2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Learning Sim-to-Real Dense Object Descriptors for Robotic Manipulation

Hoang-Giang Cao[1], Weihao Zeng[1,2], and I-Chen Wu[†1,3]

[1]*Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan*
[2]*School of Computer Science, Carnegie Mellon University, United States*
[3]*Research Center for IT Innovation, Academia Sinica, Taiwan*

***Abstract*— It is crucial to address the following issues for ubiquitous robotics manipulation applications: (a) vision-based manipulation tasks require the robot to visually learn and understand the object with rich information like dense object descriptors; and (b) sim-to-real transfer in robotics aims to close the gap between simulated and real data. In this paper, we present Sim-to-Real Dense Object Nets (SRDONs), a dense object descriptors that not only understands the object via appropriate representation but also maps simulated and real data to a unified feature space with pixel consistency. We proposed an object-to-object matching method for image pairs from different scenes and different domains. This method helps reduce the effort of training data from real-world by taking advantage of public datasets, such as GraspNet. With sim-to-real object representation consistency, our SRDONs can serve as a building block for a variety of sim-to-real manipulation tasks. We demonstrate in experiments that pre-trained SRDONs significantly improve performances on unseen objects and unseen visual environments for various robotic tasks with zero real-world training.**

I. Introduction

Vision-based robotics reinforcement learning methods have enabled solving complex robotics manipulation tasks in an end-to-end fashion [1], [13], [12]. The ability to understand unseen objects is one of crucial issues for robotics tasks. Although object segmentation is helpful, object-level segmentation ignores the rich structures within objects [7]. A better object-centric descriptor is critical for ubiquitous robotics manipulation applications. Dense Object Nets (DONs) can learn object representation useful for robotics manipulation in a self-supervision manner [7]. The learned dense descriptors enabled interesting robotics applications, such as soft body manipulation and pick-and-place from demonstrations [5], [18]. DONs are trained with matching and non-matching pixel coordinates in pairs of images. However, in the data generation process in DONs, since each pair of individual training images comes from the same object configuration, this makes it hard to be used in different object configurations. Object configuration is the setup of the object's positions in a scene. Thus, it becomes vital to learn explicitly from different object configurations for reliable object-centric descriptors.

Due to the difficulty of collecting a large amount of data in the real-world, which is crucial for learning-based robotics applications, we often train agents in the simulation and migrate to the real-world, but this poses the sim-to-real gap problem [10]. To successfully deploy learning-based robotics tasks into the real-world, a good dense object descriptors method needs not only to represent objects well but also represent simulation and real objects consistently. Prior works focus on either solving the representation problem [7] or the sim-to-real gap problem [9].

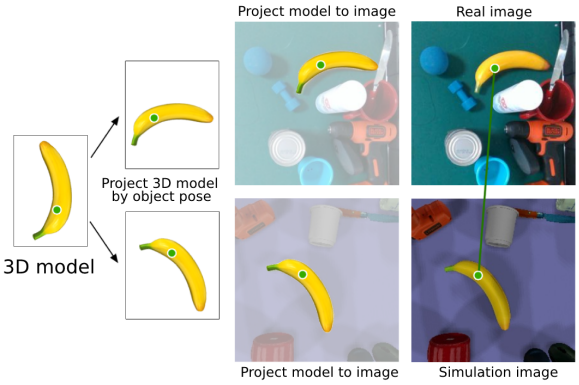


Fig. 1: Object-to-object matching.

In this paper, we present SRDONs (Sim-to-Real Dense Object Nets), a dense object descriptors with sim-to-real consistency. To represent rich object structures with sim-to-real consistency, we utilize object poses and object meshes to automatically generate matching and non-matching pixel coordinates for image pairs from different object configurations (where the images come from different scenes) and different data domains (simulation or real-world). Such data generation process enables training SRDONs using readily available large-scale public dataset. SRDONs explicitly learn dense object descriptors from different object configurations in the simulation and the real-world. The resulting dense object descriptors can effectively represent simulation and real-world object information in a pixel consistent manner. Furthermore, SRDONs exhibit great generalization ability from our experiments. And, SRDONs perform well on unseen objects in unseen visual environments in simulation and the real-world.

Contributions. The main contributions of this paper can be summarized as follows: 1) We present SRDONs, a dense object descriptors representation with sim-to-real consistency

[†]Correspondence.
Code and data: `https://github.com/hgiangcao/SRDONs`

and generalization ability. 2) We propose the matching pixel method for image pairs from different object configurations and different data domains, which helps reduce the effort of training data from real-world by taking advantage of public datasets, such as GraspNet [6]. 3) In experiments, we demonstrate the effectiveness of SRDONs in sim-to-real transfer on several robotics tasks and achieve high sim-to-real performances with unseen objects, unseen visual environments, and zero real-world training.

II. Background

A. Dense Object Descriptors

The ability to recognize and interact with unseen objects is critical in robotics applications. Recent works have explored unseen object segmentation [19], [20]. However, the segmentation objective disregards the rich information within objects, e.g. curves and flat surfaces. Dense object descriptors by [7] is a promising direction for providing such ability.

Dense Object Nets (DONs) is a self-supervised method for generating dense object descriptors useful for robotics applications. DONs learns from point to point correspondence, and are able to provide rich information within objects. Recent works demonstrated the effectiveness of DONs in many challenging robotics applications. The work in [18] presented a method for manipulating ropes using dense object descriptors. Another work by [5] enforced an additional object-centric loss to enable multi-step pick-and-place from demonstration; while in [21] imposed an additional object loss to achieve goal-conditioned grasping. Cao *at el.* [4] proposed Cluttered Object Descriptors (CODs) to represent the objects in a cluttered for robot picking cluttered objects.

B. Sim-to-Real Transfer

Training learning-based methods in the real-world are costly for robotics applications. Hence, bridging the gap between simulation and real-world is critical for ubiquitous learning-based robotics applications. Recent works mainly focus on domain randomization and domain adaptation.

In domain randomization, we randomize the simulation so that the policies are robust enough to handle real-world data [22]. Such methods include randomizing textures, rendering, and scene configurations [9]. However, domain randomization requires careful task-specific engineering in selecting the kind of randomization.

In domain adaptation, we map simulation and real-world data into a knowledge preserving unified space. We can directly map simulation images to real-world images, where GANs are commonly applied [2], [17], [9]. However, GAN-based methods do not generalize well to unseen objects and scenes [9], which is crucial for many robotics applications. Prior works have also explored learning domain invariant features [8], [16], [3]. The work in [11] employed the temporal nature of robot experience. Another approach by [8] separated task-specific and domain-specific knowledge via adversarial training.

While many of these prior works for dense object descriptors only used data in the same domain (only real data or simulation data) for training, we train the descriptor to match the pixels of the object between simulation and real-world images. Therefore, our SRDONs not only learn the useful object descriptors but also map the simulation and real-world images into a unified feature space with pixel consistency, addressing the sim-to-real problem.

III. SRDONs: Sim-to-Real Dense Object Nets

A. Object-to-Object Matching

Previous works [7], [5], [4], by using 3D TSDF (truncated signed distance function) reconstruction, can only generate matching pixels of static scenes in the same data domain. Here, we proposed an object-to-object matching method with object poses and 3D models, which can generate matching points for images from different scenes and data domains. Additionally, while other methods required collecting training data by running real robots, our method reduces time and cost by taking advantage of public datasets.

As illustrated in Figure 1, we compute the pixel coordinates corresponding to the 3D vertices on the same object model in each image to find matching points. Suppose an image contains a set of objects, $O = (O_1, O_2, O_3, \ldots, O_n)$, and pose annotations for each object, $\Phi = (\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_n)$. The 3D model O_i is given by a set of vertices $\mathbf{V_i} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})^T$. To associate 3D model vertices with 2D pixel coordinates, we also need the projection matrix $\mathcal{P}_i$ for each object O_i, where $\mathcal{P}_i$ is computed from the intrinsic matrix, K, and extrinsic matrix, E_i. K deals with the camera properties, and is known from the camera properties; E_i represents the translation, t_i, and the orientation R_i of object O_i with respect to the camera. E_i is computed from the object pose Φ_i. Note the pose annotations are in the camera frame, so we do not need to consider camera transformations.

The projection matrix $\mathcal{P}_i$ is:

$$\mathcal{P}_i = KE_i \quad \text{with} \quad E_i = [R_i|t_i] \tag{1}$$

We then project all 3D vertices $\mathbf{V_i}$ of the object O_i onto the image coordinate system to get their 2D corresponding pixel coordinates $(\mathbf{u}, \mathbf{v})^T$.

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{1} \end{bmatrix} = \mathcal{P}_i \begin{bmatrix} \mathbf{X'} \\ \mathbf{Y'} \\ \mathbf{Z'} \\ \mathbf{1} \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} \mathbf{X'} \\ \mathbf{Y'} \\ \mathbf{Z'} \end{bmatrix} = \Phi_i \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} \tag{2}$$

Method	Robotic sampling	Multi-objects (classes)	Sim to Real
Original DONs[7]	Yes	Yes (3)	No
MCDONs[5]	Yes	Yes (8)	No
LE DONs[14]	Yes	No	No
MODONs[21]	No	Yes (16)	No
Our SRDONs	No	Yes (28)	Yes

TABLE I: Comparison of different datasets.

For a pair of images both containing some objects, we randomly sample from the object models a subset of their 3D vertices and calculate their corresponding 2D pixel coordinates in each image. The pixel coordinates in each image corresponding to a 3D model vertex are considered as matching pixel coordinates. To deal with occlusion, we assign the pixel coordinate to the vertex closest to the camera in Euclidean distance. Object-to-object matching enables generating matching in a variety of scenarios: different scene matching (dynamic scenes), sim-to-real matching, and multiple matching (finding matching between one object in an image with multiple of the same objects in another).

B. Contrastive Loss

We employ the contrastive loss from [7] to enable self-supervised learning for SRDONs. Given an image $I \in \mathbb{R}^{W \times H \times d}$ where d can be either $3, 4$ depending on whether the input is RGB or RGBD, we map I to a dense descriptor space $\mathbb{R}^{W \times H \times D}$. Each pixel in I has a corresponding D-dimensional feature vector. Given a pair of images, the matching pixels coordinates, and the non-matching pixels coordinates, we optimize the dense descriptor network, f, to minimize the L2 distances between descriptors of matching pixels, and keep descriptors of non-matching pixels M distance apart.

$$\mathcal{L}_{\text{m}}(I_a, I_b) = \frac{1}{N_{\text{m}}} \sum_{N_{\text{m}}} \|f(I_a)(u_a) - f(I_b)(u_b)\|_2^2 \tag{3}$$

$$\mathcal{L}_{\text{nm}}(I_a, I_b) = \frac{1}{N_{\text{strict\_nm}}} \sum_{N_{\text{nm}}} max(0, M - \|f(I_a)(u_a) - f(I_b)(u_b)\|_2)^2 \tag{4}$$

$$\mathcal{L}(I_a, I_b) = \mathcal{L}_{\text{m}}(I_a, I_b) + \mathcal{L}_{\text{nm}}(I_a, I_b) \tag{5}$$

where "m" is matching, and "nm" is non-matching; N_{m} is the number of matches, and N_{nm} is the number of pairs of non-matching pixels ; $f(I)(u)$ is the descriptor of I at pixel coordinate u; $N_{\text{strict\_nm}}$ is the number of pairs of non-matching descriptors within M distance to each other, namely the number of non-zero terms in the summation term in Equation 4.

C. Data Collection and Training SRDONs

Previous works [7], [5] required the use of a real robot arm to collect data. In contrast, as described in Subsection III-A, our proposed matching method enables to use real data from public datasets and simulated data generated from the simulation. By this approach, we not only have easy access to diverse objects but also reduce the time and cost of real training data collection. Table I compares our dataset with other works.

Real Data. In this paper, we mainly use the real data from GraspNet [6]. The dataset provides 97,280 RGBD images of 88 objects over 190 cluttered scenes. Each scene contains 9-10 objects placed at random positions on a tabletop. They capture 256 images per scene with different view poses and recorded the camera pose, the 6D pose of each object corresponding to each image. However, to make the view of each scene sufficiently different, we downsampled the number of images to 50 images per scene.

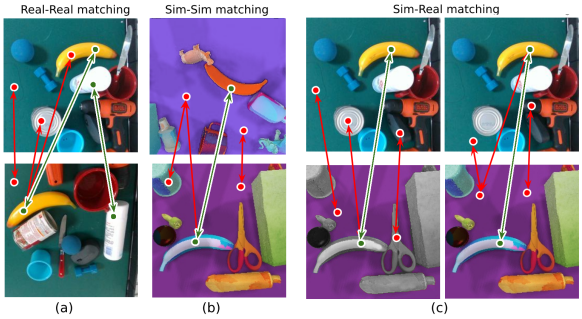


Fig. 2: Different pairing types for training the descriptors. (a) Real-Real pairing. (b) Sim-Sim pairing. (c) Sim-Real pairing without (left) and with texture randomization (right). Green lines indicate pairs of match points; while red lines indicate pairs of non-match points.

Simulation Data. We use V-REP simulator to generate the simulation image. For each scene, we randomly drop 9 to 10 objects on a table. We then use a camera to capture the RGBD images and record the camera poses and object poses with the same format as GraspNet dataset. We also apply texture randomization and background randomization for generalization purpose.

Pairing Images. With the proposed object-to-object matching in subsection III-A, we can generate matching pixel coordinates for any pair of images independent of data domain (simulation or real-world). We have 3 different types of pairing: (a) Sim-Sim: a pair of images is sampled from simulated images. (b) Real-Real: a pair of images is sampled from real images. (c) Sim-Real: one image is sampled from simulated images, and the other comes from the real images. Figure 2 shows some examples of different pairing types.

Training. During each training step, we uniformly sample pairing types (Sim-Sim, Real-Real, and Sim-Real). Once a type has been sampled, we then choose whether the two images are from the same scene or different scenes (with the probability of 30% and 70%, respectively). For Sim-Sim and Real-Real matching, two images may come from the same or different scenes, while for Sim-Real matching, two images have to come from different scenes, since they come from different data domains. For each pair of images, we sample 1000 pairs of matching points, and 5000 pairs of non-matching points (object to object, object to background, background to background). More details about collecting data and training are provided in the *accompanying video*.

D. SRDONs for Robotics Learning Tasks

We want to use SRDONs to serves as a building block for robotic tasks. The work in [4] proposed a network structure that can use the intermediate layers of the descriptor network for training a reinforcement learning task. We adopt their method, and extend to supervised learning method.

To use SRDONs for training the reinforcement learning (RL) task, we simply apply the same structure as proposed

	Input	Sim-Sim Rd-100	Real-Real	Sim-Real Rd-0	Sim-Real Rd-100
Original DONs[7]	RGB	0.155 / 0.265	0.430 / 0.203	0.143 / 0.281	0.140 / 0.289
MODONs[21]	RGB	0.157 / 0.267	**0.947 / 0.027**	0.189 / 0.264	0.164 / 0.279
CODs [4]	RGBD	**0.939 / 0.063**	0.256 / 0.238	0.424 / 0.216	0.389 / 0.230
SRDONs - Rd-0	RGB	0.248 / 0.241	0.935 / 0.063	0.695 / 0.140	0.258 / 0.264
SRDONs - Rd-80	RGB	0.857 / 0.095	0.910 / 0.085	0.915 / 0.092	0.908 / 0.098
SRDONs - Rd-100	RGB	0.899 / 0.082	0.904 / 0.084	0.915 / 0.086	0.936 / 0.088
SRDONs - Rd-0	RGBD	0.248 / 0.248	0.941 / 0.059	0.684 / 0.140	0.247 / 0.269
SRDONs - Rd-80	RGBD	0.908 / 0.079	0.917 / 0.082	**0.925 / 0.084**	**0.941 / 0.086**
SRDONs - Rd-100	RGBD	0.911 / 0.077	0.913 / 0.076	0.911 / 0.089	0.933 / 0.090

TABLE II: Evaluate matching results with different methods (in accuracy/error distance). For fairness, we use the same descriptor dimension (D=8) for all models. The above results are also visualized as in the *accompanying video*.

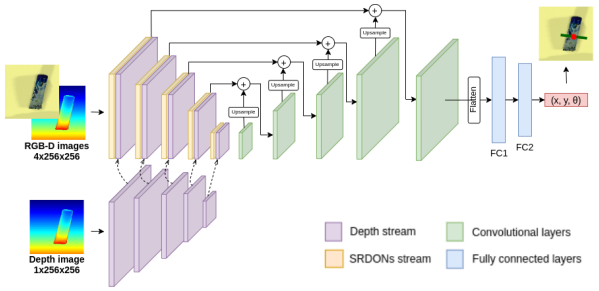


Fig. 3: The architecture of using SRDONs in grasping tasks.

in [4] (further details in the *accompanying video*), which is based on actor-critic RL, namely PPO. In the supervised learning task, we slightly modify the network structure. We first remove the critic head, and then, change the actor head which originally is fully convolutional layers to fully connected layers. Specifically, in our experiment, we use the grasping task to demonstrate the use of SRDONs with supervised learning. Grasping task requires the agent to predict the grasping position (x, y) and the top-down grasping pose (θ) to grasp the object. Therefore, the output of the network in this task is the pose (x, y, θ) $\in \mathbb{R}^3$. Figure 3 shows how we combine the intermediate layer of SRDONs and the depth stream in a U-Net structure.

IV. Experimental Results

We conduct experiments to evaluate SRDONs performances on providing good sim-to-real object descriptors in Subsection IV-A. Then we use a pre-trained SRDONs as the building block for solving two robotic manipulation tasks: two-fingered grasping in Subsection IV-B and picking cluttered objects with suction in Subsection IV-C. The robotic tasks are both trained entirely in the simulation, and directly tested in the real-world with zero real-world training.

A. Evaluation of Sim-to-Real Object Descriptors

We evaluate the performance of descriptors by finding matching interest points, as in [7]. We employ two evaluation metrics: the accuracy of matching correct objects, and the matching error distance normalized by image diagonal distance. Given a pair of source and target images and a point p on the source image that belongs to an object O, let p^* indicate the true match point in the target image (disregarding the case of occlusion as described above), and p' indicate the best match point by using a given descriptor. For the former, the object matching accuracy of matching the correct objects, i.e., p' and p^* are on the same object in the target image. For the latter, the matching error distance is the average distance between p^* and p'.

In the experiments, we use the following methods as the baselines: (a) The original DONs [7]. (b) The Multi-Object DONs (MODONs) [21]. (c) The Cluttered Object Descriptors (CODs) [4]. When training the original Dense Object Nets, we use Real-Real in the same scenes; for the Multi-Object DONs, we use Real-Real pairing in both the same and different scenes; for the CODs, we use Sim-Sim in both the same and different scenes; and our proposed SRDONs uses Sim-Real paring only. We also evaluated the effects of randomizing object textures by randomizing 0%, 80%, and 100% of the object textures when training SRDONs, denote as Rd-0, Rd-80, and Rd-100, respectively. Additional training details and matching results are reported in the *accompanying video*.

Sim-to-Real Finding Matching Points. To evaluate the matching performance, we select 500 unseen image pairs for each type of pairing. For each pair of images, we sample 1000 matching points and evaluate the matching performances. Table II shows the experimental results of finding matching points of objects in the same domain (Sim-Sim Rd-100, Real-Real) and different domains (Sim-Real Rd-0 and Rd-100). (Note that Sim-Sim Rd-0 is less interesting so ignored in the table.) We can see that the original DONs method (in the second row), which trained with same scenes only, fails to represent multi-object scenes. In different domains like Sim-Real, our SRDONs shows the best performance in the rightmost two columns. In the same domains like Sim-Sim and Real-Real, the result of our SRDONs are close to other methods which are trained with these specific types of paring, while our method used Sim-Real pairing only. We visualize the descriptors of different methods in Figure 4, which shows that SRDONs

can represent objects in simulation and real-world images with pixel consistency. Furthermore, texture randomization enables SRDONs to focus on the object geometry rather than color, as shown by the consistent object representation under texture randomization.

For training the descriptors, we leverage public datasets like GraspNet, however, our SRDONs also works with unseen objects. Figure 5 shows the result of our SRDONs when performs testing on unseen objects. We can see that DONs fails to represent and find the matching points with unseen objects in a multiple-object scene. In contrast, our SRDONs is able to represent objects in the images consistently in the representation space and perform better matching. Moreover, the matching performance of our method is improved by adding texture randomization and depth information, which are not considered in other DONs-based methods.

Sim-to-Real Multi-Object Consistent Evaluation. We conduct the experiment to verify that SRDONs are able to represent objects in simulation and real-world images with object consistency. We use 100 unseen images from both simulation and real images. Each image contains 9 to 10 objects. We feed these images through the SRDONs, and randomly select 1000 pixel-descriptors per image. Then, we use t-SNE to project the selected pixel-descriptors into two-dimensional for visualization, as shown in Figure 6a and 6b. In particular domain, the descriptors of the same objects are clearly distinguishable from the other objects. While, in different domains, the descriptors of the same object from the real-world and the simulation reside in similar regions.

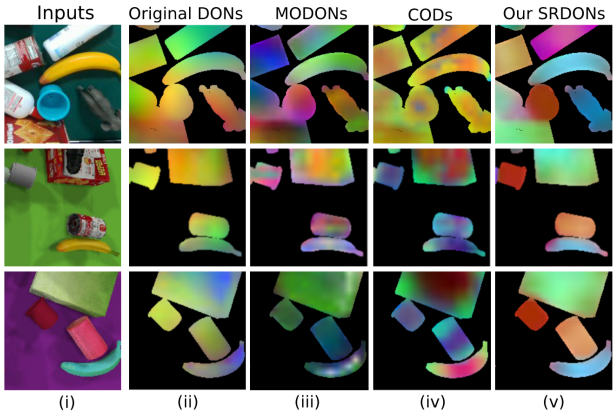


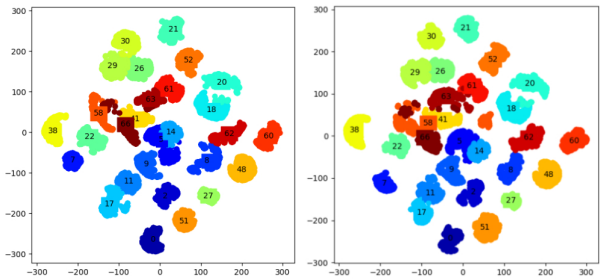
Fig. 4: Evaluate sim-to-real descriptor translation. (i): different inputs: real image (top), simulated image without texture randomization (middle) and with texture randomization (bottom). (ii)-(v): descriptors generated by different methods Our SRDONs is able to represent the objects consistently in different inputs. The colors of these descriptors are produced in a similar way to t-SNE.

B. Object Grasping

We used SRDONs as a building block in a robotics two-fingered grasping task to grasp an object that placed at randomly pose on the table. We use the supervised learning

(a) Original DONs - RGB. (b) SRDONs - RGBD - Rd-80.

Fig. 5: Compare performances on *unseen* objects between (a) the original DONs trained with RGB input and (b) SRDONs trained with RGBD input with Rd-80 setting. In each subfigure, the top two images are inputs from different scenes, and the bottom two images are the corresponding descriptors of the above inputs. Green lines indicate match points based on the descriptors.



(a) (b)

Fig. 6: Clustering by t-SNE of object pixel-descriptors produced by SRDONs in simulation (a) and real (b) images. The points from the same object are marked in the same color.

(described in Subsection III-D) to predict the grasping pose, though applying reinforcement learning with continuous action spaces (DDPG [15]) or discretizing the continuous action space will also work in this task.

In the simulation, we use Mean Square Error (MSE) between the ground-truth and the predicted grasping poses angle in radiance as the evaluation metrics in training and testing. In the real-world experiments, we do a real grasping by the robot and measure the success grasp rate.

We use the following methods as baselines: (a) Domain randomization methods with RGB and RGBD as the inputs, denoted by RGB and RGBD respectively. The model is similar to Figure 3, however, we replace the SRDONs stream by RGB stream, which is a trainable ResNet34_8s. (b) The model is similar to in Figure 3, but we remove the SRDONs and using depth only, denoted by Depth. (c) The method proposed by [4] for picking cluttered objects in simulation, denoted by CODs. (d) Our supervised learning method proposed in Subsection III-D, denoted by SRDONs.

In the simulation, we generated the ground truth grasping orientations, e.g., we place objects on the table in such a way that the grasping orientation is zero, where the robot

can grasp the objects with zero z-rotation. We captured observations from different camera poses, and calculated the grasping location and the z-rotation (x,y,θ) labels via camera poses. For training data, we captured 50 RGBD images for each 28 objects from the GraspNet train slipt. We then test the grasping methods with 15 objects from the GraspNet test splits. In the real-world, we use a parallel gripper to grasp a single object that is placed on the table at random position and orientation. Each model performs 20 grasping trials (repeat twice with 10 objects collected from our lab).

Table III shows the results for grasping orientations prediction in both simulation and real-world environments. In the simulation, SRDONs achieve the minimal average error of 0.15 radiance (8.59 degrees) on unseen objects. Furthermore, the agent with SRDONs also outperformed others in the real-world by achieving 90% success grasp rate without any further training. We can see that the models with depth information performed better than the others that use RGB input only. Training details and real experiment videos are provided in the *accompanying video*.

Method	Sim train (radiance)	Sim test (radiance)	Real world
RGB	0.09	0.49	65%
RGBD	0.061	0.35	85%
Depth	0.08	0.36	85%
CODs	0.071	0.39	50%
SRDONs	**0.035**	**0.15**	**90%**

TABLE III: Result of grasping in simulation and real-world.

C. Picking Cluttered General Objects

Now, we used SRDONs as a building block for a more complex robotics picking task. Similarly as [4], we train an agent with reinforcement learning to pick cluttered objects with a suction pad. We have two metrics for evaluating the performance. The first is the rate of completion for all runs. A run is completion if all objects are picked before the episode terminates. The second is the average number of objects picked in all runs. In this picking cluttered objects task, we use the similar baselines to those in grasping task in Subsection IV-B, but with reinforcement learning version.

In the simulation, we train each method in with 10 random objects sampled from GraspNet train split. We then test with 20 and 30 objects from GraspNet test splits, and novel household objects. In the real-world, we directly use the trained policy in the simulation to pick 10 novel household objects without any fine-tuning.

The experimental results in the simulation are shown in Table IV and Table V. Our method with SRDONs clearly out-performed other methods on all of the metrics, and are also efficient to be generalized to more cluttered scenarios with unseen objects. When directly applying the trained policy in the simulation to the real-world testing, our method can successfully pick all 10 objects within 12.81 steps (78.1% success pick rate), which is better than other methods (as shown in Table VI). Training details and real experiment videos are provided in the *accompanying video*.

Dataset	Grasp-Net	Grasp-Net	Novel objects
#obj	20	30	20
RGB	33.8%	24.5%	16.1%
RGBD	39.2%	25.1%	43.6%
Depth	89.2%	77.6%	68.3%
CODs	95.3%	92.9%	95.1%
SRDONs	**97.8%**	**94.1%**	**97.5%**

TABLE IV: Picking completion rates in simulation.

Dataset	Grasp-Net	Grasp-Net	Novel objects
#obj	20	30	20
RGB	15.8	19.5	12.9
RGBD	16.9	23.4	16.1
Depth	18.9	25.5	18.1
CODs	19.1	28.3	18.9
SRDONs	**19.7**	**29.6**	**19.2**

TABLE V: Average number of picked objects in simulation.

Method	Completion rate	Success rate	Average step
RGB	63.63%	61.12%	16.36
RGBD	90.9%	70.97%	14.90
Depth	72.72%	62.15%	16.09
CODs	72.72%	61.8%	16.18
SRDONs	**100%**	**78.1%**	**12.81**

TABLE VI: Result of picking objects in real-world.

V. Conclusion

This paper presents SRDONs, a dense object descriptors representation with sim-to-real consistency. Our method addresses both of the object representation problem and the sim-to-real gap problem. Through experiments, we demonstrated that our method can provide useful object information while representing simulation and real-world objects with pixel consistency. We showed that SRDONs enabled zero-shot sim-to-real transfer in robotic manipulation tasks on unseen objects and unseen visual environments. With the representation power of SRDONs, we expect to accelerate the sim-to-real deployment process for robotics applications.

A challenging case that we observed in this study is occluded point matching, which is commonly a challenge for most robotic tasks with cluttered objects. In occluded point matching, our SRDONs can match to the correct object, but not at the precise location. We expect that adding more constraints like region matching, or geometry consistency can improve the performance of occluded point matching.

REFERENCES

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017.

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29:343–351, 2016.

[4] Hoang–Giang Cao, Weihao Zeng, and I–Chen Wu. Reinforcement learning for picking cluttered general objects with dense object descriptors. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6358–6364, 2022.

[5] Chun-Yu Chai, Keng-Fu Hsu, and Shiao-Li Tsao. Multi-step pick-and-place tasks using object-centric dense correspondences. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4004–4011, 2019.

[6] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020.

[7] Peter Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *Conference on Robot Learning*, 2018.

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[9] Daniel Ho, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. Retinagan: An object-aware approach to sim-to-real transfer, 2020.

[10] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Florian Golemo, Melissa Mozifian, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Perspectives on sim2real transfer for robotics: A summary of the r: Ss 2020 workshop. *arXiv preprint arXiv:2012.03806*, 2020.

[11] Rae Jeong, Yusuf Aytar, David Khosid, Yuxiang Zhou, Jackie Kay, Thomas Lampe, Konstantinos Bousmalis, and Francesco Nori. Self-supervised sim-to-real adaptation for visual robotic manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2718–2724. IEEE, 2020.

[12] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

[13] Alexander Khazatsky, Ashvin Nair, Daniel Jing, and Sergey Levine. What can i do here? learning new skills by imagining visual affordances. *arXiv preprint arXiv:2106.00671*, 2021.

[14] Andras G. Kupcsik, Markus Spies, Alexander Klein, Marco Todescato, Nicolai Waniek, Philipp Schillinger, and Mathias Bürger. Supervised training of dense object nets using optimal descriptors for industrial robotic applications. *CoRR*, abs/2102.08096, 2021.

[15] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016.

[16] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[17] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. *CoRR*, abs/2006.09001, 2020.

[18] Priya Sundaresan, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Michael Laskey, Kevin Stone, Joseph E. Gonzalez, and Ken Goldberg. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. *CoRR*, abs/2003.01835, 2020.

[19] Yu Xiang, Christopher Xie, Arsalan Mousavian, and Dieter Fox. Learning rgb-d feature embeddings for unseen object instance segmentation. *arXiv preprint arXiv:2007.15157*, 2020.

[20] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics*, 2021.

[21] Shuo Yang, Wei Zhang, Ran Song, Jiyu Cheng, and Yibin Li. Learning multi-object dense descriptor for autonomous goal-conditioned grasping. *IEEE Robotics and Automation Letters*, 6(2):4109–4116, 2021.

[22] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744. IEEE, 2020.