2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Finetuning Pre-trained Model with Limited Data for LiDAR-based 3D Object Detection by Bridging Domain Gaps

Jiyun Jang, Mincheol Chang, Jongwon Park, and Jinkyu Kim

***Abstract*— LiDAR-based 3D object detectors have been largely utilized in various applications, including autonomous vehicles or mobile robots. However, LiDAR-based detectors often fail to adapt well to target domains with different sensor configurations (e.g., types of sensors, spatial resolution, or FOVs) and location shifts. Collecting and annotating datasets in a new setup is commonly required to reduce such gaps, but it is often expensive and time-consuming. Recent studies suggest that pre-trained backbones can be learned in a self-supervised manner with large-scale unlabeled LiDAR frames. However, despite their expressive representations, they remain challenging to generalize well without substantial amounts of data from the target domain. Thus, we propose a novel method, called Domain Adaptive Distill-Tuning (DADT), to adapt a pre-trained model with limited target data (≈100 LiDAR frames), retaining its representation power and preventing it from overfitting. Specifically, we use regularizers to align object-level and context-level representations between the pre-trained and finetuned models in a teacher-student architecture. Our experiments with driving benchmarks, i.e., Waymo Open dataset and KITTI, confirm that our method effectively finetunes a pre-trained model, achieving significant gains in accuracy.**

I. INTRODUCTION

LiDAR-based 3D object detection has emerged as a fundamental task in autonomous driving (AD) and robotics, and recent works [1], [2], [3], [4], [5], [6] have achieved promising results. However, such models must be trained with large-scale annotated data, which is expensive and time-consuming. Moreover, their performance is often limited to in-domain data distribution, as discussed in literature [7], [8], [9], [10], [11], [12] – they may not adapt well to target domains with different sensor configurations (e.g., types of sensors, sensor's spatial resolution, density, and FOVs) or geometric location shifts (e.g., inferencing in different cities or countries). A common practice to address this issue would be new (large-scale) data collection and annotation in target domains, which are laborious and costly. Thus, it is highly demanded that a model can be continuously adapted well to target domains without needing large-scale annotated data.

Recent studies explored self-supervised representation learning on large-scale unlabeled point clouds [16], [17], [18], [19], [14], showing promising results in the 3D detection downstream task. For example, AD-PT [14] suggested a general representation model pre-trained with 1M unlabeled driving scenes from ONCE [20], followed by diverse data augmentation for robustness, showing notable performance on various AD benchmarks. However, we empirically observe that they still need substantial amounts of data to finetune the pre-trained model in a target domain (with domain gaps from source domains). For example, we provide UMAP [13] visualizations in Fig. 1 (a) where we compare (i) 2D Bird's Eye View (BEV) features of AD-PT model finetuned with 100% labeled KITTI [15] data (blue dots) and (ii) those from AD-PT models finetuned with different percentages (1%, 5%, 20%) of labeled data (red dots). This demonstrates that the mismatch between pre-training (source) and downstream (target) datasets can negatively affect the performance of downstream tasks with limited data, sub-optimally leveraging the expressive pre-trained models.

We argue that a naive finetuning approach of a pre-trained model may harm the robustness of a model against distribution shifts, providing degraded performance (though the pre-trained model was trained with large-scale data).

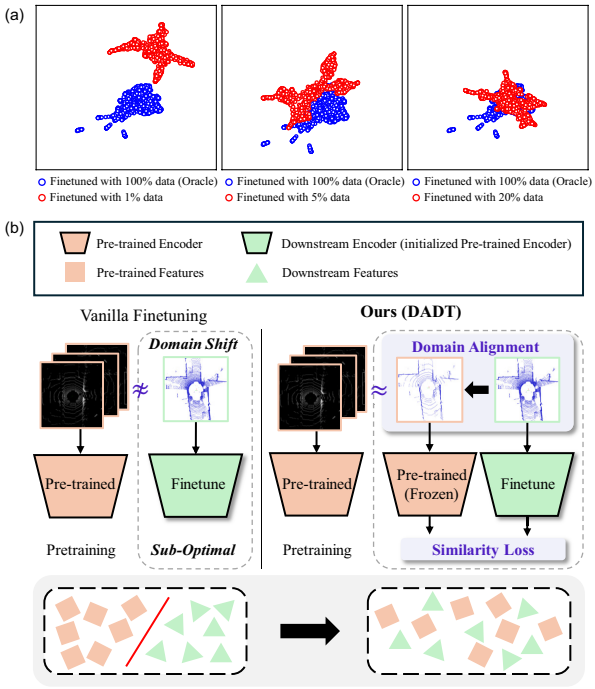


Fig. 1: **(a) UMAP [13] Visualizations.** We compare 2D BEV features between (i) the oracle model (a pre-trained model by AD-PT [14] finetuned with the whole KITTI [15] training data) and (ii) similarly finetuned models but with smaller datasets. **(b) An Overview of Architectures.** Conventional finetuning approaches (left) and our proposed Domain Adaptive Distill-Tuning (DADT) approach (right).

J. Jang, M. Chang, and J. Kim are with Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea.
J. Park is with Autonomous Driving Center, Hyundai Motor Company R&D Division.

Inspired by recent work in the domain generalization task, we opt for a strategy where models learn similar features to those of approximation of "oracle" representations, which can be generalized well across any domain. Specifically, given a large pre-trained model as an approximation, we finetune a model with an objective of two components: (i) the original object detection task (i.e., Empirical Risk Minimization objective) and (ii) a regularization term between the pre-trained model (i.e., approximation of "oracle") and the target model. We simultaneously leverage the pre-trained model as the initialization and approximation of the oracle model.

Here, as shown in Fig. 1 (b), we propose a novel finetuning approach called Domain Adaptive Distill-tuning (DADT), which is based on teacher-student architecture where a teacher network utilizes the frozen pre-trained backbone with density-aligned LiDAR inputs (target domain's LiDAR points are resampled to match those of source domain) and a student network finetunes its backbone (initialized from the pre-trained backbone) with the original density-non-aligned LiDAR inputs. We further use two BEV-based regularization terms, i.e., (i) object similarity loss and (ii) context similarity loss, to tie representations both for the teacher and student network together during the finetuning step, retaining oracle model's generalizable representations and preventing it from overfitting. Our extensive experiments with driving benchmarks, such as the Waymo Open dataset [21] and KITTI [15], demonstrate that our method effectively finetunes a pre-trained model with limited target data ($\approx$ 100 LiDAR frames), achieving significant gains in accuracy. Our contributions are summarized as:

- We propose a novel approach, called Domain Adaptive Distill-tuning (DADT), which aims to leverage and retain the representation power of a pre-trained model to effectively adapt to target domains with limited data.
- We propose a teacher-student architecture to alleviate distributional misalignments between the source (or data for pre-training) and target domains (or data for finetuning), followed by regularizations to align representations of the teacher and student networks.
- We conduct extensive experiments with driving benchmarks, including Waymo Open dataset and KITTI, to demonstrate the effectiveness of our proposed method.

II. RELATED WORK

A. Self-Supervised Pre-Training in 3D Object Detection

Self-supervised pre-training [22], [23], [24] has drawn considerable attention in contemporary research on LiDAR-based 3D object detection, due to its efficacy in learning point cloud representation without labels and transferability to downstream task with small data. Notably, GCC-3D [16] introduces a framework incorporating geometry-aware contrast in contrastive learning paradigm. PointContrast [25], ProposalContrast [17] leverage point-level and region-level contrast to find correlation between different views. In the context of masked autoencoders (MAE), MAEs such as VoxelMAE [19], Occupancy-MAE [26] employ voxel-level masking to reconstruct masked points with decoder. Recently, GD-MAE [18] and MV-JAR [27] adopt masking approaches based on transformer architectures. Different from previous works that pretrain and finetune with the same dataset, AD-PT [14] proposes a diversity-based pretraining on ONCE [20] dataset to learn unified representations, enabling finetuning on multiple AD datasets. Though impressive, AD-PT suffers from suboptimal performance during finetuning stage due to ill-posed domain shift between LiDAR datasets.

B. Unsupervised Domain Adaptation in Point Clouds

To adapt a source trained 3D LiDAR-based detector to unseen target domain, Unsupervised Domain Adaptation (UDA) addresses the domain gap between labeled source domain and unlabeled target domain. Wang, et al [7] analyzes variance of object sizes between source and target domains and proposes a statistical normalization to handle the gap. 3D-COCO [9] proposes a contrastive co-training using bird's eye view (BEV) features to progressively learn transferable knowledge. ST3D [8] leverages self-training to reduce source domain bias and enhance quality of pseudo labels in target domain. However, prior works demonstrate constrained performance since they overlook beam-induced domain gap. LiDAR Distillation [11] addresses the discrepancy in LiDAR beams by generating a pseudo low-beam data by downsampling and transferring knowledge of source model from a high-density data to a low-density data. DTS [12] extends to various settings of point cloud densities including low-to-high density adaptation by proposing Random Beam Random Sampling and object-graph consistency to match the density of the source domain and target domain.

C. General Model Finetuning

After many pretraining algorithms with large amounts of unlabeled data are proposed, recent literature also focuses on transferring the general pre-trained model's representation to downstream tasks. SCL [28], Bi-tuning [29], Core-tuning [30], and COIN [31] propose finetuning methods using supervised contrastive loss to improve performance in classification tasks. Li et al. [32] presents L2 norm regularize of parameters between pre-trained and downstream model to improve performance, and AT [33], DELTA [34] presents behavior-based regularization loss that uses attention to reduce feature map discrepancy. DR-Tune [35] selects features with semantics from the pre-trained model's general features using semantic calibration, presents a distribution regularization method using labels, and demonstrates performance improvement. However, most of the above works are studied in 2D classification, and a general model finetuning method has not been proposed for 3D object detection tasks. We confirm the existence of a density domain shift between the pretrain and finetuning datasets. Thus we propose a general finetuning framework (DADT) with limited data in 3D object detection by bridging domain gaps.

III. METHODOLOGY

A. Problem Statement

The goal is to solve the domain shift with a few downstream data and proceed with the downstream task under the

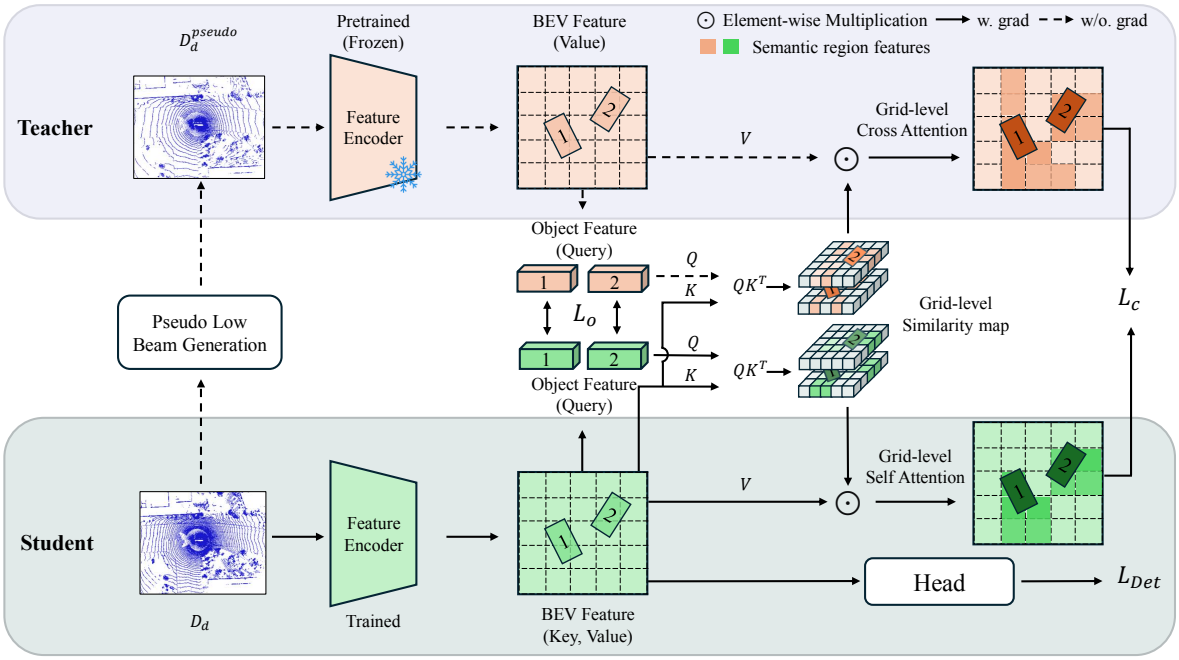


Fig. 2: **Overview of our proposed Domain Adaptive Distill-Tuning (DADT) framework.** DADT has a teacher-student architecture for reducing a density-driven representational gap. Downstream dataset D_d is downsampled using Pseudo Low Beam Generation to create a D_d^{pseudo} similar to pretrain dataset. To supervise and regularize the student's BEV feature distribution to match the teacher's general representation feature distribution, BEV object similarity loss is used to make the same objects in the teacher and student have similar features, and BEV context similarity loss is used to find the grid-similarity of the objects to highlight the semantic features of the objects in the current scene and make them similar.

assumption that there is a beam-induced discrepancy between the pretrain domain and the downstream task domain. Let the pretrain dataset be D_p and the downstream dataset be $D_d = \{(X_i, Y_i)\}_{i=1}^{N}$ (N is small). N is the sample number of the downstream domain D_d while X_i and Y_i denote ith point cloud and its 3D bounding box label (center, dimensions, heading), respectively. 3D object detection model consists of 3D encoder, BEV encoder, and detection head. We define a pre-trained model's backbone as $f_{\theta^p}(\cdot)$, consisting of 3D and BEV encoders, and its detection head as $g_{\phi^p}(\cdot)$, which are parameterized by θ^p and ϕ^p respectively. Then, we aim to finetune a downstream model $f_{\theta^d} \cdot g_{\phi^d}(\cdot)$, where f_{θ^d} is initialized by f_{θ^p}, and g_{ϕ^d} is randomly initialized.

Vanilla Finetuning. Vanilla finetuning initializes f_{θ^d} as f_{θ^p}, randomizes g_{ϕ^d}, and proceeds learning directly. Vanilla optimizes the following objective function.

$$\theta_*^d, \phi_*^d = \arg\min_{\theta^d, \phi^d} \mathrm{L}\left(f_{\theta^d} \cdot g_{\phi^d}; D_d\right) \tag{1}$$

However, due to the density domain shift, f_{θ^p} cannot learn to represent the BEV feature F for D_d well in limited data. Therefore, it is necessary to supervise f_{θ^d} to represent D_d feature well while utilizing the representation of f_{θ^p} as an encoder.

B. Teacher-Student Architecture for Reducing a Density-driven Representational Gap

Recent studies have repeatedly reported a potential domain shift by differences in the point density of a LiDAR sensor.

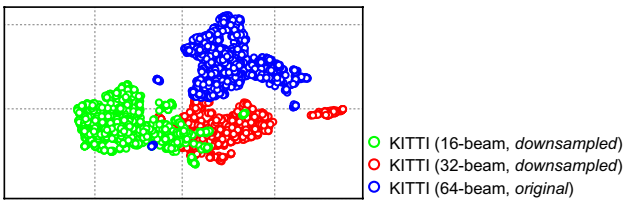


Fig. 3: **UMAP [13] Visualizations.** We compare representational differences depending on the beam density (i.e., 16, 32, 64-beams). We use KITTI data, downsampling it into a lower-density LIDAR point cloud.

Our experiment also confirms this as shown in Fig. 3, where we visualize embeddings from a pre-trained backbone by UMAP [13] with different point-density inputs. This necessitates aligning the point density between data for pre-training and finetuning. To address this issue, we utilize the teacher-student learning strategy, where a teacher network takes the density-aligned LiDAR inputs, and a student network uses the original density-non-aligned LiDAR inputs. Formally, we downsample D_d by adopting a pseudo low beam data generation approach [11] to create D_d^{pseudo}, thus having a point density similar to D_p.

Specifically, we first transform LiDAR points from Cartesian to Spherical coordinates system:

$$r = \sqrt{x^2+y^2+z^2},\ \phi = \arctan\frac{z}{\sqrt{x^2+y^2}},\ \theta = \arcsin\frac{y}{\sqrt{x^2+y^2}} \tag{2}$$

where x, y, z are Cartesian coordinates of D_d, and ϕ, θ are

inclination and azimuth angles, respectively. Given ϕ, we use K-means clustering to assign points to beam labels and uniformly downsample beam clusters along ϕ. Such downsampled points are then reverted into Cartesian coordinates, yielding D_d^{pseudo}.

C. BEV-based Similarity Losses

Given the teacher-student networks, the teacher network utilizes the frozen pre-trained backbones with density-aligned inputs, and the student finetunes its backbone (initialized from the pre-trained model) with density-non-aligned inputs. We use the following two regularization losses, i.e., (1) object similarity loss and (2) context similarity loss, to finetune the student network, retaining its representational distribution similar to the teacher network and preventing from overfitting.

Object Similarity Loss. Inspired by recent LiDAR-based studies [9], [11], [12], we apply the following object similarity loss:

$$\mathrm{L}_o = \sum_{c \in \mathrm{C}} \frac{1}{N_c} \sum_{i=1}^{N_c} \left\| z_i^{\mathrm{T}} - z_i^{\mathrm{S}} \right\|_2, \tag{3}$$

where N_c is the number of objects for a given class $c \in \mathrm{C}$. The object BEV features z_i^T and z_i^S for an object i are extracted from feature encoders of the teacher T and student S networks, respectively. We use a ground truth bounding box to extract the features of each agent from BEV representation. Note that we use the sum of per-class normalized losses to address a class imbalance problem during the finetuning step.

Context Similarity Loss. Further, in addition to object-wise similarity loss, we use attention-based context similarity loss L_c as follows:

$$\mathrm{L}_c = \sum_{c \in \mathrm{C}} \mathtt{MSE}(a_c^{\mathrm{T}}, a_c^{\mathrm{S}}) \tag{4}$$

where a_c^{T} and a_c^{S} are the average of the attended BEV features for a given class $c \in \mathrm{C}$ from the teacher and student networks, respectively. Formally, we use an attention [36] module to obtain the attended feature a_c^{S} from the student network: i.e., the object BEV feature z_i^{S} is used as the query vector, and the student's BEV features of each grid $F^{\mathrm{S}} \in \mathrm{R}^{h \times w \times d}$ are used as the key and the value vectors as follows:

$$a_c^{\mathrm{S}} = F^{\mathrm{S}} \odot \left(\frac{1}{N_c} \sum_{i=1}^{N_c} z_i^{\mathrm{S}} \cdot F^{\mathrm{S}} \right) \tag{5}$$

Similarly, we compute a_c^{T} by applying cross-attention with the following query, key, and value vectors.

$$a_c^{\mathrm{T}} = F^{\mathrm{T}} \odot \left(\frac{1}{N_c} \sum_{i=1}^{N_c} z_i^{\mathrm{T}} \cdot F^{\mathrm{S}} \right) \tag{6}$$

Specifically, we can get the grid-level similarity map for an object by performing a dot product between the object BEV feature and the whole BEV feature. Then, we can calculate the grid-level similarities for all objects in class C and average these similarities, called Context Similarity. With element-wise multiplication of the BEV feature and Context similarity for class C, we can get the features that emphasize the semantic region by multiplying the similarity of the values. Therefore, we can regularize the student network to focus on the semantic region features and retain its representational distribution similar to the teacher network by applying the MSE loss of a_i^{T} and a_i^{S}.

Loss Function. Ultimately, we minimize the following loss function L:

$$\mathrm{L} = \mathrm{L}_{Det} + \lambda_c \mathrm{L}_c + \lambda_o \mathrm{L}_o \tag{7}$$

where λ_c and λ_o are hyperparameters to control the weight of each term.

IV. EXPERIMENTS

Datasets. To evaluate the effectiveness of our proposed method, we use two widely-used public datasets: KITTI [15] and Waymo Open Dataset [21]. The former was collected with a 64-beam Velodyne LiDAR sensor in Germany, providing 7,481 annotated LiDAR frames (3,712 for training and 3,769 for validation). The latter contains annotated 19M LiDAR frames (15M for training and 4M for testing) collected with multiple sensors (a single 64-beam and four 200-beam LiDAR sensors). A subset of a few frames are uniformly sampled to evaluate our model under limited data scenarios.

Baseline Models. While our method generally applies to various 3D LiDAR-based object detection models without notable restrictions, we use the following two commonly-used detectors: SECOND [1], PV-RCNN++ [4]. Also, we utilize the pre-trained AD-PT [14] model trained on ONCE [20] dataset as our Oracle model. Note that ONCE dataset is a large-scale driving dataset collected in China with a 40-beam LiDAR, including various environmental conditions (e.g., day/night and sunny/rainy scenes). This model is ideal for our evaluation since (i) pre-trained models are publicly available for researchers to easily access for reproduction and (ii) ONCE dataset has a potential domain gap with our evaluation datasets (i.e., KITTI and Waymo Open Dataset) due to their differences in sensor configurations (40-beam LiDAR vs. 64-beam and 200-beam customized LiDAR sensors) and locations (China vs. Germany and USA). we finetune these pre-trained models with limited amounts of target data, denoting them as baselines.

Implementation Details. To finetune LiDAR-based 3D object detectors, we set $\lambda_{atten} = 1.0$ and $\lambda_{gt} = 1.0$, optimized by a grid search. As we randomly select only a few frames to finetune models, the model's performance might be varied regarding randomness. Thus, we report average scores from multiple independent runs for all experiments. We conduct each experiment on four NVIDIA 3090 GPUs with a batch size of 32 and 12 for KITTI and Waymo, respectively. Following the recent work, i.e., 3DTrans [37], we use the same augmentation and optimization techniques for all models.

Evaluation Metrics. We use the standard practice in evaluating LiDAR-based 3D object detectors regarding datasets. With KITTI datasets, we report the mean Average Precision

TABLE I: **3D Detection Accuracy Comparison with Limited Data.** We compare the detection performance of finetuned models with limited data, from 32 to 128, in terms of two metrics: AP and APH. Note that we use the Waymo [21] validation set to measure scores. All models are based on the SECOND [1].

Finetuning	# of Data	AP↑ / APH↑			
		Overall	Vehicle	Pedestrian	Cyclist
Baseline	32	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00	00.00 / 00.00
DADT (ours)	32	04.44 / 03.81 (4.44↑ / 3.81↑)	10.91 / 10.66 (10.91↑ / 10.66↑)	01.66 / 01.15 (1.66↑ / 1.15↑)	00.77 / 00.71 (0.77↑ / 0.71↑)
Baseline	64	02.75 / 02.71	05.98 / 05.90	00.01 / 00.00	02.26 / 02.23
DADT (ours)	64	23.73 / 21.48 (20.98↑ / 18.77↑)	25.74 / 25.29 (19.76↑ / 19.39↑)	17.05 / 11.64 (17.01↑ / 11.64↑)	28.39 / 27.51 (26.13↑ / 25.28↑)
Baseline	96	15.02 / 14.11	14.76 / 14.51	07.22 / 05.21	23.07 / 22.62
DADT (ours)	96	33.77 / 30.55 (18.75↑ / 16.44↑)	37.65 / 37.03 (22.89↑ / 22.52↑)	26.33 / 18.30 (19.11↑ / 13.09↑)	37.34 / 36.32 (14.27↑ / 13.7↑)
Baseline	128	33.07 / 30.34	34.96 / 34.35	27.26 / 20.52	37.00 / 36.16
DADT (ours)	128	40.22 / 36.75 (7.15↑ / 6.41↑)	46.48 / 45.71 (11.52↑ / 11.36↑)	33.33 / 24.67 (6.07↑ / 4.15↑)	40.85 / 39.89 (3.85↑ / 3.73↑)

TABLE II: **3D Detection Accuracy Comparison on KITTI Dataset.** We report scores in terms of 3D Average Precision (AP), with the IoU threshold set to 0.7 for Cars and 0.5 for Pedestrians and Cyclists.

Finetuning	# of Data	mAP (Mod.)↑	
		SECOND	PV-RCNN++
Baseline	1% (37)	16.76	15.85
DADT (ours)	1% (37)	21.66 (4.90↑)	20.21 (4.36↑)
Baseline	2% (74)	29.60	40.94
DADT (ours)	2% (74)	31.28 (1.68↑)	44.49 (3.55↑)
Baseline	3% (111)	38.91	49.53
DADT (ours)	3% (111)	42.24 (3.33↑)	54.73 (5.20↑)

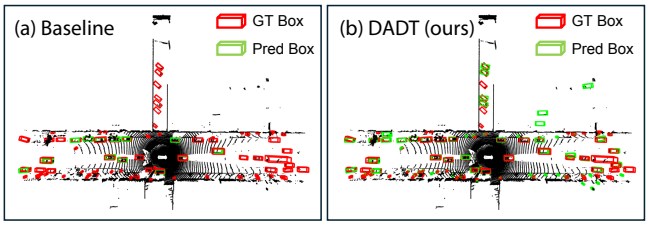


Fig. 4: **Examples of Detected 3D Objects from (a) baseline model and (b) ours.** Red boxes and green boxes denote ground-truth and predicted bounding boxes, respectively.

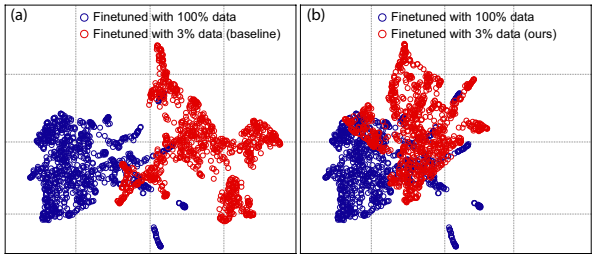


Fig. 5: **UMAP Visualizations** of the Oracle (a pre-trained backbone finetuned with 100% KITTI training data) and those from our baseline and ours. KITTI validation set is used for visualization.

(mAP) using 40 recall positions for three categories (i.e., cars, pedestrians, and cyclists) for easy, moderate, and hard cases. For the Waymo Open Dataset, we use Average Precision (AP) and Average Precision with Heading (APH) under Level 1 setting for vehicles, pedestrians, and cyclists.

A. Effect of Finetuning Pre-trained Model with Limited Data

Evaluation on Waymo Open Dataset. We start to evaluate the effectiveness of our approach by measuring detection accuracies on the Waymo [21] validation set regarding two primary metrics (i.e., AP and APH), as reported in Table I. Based on our baseline model, which is built upon SECOND architecture pre-trained with the AD-PT method on the ONCE [20] dataset, we compare our proposed finetuning approach with conventional common practice, denoted as a vanilla model. We report 3D detection scores with different numbers of limited data, i.e., from 32 to 128 LiDAR frames ra ndomly sampled from training dataset, evaluating with the 20% validation set. We observe in Table I that our method consistently shows improvements by a large gap, 4.44–20.98 and 3.81–18.77 higher AP and APH, respectively. This confirms that our method is effective for finetuning a pre-trained model with limited data. In Fig. 4, we also compare the 3D box prediction results on Waymo validation set with Vanilla (baseline) and DADT models trained on 64 frames. As illustrated in Figure 4, we can see that baseline is able to detect some nearby cars, but not the rest of the objects. However, in case of DADT, it detects not only nearby objects but also distant objects. To demonstrate the effectiveness of our proposed context similarity loss, we also visualize context similarity of the models trained with 96 frames in Fig. 6. We observe that our DADT highlights objects and its nearby environment.

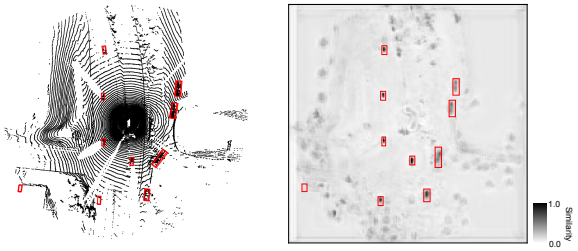
Evaluation on KITTI Dataset. Further, our experiment with the KITTI [15] dataset shows significant gains consistent with our Waymo Open dataset results. As shown in Table II, we evaluate models with different numbers of LiDAR frames (from a training set) and tested with the whole validation set. In this experiment, we use two kinds of 3D LiDAR-based object detection, i.e., SECOND [1] and PV-RCNN++ [4], which are pre-trained with AD-PT on ONCE [20] dataset.

UMAP Analysis. Our proposed model regularizes the finetuning process by constraining the representations of the LiDAR point cloud so that they are similar to those of pre-trained models. To verify this, we use UMAP [13] to visualize and compare embeddings. As expected, we observe in Fig. 5 that a naive supervised finetuning often guides the model to produce distributional shifts, overfitting to the small dataset (thus, will not generalize well to unseen data). However, our approach guides the model to produce representational distributions similar to the Oracle model (supervised finetuned model with 100% LiDAR frames), preventing overfitting.

Effect of Each Component. In Table III, we provide our

TABLE III: **Ablation Study.** We evaluate the effect of finetuned models with and without the following three components: (i) Pseudo Low Beam Generation (PLBG), (ii) the use of object similarity loss L_o and (iii) context similarity loss L_c. Data: Waymo [21]. PLBG†: Pseudo Low Beam Generation.

PLBG†	Regularizer		AP↑			
	Use of L_o	Use of L_c	Vehicle	Pedestrian	Cyclist	Avg.
✓	✓	✓	37.65	26.33	37.34	33.77
✓	✓	✗	35.69	24.72	35.61	32.01 (1.76↓)
✓	✗	✓	36.26	24.52	35.78	32.19 (1.58↓)
✗	✓	✓	36.55	26.19	36.43	33.06 (0.71↓)
✗	✗	✗	14.76	7.22	23.07	15.02 (18.75↓)



(a) Input LiDAR Frames (b) Context Similarity Heatmap

Fig. 6: **Example of Context Similarity Heatmap.** Red boxes denote ground-truth bounding boxes.

ablation study results, which show the effect of our module and two regularization functions: (i) Pseudo Low Beam Generation, (ii) the use of L_o and (iii) L_c. We use the Waymo Open dataset (finetuning with 96 LiDAR frames from the training set and evaluating with the 20% validation set). Our experiment shows that each building block has meaningful contributions, while their combinations offer significant synergy, outperforming the baseline with a large gap (compare 2-4th rows vs. bottom row).

B. Performance in Continual Training Scenarios

Continual Finetuning. Further, as our proposed approach keeps the original representational distribution while learning new data, our model might be advantageous for continuously adapting various datasets sequentially, learning more robust yet effective domain-invariant features. To evaluate the model's ability to adapt continually, we experiment with a scenario where a model is first trained on the Waymo Open dataset (with 96 LiDAR frames) with EMA [38], followed by training on the KITTI dataset (with 111 LiDAR frames). As shown in Table IV, our model provides better detection performance than direct transfer and vanilla supervised finetuned models.

Semi-supervised Learning. We also evaluate the performance in the semi-supervised learning setup, where we first pretrain the model using 3% labeled data from KITTI [15] training set and conduct further training by creating pseudo labels based on ST3D [8] for the rest of training data. As we summarized in Table V, our model shows better performance than our baseline model, showing closer to a model, which is trained from scratch with 100% labeled data (68.29 vs. 70.84 in terms of avg. mAP. Compare 2nd and 3rd rows). This demonstrates that it is possible to maintain high performance by labeling only limited data and then learning through self-training on unlabeled data. It also suggests that in the real industry, high-performance models can be made by online learning by finetuning a model with limited labels and self-training with the online LiDAR frames collected.

TABLE IV: **Performance of Continual Finetuning.** Given a model finetuned on the Waymo Open dataset (with 96 LiDAR frames from the training set), we further evaluate the ability to adapt to a new dataset (i.e., KITTI) continuously. Note that direct transfer indicates a model that is first trained on the Waymo Open dataset, directly followed by training on the KITTI dataset.

Waymo → KITTI	Car AP↑		
	Easy	Moderate	Hard
Direct Transfer	78.05	63.67	59.35
Baseline	81.19	64.02	59.54
DADT (ours)	83.84 (2.65↑)	66.07 (2.05↑)	60.89 (1.35↑)

TABLE V: **KITTI Performance Comparison in Semi-supervised Learning Setting.** All models are based on PV-RCNN++ [4] architecture and initialized from the pre-trained model with AD-PT [14] on the ONCE [20] dataset, except the scratch model, which is trained from random initialization. *Abbr.* S: Supervised (use the whole 3,712 training data), SS: Semi-supervised (use the combination of the few 111 labeled data and 3,712 pseudo-labeled data).

Model	S|SS	mAP (Mod.)↑			
		Car	Pedestrian	Cyclist	Avg.
Baseline	SS	76.68	53.71	68.10	66.83
DADT (ours)	SS	79.47 (2.79↑)	55.67 (1.96↑)	69.72 (1.62↑)	68.29 (1.46↑)
Scratch	S	84.51	56.93	71.09	70.84

V. CONCLUSION

In this paper, we introduce DADT, a distillation based domain adaptive finetuning framework for 3D LiDAR-based detection under limited target data. Our framework employs pseudo beam generation and novel BEV attention-based regularizer to effectively alleviate serious domain shift present in the finetuning of general pre-trained detection model with limited data. We comprehensively validate the effectiveness and practicality of our framework as it substantially improves performance of various baselines on Waymo and KITTI datasets and can be applicable to different problem settings.

ACKNOWLEDGMENT

This work was supported by Autonomous Driving Center, Hyundai Motor Company R&D Division. This work was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education(NRF-2021R1A6A1A13044830, 15%) and supported by Institute of Information & communications Technology Planning & Evaluation grant funded by the Korea government(MSIT) (RS-2022-II220043, 15%, IITP-2024-RS-2024-00397085, 15%).

REFERENCES

[1] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[2] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[3] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[4] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection," *International Journal of Computer Vision*, vol. 131, no. 2, pp. 531–551, 2023.

[5] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[6] Y. Zhou and O. 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*, 2018, pp. 4490–4499.

[7] Y. Wang, X. Chen, Y. You, L. E. Li, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. 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*, 2020, pp. 11 713–11 723.

[8] J. Yang, S. Shi, Z. Wang, H. Li, and X. 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, pp. 10 368–10 378.

[9] Z. Yihan, C. Wang, Y. Wang, H. Xu, C. Ye, Z. Yang, and C. Ma, "Learning transferable features for point cloud detection via 3d contrastive co-training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 493–21 504, 2021.

[10] W. Zhang, W. Li, and D. Xu, "Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6769–6779.

[11] Y. Wei, Z. Wei, Y. Rao, J. Li, J. Zhou, and J. Lu, "Lidar distillation: Bridging the beam-induced domain gap for 3d object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 179–195.

[12] Q. Hu, D. Liu, and W. Hu, "Density-insensitive unsupervised domain adaption on 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 556–17 566.

[13] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[14] J. Yuan, B. Zhang, X. Yan, T. Chen, B. Shi, Y. Li, and Y. Qiao, "Ad-pt: Autonomous driving pre-training with large-scale point cloud dataset," *arXiv preprint arXiv:2306.00612*, 2023.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[16] H. Liang, C. Jiang, D. Feng, X. Chen, H. Xu, X. Liang, W. Zhang, Z. Li, and L. Van Gool, "Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3293–3302.

[17] J. Yin, D. Zhou, L. Zhang, J. Fang, C.-Z. Xu, J. Shen, and W. Wang, "Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 17–33.

[18] H. Yang, T. He, J. Liu, H. Chen, B. Wu, B. Lin, X. He, and W. Ouyang, "Gd-mae: generative decoder for mae pre-training on lidar point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9403–9414.

[19] C. Min, D. Zhao, L. Xiao, Y. Nie, and B. Dai, "Voxel-mae: Masked autoencoders for pre-training large-scale point clouds," *arXiv preprint arXiv:2206.09900*, 2022.

[20] J. Mao, M. Niu, C. Jiang, H. Liang, J. Chen, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, H. Xu, and C. Xu, "One million scenes for autonomous driving: Once dataset," 2021.

[21] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. 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*, 2020, pp. 2446–2454.

[22] T.-Y. Pan, C. Ma, T. Chen, C. P. Phoo, K. Z. Luo, Y. You, M. Campbell, K. Q. Weinberger, B. Hariharan, and W.-L. Chao, "Pre-training liDAR-based 3d object detectors through colorization," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=fB1iiH9xo7

[23] J. Sun, H. Zheng, Q. Zhang, A. Prakash, Z. Mao, and C. Xiao, "CALICO: Self-supervised camera-liDAR contrastive pre-training for BEV perception," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=U7iiF79kI3

[24] O. Shrout, O. Nitzan, Y. Ben-Shabat, and A. Tal, "Patchcontrast: Self-supervised pre-training for 3d object detection," *arXiv preprint arXiv:2308.06985*, 2023.

[25] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 574–591.

[26] C. Min, L. Xiao, D. Zhao, Y. Nie, and B. Dai, "Occupancy-mae: Self-supervised pre-training large-scale lidar point clouds with masked occupancy autoencoders," *IEEE Transactions on Intelligent Vehicles*, 2023.

[27] R. Xu, T. Wang, W. Zhang, R. Chen, J. Cao, J. Pang, and D. Lin, "Mv-jar: Masked voxel jigsaw and reconstruction for lidar-based self-supervised pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 13 445–13 454.

[28] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=cu7IUiOhujH

[29] J. Zhong, X. Wang, Z. Kou, J. Wang, and M. Long, "Bi-tuning of pre-trained representations," *arXiv preprint arXiv:2011.06182*, 2020.

[30] Y. Zhang, B. Hooi, D. Hu, J. Liang, and J. Feng, "Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 848–29 860, 2021.

[31] H. Pan, Y. Guo, Q. Deng, H. Yang, J. Chen, and Y. Chen, "Improving fine-tuning of self-supervised models with contrastive initialization," *Neural Networks*, vol. 159, pp. 198–207, 2023.

[32] L. Xuhong, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2825–2834.

[33] N. Komodakis and S. Zagoruyko, "Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2017.

[34] X. Li, H. Xiong, H. Wang, Y. Rao, L. Liu, Z. Chen, and J. Huan, "Delta: Deep learning transfer using feature map with attention for convolutional networks," *arXiv preprint arXiv:1901.09229*, 2019.

[35] N. Zhou, J. Chen, and D. Huang, "Dr-tune: Improving fine-tuning of pretrained visual models by distribution regularization with semantic calibration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1547–1556.

[36] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[37] D. D. Team., "3dtrans: An open-source codebase for exploring transferable autonomous driving perception task." https://github.com/PJLab-ADG/3DTrans, 2023.

[38] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.