Attentive Prototypes for Source-free Unsupervised Domain Adaptive 3D Object Detection

Deepti Hegde
Johns Hopkins University

Vishal M. Patel
Johns Hopkins University

Abstract

*3D object detection networks tend to be biased towards the data they are trained on. It has been demonstrated that the evaluation on datasets captured in different locations, conditions or with sensors of different specifications than that of the training (source) data results in a drop in model performance due to the domain gap with the test (or target) data. Current methods for adapting to the target domain data either assume access to source data during training, which may not be available due to privacy or memory concerns, or require a sequence of LiDAR frames as an input. We propose a single-frame approach for source-free, unsupervised domain adaptation of LiDAR-based 3D object detectors that uses class prototypes to mitigate the effect of pseudo-label noise. Addressing the limitations of traditional feature aggregation methods for prototype computation in the presence of noisy labels, we utilize a transformer module to identify outlier regions that correspond to incorrect, over-confident annotations, and compute an **attentive class prototype**. The losses associated with noisy pseudo-labels are down-weighed in the process of self-training. We demonstrate our approach on two recent object detectors and show that our method outperforms recent source-free domain adaptation works as well as those that leverage source information during training. The code will be made available.*

1. Introduction

LiDAR datasets [1, 5, 9, 12, 25, 37] have facilitated the development of extremely effective data-driven perception algorithms for autonomous driving [20], but come with their own challenges. The weather conditions and the location of data capture lead to biases in the dataset due to the specific dimensions of roads, vehicles, and the driving conventions of the area [42]. Different LiDAR sensors possess different rates of return and produce point clouds with varying densities, leading to another set of inherent biases. This results in a distribution gap between pointcloud datasets. A 3D object detection network [27, 34–36, 48, 54, 56] trained on a particular dataset will drop in performance when evaluated on samples from a dataset with a different distribution [42]. We call the training and test datasets in this scenario as the source and target domain datasets respectively. One may argue that making use of a large, diverse source domain could solve this problem, however there will always be samples from an unseen distribution, and collecting every possible type of LiDAR scene is impractical at best. The need for robust perception systems in autonomous driving thus becomes extremely important, where there is a high likelihood of encountering scenes from a different distribution.

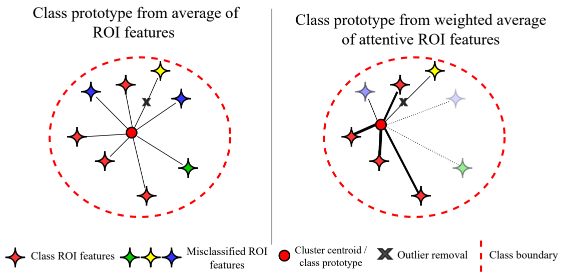
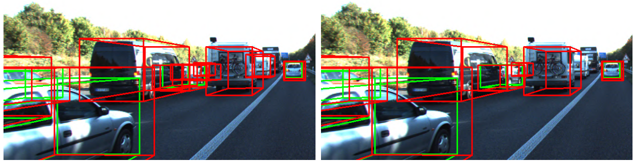

Class prototype from average of ROI features
Class prototype from weighted average of attentive ROI features
Class ROI features
Misclassified ROI features
Cluster centroid / class prototype
Outlier removal
Class boundary

Figure 1. **Left:** Visual representations of prototype computation. In the case of noisy labels, features corresponding to mis-labeled regions that are not discarded by outlier removal contribute to the class prototype. With the proposed method, only salient region features are considered for prototype computation. The opacity of the features represents the attention weights, and the width of the connecting lines represents the combination weights for computing the average.

Unsupervised domain adaptation has been broadly successful in addressing this problem for both 2D [14, 24, 31] and 3D [2, 21, 33, 46, 51] object detection networks, but depends on annotated source-domain data during adaptation, limiting its applicability in scenarios where it is unavailable due to privacy concerns or memory constraints. Recently, an alternate "source-free" setting for domain adaptation has been proposed [16, 19, 52], in which the source-trained

This work was supported by ARO grant W911NF2110135.



(a) All predictions

(b) *thresh = 0.7*

Figure 2. 3D bounding box predictions of the object detector [35] trained on Waymo [37] data and tested on KITTI [9]. Ground truth annotations are in green and predictions are in red. Thresholding (b) fails to remove all false positives present in (a). Self-training with these pseudo-labels leads to the reinforcement of errors.

model is adapted by training only on unlabelled target-distribution samples. pseudo-label based self-training has been successful in unsupervised and semi-supervised domain adaptive works [2,21,45,57] and lends itself to the setting of source-free adaptation. In this approach, the predictions of the source model are utilized as pseudo-labels [18] to supervise the network on un-annotated target data. However, current self-training-based methods rely on confidence thresholding to filter noisy pseudo-labels. As illustrated in the example from Figure 2, the use of high thresholds (as is general practice) results in training the model on easy samples and incorrect labels of high confidence that contribute to the enforcement of errors during adaptation. We suggest that a self-training framework that can deal with label-noise that requires only pseudo-annotated target data is an effective strategy for source-free domain adaptation.

We propose AttProto, an unsupervised, source-free domain adaptation framework for 3D object detection that addresses the issue of incorrect, over-confident pseudo-labels during self-training through the use of class prototypes. Objects belonging to the same object categories share geometric properties and similar feature representations. Given a representative class prototype, incorrectly labelled regions may be discarded based on the distance from the prototype. In the presence of label noise, standard feature aggregation methods of prototype computation [13, 15, 49, 55] are ineffective, since features corresponding to incorrectly labeled regions could contribute to the final prototype. This is illustrated in the left side of Figure 1. Equal importance is given to all annotated regions during aggregation, resulting in corrupted class prototypes in the presence of noise. Inspired by the high representative power of self-attention and recent works that make use of transformers to focus on salient inputs [8, 40], we calculate an attentive class prototype by using a transformer to identify salient regions-of-interest and combine their associated feature vectors using prediction entropy weights that represent the uncertainty of the classification branch for each sample. An illustration of the proposed prototype computation method can be seen on the right side of Figure 1. Once the attentive class prototype is calculated, the class predictions corresponding to incorrect pseudo-labels, which are identified by calculating the similarity with the class prototype, are down-weighed to prevent reinforcing errors during self-training. We demonstrate our result on several domain shift scenarios and compare the performance. Our contributions are as follows

- We propose the **attentive prototype** for learning representative class features in the presence of label noise by leveraging self-attention through a transformer block and perform source-free unsupervised domain adaptation of 3D object detection networks that mitigates the effect of label noise during self-training by filtering incorrect annotations.
- We demonstrate our method on two recent object detectors, SECOND-iou [48], and PointRCNN [35] for five domain shift scenarios and outperform recent source-free and standard domain adaptation works.

2. Related Works

3D object detection. The seminal works PointNet [27] and PointNet++ [28] for the hierarchical feature extraction of point clouds have spurred numerous deep neural networks for the task of 3D object detection that can be broadly categorised as voxel-based methods [17, 48, 56], which divide the pointcloud into volumetric grids before performing feature extraction, and point-based methods [35, 53] for 3D object detection, which operate directly each point in the 3D scene. In [48], Yan *et al.* propose SECOND, a single stage, voxel-based method that utilizes 3D sparse convolutions and a Region Proposal Network (RPN) head to predict the location and category of objects in a LiDAR scene. PointRCNN is a two-stage network that generates 3D bounding box proposals followed by a refinement stage similar to [30].
Unsupervised domain adaptation. The dominant unsupervised domain adaptation (UDA) approaches for 2D object detection are domain adversarial training [6, 32], distribution alignment [7, 47], and pseudo-label based self-training [14, 21, 33, 51]. Domain adversarial adaptation leverages a discriminator to learn a domain invariant feature space. Domain alignment methods take a more direct approach, and use distance measurements such as maximum mean discrepancy [10] to minimize the distribution gap between domains. The source-free setting rules out domain adversarial and domain alignment methods since they require samples from both the source domain and target domain. Thus, in this work focus on pseudo-label refinement through self-training.

Self-training [22] is an UDA method in which unlabelled target data is used for supervision by annotation with a pre-trained model. ST3D [51] is a self-training approach for 3D domain adaptive object detection where the network is adapted by training with a proposed curriculum data augmentation algorithm using pseudo-labels generated with a quality-aware memory bank. While showing promising results in some cases, in others this method is outperformed by simple pseudo-label based self-training, and depends on boosting with statistical normalization [42], a weakly supervised method, for its best reported results. The authors follow up this work with ST3D++ [50] which proposes improvements to the quality criterion for memory bank creation. While improving on performance, the method utilizes source data during adaptation. In [3], Caine *et al*. propose a method using the base network of Pointpillars [17] that trains a student network with a combination of source labels and target pseudo-labels obtained from a teacher network and filtered by a threshold. However, thresholding methods for pseudo-label collection from the source model may lead to training the model with incorrect labels that have high confidence and discarding correct labels with confidence that falls below the threshold.

Source-free domain adaptation. Source-free approaches for domain adaptation [16, 33, 52] only use unlabeled target data network models pre-trained on the source data during adaptation. Kundu *et al*. [16] propose a UDA method which does not require information about the category-level gap between domains, and consists of a procurement stage in which a generative classifier equips the model to reject out-of-distribution samples, and a deployment stage. Saltori *et al*. [33] propose a source-free UDA method on PointRCNN [35], utilizing a tracking-based scoring system to evaluate the quality of pseudo-labels at different scales. This method depends on the use of multiple frames for a single forward pass through the network.

Prototype learning. Learning representative features of a class or group of samples has been a well explored problem in pattern recognition. Originally calculated by aggregating hand-crafted features to form class exemplars [15], recent approaches use convolutional neural networks for more representative feature extraction. This method has seen success in a variety of tasks, including classification [49], zero-shot recognition [13], and domain adaptive 2D segmentation [55]. We argue that when utilizing pseudo-labels, clustering and outlier removal are insufficient in filtering out noise before the aggregation step, leading to corrupted class prototypes. We thus propose a transformer-based approach that leverages self-attention and predictive entropy to generate attentive class prototypes.

3. Proposed method

Consider an object detector network ϕ_s trained on a source dataset consisting of N sample-label pairs $\{X^S, Y^S\} = \{x_i^S, y_i^S\}_{i=1}^N$, where Y^S contains the annotations of objects in a 3D scene, consisting of the dimensions $\{l, w, h\}$, position $\{c_x, c_y, c_z\}$ and category of each bounding box. We aim to adapt this network model in absence of the source data to an unlabeled target dataset $X^T = \{x_j^T\}_{j=1}^M$ of size M and corresponding pseudo-labels $Y^{Ps} = \{y_j^{Ps}\}_{j=1}^M$ generated by thresholding the predictions of ϕ_s. The proposed domain adaptation method consists of a prototype computation and similarity-based refinement, implemented with an iterative training strategy. A visual representation of this framework is seen in Figure 3.

3.1. Transformer for prototype computation

Transformers have proven to be extremely effective for vision tasks [4, 8, 23, 39, 44]. By virtue of the self-attention mechanism, transformers have the ability to learn long-range relationships between elements in a sequence [8]. We leverage this representative power to learn the relationships between the region features of a given object category to form a class exemplar, or prototype.

In the presence of noisy labels, learning class representations through feature clustering methods like those in [13, 15, 49, 55] may compute corrupted class prototypes. In object detection, region features of different classes may be similar (such as the “Car” and “Truck” categories, see Figure 2), rendering outlier removal methods ineffective in cases of mis-classification. We use a transformer module to determine the useful elements in a sequence for a given task. We supplement the classifier branch of the object detection network with a transformer-based classification branch. The transformer utilizes self-attention to focus on salient regions-of-interest for prototype computation by learning the cross-correlation between regions. Transformers have proven to be very effective at learning global context [29]. We draw parallels to global and local context in transformers for language and consider context here to be the category of each region. In this case, global context becomes useful in weeding the uniquely different non-salient features. The MLP head performs classification of regions of interest to train the transformer on the objective of classifying region features.

Consider the set of features of the regions-of-interest (ROIs) $R_{feat} = \{f_i\}_{i \in N_{roi}}$ consisting of feature vectors generated by the object detector ϕ_s of meaningful 3D regions in the scene. In order to create a representative prototype, we take inspiration from [8] and send the ROI features as tokens to a transformer module consisting of a linear embedding layer and a set of transformer encoders, followed by a multi-layer perceptron (MLP) head. The encoder contains alternating multi-head attention blocks and feed-forward blocks, with interspersed normalization layers and residual connections, as depicted in Figure 3.

The input to the transformer module is the set of N_{roi}^+ “positive” ROI features, that is, features associated with the object category. These are obtained from the pseudo-annotations. Due to the noisy nature of the pseudo-label an-

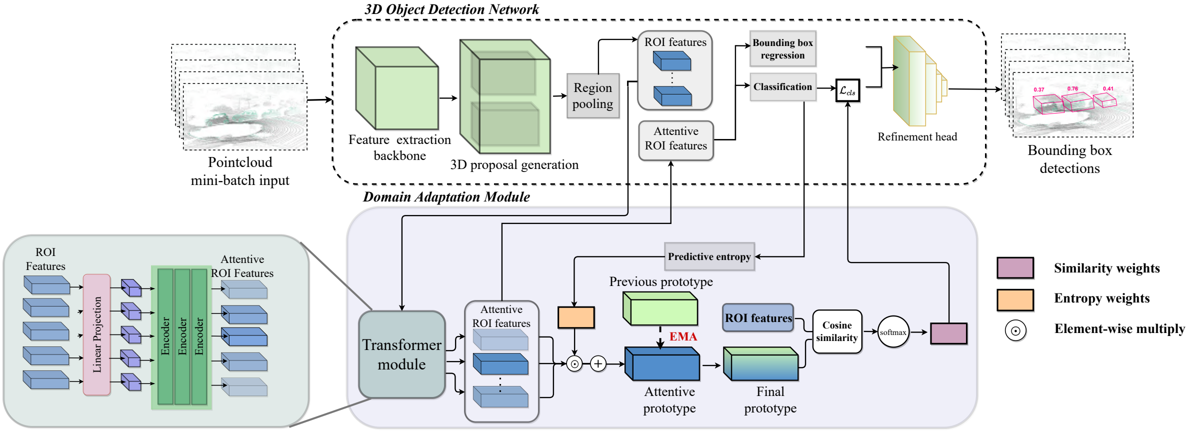


Figure 3. A visual overview of the proposed domain adaptation framework. The object detector is initialised with the source-trained model and used for region-proposal and feature extraction by the domain adaptation block. This consists of a transformer module to generate attentive features (see lower left side), prototype computation, and calculation of the output classification loss weights using cosine similarity.

notations, the set of features will contain regions associated with background points or objects outside of the category. We seek to identify these incorrect ROIs through adaptation.

The input sequence $\boldsymbol{f}$ of ROI features to the transformer module is of length N'_{roi}, where $N'_{roi} < N_{roi}$. Each feature f_i in $\boldsymbol{f}$ (comparable to the image patches in [8]) is sent as a token to the linear projection layer, which gives a set of feature embeddings $\boldsymbol{f'} \in \mathbb{R}^{N'_{roi} \times D}$ that are input to a series of encoders consisting of multi-head self attention layers [40] and fully connected layers.

The feed-forward block is a two-layers multi-layer perceptron (MLP) with the GELU [11] non-linearity function. Following the forward pass through L such encoders, the final output of the transformer module is a set of attentive region features $\{f^i_{att}\}_{i \in N'_{roi}}$. By virtue of the self-attention mechanism, we learn the cross-correlation between positive and negative region features, and in turn the ROIs that contribute salient information for prototype computation. The attended region features are passed to the classification branch. The transformer is thus optimized to perform classification of ROIs through the detection head under a cross entropy loss.

Predictive entropy. A classifier supervised with noisy pseudo-labels will generate predictions with a higher level of uncertainty. This uncertainty may be quantified as predictive entropy [26, 41]. Using the predictive entropy of the classifier allows us to disregard regions that the classifier is uncertain about. This uncertainty stems from being supervised by inconsistent and noisy labels, and thus can further aid in identifying salient ROIs. As a way to obtain insight on how informative each region feature is for the bounding box categorization task, we form the attentive class prototype as the weighted average of the attentive region features $\boldsymbol{f}_{att}$ weighted with the predictive entropy of the classifier branch C_{ϕ_s} of the object detection network parameterized by Φ. The predictive entropy of a classifier under a domain shift is a useful metric in estimating the uncertainty [26]. Instead of directly utilizing the uncertainty of the model to weigh samples as in [26], we weigh each attentive region feature with the confidence of the associated prediction. From [41], the predictive entropy is given by

$$\mathcal{H}(Y|\boldsymbol{x};\Phi) = -\sum_{c=1}^{N_c} p_\Phi(Y=c|\boldsymbol{x}) \log\{p_\Phi(Y=c|\boldsymbol{x})\},$$

where N_c is the number of classes The entropy weights for each ROI are denoted by the vector $\boldsymbol{E}$ which is obtained by

$$\boldsymbol{E} = 1 - \frac{\mathcal{H}(Y|\boldsymbol{x};\Phi)}{\sum_{x \in X} \mathcal{H}(Y|\boldsymbol{x};\Phi)}. \tag{1}$$

The higher the entropy of a prediction, the more uncertain the network is about the ROI. We wish to down-weigh the losses associated with high-entropy regions. The attentive prototype as the entropy weighted average of the attentive ROI features is thus obtained by

$$F_{att} = \frac{1}{N^+_{roi}} \sum_{i=1}^{N^+_{roi}} E_i f^i_{att}. \tag{2}$$

Computing the final prototype. During the process of training the object detector for adaptation, samples from the target dataset $X^T = \{x^T_j\}^M_{j=1}$ are sent in mini-batches. Each attentive prototype computed in an iteration is combined with the prototoype computed in the previous iteration through exponential moving average (EMA). The initial prototype at the first iteration of the first epoch is simply the average of the positive ROI features. The final attentive prototype at each iteration j is thus given by

$$F^{j}_{final} = \alpha F^{j-1}_{final} + (1-\alpha) F^{j}_{att}, \tag{3}$$

3.2. Similarity-based refinement

Once the representative class prototype is obtained, we use it as a soft-filter to identify region features that are dissimilar and thus far away from each other in the feature space. To do this, the distance of each positive ROI in R_{feat} from its corresponding prototype is calculated using the metric of cosine similarity such that

$$d^{i} = \frac{\sum_{k=1}^{K} F^{k}_{final} \cdot f^{i,k}}{\sqrt{\sum_{k=1}^{K} (F^{k}_{final})^2}\sqrt{\sum_{k=1}^{K} (f^{i,k})^2}}, \tag{4}$$

where K is the feature dimension and i ranges from 1 to the number of ROIs, N_{roi}. Cosine similarity is chosen due to the sparse nature of the features. It is a popular metric for measuring the distance between features in a latent space [13, 49, 55]. The classification loss corresponding to each positive region of interest is multiplied by a similarity weight, computed by taking the softmax of the cosine distance. The loss corresponding to the classification task in the object detection network is thus given by,

$$\mathcal{L}'_{cls} = \frac{\beta}{N_{roi}}(\sum_{i \in roi^{+}} d^{i} \ell^{i}_{cls} + \sum_{j \in roi^{-}} \ell^{j}_{cls}) \tag{5}$$

where β and γ are constants, $\ell^{i/j}_{cls}$ corresponds to the region-wise loss of the bounding box classifier, where i indexes the positive regions and j indexes the negative regions. With this similarity-based down-weighing, the losses corresponding to regions that have been identified as incorrect through prototype matching will be down-weighed and not contribute to training. As the representative prototype improves with each epoch, the network becomes better at soft-filtering incorrect regions and avoids reinforcing the error in pseudo-labels.

4. Experiments

We demonstrate our domain adaptation framework on two object detection networks, the voxel based network SECOND-iou [51], [48], and PointRCNN [35], a two stage point-based network. In this section, we explain the details of the experiments and the datasets used.

Datasets. In order to simulate the various domain shifts, we consider three popular large-scale autonomous driving datasets with considerable domain gaps among them, the Waymo Open Dataset [37], the KITTI dataset [9], and the nuScenes dataset [1]. The largest dataset among these is Waymo, with more than 230K annotated LiDAR frames collected across six US cities, of which we use approximately 50K due to memory constraints in a 40K/10K training/validation (test) split. We follow the standard sampling procedure for training with a subset of the Waymo dataset, and maintain the same splits. The nuScenes dataset consists of 34,149 frames which we utilize in 28K/6K split. The smallest dataset is KITTI, consisting of 7,481 (3K/3K split) annotated LiDAR frames collected from Germany. All training and validation splits used are official and consistent with that used by other works in the 3D object detection literature. We use an inductive setup, in which test data is not seen during adaptation.

Object detection networks. The network SECOND-iou [48,51] is a two stage voxel-based 3D object detector which uses a PointNet [27] backbone to extract voxel features from groupings of points and consists of a grouping layer, a region proposal network (RPN) and an ROI refinement head. It is a modified version of SECOND [48] proposed in [51], with the extra refinement head. The region features for prototype computation are obtained at the output of the RPN head, and the classification loss at this stage is down-weighed during adaptation. PointRCNN is a two stage point-based network with a similar PointNet backbone [27] for feature extraction that generates 3D region proposals through a bottom-up approach through foreground segmentation. This is followed by a refinement network. We implement adaptation for PointRCNN in this second stage. Our method is agnostic to the underlying object detection network, and only assumes the existence of a region proposal head

Online training procedure. The network is initialized with the source-trained model and trained the pseudo-annotated target data. Every k iterations, the network is inferenced to obtain predictions that are filtered by a low threshold value t, which are then used to update the pseudo-ground truth annotations. In this way, the network is supervised by progressively refined pseudo-labels. We follow a similar thresholding process as [51] during each inference stage.

Implementation details. For the implementation of SECOND-iou, we use the public pyTorch repository OpenPCDet [38], and the official code release from [35] for the implementation of PointRCNN. We perform experiments with a 48GB Quadro RTX 8000 GPU and a 16GB GeForce RTX 2080 GPU. The network is trained from scratch and randomly initialized. During adaptation, the 3D object detection network is initialized with the source trained model and the transformer module is randomly initialized. We follow the lengths of training recommended by the authors in the case of each object detector network. Each network is trained with a cyclic Adam optimizer with an initial learning rate [ADD].

Table 1. A tabular comparison of 3D average precision (AP) results of the "Car" object for the adaptation of two object detection networks SECOND-iou [48], and PointRCNN [35] against recent DA methods. Where the target dataset is KITTI, we evaluate with the official metric across 3 difficulty categories for an IoU threshold of 0.7. In the case of the nuScenes target dataset, we average across the various difficulty categories of the official metric. The best results are in bold blue type, the second best results are in red type. The $*$ indicates that source-data or information was used during adaptation.

SECOND-iou			Point-RCNN				
Domain Shift	Method	AP	Domain Shift	Method	AP		
					easy	mod.	hard
Waymo to nuScenes	DT	15.25	Waymo to KITTI	DT	13.11	12.10	12.24
	ST	17.80		ST	21.13	19.29	18.28
	*SN [42]	18.57		*SN	48.7	47.1	**49.7**
	ST3D [51]	20.19		SF-UDA^{3D} [33]	-	-	-
	AttProto	**20.38**		AttProto	**62.11**	**53.08**	46.64
	Oracle	32.64		Oracle	81.61	74.36	74.49

Domain Shift	Method	AP			Domain Shift	Method	AP		
		easy	mod.	hard			easy	mod.	hard
nuScenes to KITTI	DT	18.37	17.31	16.09	nuScenes to KITTI	DT	10.59	10.76	10.64
	ST	53.03	37.71	35.18		ST	22.21	11.56	11.90
	*SN	22.03	18.51	18.04		*SN	60.35	54.82	52.78
	ST3D	70.95	54.13	**51.78**		SF-UDA^{3D}	68.80	49.80	45.00
	AttProto	**71.34**	**56.51**	45.86		AttProto	**69.10**	**60.22**	**53.16**
	Oracle	84.86	68.93	67.38		Oracle	81.61	74.36	74.49

Table 2. A tabular comparison of 3D average precision (AP@R40) results of the "Car", "Pedestrian", and "Cyclist" categories for the adaptation of SECOND-iou [48] against recent domain adaptation methods. We evaluate with the official metric across the difficulty category "moderate" for an IoU threshold of 0.7. The best results are in bold blue type, the second best results are in red type. The $*$ indicates that source-data was used during adaptation.

Domain shift	Method	AP		
		Car	Ped.	Cyc.
Waymo to KITTI	DT	58.03	45.13	46.08
	*SN	59.20	50.44	41.43
	ST3D	62.19	48.33	46.09
	*ST3D++	65.10	53.87	**53.43**
	AttProto	**66.86**	**55.38**	48.46
	Oracle	73.45	41.33	60.32

5. Results

In this section we demonstrate the results of our domain adaptation framework and compare it against four domain adaptation methods; *Direct transfer (DT)*: Inference of the source trained model on target data, *Current SOTA*: ST3D [51] by Yang *et al.* and SF-UDA^{3D} [33] by Saltori *et al.* for their corresponding base networks, *Statistical normalization (SN)* [42]: weakly supervised approach that uses the target domain bounding-box statistics, *Pseudo-label self-training (ST)*: Re-training the object detector on thresholded source-model generated pseudo-labels. For reference we, also compare with the "oracle" results, which are obtained by training the detector with the ground truth labels of the target domain, indicating the possible upper bound of performance after adaptation. While ST3D++ [50] is closely related to [51], it is not a source-free method, and trains the network with source-dataset samples along with unlabelled samples. Nevertheless, we compare our performance against this work and find that we outperform it in several categories. We maintain the experimental settings of [33,43,51] and demonstrate our results on the "Car" category for established domain shift scenarios. For a comparison with [50], we demonstrate our framework on more categories.

5.1. Comparison with state-of-the-art

Quantitative analysis. We report the mean average precision (AP) of 3D bounding boxes across various difficulty settings for the domain shift scenarios Waymo → KITTI,

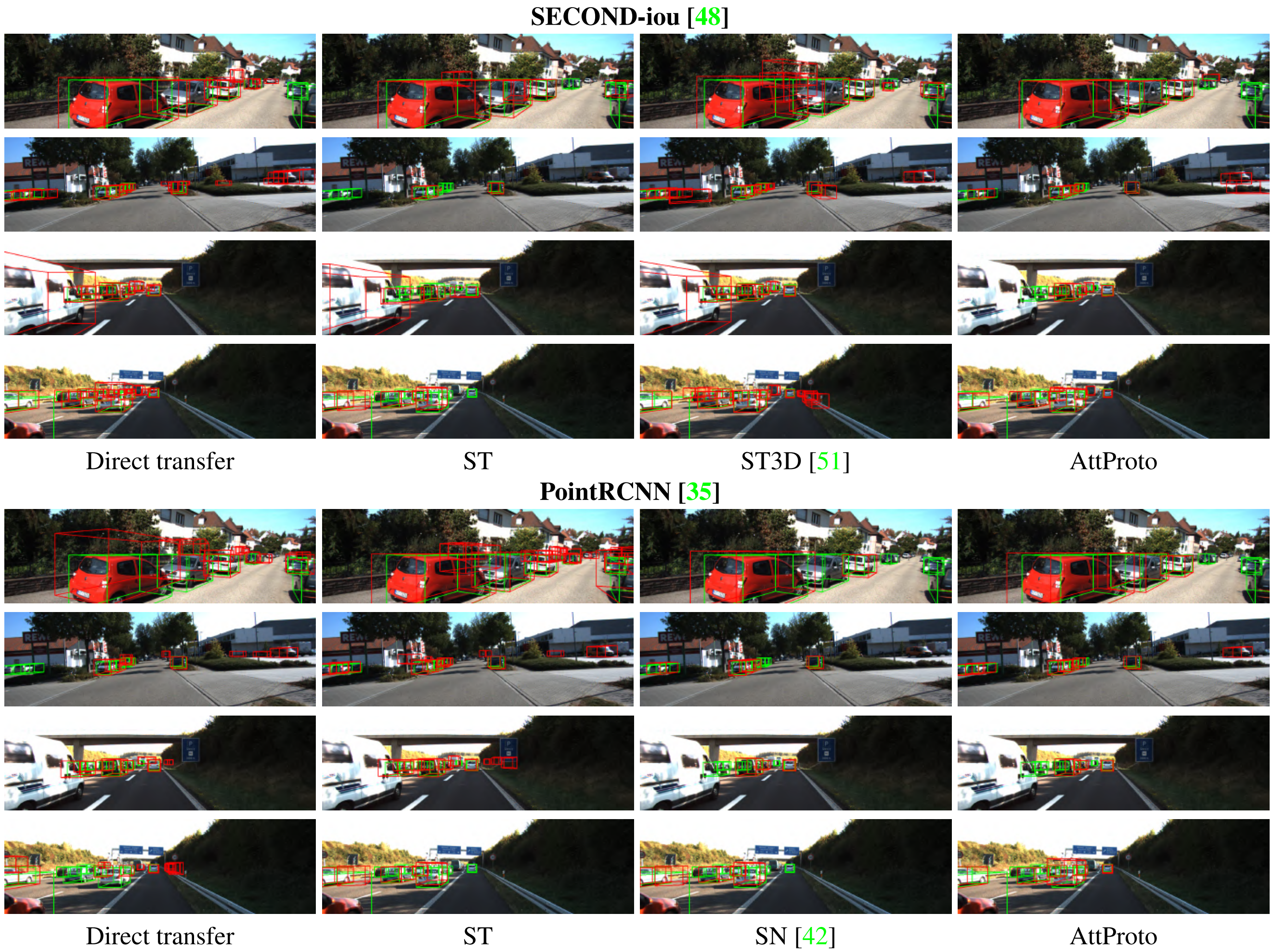


Figure 4. A qualitative comparison of our bounding box predictions of the "Car" class for the adaptation of SECOND-iou [48] (rows 1-4) and PointRCNN [35] (rows 5-8), with direct transfer (DT), pseudo-label self-training (ST), statistical normalization (SN) [42], and ST3D [51]. Ground truth annotations are in green and predictions are in red. Best viewed zoomed in and in color.

Table 3. Ablation results. A comparison of 3D mAP performance for the "Car" category of the adaptation of SECOND-iou using different prototype computation methods.

Prototype	nuScenes to KITTI		
	easy	mod.	hard
Average	65.00	47.55	40.24
Self-attention	65.01	47.54	42.43
Transformer	69.26	50.09	45.11
Transformer + entropy weight	71.56	52.12	45.86

nuScenes→ KITTI, and Waymo→ nuScenes to be consistent with the reports of [33]. Where the target dataset is KITTI, we use the official evaluation metrics detailed in [9] with easy, moderate, and hard difficulty categories based on the distance and level of occlusion of the object from the sensor, with an IoU threshold of 0.7. Where the target dataset is nuScenes, we use the metrics of [1], and average across difficulties as is done in [51] and [33]. Due to the lack of a code repository for SF-UDA^{3D} at the time of writing, we compare with the reported numbers. We implement our method with SECOND-iou for comparison with ST3D and with PointRCNN for comparison with SF-UDA^{3D} to be consistent with their base object detector networks. This can be seen in Table 1. We demonstrate the best results using both object detection networks in most categories, beating the weakly supervised approach of SN as well as [33] and [51]. In Table 2, we perform domain adaptation for three object categories "Car", "Pedestrian", and "Cyclist" for the domain shift of Waymo → KITTI and outperform with the current state-of-the art UDA approaches ST3D and ST3D++. Despite ST3D++ using source data during adaptation, we outperform it in the cases of Car and Pedestrian, and perform second best in the Cyclist category.

Qualitative analysis. We further demonstrate the effectiveness of our domain adaptation framework with those

Table 4. A comparison of 3D mAP performance for the "Car" category of the adaptation of SECOND-iou with and without updating the region features with the transformer output values during classification.

Transformer	Entropy weighing	RoI update	Waymo to KITTI		
			easy	mod.	hard
✓	✓	✗	76.77	65.96	61.52
✓	✓	✓	79.12	66.86	62.29

of recent methods through a visual comparison of the predicted bounding boxes for the Waymo → KITTI task for both object detection networks in Figure 4. The problems of direct transfer (DT) of the source model are localization and over-confident false positives (see column 1), where objects such as large vans are incorrectly identified as a "Car". This problem is mitigated only partially by pseudo-labelling methods and by ST3D, and is better addressed by our approach. We also demonstrate improved localization of correctly classified objects, as can be seen in row 1.

5.2. Ablation study

We analyze the contribution of the different segments of our domain adaptation framework. In Table 3, we compare the 3D AP performance of the network using four different prototype computation methods for the SECOND-iou object detector; *(i) Average*: Class prototype is computed by taking the mean of positive region features, *(ii) Self-attention*: Prototype is computed by taking the mean of attentive region features, which are the result of sending region features to a single multi-head self attention block. *(iii) Transformer*: Prototype is computed by taking the mean of attentive region features, which are the result of sending region features to the transformer module. *(iv.) Transformer with entropy weights*: Prototype is computed by taking the prediction entropy weighted mean of transformer generated attentive region features. This is the best performing approach. In Table 4, we demonstrate the importance of updating the region features after the forward pass through the transformer module. The network benefits from updating the positive region features with the output of the transformer block before the region classification branch, as is apparent in the relative performance with the RoI feature update.

In order to visualize the quality of the pseudo-labels at different stages of training, we plot the confidence score histogram of 500 correct ($IOU > 0.6$) and incorrect pseudo-labels generated by the source only model and 3 consecutive meta-iterations. In the case of correct pseudo-labels, we desire that the model generate confident labels with higher IoU scores (distribution should lean right) and in the case of incorrect pseudo-labels, be uncertain about generated labels with low IoU scores (distribution should lean left). We observe this in the labels generated after adaptation. They are less noisy than that of the source-generated labels, indicated by the fact that incorrect samples of meta-iteration 3 tend to be distributed with lower confidence than that of DT labels. This shows that the quality of the pseudo-labels improves after adaptation.

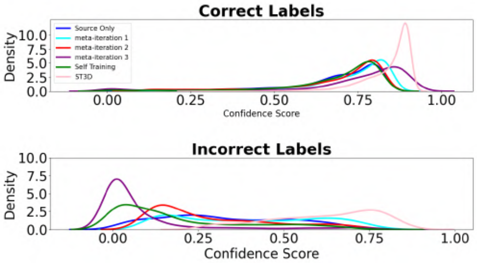


Figure 5. Histogram plot of confidence scores of correct ($IOU > 0.6$) and incorrect PL at direct transfer of the source-only model, self-training (ST), ST3D [51], and at each stage of training of the proposed method.

6. Limitations

Although our proposed approach out-performs existing source-free and source-trained UDA approaches overall, there are cases in which it under-performs, namely in the "hard" category. "Hard" objects are characterized by high occlusion, truncation, and far distance from the LiDAR sensor, and tend to be underrepresented in the dataset. Additionally, poor localization performance of the source-trained network may result in even fewer instances of these samples in the the pseudo-annotated dataset. Underrepresented samples contribute less to the class prototype, thus affecting self-training. We plan to target this limitation in follow-up work by exploring techniques such as label-free augmentations.

7. Conclusions and future work

In this paper we proposed a source-free domain adaptation framework for unsupervised domain adaptive 3D object detectors that uses a transformer module to compute an attentive class prototype to perform pseudo-label refinement during self-training. Our method outperforms other recent domain adaptation networks for several different domain shifts. We mainly address pseudo-label noise related to the false positive mis-classification of regions in the 3D scene, and not the dimensions of the bounding box. A factor that contributes the drop in performance upon domain shift is the difference in average size of the vehicles in different locations [42]. While they address this by the weakly supervised approach of statistical normalization, in the future we hope to provide a fully unsupervised solution.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Q. Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020. 1, 5, 7

[2] Benjamin Caine, R. Roelofs, Vijay Vasudevan, Jiquan Ngiam, Yuning Chai, Z. Chen, and Jonathon Shlens. Pseudo-labeling for scalable 3d object detection. *ArXiv*, abs/2103.02093, 2021. 1, 2

[3] Benjamin Caine, Rebecca Roelofs, Vijay Vasudevan, Jiquan Ngiam, Yuning Chai, Z. Chen, and Jonathon Shlens. Pseudo-labeling for scalable 3d object detection. *ArXiv*, abs/2103.02093, 2021. 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[5] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019. 1

[6] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. pages 3339–3348, 06 2018. 2

[7] Fatemeh Dorri and Ali Ghodsi. Adapting component analysis. In *2012 IEEE 12th International Conference on Data Mining*, pages 846–851, 2012. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 3, 4

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 5, 7

[10] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 2

[11] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[12] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. *arXiv preprint arXiv:2006.14480*, 2020. 1

[13] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *Proceedings of the European conference on computer vision (ECCV)*, pages 118–134, 2018. 2, 3, 5

[14] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G. Macready. A robust learning approach to domain adaptive object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 480–490, 2019. 1, 2

[15] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998. 2, 3

[16] Jogendra Nath Kundu, Naveen Venkat, V. RahulM., and R. Venkatesh Babu. Universal source-free domain adaptation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4543–4552, 2020. 1, 3

[17] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 2, 3

[18] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 2

[19] Xianfeng Li, Weijie Chen, Di Xie, Shicai Yang, Peng Yuan, Shiliang Pu, and Yueting Zhuang. A free lunch for unsupervised domain adaptive object detection without source data. In *AAAI*, 2021. 1

[20] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A Chapman, Dongpu Cao, and Jonathan Li. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3412–3432, 2020. 1

[21] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gong-Duo Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. *ArXiv*, abs/2107.11355, 2021. 1, 2

[22] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Citeseer, 2006. 2

[23] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 3

[24] Poojan Oza, Vishwanath A. Sindagi, Vibashan VS, and Vishal M. Patel. Unsupervised domain adaptation of object detectors: A survey, 2021. 1

[25] Matthew Pitropov, Danson Evan Garcia, Jason Rebello, Michael Smart, Carlos Wang, Krzysztof Czarnecki, and Steven Waslander. Canadian adverse driving conditions dataset. *The International Journal of Robotics Research*, 40(4-5):681–690, 2021. 1

[26] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8505–8514, 2021. 4

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 1, 2, 5

[28] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[29] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021. 3

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 2

[31] Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik G. Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 780–790, 2019. 1

[32] Kuniaki Saito, Y. Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6949–6958, 2019. 2

[33] Cristiano Saltori, St'ephane Lathuili'ere, N. Sebe, E. Ricci, and Fabio Galasso. Sf-uda3d: Source-free unsupervised domain adaptation for lidar-based 3d object detection. *2020 International Conference on 3D Vision (3DV)*, pages 771–780, 2020. 1, 2, 3, 6, 7

[34] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1

[35] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 1, 2, 3, 5, 6, 7

[36] Shaoshuai Shi, Zhe Wang, X. Wang, and Hongsheng Li. Part-a2 net: 3d part-aware and aggregation neural network for object detection from point cloud. *ArXiv*, abs/1907.03670, 2019. 1

[37] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1, 2, 5

[38] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. `https://github.com/open-mmlab/OpenPCDet`, 2020. 5

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 4

[41] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE, 2014. 4

[42] Yan Wang, X. Chen, Yurong You, Li Erran, Bharath Hariharan, M. Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11710–11720, 2020. 1, 3, 6, 7, 8

[43] Yan Wang, Xiangyu Chen, Yurong You, Li Erran, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 6

[44] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[45] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 2

[46] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R. Qi, and Dragomir Anguelov. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1

[47] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2272–2281, 2017. 2

[48] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. 1, 2, 5, 6, 7

[49] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3474–3482, 2018. 2, 3, 5

[50] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. 08 2021. 3, 6

[51] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 5, 6, 7, 8

[52] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021. 1, 3

[53] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 2

[54] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Ipod: Intensive point-based object detector for point cloud. 12 2018. 1

[55] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12409–12419, 2021. 2, 3, 5

[56] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 1, 2

[57] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 2