-Tuning: The process of downstream task fine-tuning is illustrated in the gray and blue parts of Fig. 1. The representation learning step only makes the model learn a suitable feature representation method for the radar detection points-based semantic segmentation task. That is, the model outputs the extracted features rather than the predicted classes of the points. As a result, downstream task fine-tuning is applied to make the model further learn how to classify the points using the learned features.

During fine-tuning, the projection head is replaced by a point-wise classification head, as shown in Fig. 1. Note that the variables in the backbone is frozen during the fine-tuning process and only the parameters in the newly connected head is adjustable. This training process is a typical transfer learning (using the labeled training data again) with cross entropy loss (L^{CE}) for point-wise classification.

Finally, DBSCAN is employed on the classified points, and a grid search with the evaluating metric of $\mathrm{mAP}_{0.5}$ is employed to find the best clustering parameters.

C. Semi-Supervised Setting

In the fully-supervised setting, a large amount of unlabeled training data are not used. In order to use these unlabeled points to enhance the segmentation accuracy of the network, we incorporate the "pseudo labels" into the training process. Note that the "pseudo labels" are only generated for the radar detection points without a label.

A simple way to generate pseudo labels is to apply the model trained in fully-supervised setting to predict the class of each point, and regard it as the groundtruth label to help define the positive and negative samples. Note that the prediction is not 100% accurate. If the accuracy is too low, the model may learn wrong representations for the points. In practice, a confidence threshold $T \in (0, 1)$ can be employed to increase the accuracy of the generated pseudo labels. The idea is that only prediction with a probability higher than the threshold will be retained for the training process. The detailed training process of contrastive learning and downstream task fine-tuning in the semi-supervised setting is the same as that in the fully-supervised setting, as described in Section III-B.

D. Joint Training of Representation Learning and Downstream Task

Inspired by [20], we merge the two steps of contrastive learning into a single overarching procedure. The process of joint training is illustrated in Fig. 1. Note that it can be performed in either fully-supervised setting or semi-supervised setting. Specifically, the projection head and the point-wise classification head are connected to the feature extractor at the same time, and the total loss is calculated as

$$L = L^{\mathrm{NCE}} + \alpha L^{\mathrm{CE}}, \tag{3}$$

where $\alpha > 0$ is a weighting hyper-parameter. By adopting joint training instead of non-joint training, an improved performance can be obtained. More details are discussed in Section IV.

E. Inference

During the inference process, the projection head of the model, the queue for storing positive and negative samples and the minibatch selecting process are all discarded. As illustrated in Fig. 2, the model takes the point cloud as input, where the feature extractor extracts point-wise features and the point-wise classification head predicts per-class score for each point. Finally, the DBSCAN is applied to obtain the predicted instance information.

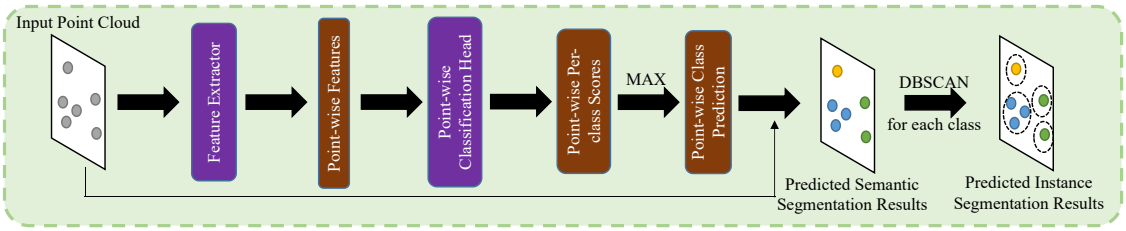

Fig. 2. The inference process of our proposed contrastive learning based model. During the inference process, only the point-wise classification head is connected, and DBSCAN is applied based on the semantic segmentation results to obtain the instance information.

IV. EXPERIMENTS

In this section, a brief description of the dataset is presented, and implementation details including the network architecture and hyper-parameters are introduced. Experimental results are presented with analyses and discussions.

A. Dataset

We choose RadarScenes [5] as the dataset of our experiments. There are 158 sequences of data containing 1,556,684 frames of radar detection point clouds, and the coordinates, velocity, radar cross section (RCS) as well as other features of points are provided. Only dynamic points are adopted as only road users are considered, and the 11 categories of non-static objects are merged into 5 (i.e., car, pedestrian, group of pedestrians, large vehicle and two-wheeler; as a result, $n_{\text{class}} = 5$) due to lack of data in some classes. These settings are suggested by the guidance of RadarScenes dataset.

In our experiments, the dataset is split into $8:1:1$, which is the ratio of frames in the training set, validation set and test set. Furthermore, we divide the training set into different proportions of labeled data and unlabeled data, whose labels are discarded in our experiments to simulate the situation that most of the data do not have any annotation.

B. Experiment Configurations

In this study, we selected PointNet++ [17] as the backbone of our model. The model structure is the same as that in [3], which is shown in Fig. 3(a). The projection head used in representation learning is a two-layer MLP with ReLU, and l_2 normalization is applied to the output of MLP, whose structure is illustrated in Fig. 3(b); the point-wise classification head is another two-layer MLP with Batch Normalization, ReLU and Dropout, as shown in Fig. 3(c).

PointNet++ can only take a batch of point clouds with the same number of points, so we set the sample size (N_{sample}) to 100 after calculating some statistics of the frames, i.e., 100 points are sampled from all points in each frame; in most case, there are not enough points in a frame, so some points may be sampled more than once. During representation learning, the duplicated points will be removed before selecting points in a minibatch, where the number of selected points (n_{point}) is set as 250. Therefore, $N = 50$ points of each class are selected in a minibatch.

The scheduler of cosine annealing warm restarts and the ADAM optimizer is adopted during training. The initial learning rate is 1e-2 in representation learning and joint training, while it is set to 5e-4 during downstream task fine-tuning. In fully-supervised setting, the batch size is 512, and the minibatch size ($n_{\text{minibatch}}$) is set to 32; in semi-supervised setting, batch and minibatch sizes are doubled.

C. Experimental Results and Analysis

Following [3], we choose the mean coverage (mCov) and mean average precision with the IoU threshold of 0.5 ($\text{mAP}_{0.5}$) as the evaluation metrics of our experiments for fair comparison. Table I exhibits the results on test data with different training strategies and methods, including the clustering-based classification method and semantic segmentation-based clustering method proposed in [3]. The former one is the most popular method in the industry and the latter achieves state-of-the-arts performance, so we employ them as our baselines of supervised learning. Correspondingly, for the baseline method of contrastive learning,

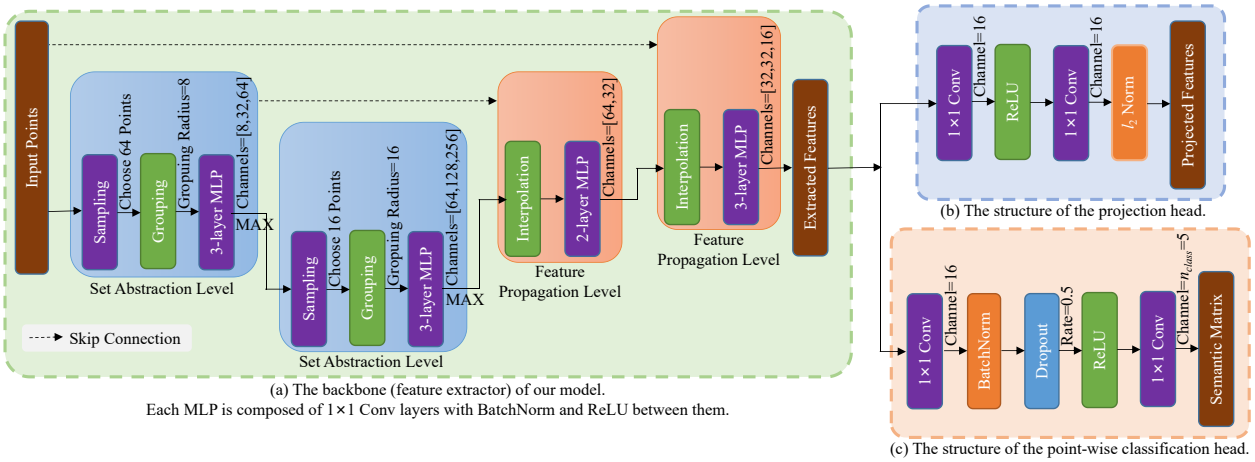

Fig. 3. The detailed configuration of our proposed contrastive learning based model, including that of the feature extractor and the two heads. The structure and hyper-parameters of the feature extractor is the same as that in [3].

TABLE I

RESULTS OF DIFFERENT STRATEGIES FOR INSTANCE SEGMENTATION ON TEST DATA

Learning Strategy				mCov/%	$mAP_{0.5}$/%
Supervised Learning	5% Training Data	Clustering + Random Forest Classifier Classification		73.18	68.19
	100% Training Data			79.54*	76.09*
	5% Training Data	Semantic Segmentation + Clustering		75.59	70.86
	100% Training Data			82.21*	77.96*
Contrastive Learning	5% Labeled Data	Baseline: Clustering + Classification for Defining Positive and Negative Samples		71.92	66.23
	5% Labeled Data	Non-joint Training (Representation Learning + Downstream Task Fine-tuning)	Fully-supervised Setting	77.47	72.41
			Semi-supervised Setting	78.58	73.82
		Joint Training	Fully-supervised Setting	77.57	72.83
			Semi-supervised Setting	79.00	74.01
	10% Labeled Data	Non-joint Training (Representation Learning + Downstream Task Fine-tuning)	Fully-supervised Setting	79.43	74.83
			Semi-supervised Setting	80.03	75.61
		Joint Training	Fully-supervised Setting	79.59	75.06
	20% Labeled Data	Non-joint Training (Representation Learning + Downstream Task Fine-tuning)	Fully-supervised Setting	80.72	76.48
			Semi-supervised Setting	81.06	76.82
	40% Labeled Data		Fully-supervised Setting	81.76	77.66
	100% Labeled Data		Fully-supervised Setting	**82.73**	**79.01**

* The results are from [3].

we apply the former idea to generating pseudo labels and defining the positive and negative samples.

Specifically, the point cloud is clustered by DBSCAN first and a random forest classifier trained on the labeled training data is applied on unlabeled training data to generate pseudo labels, according to which the positive and negative samples are defined. After that the whole training dataset is sent to the model and contrastive learning is performed, and the details are the same as that described in Section III-B. The reason of the poor performance is the high inaccuracy in the generated pseudo labels, which proves the importance of defining appropriate positive and negative samples.

For contrastive learning with different proportions of labeled training data, some of the experiments are omitted after we found the trend of the results obvious.

By comparing the results of different training strategies, we can draw the following three conclusions from Table I:

- **Contrastive learning performs better than supervised learning in case that few labeled data are provided.** The model trained in contrastive learning (non-joint training in fully-supervised setting) strategy attains around 2% improvement in performance compared to that trained in supervised learning when 5% of training data are used. By contrasting features of positive and negative samples, the model learns to extract more proper features helpful to the later semantic segmentation task, but traditional supervised learning suffers from the lack of training data.
- **Contrastive learning in semi-supervised setting outperforms that in fully-supervised setting.** When using 5% labeled training data, there exists an 1.5% gain after changing the fully-supervised setting to semi-supervised setting in both non-joint training and joint training. The phenomenon shows an evidence that more training data is beneficial to contrastive learning even if their labels are not completely correct. In addition, as the proportion of labeled training data increases, the performance gain decreases because there are less unlabeled training data. This inspires us that we could get performance improvement by collecting more frames of radar points even though they are unlabeled.
- **Joint training performs better than non-joint training.** By merging instead of separating the two steps of contrastive learning, there is a little improvement (0.2%-0.4%) on performance, which may be caused by the close relation between the representation learning and downstream task learning. The downstream task learning makes the feature extractor learn to extract features that are easy to be classified. For example, the features of points in different classes lie in different regions of the feature space, which is similar to the purpose of representation learning. The representation learning whose aim is to make the extracted features distributed like clusters in the feature space also makes the points easy to be classified. As a result, the two tasks in joint training can help each other so that the training is facilitated.

It can be seen from Table I that the performance of our method with 40% labeled training data is close to that of supervised learning with 100% training data. Moreover, when all training data are labeled, our method outperforms the supervised learning strategy [3] by about 0.5% mCov and 1% $mAP_{0.5}$, showing the superiority of our strategy.

The instance segmentation results of an example frame is shown in Fig. 4. It is obvious that the model trained in joint/semi-supervised contrastive learning achieves the highest accuracy, while the model trained in supervised learning makes the most mistakes in the example frame.

Although the training time of different training strategies of contrastive learning in Table I varies greatly, all the models have similar inference time (about 25ms per frame on an Intel Core i5-6300HQ CPU and with 8GB RAM) because the structure of feature extractor and the point-wise classification head is the same. Also, the storage memory of the models, as well as the number of parameters in the models is the same, which is 331KB and 75.245K respectively.

V. CONCLUSION

A contrastive learning strategy is proposed for radar detection points-based instance segmentation, to address the challenge of insufficient point-wise annotation of radar detection

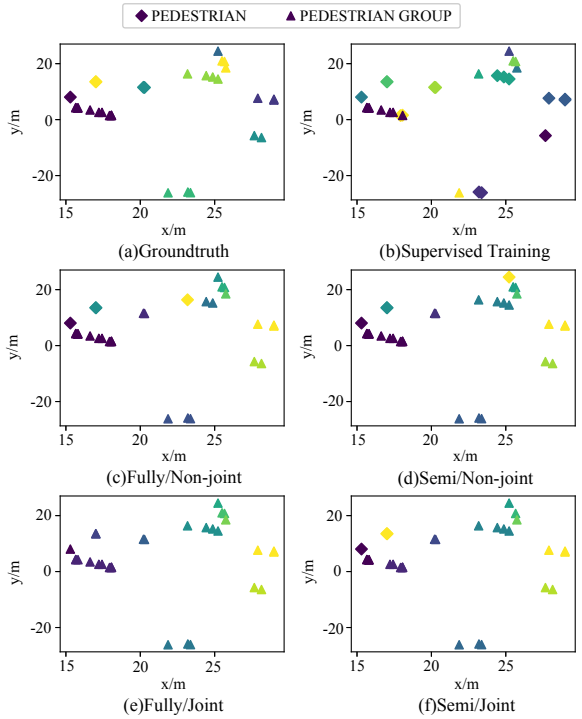

Fig. 4. The instance segmentation results of an example frame. 5% labeled training data are used to train the models. As sampling points are different during the inference process of each model, some points are missing in some subfigures. Different markers denotes different classes, while different colors in the same class represents different instances in this class.

points. We experimented that four different training settings, including fully-supervised setting/non-joint training, semi-supervised setting/non-joint training, fully-supervised setting/joint training, and semi-supervised setting/joint training, can be applied on the proposed contrastive learning strategy while keeping a small model size and low inference time. This result shows that it is feasible to embed the proposed model on an automotive radar-based ADAS product.

Further, we demonstrated via experiments that the performance of our proposed contrastive learning strategy with a few labeled training data is comparable to that of the supervised learning strategy with a large number of labeled training data. Moreover, when only a small amount of labeled training data is available, the performance of the supervised learning strategy becomes lower than that of ours, proving the superiority of our method.

It is acknowledged the proposed training strategy requires a longer training time compared to supervised learning. To reduce the training time is among our future work.

REFERENCES

[1] S. Alland, W. Stark, M. Ali, and M. Hegde, "Interference in automotive radar systems: Characteristics, mitigation techniques, and current and future research," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 45–59, 2019.

[2] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue, "Towards deep radar perception for autonomous driving: Datasets, methods, and challenges," *Sensors*, vol. 22, no. 11, p. 4208, 2022.

[3] J. Liu, W. Xiong, L. Bai, Y. Xia, T. Huang, W. Ouyang, and B. Zhu, "Deep instance segmentation with automotive radar detection points," *IEEE Trans. Intell. Veh.*, 2022.

[4] O. Schumann, J. Lombacher, M. Hahn, C. Wöhler, and J. Dickmann, "Scene understanding with automotive radar," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 188–203, 2019.

[5] O. Schumann, M. Hahn, N. Scheiner, F. Weishaupt, J. F. Tilly, J. Dickmann, and C. Wöhler, "Radarscenes: A real-world radar point cloud data set for automotive applications," in *Proc. Int. Conf. Inf. Fusion*, 2021, pp. 1–8.

[6] A. Ouaknine, A. Newson, J. Rebut, F. Tupin, and P. Perez, "Carrada dataset: Camera and automotive radar with range-angle-doppler annotations," in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 5068–5075.

[7] Y. Wang, Z. Jiang, X. Gao, J.-N. Hwang, G. Xing, and H. Liu, "Rodnet: Radar object detection using cross-modal supervision," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 504–513.

[8] H. Liu, C. Lin, D. Wu, and B. Gong, "Slice-based instance and semantic segmentation for low-channel roadside LiDAR data," *Remote Sens.*, vol. 12, no. 22, p. 3830, 2020.

[9] F. Zhang, C. Guan, J. Fang, S. Bai, R. Yang, P. H. Torr, and V. Prisacariu, "Instance segmentation of LiDAR point clouds," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2020, pp. 9448–9455.

[10] P. Xiong, X. Hao, Y. Shao, and J. Yu, "Adaptive attention model for LiDAR instance segmentation," in *Int. Symp. Vis. Comput.*, 2019, pp. 141–155.

[11] J. Behley, A. Milioto, and C. Stachniss, "A benchmark for LiDAR-based panoptic segmentation based on kitti," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2021, pp. 13 596–13 603.

[12] S. Gasperini, M.-A. N. Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari, "Panoster: End-to-end panoptic segmentation of LiDAR point clouds," *IEEE Robot. Autom.*, vol. 6, no. 2, pp. 3216–3223, 2021.

[13] C. Lin, H. Liu, D. Wu, and B. Gong, "Background point filtering of low-channel infrastructure-based LiDAR data using a slice-based projection filtering algorithm," *Sensors*, vol. 20, no. 11, p. 3054, 2020.

[14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, vol. 96, no. 34, 1996, pp. 226–231.

[15] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Supervised clustering for radar applications: On the way to radar instance segmentation," in *IEEE Mtt-S Int. Conf. Microwaves Intell. Mobil.*, 2018, pp. 1–4.

[16] H. Liu, Z. Dai, D. So, and Q. Le, "Pay attention to MLPs," *Adv. neural inf. proces. syst.*, vol. 34, 2021.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neural Inf. Proces. Syst.*, vol. 30, pp. 5099–5108, 2017.

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Int. Conf. Machin. Learn.*, 2020, pp. 1597–1607.

[19] X. Li, Y. Zhou, Y. Zhang, A. Zhang, W. Wang, N. Jiang, H. Wu, and W. Wang, "Dense semantic contrast for self-supervised visual representation learning," in *Proc. ACM Int. Conf. Multimed.*, 2021, pp. 1368–1376.

[20] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 7303–7313.

[21] X. Zhao, R. Vemulapalli, P. A. Mansfield, B. Gong, B. Green, L. Shapira, and Y. Wu, "Contrastive learning for label efficient semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 10 623–10 633.

[22] A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.