ADAPTIVE MULTI-PROTOTYPE GROUPING ALIGNMENT FOR DOMAIN ADAPTIVE 3D DETECTION

Anonymous authors
Paper under double-blind review

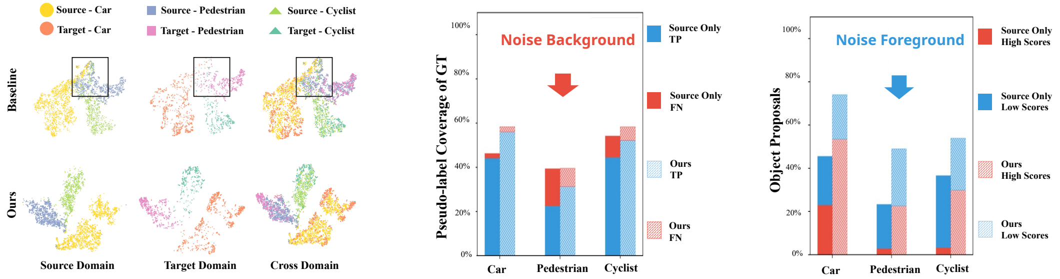
ABSTRACT

3D object detection is crucial for autonomous driving and robotics, but models often perform poorly when deployed in new environments due to domain shifts. While 3D unsupervised domain adaptation methods aim to address this issue, they still struggle with two key challenges: insufficient cross-domain feature alignment and ambiguous foreground-background decision boundaries. In this paper, we propose an Adaptive Multi-Prototype Grouping Alignment framework that addresses these challenges. Our method automatically discovers and dynamically updates feature groups, enabling adaptively cross-domain feature alignment in a fine grained manner. Additionally, we develop two techniques to address the issue of ambiguous foreground-background decision boundaries: Noise Background Hybrid Augmentation that leverages labeled source instances to enhance adaptation in uncertain regions, and Noise Foreground Contrastive Learning that improves foreground-background discrimination by pushing low-confidence features away from prototypes. We conduct extensive experiments on multiple cross-domain benchmarks and the results demonstrate the superiority of our method over the state-of-the-art methods.

1 INTRODUCTION

As a fundamental task in computer vision, 3D object detection plays a crucial role in robotic navigation and autonomous driving Arnold et al. (2019); Guo et al. (2020); Li et al. (2020); Malavazi et al. (2018); Ahmed et al. (2018). With the development of large-scale annotated datasets and deep learning techniques, this task has achieved remarkable success Lang et al. (2019); Shi et al. (2019; 2023). However, in real-world scenarios, models often suffer from severe performance degradation due to domain shift between the training data and deployment environments Wang et al. (2020). These shifts typically stem from sensor discrepancies, scene diversity, geographical variations, and weather conditions. A straightforward approach would be to annotate large-scale datasets for each target domain, but the high annotation cost makes it impractical for large-scale deployment. Consequently, Unsupervised Domain Adaptation (UDA) on 3D detection Yang et al. (2021; 2022); Hu et al. (2023); Chen et al. (2024a) has become a critical research topic, aiming to align cross-domain feature spaces and transfer knowledge from the source domain to the target domain without requiring target domain labels.

Recently, several self-training 3D UDA Yang et al. (2021); Hu et al. (2023); Chen et al. (2024b) methods have gained attention for their excellent performance. These methods generate pseudo labels for the target domain using pretrained models and perform iterative training to reduce domain shift through implicit feature alignment. Existing research mainly adopts single-factor alignment strategies that focus on specific domain factors, such as object size Wang et al. (2020); Yang et al. (2021) or point cloud density Hu et al. (2023); Wei et al. (2022). However, despite progress in bridging the domain gap, two challenges remain unsolved: (1) **Insufficient cross-domain feature alignment**: Cross-domain feature differences result from multiple factors. Existing single-factor alignment strategies are insufficient to address the distribution differences in the feature space. As shown in Fig. 1a, existing methods often confuse features from different categories in the target domain, leading to performance degradation. Moreover, these methods mainly focus on **single-category 3D UDA** Yang et al. (2019); Shin et al. (2024); Chen et al. (2024b). In more complex and realistic multi-category scenarios Chen et al. (2023), they frequently struggle to maintain good performance due to inter-category feature interference. (2) **Ambiguous foreground-background**



(a) Insufficient feature alignment

(b) Noise Background

(c) Noise Foreground

Figure 1: (a) Visualization of BEV features showing confusion across different categories and scattered distribution patterns in the baseline method Yang et al. (2022); (b) Noise Background: Lots of positive samples are incorrectly labeled as negative in the pseudo-labeling process. (c) Proposals contain substantial noise with low-confidence foreground predictions that are easily confused with background.

decision boundaries: Due to the physical limitations of LiDAR, point clouds from distant objects and occluded areas are often sparse, making the decision boundary between foreground and background ambiguous. As shown in Fig. 1b, the high-confidence filtering mechanism of pseudo labels misclassifies many positive samples as negative, leading to the **noisy background** issue. Moreover, as shown in Fig. 1c, many easily confused background proposals are predicted as low-confidence foreground during training, further blurring the decision boundary and leading to the **noisy foreground** issue.

An intuitive solution to the issue of insufficient cross-domain feature alignment is to use prototypes to align cross-domain features, as explored in 2D UDA Liu et al. (2025); Lin et al. (2022). However, we argue that two points need to be considered: (1) Point cloud features are influenced by multiple factors, such as observation angles, distances and etc. These factors cause features to exhibit multi-center distribution characteristics. As shown in Fig.1a, samples of the category appear scattered in the feature space. It is challenging to align features using a universal prototype. (2) In a multi-prototype setting, it is necessary to consider how to assign features to the corresponding prototypes. Although Li et al. (2023) proposes a heuristic grouping strategy based on observation angles, this fixed grouping strategy is suboptimal when facing complex domain differences and struggles to adaptively align transferable features. In practice, such rigid grouping tends to cluster near dense objects and far sparse objects from the same viewpoint. Moreover, this method is designed for single-category tasks and requires further exploration in multi-category scenarios.

For the issue of ambiguous foreground-background decision boundaries, Li et al. (2023); Zhang et al. (2024) attempt to add high-confidence pseudo labels to unreliable background regions. However, pseudo labels themselves contain noise. Additionally, due to the high-confidence filtering mechanism, pseudo labels often focus on simple patterns and lack diversity, limiting model generalization. These methods also fail to address the noisy foreground issue.

Based on these observations, we propose a novel Adaptive Multi-Prototype Grouping Alignment framework for multi-category 3D UDA to address the two issues mentioned above. For the insufficient cross-domain feature alignment issue, we design a data-driven grouping model for each category that automatically identifies feature groups with different representations. They update dynamically to adapt to domain differences. During training, we extract features from both domains and assign them to the corresponding cross-domain learnable prototypes through the grouping model. We use contrastive learning between features and learnable prototypes to reduce cross-domain feature distance within groups while expanding inter-group differences. Moreover, we design two important components to address the issue of ambiguous foreground-background decision boundaries: (1) We propose the Noise Background Hybrid Augmentation strategy to alleviate the noisy background issue by introducing source labeled instances with high-confidence pseudo labels to background regions potentially containing positive samples. The introduction of source instances mitigate the noise in pseudo labels while enhance their diversity. (2) We propose a Noise Foreground Contrastive Learning strategy that leverages prototype guidance to push low-confidence ambiguous features away from decision boundaries, effectively mitigating the noisy foreground issue.

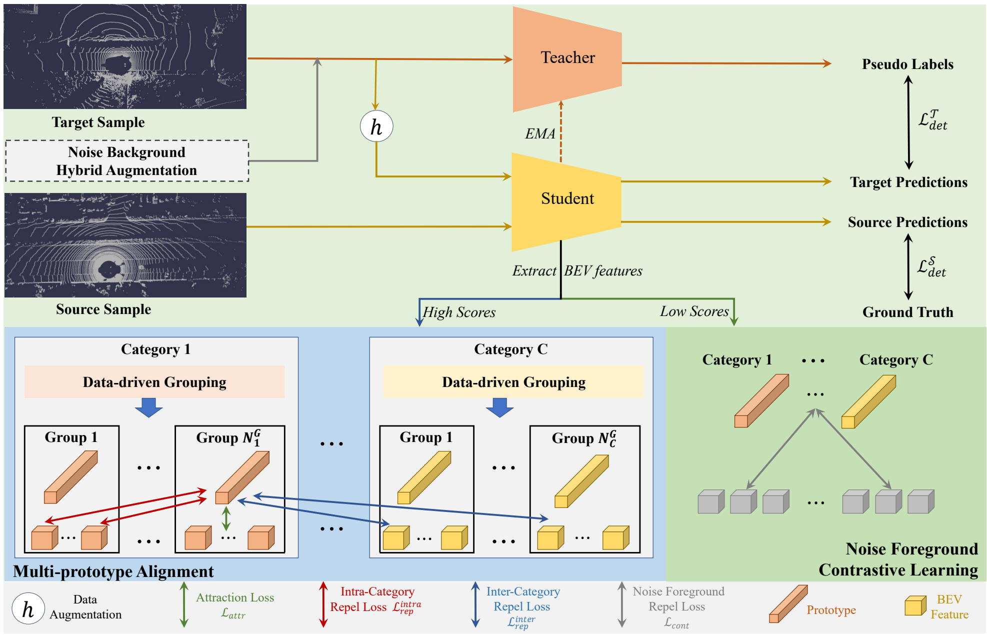


Figure 2: Overview of the proposed method. During training, we first group BEV features using our Data-driven Grouping method, and then align these features with corresponding category prototypes through Multi-prototype Alignment. Subsequently, we further optimize feature representations with Noise Foreground Contrastive Learning, and also enhancing the quality of pseudo labels using Noise Background Hybrid Augmentation. Additionally, both the grouping model and target domain pseudo labels are updated every K epochs to adapt to evolving feature distributions.

The main contributions of this paper include:

- We propose a novel Adaptive Multi-Prototype Grouping Alignment framework for multi-category 3D UDA that automatically identifies feature groups with different representations for effective cross-domain alignment.
- We design two components to address ambiguous foreground-background decision boundaries: a Noise Background Hybrid Augmentation strategy that integrates source labeled instances to enhance pseudo label quality, and a Noise Foreground Contrastive Learning strategy that pushes low-confidence features away from decision boundaries using prototype guidance.
- Extensive experiments on 3D UDA scenarios across multiple datasets validate the effectiveness of our method.

2 Related Work

LiDAR-based 3D Object Detection. LiDAR-based 3D object detection has been widely studied in recent years, which aims to detect objects in 3D space using point clouds. The existing methods can be roughly grouped into two categories: point-based methods and voxel-based methods. Point-based methods Shi et al. (2019) leverage the rich information in original point clouds and typically use PointNet series Qi et al. (2017a;b) to directly extract point-wise features. On the other hand, voxel-based methods Yan et al. (2018); Lang et al. (2019); Zhou & Tuzel (2018) divide point clouds into regular grids, encode features using sparse convolution, and map them to bird's eye view space. These methods usually achieve faster processing speeds. Some methods also combine the advantages of both points and voxels Shi et al. (2020; 2023). However, these excellent methods often experience sharp performance drops when applied to target domains Wang et al. (2020). Since current state-of-the-art methods are primarily voxel-based, we adopt SECOND-IOU Yan et al. (2018); Yang et al. (2021) and PVRCNN Shi et al. (2020) as our base detector for experiments.

Domain Adaptive 3D Object Detection. In recent years, UDA has been widely explored in 2D Ganin & Lempitsky (2015); Zhuo et al. (2022); Zhi et al. (2024); Deng et al. (2023), mainly using adversarial training and statistical methods Creswell et al. (2018); Long et al. (2017). Recently, several methods have also been proposed for 3D UDA. SN Wang et al. (2020) reduces size deviation by adjusting source domain data based on target domain size information. Since target domain size information is often hard to obtain, PLS Chen et al. (2024a) uses pseudo label information to guide source domain data scaling, while Sailor Malić et al. (2023) adapts anchor boxes to target domain size by searching for optimal features. LD Wei et al. (2022) uses point cloud distillation to adapt models from high-density to low-density point clouds. DTS Hu et al. (2023) proposes a teacher-student framework with point cloud beam random resampling to address differences in point cloud density. The ST3D series Yang et al. (2021; 2022) introduces a self-training pipeline that updates pseudo labels using a memory bank. MLC-net Luo et al. (2021) presents a teacher-student framework for multi-level consistency alignment. 3D-CoCo Yihan et al. (2021) employs cross-domain contrastive learning for transferable representations. GPA Li et al. (2023) groups features by observation angle for domain alignment. ReDB Chen et al. (2023) extends 3D UDA to multi-category settings, addressing unreliable pseudo-labels and class imbalance through cross-domain examination and class-balanced self-training. PERE Zhang et al. (2024) tackles unreliable multi-category pseudo labels by replacing them and generating additional proposals, using cross-domain triplet loss for category feature alignment. Despite these advances, most methods focus on single-category domain adaptation, neglecting multi-category adaptation challenges. Additionally, existing approaches don't fully address adaptive alignment in feature space, while our method achieves comprehensive feature alignment through a data-driven grouping approach.

3 Methodology

3.1 Problem Statement

Domain Adaptive 3D Object Detection aims to adapt a 3D detector pretrained on a labeled source domain $D_S = \{X_i^S, Y_i^S\}_{i=1}^{N_S}$ to an unlabeled and target domain $D_T = \{X_i^T\}_{i=1}^{N_T}$, where X_i^S and X_i^T are the input point clouds, Y_i^S are corresponding labels, and N_S and N_T are the number of samples in each domain, respectively. The 3D box label is represented as $(x, y, z, l, w, h, \theta, c)$, where x, y, z represents the box center, l, w, h represents the box size, θ represents the orientation, and c represents the category.

3.2 Overall Framework

Our method starts by pretraining a 3D detector on the source domain D_S with two domain adaptation data augmentation methods: ROS Yang et al. (2021) and RBRS Hu et al. (2023). For stable training, we adopt a teacher-student framework Hu et al. (2023) as our base structure. This framework consists of a non-trainable teacher model and a trainable student model with identical network architecture, both initialized with the source pretrained model. The weights of the teacher model are updated using exponential moving average (EMA). Since mainstream 3D detectors Yan et al. (2018); Shi et al. (2020) typically project sparse 3D features to bird's eye view (BEV), and previous works Yihan et al. (2021); Li et al. (2023); Hu et al. (2023) have shown good transferability of BEV features, we sample BEV features from the Region of Interest (RoI) head for feature alignment.

As shown in Fig. 2, during training, we first group the BEV features with our Data-driven Grouping method. We then align features and prototypes using our Multi-prototype Alignment method. We further optimize them through Noise Foreground Contrastive Learning. We generate target domain pseudo labels using the teacher model every K epochs and update the data-driven grouping model accordingly. We also apply Noise Background Hybrid Augmentation to process the pseudo labels. The overall prototype loss $\mathcal{L}_{proto}$ is defined as:

$$\mathcal{L}_{proto} = \beta_1 \mathcal{L}_{attr} + \mathcal{L}_{rep} + \beta_2 \mathcal{L}_{cont}, \tag{1}$$

where $\mathcal{L}_{attr}$ and $\mathcal{L}_{rep}$ are the attraction and repulsion losses in feature alignment, and $\mathcal{L}_{cont}$ is the noise foreground contrastive loss. β_1 and β_2 are the balance coefficients.

To reduce error accumulation, we also train the student model with the labeled source domain:

$$\mathcal{L}_{det}^S = \mathcal{L}_{reg}^S + \mathcal{L}_{cls}^S, \tag{2}$$

where $\mathcal{L}^S_{reg}$ and $\mathcal{L}^S_{cls}$ are the regression and classification losses in the source domain, respectively. The overall training loss is:

$$\mathcal{L} = \mathcal{L}^S_{det} + \mathcal{L}^T_{det} + \lambda \mathcal{L}_{proto}, \tag{3}$$

where $\mathcal{L}^T_{det}$ is the detection loss in the target domain with pseudo labels, and λ is the weight of the prototype loss.

3.3 Data-driven Grouping

Although multi-prototypes can better represent complex category features, grouping these category features remains challenging. Inspired by the promising performance of probability distributions in unsupervised learning Shin et al. (2024); Malić et al. (2023), we use a data-driven method to adaptively group features. We implement a Gaussian Mixture Model (GMM) as the grouping model, which is dynamically updated during training. We set up GMM models with $\{N^G_c\}^C_{c=1}$ Gaussian distributions, where C is the number of categories. The number of Gaussian distributions in each GMM model equals the number of learnable prototypes for each category, enabling one-to-one correspondence.

Adaptive Group. During training, we extract BEV features of the point cloud and use GMM to group them. For the k-th BEV feature f^c_k, we find the corresponding GMM model based on its predicted category c, and then calculate the probability that it belongs to the i-th Gaussian distribution, which is also the probability that it belongs to the i-th prototype of that category, defined as:

$$p^k_i = \frac{\mathcal{N}(f^c_k; \mu_i, \Sigma_i)}{\sum_j^{N^G_c} \mathcal{N}(f^c_k; \mu_j, \Sigma_j)}, \tag{4}$$

where N^G_c is the number of Gaussian distributions in the GMM model for category c, and $\mathcal{N}(f^c_k; \mu_j, \Sigma_j)$ is the Gaussian distribution defined as:

$$\mathcal{N}(f^c_k; \mu_j, \Sigma_j) = \frac{exp(-\frac{1}{2}(f^c_k - \mu_j)^T \Sigma_j^{-1} (f^c_k - \mu_j))}{\sqrt{(2\pi)^{N^G_c} |\Sigma_j|}}, \tag{5}$$

where μ_j and Σ_j are the mean and covariance of the j-th Gaussian distribution, respectively. The group label of feature f^c_k is defined as:

$$g^k_c = \underset{i \in \{1,2,\ldots,N^G_c\}}{\arg\max}\, p^k_i \,. \tag{6}$$

This approach allows us to adaptively assign a group label to each BEV feature.

Dynamic Update. Considering that the training data fluctuates significantly in each batch, updating the GMM every batch may affect its performance. To prevent the feature distribution from being dominated by source domain features, we select equal amounts of source and target domain data to fit the GMM model parameters $\{\mu_i, \Sigma_i, \pi_i\}^{N^G_c}_{i=1}$ for each category. Both initialization and updates use the Expectation-Maximization (EM) algorithm Dempster et al. (1977). For the update stage, we use the current epoch of BEV features and the GMM model to perform the EM algorithm to obtain the updated GMM model parameters for each category, $\{\hat{\mu_i}, \hat{\Sigma}_i, \hat{\pi_i}\}^{N^G_c}_{i=1}$.

3.4 Multi-prototype Alignment

At the beginning of training, we randomly initialize a set of learnable prototypes for each category, $\{\{\mathcal{G}^c_i\}^{N^G_c}_{i=1}\}$, with the number matching the grouping model. We assign BEV features from the source and target domains to the corresponding group , and then perform intra-group attraction and inter-group repulsion between the features and prototypes.

We push features towards the corresponding prototypes to encourage similar features to cluster together. The attraction loss Li et al. (2023); Shin et al. (2024) between intra-group features and prototypes is defined as:

$$\mathcal{L}_{attr} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{|F_c|} \sum_{f^c_k \in F_c} \sum_{i=1}^{N^G_c} \mathbb{I}(g^k_c = i) \cdot (1 - \cos(f^c_k, \mathcal{G}^c_i)), \tag{7}$$

where $\mathbb{I}(\cdot)$ is the indicator function, which is 1 if the condition is true and 0 otherwise. $|F_c|$ is the number of features in category c, and f_k^c is the k-th feature in category c.

We push features away from the prototypes of other groups to enhance discriminative ability. We first separate different prototypes within the same group to ensure that each group learns distinct features. The intra-category repulsion loss Li et al. (2023) is defined as:

$$\mathcal{L}_{rep}^{intra} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{|F_c|}\sum_{f_k^c \in F_c}\sum_{i=1}^{N_c^G}\mathbb{I}(g_c^k \neq i)\cdot \max(0, \cos(f_k^c, \mathcal{G}_i^c)). \tag{8}$$

For features and prototypes of different categories, we apply repulsion to strengthen inter-category discriminative ability. The inter-category repulsion loss is defined as:

$$\mathcal{L}_{rep}^{inter} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{|F_c|}\sum_{f_k^c \in F_c}\sum_{c' \neq c}\sum_{i=1}^{N_{c'}^G}\max(0, \cos(f_k^c, \mathcal{G}_i^{c'})). \tag{9}$$

The overall repulsion loss is:

$$\mathcal{L}_{rep} = \beta_3 \mathcal{L}_{rep}^{intra} + \beta_4 \mathcal{L}_{rep}^{inter}, \tag{10}$$

where β_3 and β_4 are the balance coefficients.

3.5 Noise Foreground Contrastive Learning

During training, the model often misclassifies confusing background regions as low-confidence foreground, limiting its ability to distinguish between foreground and background. Since background features are typically dispersed and complex, making it difficult to construct effective background prototypes, we instead repel low-confidence foreground features from existing prototypes to enhance the discriminative ability. Our experiments (Table 5) show that repelling low-confidence foreground features is more effective than constructing dedicated background prototypes.

We define foreground features with confidence below T_{neg} as low-confidence foreground features and repel them from all category prototypes:

$$\mathcal{L}_{cont} = \frac{1}{|F_{low}|}\sum_{f_k^c \in F_{low}}\sum_{c=1}^{C}\sum_{i=1}^{N_c^G}\max(0, \cos(f_k^c, \mathcal{G}_i^c)), \tag{11}$$

where F_{low} represents the set of low-confidence foreground features. This loss encourages the model to push ambiguous low-confidence foreground features away from category prototypes, thereby improving the robustness and foreground-background separation.

3.6 Noise Background Hybrid Augmentation

Self-training methods typically select high-confidence predictions as pseudo labels for training to reduce noise influence. However, they inevitably miss some low-confidence positive samples, wrongly treating them as background. We propose the Noise Background Hybrid Augmentation module to mitigate this negative supervision effect.

Specifically, we extract instances from the source domain and high-confidence instances from the target domain to build a hybrid instance bank. We identify pseudo labels with confidence scores in the range $[T_{neg}, T_{pos}]$ as noise background, where T_{neg} and T_{pos} are pre-defined thresholds for selecting negative and positive samples. During training, we randomly select samples from the bank and paste them into noise background regions of the current target frame. Following Li et al. (2023); Zhang et al. (2024), we ensure selected samples match the original region through scaling and rotation operations. We use hyperparameter γ to control the probability of selecting source domain samples with labels. By enriching the target domain with diverse and reliable foreground instances, our method effectively stabilizes the self-training process.

Table 1: Cross-dataset 3D object detection results with SECOND-IoU backbone for Waymo → KITTI and nuScenes → KITTI scenarios across different difficulty levels. For fair comparison, all methods use the same pre-trained model. Methods marked with * are originally designed for single-category detection and we modified for multi-category detection. ReDB† follows the original 120-epoch training strategy from its original paper, while all other methods use 30 epochs as in their original papers. PERE‡ is our implementation based on the original paper. The best results are shown in **bold** and the second best are underlined.

Task	Method	Easy		Moderate		Hard	
		AP$_{\text{BEV}}$/AP$_{\text{3D}}$	**Closed Gap**	**AP$_{\text{BEV}}$/AP$_{\text{3D}}$**	**Closed Gap**	**AP$_{\text{BEV}}$/AP$_{\text{3D}}$**	**Closed Gap**
W → K	Source Only	46.88/28.81	-/-	41.02/25.03	-/-	38.71/23.60	-/-
	ST3D*	65.34/48.44	64.0%/46.6%	58.38/43.66	70.7%/55.2%	55.75/41.68	73.6%/58.9%
	ST3D++*	66.20/53.71	66.9%/59.1%	59.37/47.52	74.8%/66.5%	56.92/45.36	78.5%/70.5%
	DTS*	65.68/55.64	65.1%/63.5%	59.85/49.43	76.7%/72.2%	57.08/47.06	79.2%/76.6%
	ReDB†	63.41/55.46	57.3%/63.3%	56.06/47.13	61.4%/65.1%	54.11/44.93	66.3%/69.4%
	PERE‡	66.35/54.26	67.5%/60.4%	58.65/47.92	71.8%/67.6%	56.17/45.42	75.2%/71.0%
	Ours	**68.33/58.90**	**74.4%/71.4%**	**61.33/51.46**	**82.7%/78.1%**	**58.72/48.66**	**86.2%/81.6%**
	Oracle	75.79/70.96	-/-	65.57/58.88	-/-	61.93/54.27	-/-
N → K	Source Only	21.16/9.22	-/-	17.41/7.43	-/-	17.51/7.12	-/-
	ST3D*	42.65/36.09	39.3%/43.5%	37.53/30.03	41.9%/43.9%	36.51/28.05	42.8%/44.3%
	ST3D++*	50.58/41.54	53.8%/52.4%	43.35/34.30	53.8%/52.2%	42.03/32.15	55.2%/53.1%
	DTS*	53.72/37.78	59.6%/46.3%	46.26/30.95	59.8%/45.7%	44.27/29.17	60.2%/46.7%
	ReDB†	46.94/38.42	47.1%/47.3%	40.62/31.17	48.1%/46.1%	39.18/29.83	48.9%/48.1%
	PERE‡	51.19/42.32	55.0%/53.7%	44.32/35.27	55.9%/54.0%	42.47/33.56	56.2%/56.1%
	Ours	**56.91/48.33**	**65.4%/63.4%**	**48.06/39.43**	**63.6%/62.1%**	**46.08/37.33**	**64.3%/64.0%**
	Oracle	75.79/70.96	-/-	65.57/58.88	-/-	61.93/54.27	-/-

Table 2: Cross-dataset 3D object detection results with SECOND-IoU backbone for Waymo → nuScenes scenario.

Method	AP$_{\text{BEV}}$	Closed Gap	AP$_{\text{3D}}$	Closed Gap
Source Only	11.41	–	7.30	–
ST3D*	17.35	31.2%	10.68	23.8%
ST3D++*	17.44	32.0%	11.07	26.5%
DTS*	17.46	32.1%	10.60	23.2%
ReDB†	14.67	17.1%	8.20	6.3%
PERE‡	17.42	31.8%	10.30	21.1%
Ours	**17.73**	**33.0%**	**11.21**	**27.5%**
Oracle	30.45	–	21.54	–

Table 3: Performance comparison of various methods on the Waymo → KITTI adaptation scenario with PVRCNN. We report the AP at **Moderate** difficulty.

Method	AP$_{\text{3D}}$	AP$_{\text{BEV}}$
ST3D*	53.29	63.33
ST3D++*	53.17	63.99
DTS*	52.68	63.42
PERE‡	50.73	58.33
Ours	**54.49**	**64.65**

4 Experiments

4.1 Expreimental Settings

Datasets. We conduct experiments on three popular autonomous driving datasets: KITTI Geiger et al. (2012), Waymo Sun et al. (2020), and nuScenes Caesar et al. (2020). Following previous works Yang et al. (2021; 2022); Hu et al. (2023); Li et al. (2023); Shin et al. (2024); Zhang et al. (2024), we conduct experiments in three domain adaptation scenarios (Waymo → KITTI, nuScenes → KITTI, Waymo → nuScenes) and evaluate all models on the validation set of each target dataset. Please refer to the supplements for more datasets details.

Evaluation Metrics. Following Yang et al. (2021); Chen et al. (2023); Zhang et al. (2024), we use the official KITTI evaluation metric for 3D object detection over 40 recall positions with IoU thresholds of 0.7, 0.5 and 0.5 for Car, Pedestrian, and Cyclist. In the nuScenes dataset, we merge the bicycle and motorcycle categories into the Cyclist category for consistency across datasets.

We also adopt the Closed Gap Yang et al. (2021) metric, which is defined as $Closed\ Gap = \frac{AP_{model} - AP_{source}}{AP_{oracle} - AP_{source}} \times 100\%$.

Implementation Details. We use OpenPCDet Team (2020) as our codebase to pre-train detectors on the source domain. In the self-training process, we fine-tune the model for 30 epochs. We use Adam optimizer Kingma & Ba (2014) with one cycle scheduler to adjust the learning rate, which was set to 0.003 for pretraining and 0.0015 for fine-tuning. Please refer to the supplements for more implementation details.

Compared Methods. We compare our proposed method with existing state-of-the-art methods Yang et al. (2021; 2022); Hu et al. (2023); Chen et al. (2023); Zhang et al. (2024). In addition, we also report the results of **Source Only** and **Oracle**. **Source Only** indicates directly evaluating the source domain pre-trained model on the target domain, while **Oracle** is the result of fully supervised training on the target domain, serving as an upper bound for performance.

4.2 Comparison with State-of-the-art Methods

Main Results. As shown in Table 1, our method demonstrates exceptional performance across all challenging domain adaptation scenarios, consistently achieving the best results in both AP_{BEV} and AP_{3D} metrics across all difficulty levels. Particularly in the nuScenes → KITTI scenario, our method achieves average improvements of 2.27% in AP_{BEV} over the second-best method DTS Hu et al. (2023) and 4.65% in AP_{3D} over the second-best method PERE Zhang et al. (2024) respectively. As shown in Table 2, even in the challenging Waymo → nuScenes scenario with significant LiDAR beam density differences, our method consistently maintains the best performance.

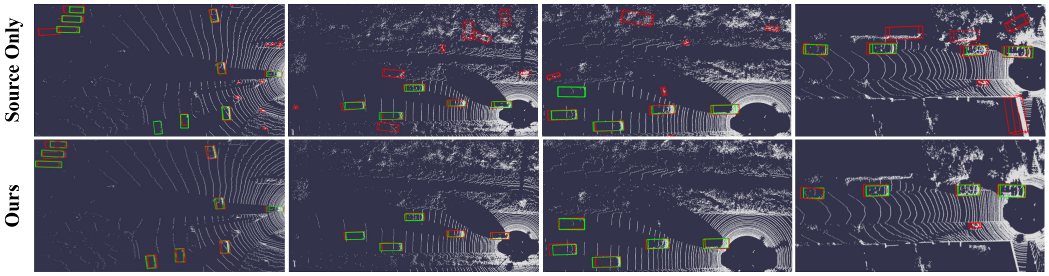


Figure 3: Qualitative results on nuScenes → KITTI. The predictions and ground truths are painted in red and green, respectively.

Evaluation Across Different Architectures. Table 3 verifies the effectiveness of our method with a different backbone architecture. Using PVRCNN Shi et al. (2020) for Waymo → KITTI adaptation, our method achieves the best average AP_{BEV} and AP_{3D}, improving upon the second-best method (ST3D Yang et al. (2021), ST3D++ Yang et al. (2022)) by 0.66% and 1.20% respectively. This demonstrates the general applicability of our method across different model architectures.

4.3 Ablation Studies

In this section, all ablation experiments are based on SECOND-IoU, conducted on nuScenes → KITTI, and evaluated for average results under moderate difficulty.

Main Ablation Studies. Table 4 shows the contribution of each component in our method. Starting with the baseline (a), adding Noisy Background Hybrid Augmentation (NBHA) in setting (b) improves AP_{3D} by 2.78% and AP_{BEV} by 4.20% by reducing noisy supervision from noisy background regions. The attraction loss in setting (c) brings additional gains of 0.47% in AP_{3D} and 0.88% in AP_{BEV}. The repulsion loss in setting (d) further improves results by 0.65% in AP_{3D} and 0.89% in AP_{BEV}. Setting (e) confirms that attraction loss and repulsion work synergistically. Setting (f) shows that without NBHA, performance drops substantially even with all other components present, highlighting NBHA's importance in addressing noise backgrounds. Our complete method (g) with Noisy Foreground Contrastive Learning (NFCL) achieves the best results, with further improvements of 1.45% in AP_{3D} and 0.96% in AP_{BEV}.

Table 4: Ablation study of different components in our method.

Setting	NBHA	Attr	Repel	NFCL	$\mathbf{AP_{3D}}$	$\mathbf{AP_{BEV}}$
(a)					34.08	41.13
(b)	✓				36.86	45.33
(c)	✓	✓			37.33	46.21
(d)	✓		✓		37.31	45.39
(e)	✓	✓	✓		37.98	47.10
(f)		✓	✓	✓	35.23	42.42
(g)	✓	✓	✓	✓	**39.43**	**48.06**

Table 5: Ablation study on the impact of NBHA. CA represents the complementary augmentation.

Method	$\mathbf{AP_{BEV}}$	$\mathbf{AP_{3D}}$
Baseline	41.13	34.08
CA	46.03	37.31
Ours	**48.06**	**39.43**

Table 6: Ablation study of different prototype strategies.

Setting	Proto	Multi	K-Means	GMM	$\mathbf{AP_{3D}}$	$\mathbf{AP_{BEV}}$
(a)					34.08	41.13
(b)	✓				36.03	47.41
(c)				✓	38.11	47.25
(d)	✓	✓	✓		38.25	45.94
(e)	✓	✓		✓	**39.43**	**48.06**

Table 7: Ablation study on the impact of NFCL.

Method	$\mathbf{AP_{BEV}}$	$\mathbf{AP_{3D}}$
Baseline	41.13	34.08
BG Proto	47.26	37.02
Ours	**48.06**	**39.43**

Analysis of Prototype Alignment Architecture. Table 6 compares different prototype strategies. Setting (b) shows that using only single prototype for alignment achieves limited improvement, indicating that single prototype struggles to represent complex category features effectively. Setting (c), which uses only GMM distribution for alignment without explicit prototypes, performs better than single prototype alignment with 38.11% $\mathrm{AP_{3D}}$, demonstrating the effectiveness of modeling feature distribution. Setting (d) combines multiple learnable prototypes with K-Means clustering, yielding comparable $\mathrm{AP_{3D}}$. Our proposed method in setting (e), which integrates GMM-based grouping with multiple learnable prototypes, achieves the best performance. This demonstrates that GMM provides more effective feature grouping than K-Means, and when combined with multiple learnable prototypes, enables more precise alignment of complex feature distributions across domains.

Analysis of Noise Background Hybrid Augmentation. Table 5 compares our NBHA with other method. Complementary Augmentation (CA) Zhang et al. (2024) only uses high-confidence instances from the target domain to fill noisy backgrounds. In contrast, our NBHA incorporates instances from the source domain in addition to target domain instances, effectively alleviating the noise by introducing more reliable supervision signals. The results demonstrate that NBHA achieves superior performance with improvements of 2.03% in $\mathrm{AP_{BEV}}$ and 2.12% in $\mathrm{AP_{3D}}$ compared with CA.

Analysis of Noise Foreground Contrastive Learning. Table 7 shows the effectiveness of our NFCL approach. Background features vary greatly across different scenes, making them difficult to represent with prototypes. While using background prototypes (BG Proto) improves over the baseline, it cannot accurately capture background distributions. Our NFCL uses noisy foreground features to optimize prototypes through contrastive learning, improving the model's ability to separate foreground from background. NFCL achieves better performance with 2.41% improvement in $\mathrm{AP_{3D}}$ and 0.80% in $\mathrm{AP_{BEV}}$ compared with BG Proto.

Qualitative Results. Fig. 3 presents qualitative results. Our method generates clean and more accurate predictions, demonstrating the effectiveness of our method.

5 Conclusion

In this paper, we propose a novel Adaptive Multi-Prototype Grouping Alignment framework for unsupervised domain adaptation on multi-category 3D detection. Our method addresses the challenges of inadequate cross-domain feature alignment and ambiguous foreground-background decision boundaries. We design a data-driven grouping model that dynamically updates feature groups and assigns learnable prototypes to each group. Additionally, we introduce Noise Background Hybrid Augmentation and Noise Foreground Contrastive Learning to enhance feature alignment and mitigate the impact of unreliable pseudo-labels. Extensive experiments validate the effectiveness of our method. In future work, we plan to explore the application on multi-modal 3D UDA tasks.

REFERENCES

Syeda Mariam Ahmed, Yan Zhi Tan, Chee Meng Chew, Abdullah Al Mamun, and Fook Seng Wong. Edge and corner detection for unorganized 3d point clouds with application to robotic welding. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 7350–7355. IEEE, 2018.

Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

Shijie Chen, Rongquan Wang, Xin Li, Yuchen Wu, Haizhuang Liu, Jiansheng Chen, and Huimin Ma. Pls: Unsupervised domain adaptation for 3d object detection via pseudo-label sizes. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6370–6374. IEEE, 2024a.

Shijie Chen, Junbao Zhuo, Xin Li, Haizhuang Liu, Rongquan Wang, Jiansheng Chen, and Huimin Ma. Cmt: Co-training mean-teacher for unsupervised domain adaptation on 3d object detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 4738–4747, 2024b.

Zhuoxiao Chen, Yadan Luo, Zheng Wang, Mahsa Baktashmotlagh, and Zi Huang. Revisiting domain-adaptive 3d object detection by reliable, diverse and class-balanced pseudo-labeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3714–3726, 2023.

Spconv Contributors. Spconv: Spatially sparse convolution library. `https://github.com/traveller59/spconv`, 2022.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35 (1):53–65, 2018.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

Jinhong Deng, Dongli Xu, Wen Li, and Lixin Duan. Harmonious teacher for cross-domain object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23829–23838, 2023.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, 2012.

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020.

Qianjiang Hu, Daizong Liu, and Wei Hu. Density-insensitive unsupervised domain adaption on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17556–17566, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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/CVF conference on computer vision and pattern recognition*, pp. 12697–12705, 2019.

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.

Ziyu Li, Jingming Guo, Tongtong Cao, Liu Bingbing, and Wankou Yang. Gpa-3d: Geometry-aware prototype alignment for unsupervised domain adaptive 3d object detection from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6394–6403, 2023.

Hongbin Lin, Yifan Zhang, Zhen Qiu, Shuaicheng Niu, Chuang Gan, Yanxia Liu, and Mingkui Tan. Prototype-guided continual adaptation for class-incremental unsupervised domain adaptation. In *European Conference on Computer Vision*, pp. 351–368. Springer, 2022.

Chenguang Liu, Yongchao Feng, Yanan Zhang, Qingjie Liu, and Yunhong Wang. Pacf: Prototype augmented compact features for improving domain adaptive object detection. *arXiv preprint arXiv:2501.08605*, 2025.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pp. 2208–2217. PMLR, 2017.

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*, pp. 8866–8875, 2021.

Flavio BP Malavazi, Remy Guyonneau, Jean-Baptiste Fasquel, Sebastien Lagrange, and Franck Mercier. Lidar-only based navigation algorithm for an autonomous agricultural robot. *Computers and electronics in agriculture*, 154:71–79, 2018.

Dušan Malić, Christian Fruhwirth-Reisinger, Horst Possegger, and Horst Bischof. Sailor: Scaling anchors via insights into latent object representation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 623–632, 2023.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 770–779, 2019.

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*, pp. 10529–10538, 2020.

Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, 131(2):531–551, 2023.

Sangyun Shin, Yuhang He, Madhu Vankadari, Ta-Ying Cheng, Qian Xie, Andrew Markham, and Niki Trigoni. Towards learning group-equivariant features for domain adaptive 3d detection. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pp. 18019–18035, 2024.

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*, pp. 2446–2454, 2020.

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

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11713–11723, 2020.

Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. In *European Conference on Computer Vision*, pp. 179–195. Springer, 2022.

Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

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*, pp. 10368–10378, 2021.

Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):6354–6371, 2022.

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*, pp. 1951–1960, 2019.

Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Advances in Neural Information Processing Systems*, 34:21493–21504, 2021.

Zhanwei Zhang, Minghao Chen, Shuai Xiao, Liang Peng, Hengjia Li, Binbin Lin, Ping Li, Wenxiao Wang, Boxi Wu, and Deng Cai. Pseudo label refinery for unsupervised domain adaptation on cross-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15291–15300, 2024.

Churan Zhi, Junbao Zhuo, and Shuhui Wang. Confusing pair correction based on category prototype for domain adaptation under noisy environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17060–17068, 2024.

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*, pp. 4490–4499, 2018.

Junbao Zhuo, Shuhui Wang, and Qingming Huang. Uncertainty modeling for robust domain adaptation under noisy environments. *IEEE Transactions on Multimedia*, 25:6157–6170, 2022.

A Appendix

In this supplementary material, we provides more details of our experiments and additional experimental results and analysis.

B Use of Large Language Models (LLMs)

We used a large language model (e.g., DeepSeek) solely as a general-purpose writing assistant for minor language polishing (grammar, phrasing, and formatting consistency). The LLM did not contribute to research ideation, problem formulation, methodological design, data analysis, result interpretation, or the drawing of scientific conclusions. All scientific content, including ideas, methods, experiments, and claims, originates from the authors. The authors verified and edited all text and take full responsibility for the content.

C Datasets

We conduct experiments on three widely used autonomous driving datasets: KITTI Geiger et al. (2012), Waymo Sun et al. (2020), and nuScenes Caesar et al. (2020). These datasets were collected using different LiDAR configurations across various geographical locations. The KITTI dataset was collected in Germany using a 64-beam Velodyne LiDAR and contains 7,481 labeled frames for training and 7,518 unlabeled frames for testing. Following Yang et al. (2021; 2022), we divide the training set into two disjoint subsets: 3,712 samples for training and 3,769 samples for validation. The Waymo dataset was collected in the United States using one 64-beam and four 200-beam LiDAR sensors, which consists of 798 sequences (over 150,000 frames) for training and 202 sequences (approximately 40,000 frames) for validation. Following Yang et al. (2021; 2022), we randomly subsample 20% of the original training set. The nuScenes dataset, collected in both the United States and Singapore using a 32-beam LiDAR, contains 28,130 samples for training and 6,019 samples for validation.

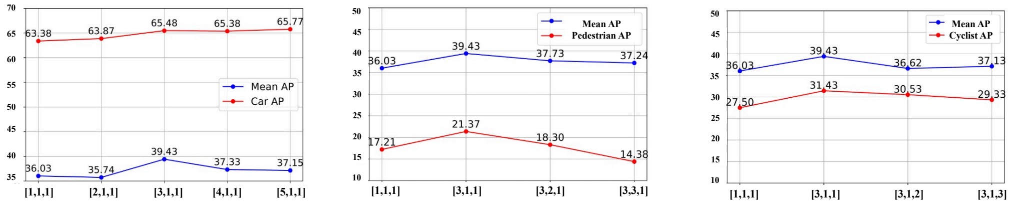
D More Implementation Details

We use OpenPCDet Team (2020) as our codebase to pre-train detectors on the source domain. In the pre-training process, the learning rate is set to 1.0×10^{-2}. In the self-training process, we fine-tune the model for 30 epochs. We use Adam optimizer Kingma & Ba (2014) with one cycle scheduler to adjust the learning rate, which was set to 0.003 for pretraining and 0.0015 for fine-tuning. we use Adam optimizer Kingma & Ba (2014) with one cycle scheduler for 30 epochs. The learning rate is set to 1.5×10^{-3}. We update pseudo labels and GMM models every $K = 2$ epochs. The threshold of pseudo labels $[T_{neg}, T_{pos}]$ are consistent with previous works Yang et al. (2021; 2022); Hu et al. (2023), between 0.2-0.5. While in the Waymo → nuScenes task, we do not update pseudo labels as previous works Yang et al. (2021; 2022) for fair comparisons. We empirically set the balance coefficients λ, β_1, β_2, β_3 ,β_4 and γ to be 0.1, 1.0, 0.5, 0.5 ,1.5 and 0.7, respectively.

E Additional Experimental Results

Sensitivity Analysis of hyperparameters. We conduct a sensitivity analysis on six key hyperparameters in our method: λ, β_1, β_2, β_3, β_4, and γ. The values used in our final model are 0.1, 1.0, 0.5, 0.5, 1.5, and 0.7, respectively, under which we achieve a 3D AP of 39.43. To evaluate the robustness of our method to hyperparameter variations, we individually vary each parameter while keeping the others fixed and report the resulting 3D AP in Table 8. The results show that our method maintains stable performance under different settings, indicating good robustness. Notably, γ and β_2 exhibit more significant influence: increasing γ from 0.5 to 0.7 (ours) leads to a substantial gain of 2.69 points in 3D AP, and decreasing β_2 from 0.5 (ours) to 0.1 brings moderate improvement. These observations suggest that careful tuning of these parameters can yield further performance gains.

Analysis of Results Under Different Training Settings. Fair comparison between different methods is non-trivial due to inconsistencies in codebases and pretraining configurations. Specifically,



(a) Car Prototype Number **(b) Pedestrian Prototype Number** **(c) Cyclist Prototype Number**

Figure 4: Class AP and mean AP (%) under varying prototype numbers for each class. (a) Increasing Car prototypes improves performance up to 3, then saturates. (b, c) Pedestrian and Cyclist perform best with a single prototype; more prototypes degrade performance due to redundancy.

Table 8: Sensitivity analysis of hyperparameters. The performance is reported in terms of 3D AP (%) under different values. The values used in our final model are highlighted in bold.

Hyperparameter	Value	3D AP (%)
λ	0.05	37.14
	0.5	37.58
	0.1 (ours)	**39.43**
β_1	0.5	37.35
	1.5	36.87
	1.0 (ours)	**39.43**
β_2	0.1	38.40
	1.0	38.27
	0.5 (ours)	**39.43**
β_3	0.1	37.53
	1.0	37.43
	0.5 (ours)	**39.43**
β_4	1.0	37.71
	2.0	38.10
	1.5 (ours)	**39.43**
γ	0.5	36.74
	1.0	38.01
	0.7 (ours)	**39.43**

DTS Hu et al. (2023) is implemented with `spconv 2.x` Contributors (2022) and adopts RBRS Hu et al. (2023) for pretraining, whereas ST3D Yang et al. (2021) is based on `spconv 1.x` and does not incorporate RBRS. Such differences introduce performance gaps unrelated to algorithmic design, making direct comparison less informative.

To ensure fairness, we re-implement and rerun all methods under a unified codebase and consistent pretraining settings. As shown in Table 9, the impact of both codebase version and RBRS is substantial. For instance, when re-running ST3D with `spconv 2.x`, its 3D AP drops from 23.27 to 20.28, indicating that the codebase upgrade alone can negatively affect performance. However, introducing RBRS consistently improves performance across both codebases. For ST3D, RBRS boosts the 3D AP from 23.27 to 28.67 under `spconv 1.x`, and from 20.28 to 30.03 under `spconv 2.x`. A similar trend is observed for ST3D++, where RBRS improves the AP from 28.40 to 33.20 under `spconv 1.x`, and from 30.94 to 34.30 under `spconv 2.x`.

These results demonstrate that both the spconv version and the use of RBRS significantly affect model performance. Importantly, our final setting—`spconv 2.x` with RBRS—not only enables a fair and modern implementation, but also achieves the best 3D AP among all combinations for both ST3D and ST3D++, validating the effectiveness and robustness of our approach.

Table 9: 3D AP (%) comparison of ST3D and ST3D++ under different combinations of `spconv` versions and RBRS usage. Abbreviations: v1 = `spconv 1.x`, v2 = `spconv 2.x`, R = with RBRS, Ours = v2 + RBRS.

Method	v1 (Orig.)	v1+R	v2	v2+R (Ours)
ST3D	23.27	28.67	20.28	**30.03**
ST3D++	28.40	33.20	30.94	**34.30**

Analysis of Prototype Numbers. We consistently observe that the group configuration [3,1,1] yields the best overall performance across all evaluated settings. This can be attributed to the varying levels of complexity and domain discrepancy among different object categories. As shown in Figure 4(a), increasing the number of prototypes for the Car class from 1 to 3 significantly improves both Car AP and mean AP. This suggests that a richer prototype representation is beneficial for modeling the higher intra-class variation and larger domain gap observed in the Car class. However, allocating more than three prototypes to the Car class introduces imbalance, which slightly degrades the mean AP.

In contrast, Figures 4(b) and (c) show that the Pedestrian and Cyclist classes perform best when assigned a single prototype. Increasing the number of prototypes for these relatively simpler classes leads to performance degradation, likely due to redundancy and overfitting. For example, the [3,3,1] and [3,1,3] configurations consistently underperform compared to [3,1,1]. This confirms that allocating excessive prototypes to low-complexity classes not only impairs their own AP but also negatively affects the overall detection performance. These findings underscore the importance of class-specific prototype allocation, where more complex classes benefit from multiple prototypes, while simpler classes are better represented with fewer.