Unsupervised Domain Adaptation for Self-Driving from Past Traversal Features

Travis Zhang*†,[1] Katie Luo*,[1] Cheng Perng Phoo[1] Yurong You[1]
Wei-Lun Chao[2] Bharath Hariharan[1] Mark Campbell[1] Kilian Q. Weinberger[1]
[1]Cornell University [2]The Ohio State University

Abstract

The rapid development of 3D object detection systems for self-driving cars has significantly improved accuracy. However, these systems struggle to generalize across diverse driving environments, which can lead to safety-critical failures in detecting traffic participants. To address this, we propose a method that utilizes unlabeled repeated traversals of multiple locations to adapt object detectors to new driving environments. By incorporating statistics computed from repeated LiDAR scans, we guide the adaptation process effectively. Our approach enhances LiDAR-based detection models using spatial quantized historical features and introduces a lightweight regression head to leverage the statistics for feature regularization. Additionally, we leverage the statistics for a novel self-training process to stabilize the training. The framework is detector model-agnostic and experiments on real-world datasets demonstrate significant improvements, achieving up to a 20-point performance gain, especially in detecting pedestrians and distant objects. Code is available at `https://github.com/zhangtravis/Hist-DA`.

1. Introduction

Self-driving cars need to detect objects like cars and pedestrians and localize them in 3D to drive safely. 3D object detection systems have advanced rapidly in accuracy, but still fail to generalize across the extremely diverse domains where vehicles are deployed: A perception system trained in sunny California may never have seen snow-covered cars, and may fail to detect these cars with disastrous consequences. Unfortunately, we cannot afford to separately annotate training data for every location a car might be driven in. We therefore need ways of adapting 3D perception systems to new driving environments without labeled training data. This is the problem of *unsupervised domain adaptation*, where the object detector must be adapted to a new *target* domain where only unlabeled data is available. In this work, we explore 3D object detection from LiDAR data and how to best adapt it to a set of diverse, real-world scenarios.

Different from prior works in unsupervised domain adaptation, we follow [13] to include the assumption that unlabeled repeated traversals of the same locations are available to the adaptation algorithm. As discussed in the prior works, such assumptions are highly realistic: for example, roads and intersections are usually visited many times by many vehicles. Prior work has shown that the additional information from repeated traversals helps 3D detection in the same domain [11, 12], and helps the perception models adapt to a new domain [13].

However, it is not readily obvious how to best utilize the repeated traversals. Rode-DA [13] uses P2-score, which is a statistic computed from repeated LiDAR scans characterizing the persistence of different areas of the 3D scene, to correct the false positive detections in self-training and better supervise the model. We argue that this method has not fully exploited the information from the P2-score and it can be used in a more principled way to guide the adaptation process.

Our key insight is that the P2-score is a perfect signal to *regularize* the feature in the detection training. Our full method is based on Hindsight [11]. Hindsight enhances LiDAR-based 3D object detection models with the spatial quantized historical (SQuaSH) features computed from repeated past traversals. The authors show that Hindsight can greatly improve the detection performance when tested within similar areas. However, the SQuaSH features do not guarantee to be invariant across different domains, resulting in limited performance gain. To prevent the SQuaSH features overfit to the training domain, we propose to add the P2-score prediction task as an auxiliary task while training the SQuaSH featurizer. Observing LiDAR points within each voxel sharing similar P2-score, we apply an extra light-weighted regression head after the SQuaSH feature, and train the head with simple P2 regression task. The regres-

*Denotes equal contribution.

†Correspondences could be directed to `tz98@cornell.edu`

sion head is only used in training and does not introduce latency overhead during testing.

Pairing with the typical self-training technique in domain adapation, we validate our method on two large, real-world datasets: Ithaca365 [2] and Lyft [4], as well as a suite of representative object detectors. Our method, which we term Historically Guided Domain Adaptation (Hist-DA), can achieve up to 20 points in improvement, most notably in difficult cases such as detecting pedestrians and far away objects. Furthermore, our method requires very little tuning to achieve strong performance for 3D object detection. Concretely, our contributions are as follows:

- Our methodology identifies a strong source of information with a high learning signal to improve self-supervised adaptation.
- We designed a model-agnostic adaptation framework to leverage repeated traversals effectively.
- We empirically validated our approach on two real-world datasets and show through ablation studies that Hist-DA is robust and generalizable.

2. Related Works

Past Traversals in Autonomous Driving Human drivers often drives through the same locations repeatedly, thus it is natural to assume that the (unlabeled) data collected for training perception systems for self-driving vehicle contains repeated traversals of different locations. Past works have leveraged this property to enhance the perception of autonomous vehicles. These include self-supervising 2D representation for visual odometry [1], uncovering mobile objects in LiDAR in an unsupervised manner [12], etc. The line of work that is directly related to ours is Hindsight [11] where the authors proposed to learn additional feature descriptors for each point in a LiDAR point cloud from the unlabeled past traversals for better downstream 3D object detection. Hindsight is simple and effective and would work with any downstream 3D detectors that consumes 3D LiDAR point cloud. In this work, we seek to adapt this family of detectors when deploying to a new domain where unlabeled past traversals are available.

Unsupervised Domain Adaptation (UDA) for 3D Object Detection. Adapting 3D object detectors to new domains where no labels are available is crucial to deployment of self-driving vehicles. The key to UDA is to understand the domain differences that the detector would encounter during deployment. SN discovers that car sizes could be a source of domain differences and propose to normalize the car sizes when training the detectors on the source domain [7]; SPG identifies point cloud density as one potential source of differences and propose to fill in point clouds during deployment [8]. Though these methods have shown remarkable progress in the problem, they all target specific domain differences, which is not feasible in all cases. One way to characterize domain differences is through the use of unlabeled data. Along this vein, ST3D [9] and ST3D++ [10] adopt conventional self-training approaches with improved filtering mechanism to stabilize adaptation whereas MLC-Net [6] achieves domain alignment via enforcing consistency between a source detector and its exponential moving average on the unlabeled data. Though these methods[6, 7, 9] are effective, they mostly assume that all the unlabeled data are i.i.d. which ignores other potential signals that could be inherent in the unlabeled data such as temporal signals [13] that are potentially useful in adaptation. In this work, we explore using unlabeled past traversals for domain adaptation. As shown in [13], these correlated data contains potent signals for aiding adaptation. However, crucially different from [13], we focus on adapting Hindsight — a family of models that uses past traversals during inference time.

3. Historically Guided Domain Adaptation

We attempt to adapt a 3D object detector to a target domain using unlabeled data. Different from typical adaptation setup [7, 9], we assume the autonomous driving system has access to multiple traversals of the same driving scenes and accurate localization information, both in the source and target domain. In section 3.1, we will clearly lay out the adaptation setup. Then, we will discuss relevant background information to clarify our proposed methodology in section 3.2. Our key insight is to leverage P2-score information from repeated traversals to adapt the detector's point features from one domain to another —akin to feature-alignment works done in the 2D space— as well as self-training to ensure stable predictions. We discuss the relevant adaptation strategies in section 3.3. Our overall method is shown in Figure 1.

3.1. Unsupervised Domain Adaptation with Repeated Traversals

Our goal is to adapt a LiDAR-based detector using repeated traversals of unlabeled point clouds $\{\boldsymbol{P}_i^t\}$ and the associated global localization $\{G_i^t\}$ from the target domain, for the i-th frame in traversal t. We assume that the source domain also has access to multiple repeated traversals, as well as the bounding box labels $\boldsymbol{b}_c$ associated with training point clouds $\boldsymbol{P}_c$, for the c-th training frame. To characterize these historical traversals, we combine the point clouds for a single traversal to create a dense point cloud in the same way as [11]. Specifically, for a single traversal in a domain that consists of a sequence of point clouds, we transform each point cloud into a fixed global coordinate system. Then, for a location l in a single frame i within traversal t every m meters along the road, the point clouds

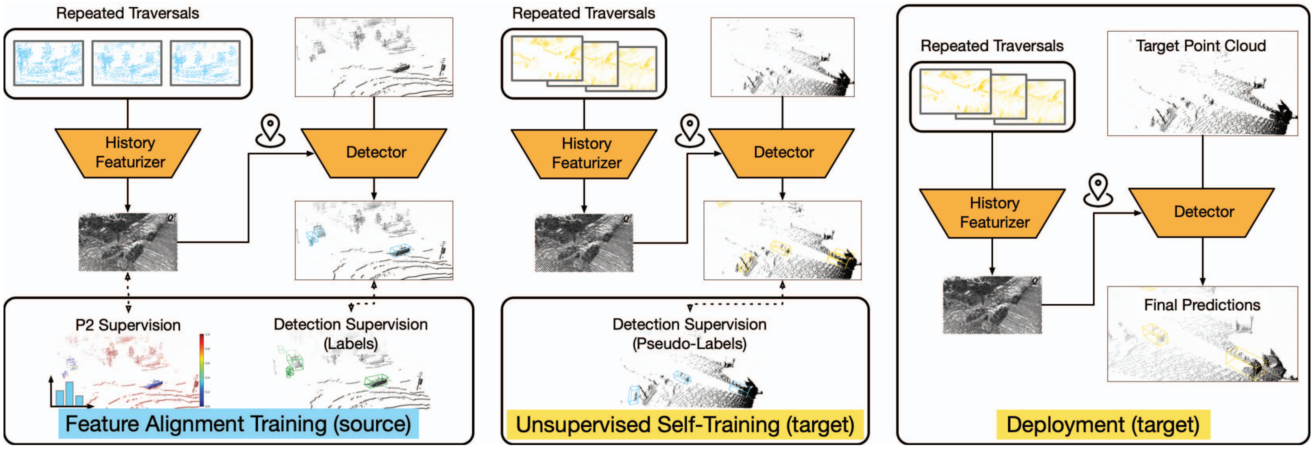

Figure 1. **Method diagram of the adaptation process.** The method is divided into source domain training, target domain unsupervised training, and finally deployment on the target domain. The repeated traversals from the source domain are colored in blue, and those from the target domain are colored in yellow. Best viewed in color.

from a range $[-H_m, H_m]$ are combined to produce a dense point cloud $\boldsymbol{D}_l^t = \bigcup_{G_i^t \in [l-H_m, l+H_m]} \{\boldsymbol{P}_t^i\}$ (with a slight abuse of notation as we use G_i^t additionally for the location i was captured).

3.2. Background

Persistency Prior Score from multiple traversals. Our goal is to exploit the inherent information from the unlabeled repeated traversals for adaptation. One source of information we can retrieve is the Persistency Prior (P2) score, that was introduced in [12]. To recap, P2-score uses entropy-based measures to quantify how persistent a single LiDAR point cloud is across multiple traversals. It is calculated using the set of dense point clouds $\{\boldsymbol{D}_l^t\}_{t=1}^T$, for $T \geq 1$ traversals of a location. For a given 3D point $\boldsymbol{q}$ around location l, we first count the number of neighboring points around $\boldsymbol{q}$ within a certain radius r in each $\boldsymbol{D}_l^t$:

$$N_t(\boldsymbol{q}) = \left|\{\boldsymbol{p}_i ; ||\boldsymbol{p}_i - \boldsymbol{q}||_2 < r, \boldsymbol{p}_i \in \boldsymbol{D}_l^t\}\right| \tag{1}$$

We can then normalize the neighbor count $N_t(\boldsymbol{q})$ across traversals $t \in \{1, ...T\}$ into a categorical probability:

$$P(t; \boldsymbol{q}) = \frac{N_t(\boldsymbol{q})}{\sum_{t'=1}^{T} N_{t'}(\boldsymbol{q})} \tag{2}$$

Using $P(t; \boldsymbol{q})$, we can then compute the P2-score $\tau(\boldsymbol{q})$ the same way as [12]:

$$\tau(\boldsymbol{q}) = \begin{cases} 0 & \text{if } N_t(\boldsymbol{q}) = 0 \; \forall t; \\ \frac{H(P(t;\boldsymbol{q}))}{\log(T)} & \text{otherwise} \end{cases} \tag{3}$$

where H is the information entropy. Intuitively, a higher P2-score corresponds to a more persistent background, while a lower P2-score corresponds to a mobile foreground object. This value is a statistic that can be calculated from the repeated traversals, and as a result, it's natural for us to use an architecture that leverages these data. One such candidate is Hindsight.

Hindsight. Hindsight [11] is an end-to-end featurizer intended to extract contextual information from repeated past traversals of the same location. The authors proposed an easy-to-query data structure used to endow the current point cloud with information from past traversals to improve 3D detection.

Given the dense point cloud $\boldsymbol{D}_l^t$, Hindsight encodes it using a spatial featurizer that results in a spatially-quantized feature tensor $\boldsymbol{Q}_l^t$. This can be applied to the T tensors, one for each traversal in location l, which is then aggregated into a single tensor $\boldsymbol{Q}_l^g$, deemed SQuaSH, using a per-voxel aggregation function f_{agg}:

$$\boldsymbol{Q}_l^g = f_{agg}(\boldsymbol{Q}_l^1, ..., \boldsymbol{Q}_l^T) \tag{4}$$

Once deployed, if the self-driving car captures a new scan $\boldsymbol{P}_c$ at a new location G_c and the SQuaSH feature at this location is $\boldsymbol{Q}_{l_c}^g$, Hindsight endows $\boldsymbol{P}_c$ by querying the features $\boldsymbol{Q}_{l_c}^g$ around it. In the work [11], the SQuaSH featurizer is trained concurrently with the object detector, using the detection loss as a signal for gradient updates.

3.3. Adaptation Strategy

Our adaptation strategy consists of P2 feature alignment training in the source domain and unsupervised self-training in the target domain.

P2 Feature Alignment Training Though the computation of P2-score is of high latency and thus it is hard to be applied online, it serves perfectly as an additional signal for offline adaptation algorithm. With P2-score, we construct a simple self-supervised learning task to adapt the SQuaSH features after deployment.

	Car				Pedestrian			
Method	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
No Adapt/ No HS	42.19	12.66	0.95	18.54	40.74	18.32	0.42	21.18
ST3D	61.63	38.70	4.73	35.89	44.37	26.94	0.00	24.97
Rote-DA	**62.85**	**41.88**	15.07	41.32	48.76	32.61	1.21	30.59
No Adapt + HS	41.88	29.31	16.40	30.29	51.16	26.41	5.80	29.99
Hist-DA (Ours)	58.44	40.03	**25.26**	**42.82**	**60.72**	**48.58**	**21.42**	**48.48**
Oracle (in domain)	73.38	56.19	39.08	57.10	55.39	37.42	14.86	40.37

Table 1. Results of adapting a detector from Lyft to Ithaca365. Metrics are reported on nuScenes mAP at 1m matching.

	Car				Pedestrian			
Method	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
No Adapt/ No HS	59.0	40.9	25.8	45.4	16.7	8.2	0.2	6.7
ST3D	**71.8**	**52.1**	30.4	**55.7**	–	–	–	–
Rote-DA	54.3	31.9	14.9	35.7	**29.6**	**34.4**	4.1	**22.0**
Hist-DA (Ours)	62.6	49.2	**34.9**	51.8	25.6	26.5	**7.9**	16.7
Oracle (in domain)	69.1	71.5	49.0	65.7	37.0	38.2	26.3	32.2

Table 2. Results of adapting a detector from Ithaca365 to Lyft. Metrics are reported at 0.7 IoU matching.

Consider a point $\boldsymbol{q}$ in a point cloud $\boldsymbol{P}_c$, we can obtain 1) its corresponding SQuaSH feature $\mathbf{Q}_l^g(\boldsymbol{q})$; 2) its corresponding P2-score $\tau(\boldsymbol{q})$. Since the SQuaSH feature is computed from the same traversals, it should contain sufficient information to reproduce the corresponding P2-score. However, the trained model might suffer from the domain difference, and thus in the target domain, it might not be able to encode sufficient information from the past traversals, including those for the P2 score. We thus construct a P2 score prediction task for the SQuaSH feature to help the model align the relevant information it extracts in the source domain to invariant information encoded in P2-scores. For each SQuaSH feature $\boldsymbol{Q}_l^g(\boldsymbol{q})$, we apply a simple MLP to predict the corresponding P2 score,

$$\hat{\tau}(\boldsymbol{q}) = \mathrm{MLP}(\boldsymbol{Q}_l^g(\boldsymbol{q})). \tag{5}$$

We compute the L1 distance between the predicted P2 score and the corresponding P2 score as the alignment loss,

$$l_{\text{alignment}} = \|\hat{\tau}(\boldsymbol{q}) - \tau(\boldsymbol{q})\|_1. \tag{6}$$

The final objective for the detector training under the source domain consists of the alignment loss $l_{\text{alignment}}$, in addition to the regular detection loss for the detector we are adapting $l_{\text{detection}}$, computed from the predicted bounding boxes $\hat{\boldsymbol{b}}$ and the labels $\boldsymbol{b}_c$. Our methodology is detector agnostic, and we do not assume the base detector or the detection loss $l_{\text{detection}}$.

Unsupervised Self-Training To stabilize finetuning in the target domain, we apply self-training in the target domain. Similar to works [5, 13] that showed the effectiveness of self-training, we leverage refined *pseudo-labels* that we generate for adaptation into the target domain.

Given an aligned detector from P2 feature alignment training, we can generate bounding boxes in the target domain. Given a point cloud $\boldsymbol{P}_c$ in the target domain, we can obtain bounding boxes $\hat{\boldsymbol{b}}$ from the detector. Similar to the source domain, we can compute the P2-score for each point cloud in the target domain, $\tau(\boldsymbol{q})$, $\boldsymbol{q} \in \boldsymbol{P}_c$. To assess the quality of a particular bounding box, we can apply a simple criteria that points within the bounding box cannot be too *persistent*, *i.e.* having P2-scores that are too high. In this work, we filter out bounding boxes that capture points with P2-scores with a 20*th*-percentile larger than 0.7:

$$\hat{\boldsymbol{b}}_{\text{final}} = \{b \in \hat{\boldsymbol{b}} | \mathrm{P}_{20}(\{\tau(\boldsymbol{q}_j)\}_{j\in b}) < 0.7\}, \tag{7}$$

with a slight abuse in notation, we denote $j \in b$ as the j-th point that is in bounding box b. This gives us the final set of pseudo-labels, $\hat{\boldsymbol{b}}_{\text{final}}$, and we compute the detection loss for the model on the pseudo-labels, and the final objective for the unsupervised training on the target domain is this pseudo-label detection loss, $l_{\text{detection}}$ computed on $\hat{\boldsymbol{b}}_{\text{final}}$ as the labels.

4. Experiments

Datasets. We experiment with two large-scale autonomous driving datasets: the Lyft Level 5 Perception dataset [4] and the Ithaca-365 dataset [2]. To the best of our knowledge, these are the only two publicly available autonomous driving datasets that have both bounding box annotations and multiple traversals with accurate 6-DoF localization. The Lyft dataset is collected in Palo Alto (California, US) and the Ithaca-365 dataset is collected in Ithaca (New York, US). We use the roof LiDAR (40/60-beam in Lyft; 128-beam in Ithaca-365), and the global 6-DoF localization with the calibration matrices directly from the raw data. We simulate adaptation scenarios in both ways: 1) train in Lyft and test in Ithaca-365; 2) train in Ithaca-365 and test in Lyft.

Source 3D Object Detectors. In the source domain, we train the default implementation of PointRCNN model with Hindsight [11] using both object detection and P2-score feature alignment training for 60 epochs. We modify the Hindsight model to predict P2-scores from the inputted dense point cloud $\mathbf{S}_l^t$. All models are trained with 4 GPUs (NVIDIA A6000).

It is worth noting that our methodology has the ability to be applied to other 3D object detectors as well, and leave this for future exploration.

Evaluation Metrics. On the Lyft dataset, we evaluate object detection performance in a bird's eye view (BEV) and use KITTI [3] metrics and conventions for 3D detection. We report average precision (AP) with the intersection over union (IoU) thresholds at 0.7 and 0.5 for Car and Pedestrians. Additionally, these evaluations are evaluated at various depth ranges. Due to space constraints, we report AP$_{BEV}$ at IoU=0.7 for Cars and Pedestrians. On the Ithaca365 dataset, the default match criterion is by the minimum distance to the ground-truth bounding boxes. We report the mean average precision (mAP) with match thresholds of 1-meter for Cars and Pedestrians. Since there are too few cyclists in the Ithaca-365 dataset to provide a reasonable performance estimate, we train and evaluate our models only on *Cars* and *Pedestrians*.

Adaptation Method Comparisons We compare the proposed methodology against the following methods with publicly available code: ST3D [9] and Rote-DA [13].

4.1. Domain Adaptation Performance Results

In Table 1 and Table 2, we show the results on adaptation from a detector trained in the Lyft dataset to the Ithaca365 dataset and vice versa. Based on the tables, we can see that our methodology, despite its simplicity, outperforms all baselines in almost all metrics in both adaptation directions. This goes to show not only that using multiple traversals serves as a strong learning signal for these models, but also that predicting P2-scores as a self-supervised learning task leads to a dramatic improvement.

Hist-DA works especially well in more challenging scenarios, specifically in the pedestrian scenario and with farther distances. Although it performs slightly worse for cars at close ranges, our methodology has a significantly stronger performance for pedestrians and for far away objects. The model even outperforms an in-domain detector in all distances for pedestrians by a substantial amount. Furthermore, due to the simple nature of the single round of self-training in Hist-DA, our method is significantly simpler to train than any of the baselines, which require many rounds of self-training. Consequentially, it is significantly simpler to tune and is faster to train. Observe that by adding in Hindsight features (+ HS), we are already able to observe performance gains over the model that doesn't leverage past traversal information. This shows that such historical features already improve adaption and are more robust across domains. By including our method, we are able to achieve the best performance by explicitly bootstrapping in the past traversal statistics in the form of P2-scores.

4.2. Qualitative Results

We visualize our adaptation results in Figure 2, and compare the detections of Hist-DA (in yellow) to detections without adaptation (in blue). Observe that detection results using Hist-DA are qualitatively better than those without adaptation, both in the shape, as well as precision and recall. In particular, for smaller actors such as pedestrians and in actors that are further away. The feature alignment training allows for more robust features that generalizes across domains, and the unsupervised self-training allows for stronger adaptation into the new domain.

4.3. Analysis

Effect of different adaptation components. We additionally ablate the different components and report our results in Table 3. Observe that adding in P2-score training is crucial to the generalizability of the features across domains. Additionally, adding in self-training ("Pseudo-Label") helps stabilize the model training in an unsupervised manner into the new domain. Although using both P2 Training and pseudo-labels and only using P2 Training have similar performances, we noticed that the number of traversals from the source to the target domain can affect the performance of including P2 Training in the target domain. This occurs because P2 scores are inherently derived from repeated traversals, and the number of traversals can affect its accuracy. To be more specific, we observed that going from Ithaca365, which had 20 traversals to Lyft, which had 5 traversals made the performance of both P2 and pseudo-labels worse than using pseudo-labels since the P2 scores derived from the target domain was introducing noise to cause the model's P2 backbone to decrease in accuracy.

Effect of historical traversals. We examine the effect of the additional information by including unlabeled, historical traversals. We report our findings in Table 4. Although directly evaluating on Ithaca365 using a PointRCNN model

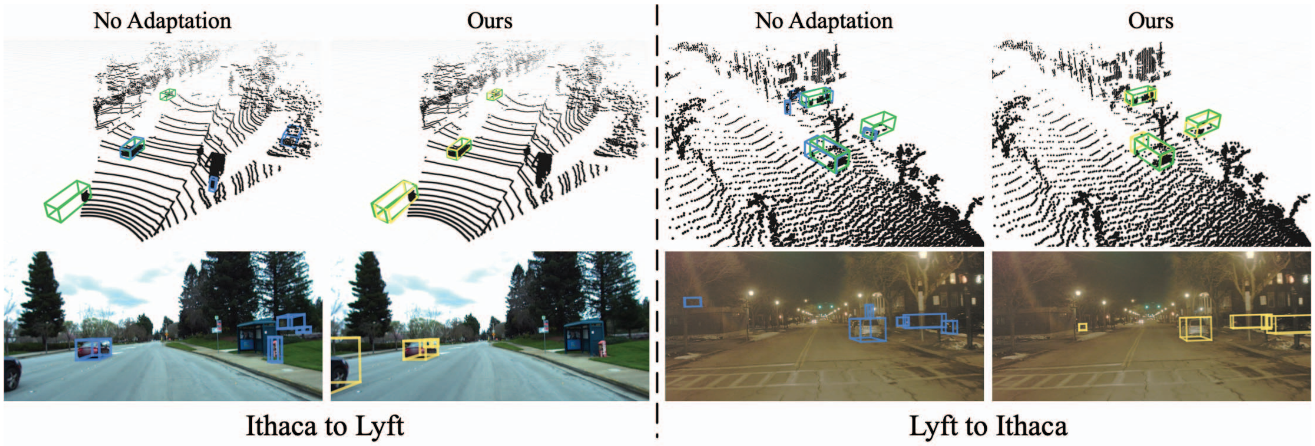

Figure 2. **Qualitative visualization of adaptation results.** We visualize one example scene (above: LiDAR, below: image, not used for adaptation) from the adaptation results from the Ithaca-365 → Lyft and Lyft → Ithaca-365 datasets. Ground-truth bounding boxes are shown in green, detection boxes of no adaptation and our method are shown in blue and yellow, respectively. Best viewed in color.

Source	Target		Car				Pedestrian			
P2 Training	P2 Training	Pseudo-Label	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
			41.88	29.31	16.4	30.29	51.16	26.41	5.8	29.99
✓			50.57	34.57	19.04	36.39	55.9	38.83	12.28	39.88
✓	✓		37.24	15.26	2.85	18.37	41.16	26.32	12.05	27.73
✓		✓	**58.44**	40.03	25.26	42.82	60.72	**48.58**	**21.42**	**48.48**
✓	✓	✓	58.06	**42.34**	**25.82**	**43.31**	**60.87**	48.42	20.41	47.92

Table 3. Ablation results adapting a detector from Lyft to Ithaca365. Metrics are reported on nuScenes mAP at 1m matching.

trained on Lyft has noticeable performances, one can see that including the Hindsight model increases the performance in both cars and pedestrians, with some distances improving by almost two-fold. On the other hand, adapting a PointRCNN model without Hindsight leads to improvements specifically for pedestrians, but performs slightly worse than (-) adapt / HS in cars. Naturally, adding both would significantly improve the performance as shown in the table, with adapting improving the pedestrian performance and the hindsight model improving the car performance.

PRCNN Model	Car				Pedestrian			
	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
baseline	42.2	12.7	0.9	18.5	40.7	18.3	0.4	21.2
(-) adapt, HS	41.9	29.3	16.4	30.3	51.2	26.4	5.8	30.0
adapt, (-) HS	54.1	22.8	2.0	27.0	52.5	31.2	1.9	32.1
Ours	**58.4**	**40.0**	**25.3**	**42.8**	**60.7**	**48.6**	**21.4**	**48.5**

Table 4. Ablation results adapting a detector from Lyft to Ithaca365. Metrics reported on nuScenes mAP at 1m matching.

Robustness of the framework. We analyze the robustness of our method to localization error and number of past traversals used in computing the historical features of the model. Results for localization error are shown in Table 5; Hist-DA is robust to minor errors in noise. We additionally report results for robustness under number of past traversals in Table 6. Observe that performance gain in adaptation can be seen with even two past traversals of an area. Additionally, our method handles higher depths better than other methods as shown in Table 1 and Table 2, since P2-score as a self-supervision task acts as a prior over the point clouds and inherently removes static objects that normal object detectors might not catch at higher depths.

5. Discussion and Future Works

In this work, we propose our method, Hist-DA, for the task of domain adaptation in self-driving object detection. Our work is able to achieve strong performance by training well aligned features from past traversal statistics, and further leverage the statistics to stabilize model outputs in the test domain in an unsupervised manner. Our method is the first to approach domain adaptation for 3D object detection under a feature alignment perspective leveraging past traversal information. Furthermore, by bringing in an architecture specifically designed to leverage such information,

Loc. Error	Car				Pedestrian			
	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
baseline	42.2	12.7	0.9	18.5	40.7	18.3	0.4	21.2
0.1 m	**58.2**	**40.8**	**24.6**	**42.6**	**59.7**	**47.8**	**21.1**	**47.5**
0.2 m	**58.2**	40.1	23.9	42.4	59.2	46.5	19.4	46.4
0.3 m	57.3	40.5	23.0	41.8	57.4	44.5	16.7	44.2
0.4 m	57.0	38.8	21.6	40.7	56.0	41.2	14.3	41.9
0.5 m	56.9	38.4	19.8	40.2	54.4	38.5	11.3	38.9

Table 5. Robustness testing on Lyft to Ithaca365, localization error. Metrics are reported on nuScenes mAP at 1m matching.

# Traversals	Car				Pedestrian			
	0-30	30-50	50-80	0-80	0-30	30-50	50-80	0-80
$N = 1$	56.0	33.1	14.0	36.3	52.8	32.1	6.3	33.9
$N \leq 2$	**59.2**	39.6	22.2	41.9	58.4	43.5	16.0	43.8
$N \leq 3$	59.0	**39.9**	24.3	42.6	59.8	46.0	19.6	46.5
$N \leq 4$	59.1	39.7	**25.2**	**42.9**	**60.6**	**48.4**	**20.9**	**48.3**

Table 6. Robustness testing on Lyft to Ithaca365, number of traversals. Metrics are reported on nuScenes mAP at 1m matching.

we show state-of-the-art performance on two large, real world datasets. Future directions include expanding this framework into other object detectors and exploring other feature alignment methods leveraging past traversals.

References

[1] Dan Barnes, Will Maddern, Geoffrey Pascoe, and Ingmar Posner. Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments. In *ICRA*, pages 1894–1900. IEEE, 2018. 2

[2] Carlos Andres Diaz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Z Luo, Yan Wang, Marc Emond, Wei-Lun Chao, Bharath Hariharan, Kilian Q. Weinberger, and Mark Campbell. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2, 4

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5

[4] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 av dataset 2019. https://level5.lyft.com/dataset/, 2019. 2, 4

[5] 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. 4

[6] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8866–8875, 2021. 2

[7] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *CVPR*, pages 11713–11723, June 2020. 2

[8] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15446–15456, 2021. 2

[9] 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. 2, 5

[10] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1951–1960, 2019. 2

[11] Yurong You, Katie Z Luo, Xiangyu Chen, Junan Chen, Wei-Lun Chao, Wen Sun, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Hindsight is 20/20: Leveraging past traversals to aid 3d perception. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Apr. 2022. 1, 2, 3, 5

[12] Yurong You, Katie Z Luo, Cheng Perng Phoo, Wei-Lun Chao, Wen Sun, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Learning to detect mobile objects from lidar scans without labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 1, 2, 3

[13] Yurong You, Cheng Perng Phoo, Katie Z Luo, Travis Zhang, Wei-Lun Chao, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Unsupervised adaptation from repeated traversals for autonomous driving. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, Dec. 2022. 1, 2, 4, 5