GPDAN: Grasp Pose Domain Adaptation Network for Sim-to-Real 6-DoF Object Grasping

Liming Zheng[1], Wenxuan Ma[1], Yinghao Cai[2,*], Tao Lu[2], Shuo Wang[2]

***Abstract*—In this paper, we propose a novel Grasp Pose Domain Adaptation Network (GPDAN) to achieve sim-to-real domain adaptation for 6-DoF grasp pose detection. The main task of GPDAN is to detect feasible 6-DoF grasp poses in cluttered scenes. A point-wise self-supervised domain classification module with point cloud mixture and feature fusion strategy is proposed as the auxiliary task to promote the feature alignment between the source and target domain through adversarial training. Experimental results on both simulation and real-world environments demonstrate that GPDAN outperforms other approaches in detecting 6-DoF grasps on the target domain, highlighting the effectiveness of GPDAN in improving the performance of 6-DoF grasp pose detectors trained in simulation and deployed in real-world environments without any further laborious labeling.**

***Index Terms*—Domain Adaptation, Grasp Pose Detection, Feature Alignment, Sim-to-real.**

I. Introduction

Robotic grasping is a fundamental capacity that is required for many robot manipulation tasks. However, the large $SE(3)$ space for the 6-DoF grasp poses makes the grasp pose detection in cluttered scenes quite challenging. Most state-of-the-art grasp pose detection methods [1]–[3] are trained with labeled data collected in simulation and applied directly to the real-world environments due to the high cost of data collection in the real world and the difficulty of labeling the theoretically infinite ground truth 6-DoF grasp labels in point clouds. Although the domain gap in point clouds is much smaller than that of other 2D visual information [4], there still exists significant reality gaps between simulated and real-world environments. For example, the spatial sizes of the objects, the density of points and noise patterns of point clouds differ between simulated and real-world data as shown in Fig. 1 [5], [6]. These reality gaps limit the performance of grasp pose detection trained in simulation when applied to the real world. The high-scored predictions may fail during execution, resulting in serious consequences in real-world environments. Domain adaptation (DA) can be a solution to this problem by aligning features in the source domain (simulation) and target domain (real world). In this paper, we propose a novel Grasp Pose Domain Adaptation Network (GPDAN) for 6-DoF grasp pose detection. With domain adaptation, the grasp pose detector is able to produce similar results for similar geometric features in these two domains so that the performance of the grasp pose detector in the real-world environments can be improved.

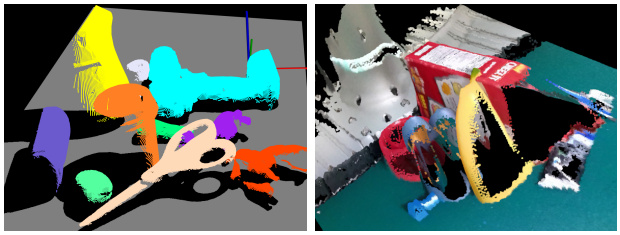

Fig. 1. Point clouds in simulation and real-world environments. (Left) The objects in the synthetic point clouds usually have a smooth surface formed by uniformly distributed points. (Right) The point clouds in the real-world scenes which are captured by an RGB-D camera inevitably contain noise patterns such as rough surfaces and holes due to various reasons such as reflectance, transparency, occlusions, etc.

Various simulation-to-reality domain adaptation methods have been proposed in many visual and robotic tasks to bridge the domain gap across different domains. A straightforward way is domain randomization, which randomizes the task-irrelevant properties such as background color, object material and dynamic parameters in simulation to force the network to extract task relevant features [7], [8]. Recent works have focused on methods based on geometric feature learning, including feature-based methods [9], [10], reconstruction-based methods [11], [12] and adversarial-based methods [13]–[15]. Through domain adaptation, the module for the main task can process data from both domains in a more uniform way, which improves the performance of grasp detection task in the target domain. Our work falls into the category of adversarial-based methods, which encourage the feature extractor to extract similar geometric features for the task of 6-DoF grasp pose detection that can confuse the domain classifier from both the source and target domains.

However, directly applying adversarial training to the task of grasp pose detection is quite challenging, because of the complexity of the grasp pose configurations and the inherent difficulties of adversarial training such as gradient vanishing

Manuscript received: April 1, 2023; Accepted: May 29, 2023.

This paper was recommended for publication by Editor Hong Liu upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0103003 and the National Natural Science Foundation of China under Grant U1913201 and 62273342.

[1]Liming Zheng and Wenxuan Ma are with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 101408, China. zlm898@163.com

[2]Yinghao Cai, Tao Lu and Shuo Wang are with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. Yinghao Cai is also with the Centre for Artificial Intelligence and Robotics (CAIR), the Hong Kong Institute of Science & Innovation, Chinese Academy of Sciences (HKISI-CAS). Corresponding author: Yinghao Cai. yinghao.cai@ia.ac.cn

Digital Object Identifier (DOI): see top of this page.

and mode collapse. On one hand, using Generative Adversarial Networks (GANs) to reconstruct the input point cloud by extracting features from the point cloud itself [16], [17] cannot achieve good performance on the 6-DoF grasp pose detection task. Point cloud is a highly structured data format and the grasp detection task heavily relies on the geometric details. The ambiguity present in the feature embeddings generated by GANs is in contrast to the geometric requirements of grasp pose detection. On the other hand, the generator-domain classifier structure widely used in adversarial-based approaches is not suitable for our task because the grasp generator network in the main task and the domain classifier in the auxiliary task have disparities in training difficulties. The domain classifier tends to converge much faster than the grasp generator, resulting in a lack of gradient flow from the domain classifier to the feature extractor. Consequently, feature alignment across domains cannot be achieved.

To this end, in this paper we propose a novel Grasp Pose Domain Adaptation Network (GPDAN) for 6-DoF grasp pose detection. The point clouds from the source and target domains are first mixed into a single cloud where the geometric features from two domains are fused. Then, for each point in the mixed cloud, a domain classifier is applied to classify which domain the point comes from, to iteratively align the geometric features in these two domains through self-supervised adversarial training. The proposed GPDAN has the potential to reduce the need for expensive data collection and labeling in real world, which is of great value in robotic applications. The contributions of this paper are summarized as follows:

- A novel Grasp Pose Domain Adaptation Network (GPDAN) for improving the performance of 6-DoF grasp pose detectors trained in simulation and deployed in real-world environments.
- A point cloud mixture and feature fusing strategy along with the point-wise domain classifier to promote the feature alignment across domains through adversarial training.
- Experimental results show that our proposed method is able to significantly improve the performance of the grasp detectors when deployed in real world.

II. Related Work

A. Grasp Pose Detection

Grasp pose detection using a parallel gripper has been extensively studied for decades. Early works, such as DexNet [18], introduced deep learning to this field. One line of this research focuses on 3-DoF grasp pose detection [19]–[21] which predicts the 2D grasp position and the rotation angle on the image plane, where the grasps are perpendicular to it. However, due to the constraint of degrees of freedom, a large proportion of feasible grasp poses is neglected, which restricts the flexibility of grasp execution. The other line of work is 6-DoF grasping [3], [22], [23], which predicts the grasp position and orientation in 3D space. 6-DoF grasping offers more flexible and comprehensive grasps that can be executed in real world. However, the enormous $SE(3)$ search space poses significant challenges for 6-DoF grasp detection.

Most state-of-the-art grasp pose detection methods are trained in a supervised manner which necessitates a large quantity of labeled training data. Therefore, simulation data is often preferred due to its convenience and low cost of collection. However, due to the domain gap, model trained with simulation data may experience significant degradation in performance when applied to the real world. To this end, this paper proposes a novel domain adaptation method for grasp pose detection to reduce the negative impacts of the domain gap on model performance.

B. Self-Supervised Domain Adaptation

Sim-to-real DA aims to improve the performance of the model trained with simulation and deployed in the real world, which has been studied in various applications such as object detection [19], [24] and pose estimation [25]. Discrepancy-based methods aim to minimize the distribution differences between features in the source and target domains. The distribution differences are usually measured using distribution loss functions such as KL divergence [24] and Maximum Mean Discrepancy (MMD) [19]. However, these methods overlook the differences in feature patterns across domains, which are crucial in grasp pose detection. Reconstruction-based methods use GANs to reconstruct the data input in a self-supervised manner. While reconstruction-based methods have shown success in tasks such as segmentation [11], classification [26], and image translation [27], these methods may not be suitable for grasp pose detection due to the geometry-sensitive nature of the task. Grasp pose detection requires to capture fine geometric details, which GAN-based reconstructions may not adequately provide. Adversarial-based methods employ domain classifiers to encourage the feature extractors to extract similar features from both source and target domains [17], [28], [29]. However, the adversarial tasks may lead to totally opposite update directions to the main task of the network, resulting in oscillation or even divergence in training. In this paper, we propose a point cloud mixture and feature fusing strategy along with the point-wise domain classifier to promote the feature alignment across domains through adversarial training. With the smoother gradients from the feature fusion module in the two domains, the challenges of adversarial training such as gradient vanishing and mode collapse can be avoided.

C. Domain Adaptation in Robotic Grasping

DA methods have also been extensively studied in grasp detection. Previous work on DA for robotic grasping has mostly focused on planar grasp detection, which is much easier than 6-DoF grasp detection. Bousmalis et al. [7] and Jing et al. [30] generated synthetic images or feature maps from one domain to the other to align the two domains so that a unified grasp pose detector can anticipate the training process. Wang et al. [19] extracted scene features from RGB-D images in both source and target domains for grasp detection and aligned them using the MMD loss with a unit Gaussian distribution. Zhu et al. [31] proposed a confidence-driven mean-teacher network that utilizes a teacher network trained on source data in a

supervised manner to provide pseudo-labels for data in the target domain, which are then used to fine-tune the student network. To the best of our knowledge, few work has been focused on domain adaptation for 6-DoF grasp pose detection. In this paper, we combine domain adaptation for point cloud and 6-DoF grasp pose detection in a unified framework where the smooth training of both tasks can be ensured.

III. METHOD

This paper addresses the challenge of improving the performance of 6-DoF grasp pose detection in the real-world using only labeled grasps from simulations. Specifically, given labeled point cloud $(\mathbb{P}^s, \mathbb{G}^s)$ from simulation and unlabeled point cloud $\mathbb{P}^t$ from the real world, where $\mathbb{P}^s$ and $\mathbb{P}^t$ are the point clouds from the source and target domains and $\mathbb{G}^s$ is the ground truth grasp labels from the source domain, the objective is to learn feature representations that are effective for 6-DoF grasping in both domains. In this paper, we propose a novel method called the Grasp Pose Domain Adaptation Network (GPDAN) to learn domain invariant features across domains.

A. Point Cloud Scaling

The dimensions of the objects in the scenes, as well as the scales of the point clouds may vary greatly across different domains. Thus, the grasp pose detector will produce ambiguous grasp predictions even for similar geometric features across domains. Here, we propose a point cloud scaling strategy to align the scales of the point clouds from different domains, which helps improve the generalization of the grasp prediction.

Given a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$ which consists of N points under the camera coordinate, a scale factor α is first uniformly sampled from the range $[\alpha_{min}, \alpha_{max}]$. In order not to change the relative poses between points in the point cloud and the camera coordinate, the scaling procedure is carried out in the coordinate system of the point cloud.

Suppose the center of the point cloud $\mathbf{P}$ is $\mathbf{x}_c$, for each point $\mathbf{p} \in \mathbf{P}$, the corresponding point $\mathbf{p}'$ in the scaled point cloud $\mathbf{P}'$ is obtained as:

$$\mathbf{p}' = \alpha\mathbf{p} + (1-\alpha)\,\mathbf{x}_c. \tag{1}$$

Along with the scaling of the point clouds, the ground truth grasp labels should also be scaled accordingly. For each grasp pose $g_{gt} \in SE(3)$ with finger width w, the orientation in the scaled grasp pose g'_{gt} is not changed, while the center coordinate of the parallel gripper $\mathbf{x}'_g$ is calculated using eq. (1) with the new finger width $w' = \alpha w$.

B. Domain Invariant Feature Learning

The architecture of GPDAN is shown in Fig. 2. Our network consists of three blocks: feature extraction, grasp pose detection for the main task and self-supervised domain classification for the auxiliary task. The objective is to learn domain invariant features which can be generalized across simulation and real-world environments.

The main task of our proposed GPDAN is to detect feasible 6-DoF grasp poses from a partially observed point cloud in cluttered scenes. The main task is trained in a supervised manner. Given a point cloud $\mathbf{P}^s \in \mathbb{P}^s$ with its ground truth grasp labels $\mathbf{G}^s_{gt} \in \mathbb{G}^s$, the feature extractor Ψ_{fe} first extracts the local geometric features of the point cloud, then a grasp pose detector Ψ_{gp} takes these features as inputs and outputs the predicted grasp poses $\mathbf{G}_{pred}$ in the scene, i.e., $\mathbf{G}_{pred} = \Psi_{gp}\left(\Psi_{fe}\left(\mathbf{P}^s\right)\right)$.

The main task of GPDAN is trained with the grasp loss $L_g\left(\mathbf{G}_{pred}, \mathbf{G}^s_{gt}\right)$, which back propagation is performed to obtain feasible grasp poses for the input scenes. Here, the discrepancy between grasp poses is measured as the average closest point distance (ADD-S) between the corresponding five control points as in [32]. There are many alternatives for the feature extractor Ψ_{fe} (e.g. [33]–[35]) and the grasp pose detector Ψ_{gp} (e.g. [2], [32], [36]). We use PointNet++ network [37] as the feature extractor and VGPN [32] as the grasp pose detector in this paper. It should be noted that the feature extractor is shared across domains. Moreover, it is shown in experiments that the proposed GPDAN can be applied along with various choices of grasp pose detectors.

An auxiliary self-supervised domain classification task is additionally designed in GPDAN. Taken randomly and independently selected point cloud $\mathbf{P}^s$ and $\mathbf{P}^t$ as inputs, the shared feature extractor Ψ_{fe} samples a subset of points $\mathbf{X}^s, \mathbf{X}^t \in \mathbb{R}^{M\times 3}$ consisting of M points from each point cloud and extracts their local features $\mathbf{F}^s, \mathbf{F}^t \in \mathbb{R}^{M\times C}$, where C is the channel number. The features are then fed into the self-supervised domain classification module, in which the two point clouds are mixed together. For each point in the mixed cloud, the domain classifier Ψ_{cls} is trained to classify which domain the point comes from, i.e., to differentiate the extracted features in these two domains.

During the training process, a Gradient Reversal Layer (GRL) [14] is applied to reverse the sign of the gradients from the domain classification module. Through GRL layer, the weight parameters in Ψ_{fe} are instead adjusted to maximize the domain classification loss to learn domain invariant features which are difficult to differentiate between the source domain and the target domain, through which the geometric features in these two domains are aligned.

In the process of point cloud mixture, two point clouds $\mathbf{X}^s, \mathbf{X}^t$ along with the features $\mathbf{F}^s, \mathbf{F}^t$ are stacked into a single mixed point cloud $\mathbf{P}^m = (\mathbf{X}^m, \mathbf{F}^m)$ with $\mathbf{X}^m \in \mathbb{R}^{2M\times 3}$ and $\mathbf{F}^m \in \mathbb{R}^{2M\times C}$, as shown in Fig. 3:

$$\mathbf{X}^m = \mathbf{X}^s \uplus \mathbf{X}^t, \mathbf{F}^m = \mathrm{BN}^s\left(\mathbf{F}^s\right) \uplus \mathrm{BN}^t\left(\mathbf{F}^t\right), \tag{2}$$

where BN^s and BN^t are batch normalization modules for the source and target domains, respectively. The binary domain label $Y^* \in \{0,1\}^{2M}$ is also generated to indicate the source or target domain of each point in the mixed cloud. Through point cloud mixture, the domain classifier is able to exploit point clouds and geometric features from both domains simultaneously.

However, there may still be significant differences in the geometric features between different domains, which the domain classifier is able to classify the points from source/target domain easily. In this case, the domain classification loss L_{cls} will soon approach to zero at the beginning of the training

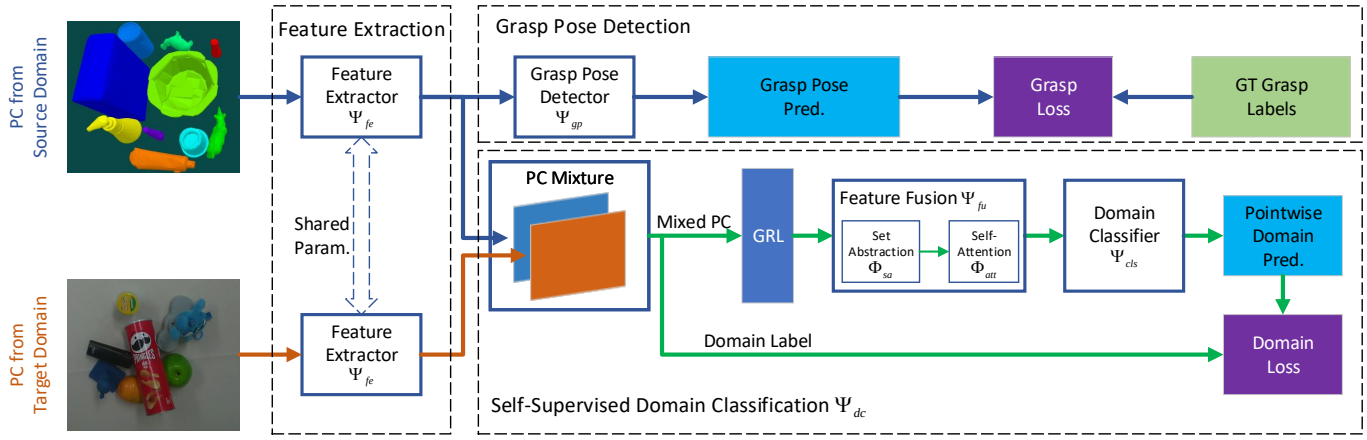

Fig. 2. The architecture of our proposed GPDAN. Point clouds from both source and target domains are first input to a shared feature extractor to obtain the feature vectors. The features from source domain are then fed into the grasp pose detector to train the main task in a supervised manner. In the self-supervised domain classification module, the features from both domains are mixed and fused to perform point-wise domain classification as the auxiliary task, where feature alignment across different domains is achieved through a gradient reversal layer (GRL).

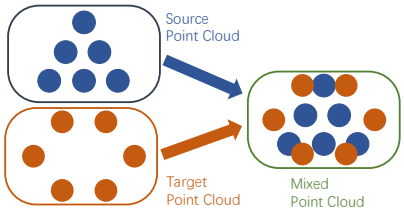

Fig. 3. The process of point cloud mixture. Point clouds from the source and target domain are stacked together at their original positions. Their corresponding feature vectors are also stacked together after batch normalization.

process. As a result, no gradient can be obtained from the domain classifier to the feature extractor. Consequently, the features from the source and the target domains are not aligned which the role of the self-supervised domain classification is negligible.

To this end, a feature fusion module Ψ_{fu} is proposed to further fuse features in the mixed cloud. With this feature fusion module, the feature of each 3D point is able to fuse features from other points that comes from two different domains. Therefore, the discrepancy between features of two domains can be reduced while the gradients from the feature learning are maintained during back propagation.

In the feature fusion module, a set abstraction module [37] Φ_{sa} is used to fuse the local features from both domains in $\mathbf{P}^m$ with its spatially neighboring points, in order to prevent the domain classifier from converging too quickly in the early stages of training. Φ_{sa} first samples S points $\mathbf{P}^m_{smp}$ from $\mathbf{P}^m$ using the farthest point sampling. The features of these sampled points are then fused together. The corresponding domain labels Y^*_{smp} for $\mathbf{P}^m_{smp}$ are also generated using Y^* and the sample indices. Furthermore, a self-attention module [38] Φ_{att} is used to fuse similar features in $\mathbf{P}^m$. The self-attention module enables uniform optimization of similar geometric features across domains. By concatenating features obtained from Φ_{att} and Φ_{sa}, we can obtain the features from the feature fusion module Ψ_{fu}, i.e. $\Psi_{fu} = \Phi_{att} \circ \Phi_{sa}$. Through the feature fusion module, the discrepancy between features across domains can be effectively reduced.

The final part of the self-supervised domain classification module is the domain classifier Ψ_{cls} which is built by several fully connected layers. The domain classification is treated as a two-class classification task. The domain classification module takes the fused D-dimensional features $\mathbf{F}_{merge} \in \mathbb{R}^{S\times D}$ from Ψ_{fu} as input, and outputs the predicted domain labels $\hat{Y} \in \mathbb{R}^{S\times 2}$. A cross entropy classification loss L_{cls} is then obtained with the binary label Y^*_{smp} through which back propagation and feature alignment across domains can be achieved. The self-supervised domain classification branch is represented as:

$$\hat{Y} = \Psi_{dc}\left(\mathrm{GRL} \circ \mathrm{MIX}\left(\Psi_{fe}\left(\mathbf{P}^s\right), \Psi_{fe}\left(\mathbf{P}^t\right)\right)\right), \quad (3)$$

where $\Psi_{dc} = \Psi_{cls} \circ \Psi_{fu}$ is the self-supervised domain classification module and MIX is the process of the point cloud mixture.

C. Iterative Training

The main task and auxiliary task are trained iteratively to align geometric features in the source and target domains which are suitable for 6-DoF grasping. The training process is shown in Algorithm 1. It is noted that we use a point-wise domain loss in the domain classification module instead of predicting the domain class of the whole point cloud. The point-wise loss produces much smoother gradients compared with using the whole point cloud. In this way, the challenges of adversarial training such as gradient vanishing and mode collapse can be avoided.

IV. Experiments

We train our GPDAN network using the ACRONYM dataset [39] as the source domain and the GraspNet-1Billion dataset [36] as the target domain. The ACRONYM dataset contains over $10,000$ scenes collected in simulation while the GraspNet-1Billion dataset contains 190 real-world scenes.

Our GPDAN network is trained with 70 epochs on a single Nvidia RTX3060 GPU which takes about 35 hours to

Algorithm 1 Iterative Training for Grasp Feature Domain Adaptation

Require: Source point cloud with ground truth grasp labels $(\mathbb{P}^s, \mathbb{G}^s)$, Target point cloud $\mathbb{P}^t$, Network Ψ_{fe}, Ψ_{gp}, Ψ_{dc}

for $\mathbf{P}^s$ in $\mathbb{P}^s$, $\mathbf{G}^s_{gt}$ in $\mathbb{G}^s$, $\mathbf{P}^t$ in $\mathbb{P}^t$ **do**

1. For grasp pose detection:

$\mathbf{G}_{pred} = \Psi_{gp}\left(\Psi_{fe}\left(\mathbf{P}^s\right)\right)$

$l_g = L_g\left(\mathbf{G}_{pred}, \mathbf{G}^s_{gt}\right)$

$w_{gp} \leftarrow w_{gp} - \eta \nabla_{w_{gp}} l_g$, $w_{fe} \leftarrow w_{fe} - \eta \nabla_{w_{fe}} l_g$

2. For self-supervised domain classification:

$\mathbf{P}^m, Y^* = \mathrm{MIX}\left(\Psi_{fe}\left(\mathbf{P}^s\right), \Psi_{fe}\left(\mathbf{P}^t\right)\right)$

$\hat{Y}, idx = \Psi_{dc}\left(\mathbf{P}^m\right)$

$Y^*_{smp} = \mathrm{INDEX}\left(Y^*, idx\right)$

$l_{cls} = L_{cls}\left(\hat{Y}, Y^*_{smp}\right)$

$w_{dc} \leftarrow w_{dc} - \eta \nabla_{w_{dc}} l_{cls}$, $w_{fe} \leftarrow w_{fe} + \eta \nabla_{w_{fe}} l_{cls}$

end for

TABLE I
RESULT OF ABLATION STUDIES

	ACRONYM (source)		GraspNet-1Billion (target)	
	SR	Recall	SR	Recall
GPDAN	0.807	0.618	**0.699**	**0.327**
w/o adaptation	0.824	0.636	0.559	0.273
scene cls.	0.833	0.573	0.597	0.301
w/o fusion	0.820	0.602	0.568	0.280
w/o scaling	0.813	0.527	0.506	0.178

complete. The range of the point cloud scaling α is set to $[0.5, 1.0]$ for ACRONYM dataset and $[0.9, 1.5]$ for GraspNet-1Billion dataset. The performance of GPDAN for grasp pose detection is evaluated in both simulation and real-world experiments. The performance of grasp pose detection approaches without domain adaptation is also evaluated to indicate the effectiveness of GPDAN.

A. Simulation Experiments

The simulation experiments are conducted in PyBullet [40]. A Robotiq-140 gripper is placed on each predicted grasp pose. We select 100 scenes from ACRONYM dataset and 45 scenes from GraspNet-1Billion dataset as the testing set in the simulator. All the objects in the scenes are assumed to have a uniformly distributed density of $1000kg/m^3$ and surface friction coefficient of 1.0. The success rate (SR) of the top-scored 5000 grasp predictions in each experiment and the recall of the ground truth grasps are evaluated in experiments. A predicted grasp pose is considered successful if the gripper can successfully grasp the object and the object is still in the gripper after shaking it quickly with an amplitude of 45 degrees for 2 seconds. Similar to [2], [32], a ground truth grasp label is considered to be covered if any successful grasp prediction lies within $2cm$ from the position of the label.

1) Ablation Studies: Ablation studies are conducted by removing modules in GPDAN to investigate the effectiveness of each module. The remaining parts of the network are trained with the same training data and experimental configurations. The results of ablation studies are shown in Tab. I.

In Tab. I, the "w/o adaptation" corresponds to the performance of a grasp pose detector trained with the same scaled point cloud data as GPDAN without domain adaptation. We use VGPN [32] as the grasp pose detector in all experiments. It is observed that the domain adaptation in GPDAN significantly improves the success rate (14%) and recall (5%) of VGPN on the target domain. There is a slight decrease in performance (1.7% on SR) of GPDAN on the source domain compared with VGPN, which can be attributed to the inherent differences across domains in object shapes, noise patterns and etc.

The "scene cls." term corresponds to the performance by predicting the domain class of the individual point cloud from the source/target domain. That is, we extract the global features of the whole source/target point cloud and use the domain classifier to identify which domain the scene point cloud belongs to, without point cloud mixture and feature fusion. This strategy is widely used in domain adaptation [24], [41]. Since the domain classifier converges quickly at the beginning of the training, which makes the adversarial learning strategy do not contribute much to the feature alignment across domains. The performance on the target domain does not significantly improve compared to the result without domain adaptation.

The "w/o fusion" term corresponds to the performance of GPDAN without feature fusion. The point-wise domain classifier also converges quickly due to the large differences between features in two domains. Without feature fusion, the adversarial learning strategy is also negligible. The network achieves similar performance as the supervised training of VGPN using data from the source domain.

It is noted in Tab. I that without point cloud scaling, the point-wise domain classification can not work well either. The dimensions of the objects strongly influence the extracted geometric features. The domain adaptation is unable to bridge the gap between features across domains, which results in even worse performance on the target domain compared with the performance without adaptation.

2) Comparative Experiments: The results of comparative experiments are shown in Tab. II, where AC and GN are ACRONYM and GraspNet-1Billion datasets, respectively. "AC $\rightarrow$ GN" indicates the domain adaptation from ACRONYM dataset to GraspNet-1Billion dataset and vice versa. "CGN" stands for the Contact-GraspNet [2]. "l.b. S" stands for the lowest grasp score obtained from the top-scored 5000 grasp predictions, which lies between $[-1, 1]$ as defined in [32]. The higher the score, the more confident is the grasp prediction.

In all experiments, the point cloud scaling module is used to scale all the point clouds at the training phase. In addition to success rate and recall of the top-scored 5000 grasp predictions, the success rates of the highest 500 grasps are also evaluated in experiments. Since only the top-scored grasps are executed by the robot, SR (500) indicates if the highly-confident grasps predicted by the detector will succeed in grasping the objects.

It is observed in Tab. II that with domain adaptation from AC to GN, the SR of our GPDAN on the target domain is 14% higher than that of VGPN and 17.3% higher than that of Contact-GraspNet. The SR drops slightly from 82.4% to 80.7% on the source domain, which implies that GPDAN is able to improve the performance of grasp detection on the

TABLE II
RESULT OF COMPARATIVE EXPERIMENT

Method (Dataset)	ACRONYM				GraspNet-1Billion			
	SR	SR (500)	Recall	l.b. S	SR	SR (500)	Recall	l.b. S
GPDAN (AC → GN)	0.807	0.934	0.618	0.67	**0.699**	**0.946**	**0.327**	**0.51**
VGPN [32] (AC)	**0.824**	0.956	**0.636**	**0.71**	0.559	0.914	0.273	0.49
VGPN [32] (AC w/o scaling)	0.820	**0.961**	0.625	0.69	0.482	0.791	0.213	-0.41
CGN [2] (AC)	0.786	0.836	0.512	-0.61	0.526	0.758	0.279	-0.84
CGN [2]+DA (AC → GN)	0.753	0.802	0.607	-0.76	0.584	0.782	0.257	-0.62
VGPN [32]+UDF (AC → GN)	0.816	0.946	0.623	0.70	0.687	0.886	0.335	0.44
VGPN [32] (GN)	0.501	0.806	0.188	**0.59**	**0.775**	**0.832**	**0.438**	**0.72**
GPDAN (GN → AC)	**0.607**	**0.835**	**0.227**	0.52	0.753	0.801	0.421	0.70

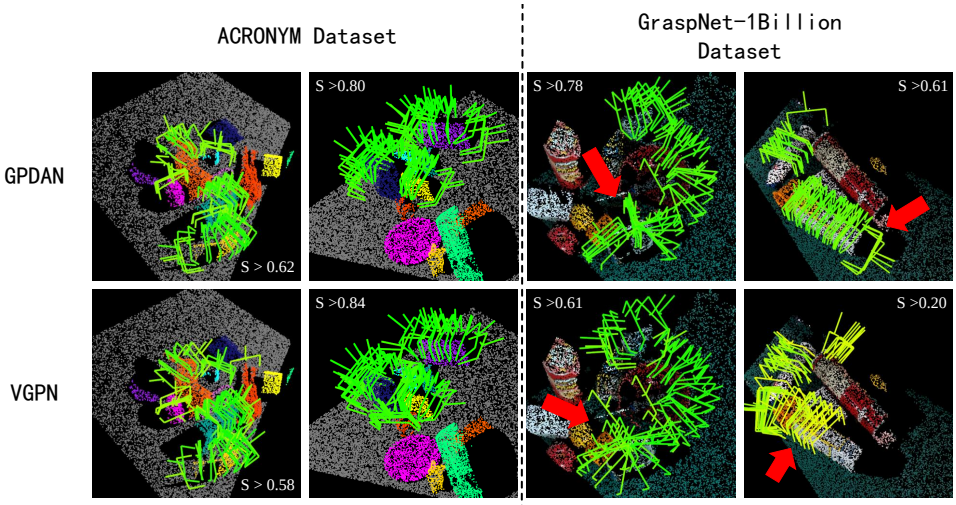

Fig. 4. Examples of grasp pose predictions of GPDAN (the first row) and VGPN (the second row). Each column shows the prediction results of the same scene. The first two columns are the results in the source domain (ACRONYM) while the last two columns are results in the target domain (GraspNet-1Billion). Only the predicted grasps with scores which are higher than a threshold are shown. It is observed that GPDAN and VGPN achieve comparable performance on the source domain while GPDAN is able to generate more feasible and comprehensive grasp poses on the target domain.

target domain without significantly affecting the performance on the source domain. Moreover, it is observed that the lowest grasp scores obtained on the target domain is larger than that of VGPN, which indicates that the network is more confident about the grasp predictions since the grasp detector is better adapted to the features in the target domain. Examples of grasp pose predictions of GPDAN and VGPN are shown in Fig. 4. It can be observed that GPDAN and VGPN achieve comparable performance on the source domain while GPDAN is able to generate more feasible and comprehensive grasp poses on the target domain.

To evaluate the generalization ability of the proposed adversarial training strategy, we replace the VGPN with CGN. Other parts of the network is trained with the same configurations as GPDAN. The results are indicated by "CGN + DA (AC → GN)" in Tab. II. It can be seen that the success rate on the target domain is also higher than the original CGN ("CGN (AC)"), which shows that our proposed adversarial training strategy can be generalized to different grasp pose detectors.

Furthermore, to evaluate the effectiveness of our proposed point cloud mixture strategy and feature fusion, we replace the self-supervised domain classification module in GPDAN with a geometry-aware unsigned distance field (UDF) prediction strategy similar to [42], i.e. the Ψ_{dc} in GPDAN is replaced with a UDF prediction module in this network. Due to the ambiguity of the global features of the entire point cloud in a cluttered scene, the performance of VGPN+UDF network on the target domain is inferior to that of GPDAN. Moreover, the UDF prediction strategy requires significantly more computing resources than GPDAN since it has to uniformly sample a set of points in the whole workspace and calculates the distances to the closest surface on the object.

In addition, we swap the source and target domains and retrain our GPDAN, i.e. the domain adaptation is from the GN dataset to the AC dataset. The results are indicated by "GPDAN (GN → AC) in Tab. II. The 11% increase in the success rate on the AC dataset compared to VGPN (GN) demonstrates that our method can be applied to a variety of dataset sources and has well generalization ability.

B. Robot Experiments

Physical experiments on a real robot are performed to show the effectiveness of our method in real-world environments. The robot platform we use in experiments consists of an AUBO-i5 6-DoF robotic arm, a DH-Robotics AG-95 two-finger electrical gripper attached to the end of the arm as the end effector, and an Intel RealSense L515 RGB-D camera placed on the top of the workspace, as shown in Fig. 5. With a point cloud containing 20000 points, each forward pass takes about 0.2 seconds on a single Nvidia RTX3060 GPU.

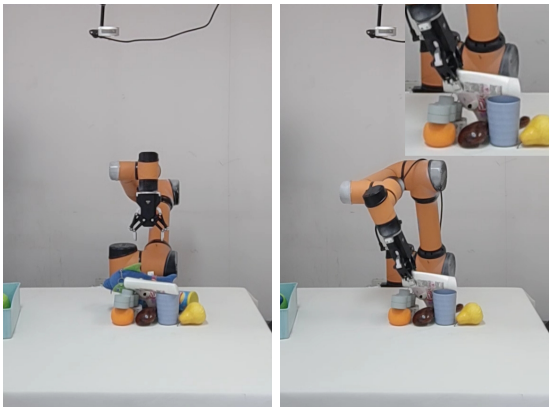

Fig. 5. Experimental settings of the real-world experiments. (Left) Multiple objects are stacked in a pile on the table. An RGB-D camera is placed above the workspace to get the observations of the scene. (Right) The robot picks an object and places it to the drop position.

TABLE III
RESULTS OF ROBOTIC EXPERIMENTS

	SR	F.A.	#Attempts
GPDAN	**0.847**	**0.745**	59
VGPN [32]	0.774	0.661	62
CGN [2]	0.719	0.561	57
Method in [36]	0.707	0.517	58

Similar to [32], the point clouds of the scenes consist of 5-10 stacked objects, which are then fed into GPDAN and grasp pose predictions are generated. Different from [32], we do not use object segmentation in GPDAN and VGPN in our experiments. For each attempt, the grasp prediction with the highest score is performed to grasp and place the object to the drop position. At most 2 grasps are executed for each object, after which the success rate of the pick-and-place process is recorded.

The result of real-world experiments is shown in Tab. III. SR is the success rate. FA is the first attempt success rate on each object. Our GPDAN outperforms other methods by over 7.3% on success rate and 8.4% on the success rate of the first attempt for each object. The result shows that the performance of GPDAN is much better compared with the performance of grasp pose detectors trained in simulation and directly deployed in real world environments. Compared with VGPN, the failures caused by the slip of the objects in the process of grasping are reduced.

V. DISCUSSIONS

To explore the differences between features extracted by the feature extractor of GPDAN and VGPN, we apply Grad-CAM++ algorithm [43] on point clouds to show the influences of the geometric features of different locations on the scores of grasp predictions, i.e. the grasp capability of features on different locations. For any point at position $\mathbf{x}_i$ in the given point cloud, with its corresponding feature vector $\mathbf{f}_i \in \mathbf{F}$ output by Ψ_{fe}, the grasp capability score r_i at this point is

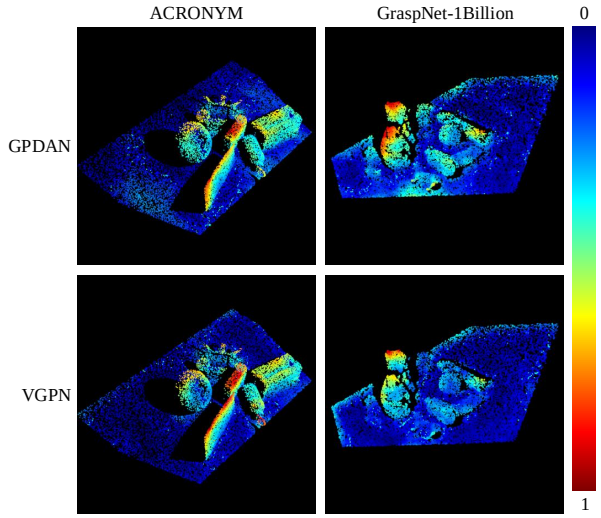

Fig. 6. The visualization of the grasp capability scores of features in different domains from GPDAN and VGPN.

calculated as follows:

$$\begin{aligned} \alpha_i &= \frac{\left(\nabla_{\mathbf{f}_i} \bar{s_g}\right)^2}{2\left(\nabla_{\mathbf{f}_i} \bar{s_g}\right)^2 + \sum_j \mathbf{f}_j \left(\nabla_{\mathbf{f}_j} \bar{s_g}\right)^3}, \\ \mathbf{w} &= \sum_i \alpha_i \cdot \mathrm{ReLU}\left(\nabla_{\mathbf{f}_i} \bar{s_g}\right), \\ r_i &= \mathbf{w}^T \mathbf{f}_i, \end{aligned} \tag{4}$$

where $\bar{s_g}$ is the mean score of predicted grasps, $\nabla_{\mathbf{f}_i} \bar{s_g} \in \mathbb{R}^{M \times C}$ is the gradient of $\bar{s_g}$ on $\mathbf{f}_i$. The results on the source and target domains are shown in Fig. 6.

It can be observed from Fig. 6 that GPDAN produces a wider range of regions with high grasp capability scores compared with VGPN, indicating that GPDAN takes into account of more information around the target objects when generating grasps. By incorporating information from a wider neighboring region, GPDAN is able to gather scene information such as the noise patterns and differences in object arrangements for alignment of the features across domains for the main grasping task. In this way, the grasp detector is able to achieve similar performances on both domains.

VI. CONCLUSION

In this paper, a novel domain adaptation method GPDAN is proposed for 6-DoF grasp pose detection. By leveraging the auxiliary task of point-wise self-supervised domain adaptation with point cloud mixture strategy in an adversarial training manner, our approach has significantly improved the performance of grasp detection on the target domain. The point cloud mixture and feature fusion strategy enable the alignment of similar geometric features across domains. Experimental results demonstrate that our proposed method outperforms state-of-the-art grasp pose detection approaches that are trained in a supervised manner as well as other domain adaptation methods, highlighting the effectiveness of our approach for improving the performance of 6-DoF grasp pose detectors trained in simulation and deployed in real-world environments.

REFERENCES

[1] A. Mousavian, C. Eppner, and D. Fox, "6-DoF GraspNet: Variational Grasp Generation for Object Manipulation," in *IEEE/CVF International Conference on Computer Vision*. IEEE, 2019, pp. 2901–2910.

[2] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.

[3] W. Wei, Y. Luo, F. Li, and et al., "GPR: Grasp Pose Refinement Network for Cluttered Scenes," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4295–4302.

[4] T. Hodaň, F. Michel, E. Brachmann, and et al., "BOP: Benchmark for 6D Object Pose Estimation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 19–35.

[5] Y. Chen, Z. Wang, L. Zou, and et al., "Quasi-Balanced Self-Training on Noise-Aware Synthesis of Object Point Clouds for Closing Domain Gap," in *European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 728–745.

[6] S. Peng, X. Xi, C. Wang, and et al., "Point-Based Multilevel Domain Adaptation for Point Cloud Segmentation," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, 2022.

[7] K. Bousmalis, A. Irpan, P. Wohlhart, and et al., "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[8] J. Matas, S. James, and A. J. Davison, "Sim-to-Real Reinforcement Learning for Deformable Object Manipulation," in *2nd Conference on Robot Learning*, vol. 87. PMLR, 2018, pp. 734–743.

[9] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey," in *IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[10] M. Wang and W. Deng, "Deep Visual Domain Adaptation: A Survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[11] Y. Chen, M. Li, and C. Yang, "Unknown Object Segmentation through Domain Adaptation," in *4th International Conference on Intelligent Autonomous Systems (ICoIAS)*. IEEE, 2021, pp. 72–77.

[12] I. Achituve, H. Maron, and G. Chechik, "Self-Supervised Learning for Domain Adaptation on Point Clouds," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2021, pp. 123–133.

[13] M.-Y. Liu and O. Tuzel, "Coupled Generative Adversarial Networks," in *30th Conference on Neural Information Processing Systems (NIPS)*, vol. 29. NIPS, 2016, pp. 469–477.

[14] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[15] A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis with Auxiliary Classifier GANs," in *34th International Conference on Machine Learning*, vol. 70. Journal Machine Learning Research, 2017, pp. 2642–2651.

[16] X. Yan, J. Hsu, M. Khansari, and et al., "Learning 6-DoF Grasping Interaction via Deep Geometry-Aware 3D Representations," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3766–3773.

[17] Z. He, B. Yang, C. Chen, and et al., "CLDA: An Adversarial Unsupervised Domain Adaptation Method with Classifier-Level Adaptation," *Multimedia Tools and Applications*, vol. 79, no. 45-46, pp. 33 973–33 991, 2020.

[18] J. Mahler, F. T. Pokorny, B. Hou, and et al., "Dex-net 1.0: A Cloud-Based Network of 3D Objects for Robust Grasp Planning Using A Multi-Armed Bandit Model with Correlated Rewards," in *IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016.

[19] H. Wang, X. Chen, and X. Lan, "An Exploration of Domain Adaptation Applying to Grasp Detection Algorithm," in *Chinese Automation Congress (CAC)*. IEEE, 2020, pp. 5332–5337.

[20] T.-T. Do, A. Nguyen, and I. Reid, "AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[21] P. Ardon, E. Pairet, R. P. A. Petrick, and et al., "Learning Grasp Affordance Reasoning Through Semantic Relations," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4571–4578, 2019.

[22] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp Pose Detection in Point Clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[23] C. Wang, H.-S. Fang, M. Gou, and et al., "Graspness Discovery in Clutters for Fast and Accurate Grasp Detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 15 964–15 973.

[24] A. K. Tanwani, "DIRL: Domain-Invariant Representation Learning for Sim-to-Real Transfer," in *4th Conference on Robot Learning (CoRL)*, vol. 155. PMLR, 2020, pp. 1558–1571.

[25] G. Shi, Y. Zhu, J. Tremblay, and et al., "Fast Uncertainty Quantification for Deep Object Pose Estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5200–5207.

[26] S. Liu, X. Luo, K. Fu, and et al., "A Learnable Self-Supervised Task for Unsupervised Domain Adaptation on Point Clouds," *Frontiers of Computer Science*, vol. 16, no. 6, 2022.

[27] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *16th IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2242–2251.

[28] Y. Ganin, E. Ustinova, H. Ajakan, and et al., "Domain-Adversarial Training of Neural Networks," *Journal of Machine Learning Research*, vol. 17, 2016.

[29] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial Discriminative Domain Adaptation," in *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2962–2971.

[30] X. Jing, K. Qian, X. Xu, and et al., "Domain Adversarial Transfer for Cross-Domain and Task-Constrained Grasp Pose Detection," *Robotics and Autonomous Systems*, vol. 145, 2021.

[31] H. Zhu, Y. Li, F. Bai, and et al., "Grasping Detection Network with Uncertainty Estimation for Confidence-Driven Semi-Supervised Domain Adaptation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9608–9613.

[32] L. Zheng, Y. Cai, T. Lu, and S. Wang, "VGPN: 6-DoF Grasp Pose Detection Network Based on Hough Voting," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7460–7467.

[33] Y. Wang, Y. Sun, Z. Liu, and et al., "Dynamic Graph CNN for Learning on Point Clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, 2019.

[34] Z. Wu, S. Song, A. Khosla, and et al., "3D ShapeNets: A Deep Representation for Volumetric Shapes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 1912–1920.

[35] J. Masci, D. oscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic Convolutional Neural Networks on Riemannian Manifolds," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 832–840.

[36] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 11 441–11 450.

[37] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++ : Deep Hierarchical Feature Learning on Point Sets in A Metric Space," in *31st Annual Conference on Neural Information Processing Systems (NIPS)*, vol. 5. NIPS, 2017.

[38] A. Vaswani, N. Shazeer, N. Parmar, and et al., "Attention is All You Need," in *31st Annual Conference on Neural Information Processing Systems (NIPS)*, vol. 30. NIPS, 2017.

[39] C. Eppner, A. Mousavian, and D. Fox, "ACRONYM: A Large-Scale Grasp Dataset Based on Simulation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6222–6227.

[40] C. Erwin and B. Yunfei, "PyBullet, A Python Module for Physics Simulation for Games, Robotics and Machine Learning," 2016. [Online]. Available: http://pybullet.org

[41] W. Zhang, W. Li, and D. Xu, "SRDAN: Scale-Aware and Range-Aware Domain Adaptation Network for Cross-dataset 3D Object Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 6765–6775.

[42] Y. Shen, Y. Yang, M. Yan, and et al., "Domain Adaptation on Point Clouds via Geometry-Aware Implicits," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 7213–7222.

[43] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 839–847.