

Sim-to-Real Grasp Detection with Global-to-Local RGB-D Adaptation

Haoxiang Ma*, Ran Qin*, Modi Shi, Boyang Gao and Di Huang[†]

Abstract—This paper focuses on the sim-to-real issue of RGB-D grasp detection and formulates it as a domain adaptation problem. In this case, we present a global-to-local method to address hybrid domain gaps in RGB and depth data and insufficient multi-modal feature alignment. First, a self-supervised rotation pre-training strategy is adopted to deliver robust initialization for RGB and depth networks. We then propose a global-to-local alignment pipeline with individual global domain classifiers for scene features of RGB and depth images as well as a local one specifically working for grasp features in the two modalities. In particular, we propose a grasp prototype adaptation module, which aims to facilitate fine-grained local feature alignment by dynamically updating and matching the grasp prototypes from the simulation and real-world scenarios throughout the training process. Due to such designs, the proposed method substantially reduces the domain shift and thus leads to consistent performance improvements. Extensive experiments are conducted on the GraspNet-Planar benchmark and physical environment, and superior results are achieved which demonstrate the effectiveness of our method. Code is available at <https://github.com/mahaoliang822/GL-MSDA>.

I. INTRODUCTION

Given its generalizability to new scenes and objects, learning-based grasp detection is being increasingly applied to complex robot manipulation tasks. In this case, a large amount of annotated data is generally required for model training. Unfortunately, it is quite difficult to obtain grasp labels from real-world, which often consumes hundreds of hours in executing grasp candidates to ensure sufficient annotations [1], [2]. To deal with this, some attempts [3]–[5] employ simulators to construct virtual scenes and generate grasp labels, effectively reducing the cost.

Although simulators conveniently enrich the scale and diversity of data, a grasp detection model directly trained with simulated data suffers performance degradation in real-world scenarios. This is due to the discrepancy of data distributions, referred to as the sim-to-real problem. Some studies [6]–[8] introduce Domain Randomization (DR) to alleviate such a performance gap. By randomly varying parameters in the simulator, *e.g.* light direction, object pose and camera perspective, the distribution of simulated data is expected to cover that in the real-world. However, as the

This work is partly supported by the National Natural Science Foundation of China (62022011), the Research Program of State Key Laboratory of Software Development Environment (SKLSDE-2023ZX-14), and the Fundamental Research Funds for the Central Universities.

Haoxiang Ma, Ran Qin, Modi Shi and Di Huang are with the State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing, China.

Boyang Gao is with the Geometry Robotics and the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China.

* Equal contribution.

[†]Corresponding author. (email: dhuang@buaa.edu.cn).

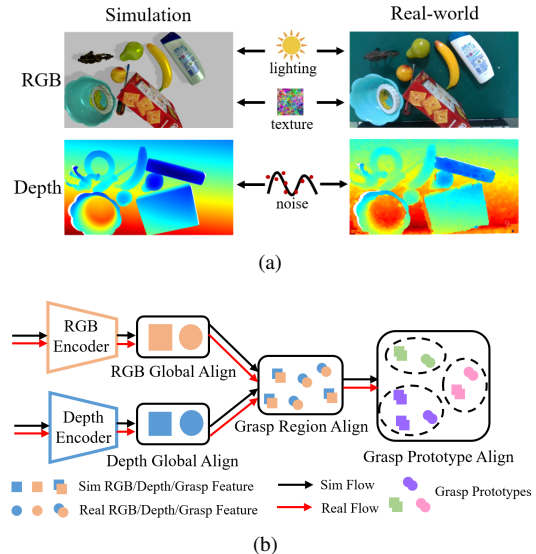


Fig. 1. (a) The domain gap occurs in sim-to-real grasp detection and (b) the proposed GL-MSDA pipeline.

true distribution is unknown, it is uncertain that DR can perform data augmentation tailored to the target domain and many uncorrelated data are simultaneously generated during this process, making these solutions less efficient. With the advancements of transfer learning, Domain Adaptation (DA) methods have been investigated in grasp detection [9]–[12]. By making use of unlabeled real-world data, they encourage the model to learn features consistent with both simulated and real-world environments. For example, Bousmalis *et al.* [9] exploit Generative Adversarial Network (GAN) to transfer RGB images rendered by simulators to the style of reality and Fang *et al.* [10] design a domain classifier with an adversarial loss to align features between source and target domains.

Despite that promising results are reported in sim-to-real grasp detection, existing DA methods still have two limitations. First, in these methods alignment is only performed on the RGB modality [9], [10], [13] while RGB-D sensors have been widely adopted in recent grasp detection systems [14], [15]. As shown in Fig. 1 (a), gaps in RGB and depth images between simulated and real-world are distinct, with RGB discrepancies mainly due to lighting and texture and depth disparities arising from noise in depth camera. To the best of our knowledge, alignment on RGB-D data is not well handled, making current DA solutions to multi-modal sim-to-real adaptation problematic. Second, previous studies [9], [10], [13] typically perform global alignment at the image-level, where local features significantly influence the detection performance [14], [16]–[18]. Considering that the

distribution of different local shapes varies, directly aligning such features inadvertently results in pulling in grasping features that correspond to entirely dissimilar local shapes, thereby incurring ambiguity.

To address the issues mentioned above, we propose a novel sim-to-real grasp detection framework, namely Global-to-Local Multi-modal Self-supervised Domain Adaptation (GL-MSDA). Fig. 1 (b) shows an overview. Specifically, GL-MSDA first introduces self-supervised rotation pre-training to enable two independent networks to learn domain invariant features from simulated and real-world RGB and depth images and then applies global domain classifiers [19] to separately align the features of simulation and real-world data of each modality. Besides global alignment, GL-MSDA incorporates a local domain classifier to align features of grasp proposals and employs consistency regularization to enforce the consistency between the results of local and global domain classifiers. Furthermore, to align local geometric features with similar shape distributions from simulation and real-world, inspired by [20], we construct local grasp prototypes by partitioning rotation angles. During training, these prototypes are continuously updated while the prototype distance between the two domains is minimized. Thanks to such designs, GL-MSDA effectively reduces sim-to-real domain shift and delivers decent performance gains. The proposed method is experimentally evaluated on the GraspNet-Planar benchmark and in physical environment with competitive results reported. Additionally, to facilitate future research, we generate a large-scale simulated grasp detection dataset based on GraspNet [18] and GraspNet-Planar [14] using the PyBullet simulator [21] and DR techniques.

II. RELATED WORK

A. Sim-to-Real Transfer

The objective of sim-to-real transfer is to narrow the performance gap between simulated and real environments. Over the past decade, it has been extensively studied across various fields, such as robot control [22], robot manipulation [7] and autonomous driving [23]. Among the methods in the literature, DR and DA have emerged as two predominant alternatives.

The DR methods [6]–[8], [24], [25] follow the assumption if sufficient diversity is presented in the simulated environment, the generalization ability of the model can be guaranteed for good performance in real-world scenarios. Besides, [26], [27] add Gaussian noise and salt-and-pepper noise to the samples generated by the simulator, replicating the real ones. Given labeled data from the source domain and unlabeled data from the target domain, the DA methods endeavor to map the features of the two domains into a shared domain-agnostic feature space, aiming to reduce the disparity between their feature distributions. Several methods [28]–[33] employ the adversarial training strategy to facilitate detectors in extracting domain-invariant features. Another way to align features is to translate target data (*e.g.* images) into source-like ones using style transfer methods [34], [35]. In addition to enhancing the similarity between simulated

and real data, some studies [13], [23] map source and target domains a predefined intermediate domain.

B. Sim-to-real Transfer for Grasp Detection

Regarding grasp detection, collecting real-world data with annotated grasps is typically expensive and time-consuming. Many studies [9], [36] conduct robotic arm grasping experiments using physical simulators like PyBullet [21] and Sapien [37], generating RGB-D images along with corresponding grasping annotations. To bridge the distribution gap between simulated and real data, the DR methods [6], [7] introduce random variations in visual parameters within the simulation environment, thereby increasing the diversity of training set to enhance model generalization. Others [36], [38]–[40] employ some high-quality simulators equipped with powerful rendering capabilities to mimic realistic grasp scenarios. On the other side, some methods apply DA to align features derived from the simulated and real-world domain. Fang *et al.* [10] employ a domain classifier and an adversarial training strategy to compel the network to learn domain-agnostic features. Bousmalis *et al.* [9] transfer the style of source domain images to match that of the target domain using GraspGAN. Some studies [11], [13] also make use of artificially defined standard simulation environments or the mean teacher network during training. Although the DA methods achieve a great success, they do not take into consideration the disparity between depth and RGB modalities in the grasping task. Furthermore, they only align the source and target domains but disregards the distribution discrepancy of the features of local grasp regions. In contrast, this paper presents a solution to grasp detection which particularly addresses the sim-to-real problem in a multi-modal RGB-D mode with both global and local feature alignment.

III. METHODOLOGY

In this section, we introduce the sim-to-real RGB-D grasp detection method proposed in this paper. Specifically, in Section III-A, we provide a brief overview of the GL-MSDA framework; in Section III-B, we describe the self-supervised rotation pre-training strategy; in Section III-C, we present the global-local adaptation module based on multi-modal domain classifiers and consistency regularization; and in Section III-D, we finally describe the local grasp prototype adaptation module.

A. Overall Framework

With labeled simulated RGB-D input, GL-MSDA takes the recently proposed RGB-D planar grasp detection network with depth prediction [14] as the baseline model and extends it by incorporating another stream for unlabeled real-world RGB-D data. The pipeline of GL-MSDA is depicted in Fig. 2.

The proposed method comprises two stage: *i.e.* Self-supervised Rotation Pre-training and Global-Local Multi-modal Adaptation. In the pre-training stage, RGB and depth images from both the simulation and real-world are mixed.

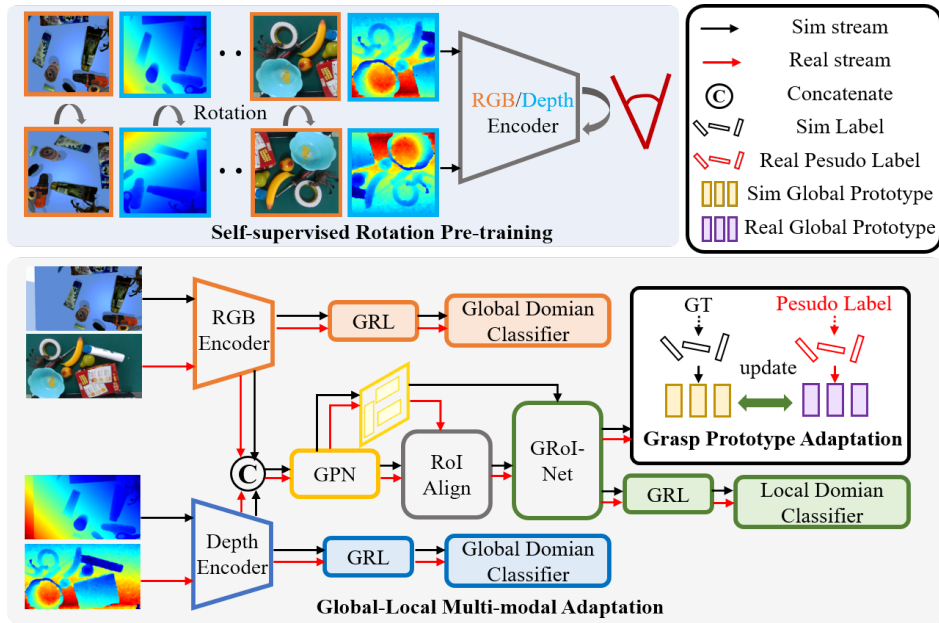


Fig. 2. Overview of the proposed GL-MSDA method.

Inspired by [41], we rotate the images and employ relative rotation angle prediction as the pretext task for self-supervised training. In the adaptation stage, we train the grasp detection network with labeled simulated data and unlabeled real-world data, initializing it with the pre-trained weights. We utilize the RGB and depth encoders to extract features from the corresponding modalities of the two domains. To fulfill image-level alignment, we introduce two global domain classifiers with Gradient Reversal Layers (GRL) [42]. The RGB and depth features are concatenated and fed into Grasp Proposal Network (GPN) to generate grasp proposals. For local grasp features, we employ the Grasp Region of Interest Network (GRoI-Net) to predict the final grasp parameters and introduce a local domain classifier to align local features from both domains. To achieve fine-grained local alignment, the Grasp Prototype Adaptation (GPA) module utilizes local grasp features to update grasp prototypes, ensuring that the same category of grasp prototypes in the simulated and real-world domains are aligned.

B. Self-supervised Rotation Pre-training

To acquire robust visual representations, some methods [41], [43] introduce image rotation as a straightforward yet effective pretext task. To narrow the gap of feature distributions of both the modalities between the simulated and real domains before training the grasp detection network, we separately pre-train the RGB and depth network by predicting relative image rotation angles. We illustrate the pre-train process using the RGB branch as an example, with the depth branch being identical except for its use of depth images as input.

As shown at the top of Fig. 2, given an RGB image I either from simulation or real-world and its randomly rotated counterpart, I' , the prediction target is the relative rotation

angle $A_{I \rightarrow I'}$. I' is obtained by the following transformation:

$$I' = \text{rot}90(I, k) \quad (1)$$

where $\text{rot}90$ represents a counterclockwise rotation of the image by $(90 \times k)^\circ$, $k \in \{0, 1, 2, 3\}$. To predict $A_{I \rightarrow I'}$, we employ the same RGB encoder to process both I and I' , resulting in features F_I and $F_{I'}$. These two features are adaptively pooled to a fixed size (*i.e.* 16×16) and then concatenated. Afterward, the concatenated feature is processed through the convolutional layer c and the fully connected layer f , ultimately yielding the relative rotation $A_{I \rightarrow I'}$. The equation is displayed below:

$$A_{I \rightarrow I'} = f(c(\text{Ada}(F_I) \parallel \text{Ada}(F_{I'}))) \quad (2)$$

Here, Ada is adaptive pooling and \parallel denotes concatenation. To facilitate the training process, we reframe angle regression as a k value classification problem, optimizing it through the cross-entropy loss function.

C. Global-Local Multi-modal Adaptation

Previous sim-to-real grasp detection methods typically focus on image-level domain adaptation. However, in the field of grasp detection, many studies [14], [16]–[18] point out the importance of features of local regions, which motivate us to design global-local alignment.

The patterns of the distribution gaps between RGB and depth data exhibit differences. Discrepancies in RGB images often arise from lighting and texture variations, while those in depth images primarily result from noise. Learning different domain gaps with a single domain classifier is challenging. To tackle this issue, we employ separate global domain classifiers for RGB and depth features. This enhances the robustness to distinct distribution shifts encountered in each modality through adversarial learning.

Given the output of RGB and depth encoders, represented as $F_I, F_D \in \mathbb{R}^{C \times H \times W}$, we construct two separate global

domain classifiers to predict domain labels, denoted as $P_I, P_D \in \mathbb{R}^{1 \times H \times W}$, for features originating from simulation and real-world, as formulated below:

$$P_m = \text{Sigmoid}(c(F_m)), m \in [I, D] \quad (3)$$

Here, c represents 1×1 convolutional layers. To extract features robust against domain shifts, we establish a min-max game. The objective of the domain classifiers is to accurately classify the origin domain of input features, while the RGB and depth encoder network strive to learn similar features for both simulated and real input, thereby confusing the domain classifiers. Given the domain label Q , the loss for the global domain classifier, denoted as L_I^{DC}, L_D^{DC} can be formulated as follows:

$$L_m^{DC} = -\frac{1}{H \times W} \sum_{i,j} [Q \log P_m^{i,j} + (1-Q) \log (1 - P_m^{i,j})], m \in [I, D] \quad (4)$$

To facilitate end-to-end training, we incorporate a GRL between the domain classifier and the encoder network. GRL works by reversing the gradient during the backpropagation phase: while the forward pass remains unchanged, the gradient sign is flipped in the backpropagation. Additionally, we introduce a local domain classifier to align multi-modal grasp features, which assists in reducing disparities within the grasping region, such as variations in object texture and shape. For the n th grasp region, denoted by domain label Q_n and prediction P_n , the local domain classification loss, denoted as L_G^{DC} , is formulated as follows:

$$L_G^{DC} = -\frac{1}{N_G} \sum_n [Q_n \log P_n + (1 - Q_n) \log (1 - P_n)] \quad (5)$$

where N_G represents the number of grasp regions, and similar to the global classifier, we insert a GRL between the grasp features and the local domain classifier. Furthermore, to improve the robustness of GPN across the simulated and real-world domains, we implement consistency regularization like [28] to reinforce the consistency of predictions from the global and local domain classifiers. Specifically, the consistency regularization loss is formulated as follows:

$$L_m^{CR} = \frac{1}{N_G} \sum_n \left\| \frac{1}{|P_m|} \sum_{i,j} P_m^{i,j} - P_n \right\|_2, m \in [I, D] \quad (6)$$

D. Grasp Prototype Adaptation

Local shapes can vary in grasp detection scenarios, which leads to significant differences in the instance-level grasp distribution. Directly aligning local grasp features between simulation and real-world ignores the internal distribution of grasp features, potentially incurring ambiguity by aligning grasp features from different local shapes. [20] proposes a prototype-based semantic alignment method to tackle a similar issue in object detection, where the features from the same category are formed into a prototype and the prototype distance in the same category is minimized between the source and target domains. However, for grasp detection, there are no explicit category divisions for constructing

prototypes. To address this problem, we introduce a Grasp Prototype Adaptation (GPA) module that generates pseudo-category prototypes for both the simulated and real-world domains based on in-plane grasp rotation angles. GPA aligns these prototypes and iteratively updates them during the training process.

The GPA module aims to minimize the distance between corresponding grasp prototypes from the simulated and real-world domains within the feature space. But two challenges arise: (1) how to divide and construct the grasp prototype; (2) the prototypes calculated within a small batch may deviate from the true grasp prototypes, rendering them unsuitable for alignment.

For the former issue, we employ the in-plane rotation angle θ of the local grasp as the division criterion. This choice is made because the distribution of in-plane rotation angles can accurately reflect the local shape distribution in planar grasp detection. We evenly divide the in-plane rotation space into L categories, and based on this division, we construct simulated and real-world grasp prototypes by averaging the features of the grasp regions within each pseudo-category. The equation for this process is as follows:

$$P_i^S = \frac{1}{|GT_i|} \sum_{r \in GT_i} F(r) \quad (7)$$

$$P_i^R = \frac{1}{|GRoI_i|} \sum_{r \in GRoI_i} F(r) \quad (8)$$

Here, P_i^S and P_i^R represent the i th prototype of simulated and real-world domains, GT_i and $GRoI_i$ denote the ground-truth grasp label and the pseudo grasp label predicted by GRoI-Net with the i th in-plane rotation angle class and $F(r)$ is the GRoI feature of region r . When calculating the simulated prototype, we directly use the ground-truth (GT) to extract the grasp region feature and obtain the in-plane angle. For real-world prototype calculation, we rely on the pseudo label predicted by GRoI-Net due to the absence of ground-truth labels.

For the latter issue, we introduce a weighted moving average to compute the global grasp prototypes, denoted as GP^S and GP^R , using the prototypes P^S and P^R from each iteration. The update process is defined as follows:

$$step = \lambda \cdot \text{sim}(P_i^{(t)}, GP_i^{(t-1)}) \quad (9)$$

$$GP_i^{(t)} = step \cdot P_i^{(t)} + (1 - step) \cdot GP_i^{(t-1)} \quad (10)$$

Here, $\text{sim}(a, b) = \frac{1}{2} \cdot \left(\frac{a^T \cdot b}{\|a\| \cdot \|b\|} + 1 \right)$ represents the cosine similarity, and λ stands for a small, fixed update step length. We multiply the fixed step length λ with the similarity between the prototype and global prototype as the weighted step length. This ensures gradual updates to the global prototypes throughout the training process. To maintain the consistency, we employ the L_2 distance to constrain the distance between the prototypes in the simulated and real-world domains, as shown below:

$$L_{GPA} = \sum_{i \in L} \|GP_i^S - GP_i^R\|_2 \quad (11)$$

E. Loss Function

By summarizing the grasp loss L_{grasp} and the losses described above, the overall loss function during training is formulated as:

$$L = L_{grasp} + \alpha (L_I^{DC} + L_D^{DC}) + \beta L_G^{DC} + \gamma (L_I^{CR} + L_D^{CR}) + \theta L_{GPA} \quad (12)$$

L_{grasp} consists of the losses from GRoI-Net and GPN, and for more details, please refer to [14].

IV. SIMULATED GRASP DATA GENERATION

In this section, we introduce the simulated grasping dataset used in this paper, namely Sim-GraspNet-Planar. Following [14], we use the object set as well as the corresponding grasp annotations in [18] to construct scenes in the PyBullet simulator. To generate a grasping scene, we first set the workspace parameters (*i.e.* desk color and specular reflection) for initialization. Then, we arbitrarily vary the number and category of objects and initialize object poses in the simulator through free fall due to gravity to the desk. Finally, the remaining parameters, such as camera angle and light direction are randomly sampled to capture RGB-D images of the scene from various perspectives. We create 500 scenes, each containing 20 different viewpoints, resulting in a total of 10,000 RGB-D images. Similar to [14], camera angles of simulated data are all less than 15° from the vertical direction of the desktop. Some scenes in our simulation dataset are shown in Fig. 3 (a). For grasp annotation, as shown in Fig. 3 (b), we employ object-level grasp labels from GraspNet-Billion [18] and project them to scene-level ones with object poses. Gripper poses that collide with either the table or objects, or those whose approach direction is near the table, are filtered out.

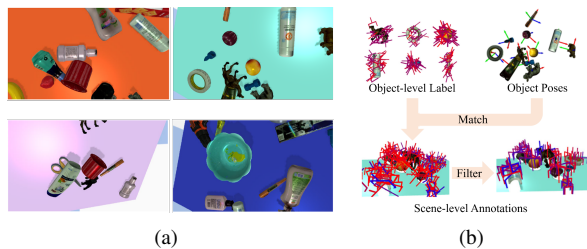


Fig. 3. (a) Visualization of scenes rendered in the simulator by DR. (b) Scene-level grasp annotation.

V. EXPERIMENTS

In this section, we evaluate the proposed GL-MSDA method with real-world input on the GraspNet-Planar benchmark [14] and also in the physical environment.

A. Protocols

For benchmark evaluation on GraspNet-Planar, we employ Average Precision (\mathbf{AP}_μ) under different friction coefficient μ as the metric and the overall result \mathbf{AP} is the average of \mathbf{AP}_μ , where μ ranges from 0.2 to 1.0 with the interval $\Delta\mu = 0.2$.

For physical evaluation, we adopt the same setting as in [14], where 25 objects of various sizes, shapes and textures

from the YCB Object Set [45] are used for single object and multi-object grasping. In the single object setting, each object is randomly placed randomly in three different poses and we record the Grasp Success Rate (GSR) as the metric. In the multi-object setting, each cluttered scene is composed of 5 objects and the grasping method works to clean the scene within 10 attempts. Besides GSR, we additionally use Scene Completion Rate (SCR) as the metric.

B. Implementation Details

The simulation data generated in Section IV serve as the source domain, while the GraspNet-Planar dataset [14] is utilized as the target domain. Our network is built upon ResNet-50 [46], with the image dimension set to $1,280 \times 720$ pixels. For the ratio of positive and negative samples and NMS threshold, we follow the parameters specified in [14]. The weights within the loss function in Eq. 12 are set as follows: $\alpha = 0.1$, $\beta = 0.1$, $\gamma = 0.1$, and $\theta = 1000.0$. Additionally, for the fixed update step length in Eq. 10, we set $\lambda = 0.001$. The number of in-plane rotation categories L is set to 12.

Our experiments are launched on 4 RTX 2080 Ti GPUs, with a batch size set to 8 and the allocation of two RGB-D images on each GPU. We apply the SGD optimizer, with the momentum and regularization parameter set to 0.9 and 0.0001, respectively. The learning rate is initially set to 0.005, and then reduced to 0.0005 after 64,000 iterations. We conduct a total of 96,000 training iterations and begin computing L_{GPA} after 56,000 iterations.

In physical evaluation, we employ a 7-DoF Agile Diana-7 robot arm, and RGB-D images are captured using an Intel RealSense D435i camera mounted at the end of the arm as shown in Fig. 4 (b). The inference process is executed on a single NVIDIA GeForce 1080 GPU.

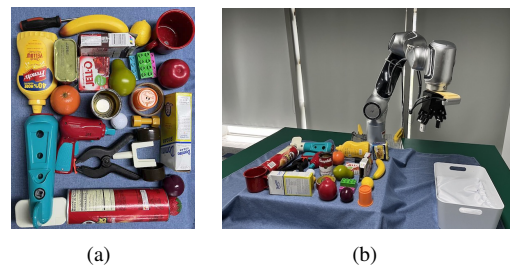


Fig. 4. (a) The 25 objects used for physical evaluation. (b) The 7-DoF Agile Diana-7 robot arm with Intel RealSense D435i camera mounted at the end.

C. Benchmark Evaluation

We conduct a comparative evaluation of our method and some representative sim-to-real ones on the GraspNet-Planar benchmark, as shown in Table I. The **Oracle** model refers to the model trained with labeled real-world data from GraspNet-Planar, which can be seen as the upper bound of the sim-to-real adaptation paradigm. Both the image-level adaptation [9] and feature-level adaptation [10] methods are taken as the counterparts. Due to the differences in the input modalities and network architecture, we replicate the DA methods utilized in the aforementioned two studies on our

TABLE I
PERFORMANCE COMPARISON ON GRASPNET-PLANAR CAPTURED BY REALSENSE.

Method	Seen			Similar			Novel		
	AP	AP _{0.8}	AP _{0.4}	AP	AP _{0.8}	AP _{0.4}	AP	AP _{0.8}	AP _{0.4}
Source Only	33.93	42.28	25.08	29.68	38.25	19.09	12.00	14.79	5.03
CycleGAN [44]	32.7	41.64	21.95	26.62	34.30	17.00	11.95	14.86	5.34
DAF [28]	37.41	46.02	28.78	33.59	41.98	24.31	12.83	15.96	5.65
GL-MSDA	39.25	48.31	30.43	34.38	43.20	24.15	13.86	17.34	5.89
Oracle	46.45	56.15	38.32	36.23	45.34	26.83	15.43	19.39	6.44

baseline. Specifically, we adopt DAF [28] for feature-level adaptation and CycleGAN [44] for image-level adaptation. It should be noted that DAF and CycleGAN only differ in the DA module compared to our method and their planar grasp detection networks are consistent with ours for fair comparison. It can be seen that GL-MSDA performs better than its counterparts and compared to the source-only model based on DR, it improves the performance by 4.68%, 5.01%, and 2.19% on seen, similar and novel objects respectively. Additional comparison results of different methods are shown in Fig. 5. Compared to the source-only model, our method exhibits a significant advantage. Regarding DAF, thanks to separate adaptation on the depth branch and fine-grained local alignment, our method works better on regions with complex shapes (elephant in the left column) or where accurate prediction of depth is required (screwdriver in the right column).

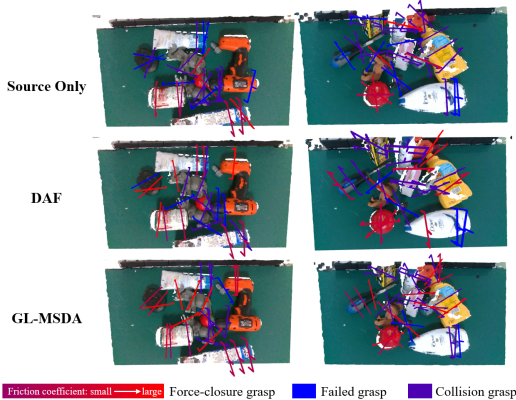


Fig. 5. Visualization of the results on GraspNet-Planar.

D. Physical Evaluation

We also compare our method to the source only model in real-robot experiments. As shown in Table II, for single object grasping, our method delivers a large improvement in terms of GSR. For multi-object grasping, our model achieves a gain of 19.83% in GSR and 16% in SCR. Furthermore, the results of multi-object scenes are close to the oracle model. These results highlight the effectiveness of our method.

E. Ablation Study

Influence of global and local domain classifier. In GL-MSDA, we employ both the global and local domain classifiers. To verify their effectiveness, we carry out an ablation study on them separately, as shown in the first and second rows of Table III. Without the global or local domain classifier, the results decrease by 0.81%, 0.57%,

TABLE II
PHYSICAL ROBOT EVALUATION RESULTS

Method	Single-Object	Multi-Object	
	GSR (%)	GSR (%)	SCR (%)
Source Only	40.00 (30/75)	43.18 (38/88)	76.00 (38/50)
GL-MSDA	62.67 (47/75)	63.01 (46/73)	92.00 (46/50)
Oracle	77.33 (58/75)	65.75 (48/73)	96.00 (48/50)

TABLE III
ABLATION STUDY OF GRASP DEPTH PREDICTION ON GRASPNET-PLANAR CAPTURED BY REALSENSE.

Method	Seen	Similar	Novel
Pre-train + Global	38.50	32.86	13.86
Pre-train + Local	38.21	33.52	13.2
Global + Local	38.61	33.33	13.84
Pre-train + Global + Local	39.02	34.09	13.89
Pre-train + Global + Local + GPA	39.25	34.38	13.86

0.69%, and 0.52%, 1.23%, 0.03%, on the seen, similar and novel set respectively. This demonstrates that for sim-to-real grasp detection, the global domain classifiers of the RGB and depth networks and the local domain classifier of the grasp features enhance alignment at multiple levels, resulting in better performance.

Influence of Self-supervised Rotation Pre-training. The self-supervised rotation pre-training scheme for RGB and depth networks helps the model learn robust and domain invariant features before training the grasp detection network. As shown in the third row of Table III, the performance drops 0.41%, 0.76% and 0.05% in the seen, similar and novel sets without pre-trained weights, indicating the necessity to the performance.

Influence of Grasp Prototype Adaptation. The GPA module aligns grasp features with similar shape distributions. As shown in the fifth row of Table III, it achieves an improvement of 0.23% on seen objects and 0.29% on similar objects, demonstrating the effectiveness of fine-grained alignment.

VI. CONCLUSION

In this paper, we propose a sim-to-real RGB-D grasp detection method, GL-MSDA. A multi-modal DA framework is designed to enhance the robustness to the domain gaps in RGB and depth modalities. The usage of local adaptation eases domain shift of instance-level grasp features between simulation and real-world. Moreover, we notice the intra-domain distribution of grasp features and present the GPA module which aligns local grasp features more sufficiently. Additionally, a simulation dataset with DR is constructed. The results of benchmark and real-robot experiments show the superiority of our method.

REFERENCES

- [1] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *IEEE International Conference on Robotics and Automation, ICRA*, 2016.
- [2] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *International Journal of Robotics Research, IJRR*, 2018.
- [3] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2018.
- [4] H. Zhang, D. Yang, H. Wang, B. Zhao, X. Lan, J. Ding, and N. Zheng, "REGRAD: A large-scale relational grasp dataset for safe and object-specific robotic grasping in clutter," *IEEE Robotics and Automation Letters, RA-L*, 2022.
- [5] 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*, 2021.
- [6] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2017.
- [7] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, W. Zaremba, and P. Abbeel, "Domain randomization and generative models for robotic grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2018.
- [8] R. Alghonaim and E. Johns, "Benchmarking domain randomisation for visual sim-to-real transfer," in *IEEE International Conference on Robotics and Automation, ICRA*, 2021.
- [9] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *IEEE International Conference on Robotics and Automation, ICRA*, 2018.
- [10] K. Fang, Y. Bai, S. Hinterstoisser, S. Savarese, and M. Kalakrishnan, "Multi-task domain adaptation for deep learning of instance grasping from simulation," in *IEEE International Conference on Robotics and Automation, ICRA*, 2018.
- [11] H. Zhu, Y. Li, F. Bai, W. Chen, X. Li, J. Ma, C. S. Teo, P. Y. Tao, and W. Lin, "Grasping detection network with uncertainty estimation for confidence-driven semi-supervised domain adaptation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2020.
- [12] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Advances in Neural Information Processing Systems, NIPS*, 2006.
- [13] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.
- [14] R. Qin, H. Ma, B. Gao, and D. Huang, "RGB-D grasp detection via depth guided learning with cross-modal attention," in *IEEE International Conference on Robotics and Automation, ICRA*, 2023.
- [15] M. Gou, H. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "RGB matters: Learning 7-dof grasp poses on monocular RGBD images," in *IEEE International Conference on Robotics and Automation, ICRA*, 2021.
- [16] H. Ma and D. Huang, "Towards scale balanced 6-dof grasp detection in cluttered scenes," in *Conference on Robot Learning, CoRL*, 2022.
- [17] F. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters, RA-L*, 2018.
- [18] H. 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*, 2020.
- [19] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research, JMLR*, 2016.
- [20] Y. Zheng, D. Huang, S. Liu, and Y. Wang, "Cross-domain object detection through coarse-to-fine feature adaptation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.
- [21] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," <http://pybullet.org>, 2016–2021.
- [22] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE International Conference on Robotics and Automation, ICRA*, 2018.
- [23] J. So, A. Xie, S. Jung, J. A. Edlund, R. Thakker, A. Agha-mohammadi, P. Abbeel, and S. James, "Sim-to-real via sim-to-seg: End-to-end off-road autonomous driving without real data," in *Conference on Robot Learning, CoRL*, 2022.
- [24] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2018.
- [25] D. Horváth, G. Erdős, Z. Istenes, T. Horváth, and S. Földi, "Object detection using sim2real domain randomization for robotic applications," *IEEE Transactions on Robotics, T-RO*, 2023.
- [26] A. Pashevich, R. Strudel, I. Kalevatykh, I. Laptev, and C. Schmid, "Learning to augment synthetic images for sim2real policy transfer," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2019.
- [27] K. Kleeberger, M. Völk, M. Moosmann, E. Thiessenhusen, F. Roth, R. Bormann, and M. F. Huber, "Transferring experience from simulation to the real world for precise pick-and-place tasks in highly cluttered scenes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2020.
- [28] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool, "Domain adaptive faster R-CNN for object detection in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018.
- [29] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Strong-weak distribution alignment for adaptive object detection," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.
- [30] V. VS, V. Gupta, P. Oza, V. A. Sindagi, and V. M. Patel, "Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.
- [31] L. Zhao and L. Wang, "Task-specific inconsistency alignment for domain adaptive object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2022.
- [32] Q. Tian, H. Sun, S. Peng, and T. Ma, "Self-adaptive label filtering learning for unsupervised domain adaptation," *Frontiers of Computer Science, FCS*, 2023.
- [33] S. Liu, X. Luo, K. Fu, M. Wang, and Z. Song, "A learnable self-supervised task for unsupervised domain adaptation on point cloud classification and segmentation," *Frontiers of Computer Science, FCS*, 2023.
- [34] H. Hsu, C. Yao, Y. Tsai, W. Hung, H. Tseng, M. K. Singh, and M. Yang, "Progressive domain adaptation for object detection," in *IEEE Winter Conference on Applications of Computer Vision, WACV*, 2020.
- [35] A. L. Rodriguez and K. Mikolajczyk, "Domain adaptation for object detection via style consistency," in *British Machine Vision Conference, BMVC*, 2019.
- [36] X. Zhang, R. Chen, A. Li, F. Xiang, Y. Qin, J. Gu, Z. Ling, M. Liu, P. Zeng, S. Han, Z. Huang, T. Mu, J. Xu, and H. Su, "Close the optical sensing domain gap by physics-grounded active stereo sensor simulation," *IEEE Transactions on Robotics, T-RO*, 2023.
- [37] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "SAPIEN: A simulated part-based interactive environment," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.
- [38] S. Iqbal, J. Tremblay, A. Campbell, K. Leung, T. To, J. Cheng, E. Leitch, D. McKay, and S. Birchfield, "Toward sim-to-real directional semantic grasping," in *IEEE International Conference on Robotics and Automation, ICRA*, 2020.
- [39] X. Li, R. Cao, Y. Feng, K. Chen, B. Yang, C. Fu, Y. Li, Q. Dou, Y. Liu, and P. Heng, "A sim-to-real object recognition and localization framework for industrial robotic bin picking," *IEEE Robotics and Automation Letters, RA-L*, 2022.
- [40] Q. Dai, J. Zhang, Q. Li, T. Wu, H. Dong, Z. Liu, P. Tan, and H. Wang, "Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects," in *European Conference on Computer Vision, ECCV*, 2022.
- [41] M. R. Loghmani, L. Robbiano, M. Planamente, K. Park, B. Caputo, and M. Vincze, "Unsupervised domain adaptation through inter-

- modal rotation for RGB-D object recognition,” *IEEE Robotics and Automation Letters, RA-L*, 2020.
- [42] Y. Ganin and V. S. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *International Conference on Machine Learning, ICML*, 2015.
- [43] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in *International Conference on Learning Representations, ICLR*, 2018.
- [44] J. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *IEEE International Conference on Computer Vision, ICCV*, 2017.
- [45] B. Çalli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. S. Srinivasa, P. Abbeel, and A. M. Dollar, “Yale-cmu-berkeley dataset for robotic manipulation research,” *International Journal of Robotics Research, IJRR*, 2017.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.