

Attribute-Based Robotic Grasping With Data-Efficient Adaptation

Yang Yang , Houjian Yu , Xibai Lou , Yuanhao Liu , and Changhyun Choi 

Abstract—Robotic grasping is one of the most fundamental robotic manipulation tasks and has been the subject of extensive research. However, swiftly teaching a robot to grasp a novel target object in clutter remains challenging. This article attempts to address the challenge by leveraging object attributes that facilitate recognition, grasping, and rapid adaptation to new domains. In this work, we present an end-to-end encoder–decoder network to learn attribute-based robotic grasping with data-efficient adaptation capability. We first pretrain the end-to-end model with a variety of basic objects to learn generic attribute representation for recognition and grasping. Our approach fuses the embeddings of a workspace image and a query text using a gated-attention mechanism and learns to predict instance grasping affordances. To train the joint embedding space of visual and textual attributes, the robot utilizes object persistence before and after grasping. Our model is self-supervised in a simulation that only uses basic objects of various colors and shapes but generalizes to novel objects in new environments. To further facilitate generalization, we propose two adaptation methods, adversarial adaption and one-grasp adaption. Adversarial adaptation regulates the image encoder using augmented data of unlabeled images, whereas one-grasp adaptation updates the overall end-to-end model using augmented data from one grasp trial. Both adaptation methods are data-efficient and considerably improve instance grasping performance. Experimental results in both simulation and the real world demonstrate that our approach achieves over 81% instance grasping success rate on unknown objects, which outperforms several baselines by large margins.

Index Terms—Deep learning in grasping and manipulation, grasping, perception for grasping and manipulation.

I. INTRODUCTION

OBJECT attributes are generalizable properties in object manipulation. Imagine how we describe a novel object when asking someone to fetch it, “Please give me the apple, a red sphere.”, we intuitively characterize the target by its appearance attributes (see Fig. 1). If an assistive robot can be similarly

Manuscript received 8 August 2023; accepted 22 December 2023. Date of publication 12 January 2024; date of current version 31 January 2024. This paper was recommended for publication by Associate Editor M. Walter and Editor J. Bohg upon evaluation of the reviewers’ comments. This work was supported in part by the NSF under Grant 2143730, in part by the Sony Research Award Program, UMII-MnDRIVE Ph.D. Graduate Assistantship, and MnDRIVE Initiative on Robotics, Sensors, and Advanced Manufacturing [DOI: 10.1109/ICRA48506.2021.9561139]. This paper is an evolved version of the conference paper [1]. (Yang Yang and Houjian Yu are joint first authors.) (Corresponding author: Yang Yang.)

The authors are with the University of Minnesota, Minneapolis, MN 55413 USA (e-mail: yang5276@umn.edu; yu000487@umn.edu; lou00015@umn.edu; liu00800@umn.edu; cchoi@umn.edu).

Supplementary material is available at <https://z.umn.edu/attr-grasp>.

This article has supplementary downloadable material available at <https://doi.org/10.1109/TRO.2024.3353484>, provided by the authors.

Digital Object Identifier 10.1109/TRO.2024.3353484

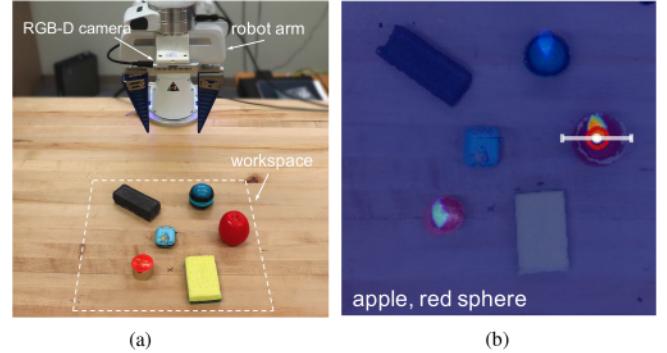


Fig. 1. *Attribute-based instance grasping*. Various objects of generic attributes are placed in the workspace, and we propose to grasp a target object by describing its attributes, e.g., “Please give me the apple, a red sphere.” (a) Objects of various attributes. (b) Grasping affordances.

commanded utilizing such object attributes (e.g., color, shape, and category name, etc.), it would allow better generalization capability for novel objects than using a discrete set of predefined category labels. Moreover, individuals learn to recognize and grasp an unknown object through rapid interactions; hence, it would be advantageous if a grasping pipeline is capable of adapting with minimal adaptation data. These factors motivate the development of attribute-based robotic grasping with data-efficient adaptation capability.

Recognizing and grasping a target object in clutter is crucial for an autonomous robot to perform daily-life tasks in the real world. Over the past years, the robotics community has made substantial progress in target-driven robotic grasping by combining off-the-shelf object recognition modules with data-driven grasping models [2], [3]. However, these recognition-based approaches presume a unique ID for each category and are likely to experience limited generalization when applied to novel objects. In contrast, we propose an attribute-based robotic grasping approach that enables a robot to grasp an attributes-specified target object. The intuition of using attributes for grasping is that the grounded attributes can help transfer object recognition and grasping capabilities across different environments. Suffering from domain shift [4], a machine learning model trained with the data in one domain is subject to limited generalization when tested in another domain. In robotic grasping, the source of domain shifts includes novel objects, new environments, perception noises, etc. To mitigate the domain shift, domain adaptation methods [5] are widely used for model transfer. These adaptation methods, on the other hand, typically require the collection of

a large adaption dataset, which is costly, inefficient, and time-consuming. To efficiently transfer our pretrained attribute-based grasping model, we present two tailored adaptation methods. Both the two proposed adaptation methods are data-efficient, requiring minimal data collecting and labeling.

Compared to recognition-based robotic grasping (i.e., employing pretrained recognition modules), the challenges of attribute-based grasping are 1) mapping from workspace images and query text of the target to robot motions, 2) associating abstract attributes with raw pixels, 3) data labeling in target-driven grasping, and 4) data-efficient adaptation to unknown objects and new scenes. In this article, we design an architecture that consists of a multimodal encoder (i.e., encoding both visual and textual data) and an affordances decoder (i.e., predicting instance grasping affordances [6]). The key aspects of our system are as follows.

- 1) We design the deep grasping neural networks that represent 3-DOF grasp poses. After encoding and fusing visual-textual representations, the networks rotate the fused features to account for different grasping angles, and then predict pixel-wise instance grasping affordances.
- 2) To learn a multimodal metric space, we employ the equation of object persistence before and after grasping; the visual embedding of a grasped object should be equal to the textual embedding of that object.
- 3) Our model learns object attributes that generalize to new objects and scenes by only using basic objects (of various colors and shapes) in simulation.
- 4) With the pretrained attribute representations, our model supports efficient adaptation with minimal data. Adversarial adaptation regulates the image encoder with augmented data of unlabeled images, whereas one-grasp adaptation updates the end-to-end model with augmented data requiring only one successful grasp trial. Both adaption approaches are data-efficient, and they can be employed independently or in combination to improve instance grasping performance.

The deep grasping model in our approach is fully self-supervised through the interactions between the robot and objects. Fig. 1 presents an example of attribute-based robotic grasping, wherein our approach successfully grounds object attributes and accurately predicts grasping affordances for an attributes-specified target object.

In our prior work [1], we proposed 1) an end-to-end architecture for learning text-commanded robotic manipulation and 2) a method of self-supervising multimodal attribute embeddings through object grasping to facilitate quick adaptation. As an evolved paper, this article presents an in-depth study of adaptation in robotic manipulation and strives to improve the autonomy of robots by achieving self-supervision and self-adaptation. The pretrained model is self-supervised in a simulation that only uses basic objects of various colors and shapes. In our adaptation framework, we make use of autonomous robots to collect raw data for adaptation. We present three core technical contributions as follows.

- 1) *A sequential adaptation scheme.* We propose a robotic grasping adaptation framework that comprises two

stackable and data-efficient adaptation methods. The adversarial adaptation and one-grasp adaptation methods aim to comprehensively adapt the model for object recognition and grasping. Through data-efficient adaptation, the robot adeptly grasps challenging objects, eliminating the need for extensive data collection.

- 2) *Data-efficient augmentation methods.* We design data augmentation methods that only require unlabeled images of candidate objects for adversarial adaptation and one-grasp data of a target object for one-grasp adaptation.
- 3) *Evaluation and analysis of robot grasping.* We evaluate the grasping model in simulated and real-world scenes with various testing objects and domain gaps, which verifies the effectiveness of our grasping model. Furthermore, the ablative analysis of the data augmentation methods shows the efficiency of our approach.

With observations from an RGB-D camera, our robot system is designed to grasp a target object following the user command containing object attributes. To our best knowledge, this is the first work that explores object attributes to improve the generalization and adaptation of deep robotic grasping models. We believe that the adaptation framework not only enhances the overall performance, but also opens up new possibilities for solving the problem in target-driven robotic manipulation.

II. RELATED WORK

A. Instance Grasping

Though there are different taxonomies, the existing work of robotic grasping can be roughly divided according to approaches and tasks: 1) model-driven [7] and data-driven [8] approaches; 2) indiscriminate [9], [10] and instance grasping [2] tasks. Our approach is data-driven and focuses on instance grasping. Typical instance grasping pipelines assume a pretrained object recognition module (e.g., detection [2], segmentation [3] [11], template matching [12], and object representation [13], etc.), limiting the generalization for unknown objects and the scalability of grasping pipelines. Our model is end-to-end and exploits object attributes for generalization. Some recent research also proposes end-to-end learning methods for instance robotic grasping. Jang et al. [14] learned to predict the grasp configuration for an object class with a learning framework composed of object detection, classification, and grasp planning. In [15], CCAN, an attention-based network, learns to locate the target object and predict the corresponding grasp affordances given a query image. Compared to these methods, the main features of our work are two-fold. First, we collect a much smaller dataset of synthetic basic objects to learn generic attribute-based grasping. Moreover, our generic grasping model is capable of further adapting to new objects and domains. Second, our approach takes a description text of target attributes as a query command, which is more flexible when grasping a novel object.

B. Attribute-Based Methods

Object attributes are middle-level abstractions of object properties and generalizable across object categories [16]. Learning

object attributes has been widely studied in the tasks of object recognition [17], [18], [19], [20], while attribute-based robotic grasping has been much less explored, except for [21], [22]. Cohen et al. [21] developed a robotic system to pick up the target object corresponding to a description of attributes. Their approach minimizes the cosine similarity loss between visual and textual embeddings as well as predicts object attributes. However, they only show generalization across viewpoints but not object categories. In [22], the proposed Text2Pickup system uses object attributes to specify a target object and removes ambiguities in the user's command. They use mono-color blocks as training and testing objects but fail to show generalization to novel objects. In contrast, our work learns generic attribute-based robotic grasping (only using synthetic basic objects) and generalizes well to novel objects and real-world scenes.

C. Model Generalization

Model generalization is one of the most important challenges in robotic manipulation. To improve model generalization, various approaches to bridging domain gaps have been proposed. Domain randomization [23] is one frequently used method, which collects more diverse data by randomizing simulation settings. Some recent research [24], [25], [26] have applied domain randomization to improve the real-world generalization of a simulation-trained robot policy. We build a simulation environment and apply domain randomization during the pre-training of a generic model. In addition to domain randomization, we propose two adaption methods following the form of domain adaptation and few-shot learning. Domain adaptation [5], a subcategory of transfer learning [27], is used to reduce the domain shift between the source and target domain when the feature space is the same but the distributions are different. Inspired by adversarial domain adaptation [28], our approach learns a domain classifier and the image encoder learns domain-invariant features to confuse the classifier. We propose an object-level augmentation method to enrich the image dataset for adversarial training, increasing the generalization of the encoder to new domains. While there exist similar work, for example, Chen et al. [29] investigated domain adversarial training in their work, their approach focuses on updating the feature adaptor and the discriminator using unlabeled data, rather than updating the grasp synthesis model. In contrast, our grasping adaptation approach, consisting of unsupervised adversarial adaptation and supervised few-shot learning, jointly updates the grasping pipeline.

Few-shot learning [30] is the paradigm of learning from a small number of examples at test time. The key of metric-based few-shot learning method, one of the most popular categories, is to supervise the latent space and learn a versatile similarity function by metric loss [31], [32]. The supervised metric space supports fine-tuning using minimal adaptation data (also known as support set), and the similarity function generalizes to unknown test data [33], [34]. Motivated by the idea of few-shot learning methods, our approach first learns a joint metric space that encodes object attributes and then fine-tunes recognition and grasping of our model when testing on novel objects.

III. PROBLEM FORMULATION

The attribute-based robotic grasping problem in this article is formulated as follows:

Definition 1: Given a query text for a target object, the goal for the robot is to grasp the corresponding object that is placed in the cluttered workspace.

To handle the natural language that is diverse and unconstrained, we assume a language attribute parser, such as [35], and make the following assumption:

Assumption 1: The query text is parsed into the keywords of object attributes as an input to the robotic grasping model.

We consider color, shape, and category name attributes in this article, while the proposed approach is extensible to other attributes (e.g., texture, pose, and functionality, etc.). In order to make object recognition tractable, we have the following assumption regarding object placement.

Assumption 2: The objects are stably placed within the workspace, and there is no stacking between objects.

While we show robotic grasping as a manipulation example in this article, the proposed attribute-based learning methods should be, in principle, extensible to other robotic manipulation skills, such as suction, pushing, and placing.

IV. LEARNING ATTRIBUTE-BASED GRASPING

Object attributes are semantically meaningful features and serve as an intermediate representation for object recognition and manipulation. In this section, we propose an end-to-end neural network for attribute-based robotic grasping. The proposed model takes as input an image of visual observation and a text of target description to predict pixel-wise instance grasping affordances. To acquire a rich dataset for training, we build a simulation environment that allows domain randomization with diverse objects. In simulation, the model is pretrained to learn instance grasping and object attributes simultaneously.

A. Learning Grasping Affordances

We formulate attribute-based grasping as a mapping from pairs of workspace images and query text to target grasping affordances. The proposed visual-textual manipulation architecture assumes no prior linguistic or perceptual knowledge. It consists of two modules, a multimodal encoder and an affordances decoder, as illustrated in Fig. 2.

Multimodal Encoder: As shown in Fig. 1(a), our robot system uses an overhead RGB-D camera to capture the workspace. The RGB-D image is projected into a 3-D point cloud and then orthographically back-projected in the gravity direction to construct a heightmap image v_{pre} of RGB and depth channel. To specify an object in the image as the grasping target, we give a text command t composed of color and/or shape attributes, e.g., “red cuboid.” The workspace image v_{pre} and query text t are the input to visual spatial encoder $\phi_{v,\text{spa}}$ and text encoder ϕ_t , respectively. We use the ImageNet-pretrained [36] ResNet-18 [37] backbone as our image encoder $\phi_{v,\text{spa}}$. We replace the first convolutional layer of the ResNet backbone with a four-channel convolutional

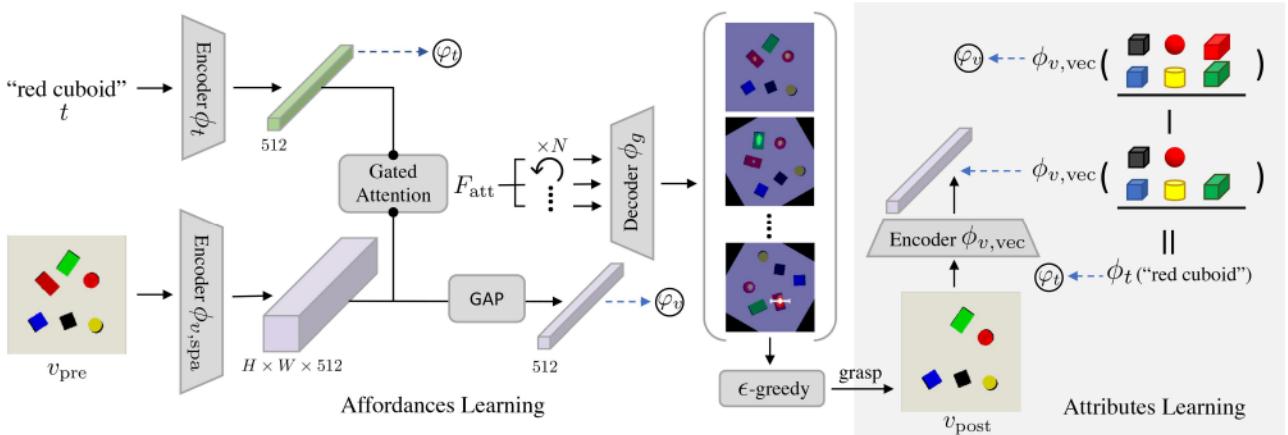


Fig. 2. *Overview of affordances and attribute learning.* The workspace image and query text are encoded separately and fused using gated-attention. The fusion matrix F_{att} is rotated by N orientations for different grasping angles and then fed into the grasping affordances decoder. The decoder learns to predict pixel-wise scores of target grasping success, and we run the ϵ -greedy grasping policy and obtain the image v_{post} after grasping. By utilizing the equation of object persistence before and after grasping, we learn a metric space where multimodal embedding vectors corresponding to similar attributes are encouraged to be closer. Note we denote the combination of $\phi_{v,\text{spa}}$ and GAP as $\phi_{v,\text{vec}}$, which encodes images to vectors.

layer to match the RGB-D heightmap input. The encoder encodes the RGB and depth observation into 3D visual matrix $\varphi_{v,\text{spa}} \in \mathbb{R}^{H \times W \times 512}$. The text encoder ϕ_t is a deep averaging network [38] represented by three fully connected layers and interleaved ReLU [39] activation functions. We first map each token in a sentence text to an embeddings vector of 128 dimension. The mean token embeddings (i.e., continuous bag-of-words [40] model) of the text are input to the three-layer MLP text encoder to produce a text vector $\varphi_t \in \mathbb{R}^{512}$. The visual matrix $\varphi_{v,\text{spa}}$ and the text vector φ_t are then fused by the gated-attention mechanism [41]: each element of φ_t is repeated and expanded to an $H \times W$ matrix to match the dimension of $\varphi_{v,\text{spa}}$. The expanded matrix is multiplied element-wise with $\varphi_{v,\text{spa}}$ to produce a fusion matrix F_{att} . The gated-attention unit is designed to gate certain pixels in the visual feature matrix matching to the text vector, resulting in the fusion matrix containing the visual features selected by the query text. By this means, we can detect different attributes of the objects in the image, such as color and shape.

Affordances Decoder: Grasping affordances decoder ϕ_g is a fully convolutional residual network [37], [42] interleaved with spatial bilinear $4 \times$ upsampling and ended with the sigmoid function. The decoder takes as input the fusion matrix F_{att} and outputs a unit-ranged map Q_g with the same size and resolution as the input image v_{pre} . Each value of a pixel $q_i \in Q_g$ represents the predicted score of target grasping success when executing a top-down grasp at the corresponding 3-D location with a parallel-jaw gripper oriented horizontally concerning the map Q_g . The grasping primitive is parameterized by a 3-D location and an angle. To examine different grasping angles, we rotate the input F_{att} by $N = 6$ (multiples of 30°) orientations before feeding into the decoder, which predicts pixel-wise scores of horizontal grasps within the rotated heightmaps. The pixel with the highest score among all the N maps determines the parameters (i.e., location and angle) for the grasping primitive to be executed. As in Fig. 2, our model predicts accurate target grasping location and valid (e.g., the selected angles for the red cuboid) target grasping angle.

The motion loss $\mathcal{L}_{\text{grasp}}$, which supervises the entire encoder-decoder networks, is the error from predictions of grasping affordances:

$$\mathcal{L}_{\text{grasp}} = \sum_{i=1}^{N_s} \left[(q_e - \bar{q}_e)^2 + \lambda_M \sum_{i \in M} q_i^2 \right] \quad (1)$$

where N_s is the size of the dataset that is collected in simulation, q_e is the grasping score in Q_g at the executed location, and \bar{q}_e is the ground-truth label (see Section IV-C). The second term ensures lower grasping scores for the pixels in background mask M (obtained from the depth image) with weight λ_M [43], and q_i is the grasping score of a background pixel.

B. Learning Multimodal Attributes

To learn generic object attributes, we perform multimodal attributes learning, where visual or textual embedding vectors corresponding to similar attributes are encouraged to be closer in the latent space. Inspired by [13], we take advantage of the object persistence: the embedding difference of the scene before and after grasping is enforced closer to the representation of the grasped object. During data collection, we record image-text data $(v_{\text{pre}}, v_{\text{post}}, t)$, where v_{pre} and v_{post} are the workspace image before and after grasping, respectively, and t is the query text that describes attributes of the grasped object.

We add one layer of global average pooling (GAP) [44], [45] at the end of the encoder $\phi_{v,\text{spa}}$ and denote the network as visual vector encoder $\phi_{v,\text{vec}}$. The output from $\phi_{v,\text{vec}}$ is a visual embedding vector that represents the average of scene features and has the same dimension of $\phi_t(t)$. We express the logic of the object persistence as an arithmetic constraint on visual and textual vectors such that $(\phi_{v,\text{vec}}(v_{\text{pre}}) - \phi_{v,\text{vec}}(v_{\text{post}}))$ is close to $\phi_t(t)$. We use the triplet loss [46] to approximate the constraint, and the set of triplets \mathcal{T} is defined as

$$\mathcal{T} = \left\{ (f_i, f_i^+, f_i^-) \mid s(a_{f_i}, a_{f_i^+}) > s(a_{f_i}, a_{f_i^-}) \right\} \quad (2)$$

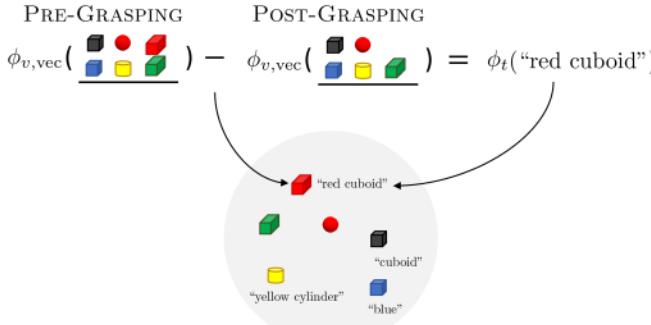


Fig. 3. *Multimodal feature space* supervised by the equation of object persistence (5). The image and text encoders are trained to produce consistent embeddings, where feature vectors corresponding to similar attributes are encouraged to be closer.

where f_i , f_i^+ , and f_i^- are random samples from the pool of vectors $(\phi_{v, \text{vec}}(v_{\text{pre}}) - \phi_{v, \text{vec}}(v_{\text{post}}))$ and $\phi_t(t)$, and a_f is an n -dimensional attribute label vector corresponding to the feature vector f (e.g., color, shape, and category name, etc.). Function $s(\cdot, \cdot)$ is an attribute similarity function that evaluates the similarity between two attribute label vectors

$$s(a_1, a_2) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(a_1^i, a_2^i) \quad (3)$$

$$\mathbb{1}(a_1^i, a_2^i) = \begin{cases} 1 & \text{if } a_1^i = a_2^i \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where a^i denotes the i th element of the label vector a , and the indicator function $\mathbb{1}(\cdot, \cdot)$ evaluates the element-wise similarity. Note that 0 indicates null attribute meaning no attribute is specified in the label. As an example, suppose we have the dictionary $\text{dict} = \{\text{"eos"} : 0, \text{"red"} : 1, \text{"black"} : 2, \text{"yellow"} : 3, \text{"cylinder"} : 4, \text{"cube"} : 5\}$; then, “red cylinder” can be represented as a label vector $a_{f_0} = [1, 4]$, and “red cube” can be represented as $a_{f_1} = [1, 5]$. The similarity between the two label vectors is computed using (3) such that $s(a_{f_0}, a_{f_1}) = s([1, 4], [1, 5]) = 0.5$. In addition, when “red” and “black” are used without any additional attribute description, they are mapped to the vectors $[1, 0]$ and $[2, 0]$, respectively. In this case, the similarity between the two labels is derived as $s([1, 0], [2, 0]) = 0$. With the triplets of embedding vectors, multimodal metric loss $\mathcal{L}_{\text{attr}}$ is defined as

$$\mathcal{L}_{\text{attr}}(\mathcal{T}) = \sum_{i=1}^{|\mathcal{T}|} \max(\|f_i - f_i^+\|^2 - \|f_i - f_i^-\|^2 + \alpha, 0) \quad (5)$$

where α is a hyperparameter that controls the margin between positive and negative pairs. By encoding workspace images and query text into a joint metric space and supervising the embeddings through the equation of object persistence (as shown in Fig. 3), we learn generic attributes that are consistent across object categories, as discussed in Section VII-A.

C. Data Collection and Training

To achieve self-supervision, we create a simulation environment in which objects are identified and grasped based on



Fig. 4. *Examples of basic objects.* Synthetic objects of various colors and shapes are used for learning object attributes and grasping affordances. To ensure shape attribute learning, we include objects having random textures.

Algorithm 1: Online Data Collection.

Initialize bounded buffer B

Notations: ϵ -greedy policy π_ϵ , our model ϕ , image v , text t , mask M , action a , and label \bar{q}_e

```

1: while collecting data do
2:   reset the simulation and randomly drop basic objects
3:   get image  $v_{\text{pre}}$  and randomly choose a command  $t$ 
4:   execute action  $a \leftarrow \pi_\epsilon(\phi; v_{\text{pre}}, t, \epsilon)$ 
5:   label  $\bar{q}_e$  according to grasping results
6:   save  $v_{\text{pre}}, t, M, a$  and  $\bar{q}_e$  into  $B$ 
7:   if successful grasp then
8:     save  $v_{\text{post}}$  into  $B$  with HER
9:   end if
10:  sample a batch from  $B$  to train the model
11: end while

```

a description of semantic attributes. We collect training data in simulation with the following procedure, as summarized in Algorithm 1. Several objects are randomly dropped into the workspace in front of the robot. Given a workspace image and a query text, the robot learns to grasp a target under ϵ -greedy exploration [47] ($\epsilon=0$ during testing, i.e., an argmax policy). We save the workspace images, query text, background masks, executed actions, and results into a bounded buffer. The ground-truth labels are automatically generated for learning grasping affordances. The label \bar{q}_e in (1) is assigned as the attribute similarity in (3) between the query text and the grasped object (0 if no object grasped). We also save the workspace image after a successful grasping for learning object attributes. To deal with sparse rewards in target-driven grasping, the hindsight experience replay (HER) technique [48] is applied. If a nontarget is grasped, we add the additional positive sample by relabeling the target text. Fig. 4 shows the basic objects of various colors (red, green, blue, yellow, and black) and shapes (cube, cuboid, cylinder, and sphere) used in our simulation. We choose these colors and shapes because they are foundational for common objects in daily life. To enrich the distribution of training data, we perturb color RGB values, randomize sizes and heights of the objects, and randomize textures of the workspace. Using the domain randomization techniques [24], we can generate a number of randomized properties in simulation and achieve a model with better generalization. At every iteration, we sample a batch from the buffer and run one off-policy training. The



Fig. 5. *Testing objects* in simulation and the real world. We use the testing objects that share similar attributes with the training objects. See Appendix for more details. (a) Simulated novel objects. (b) Real-world objects.

training loss is defined as

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{grasp}} + \lambda_a \mathcal{L}_{\text{attr}} \quad (6)$$

by combining both motion loss $\mathcal{L}_{\text{grasp}}$ (1) and metric loss $\mathcal{L}_{\text{attr}}$ (5). We train the proposed model using stochastic gradient descent with a learning rate of 10^{-4} , momentum of 0.9, and weight decay of 2×10^{-5} for 5 k iterations. Each training iteration involves capturing data, computing a forward pass, executing an action, and backpropagating. After collecting a dataset of 5 k samples, we replay the entire data for 100 epochs.

V. DATA-EFFICIENT ADAPTATION

Due to the high cost of collecting data on real robots, we often choose to train robotic models in a simulator. However, the domain gap between the source domain (e.g., simulation, trained objects) and the target domain (e.g., the real world, novel objects) frequently leads to the failure of the learned models. We propose to, in addition to randomizing the source domain in Section IV-C, adapt our learned model using data from the target domain to further alleviate the domain shifts. One typical adaptation approach is fine-tuning the pretrained model. However, the fine-tuning methods remain expensive in terms of data usage. In this section, as shown in Fig. 6, we propose two data-efficient adaptation methods: 1) adversarial adaptation, which adapts the image encoder using unlabeled images, and 2) one-grasp adaptation, which updates the end-to-end model using one grasp trial. The two adaptation methods can be either used independently or in combination for performance improvement.

A. Adversarial Adaptation

Despite that our generic model trained using the simulated basic objects shows good generalization (see Section VII-B), the visual feature shifts (e.g., objects, lighting conditions, and scene configurations, etc.) are inevitable. As a result, the image encoder is likely to produce out-of-distribution visual embeddings, leading to the failure of the grasping model. To reduce the influence of the domain shifts, we propose to use adversarial adaptation [28] to learn domain-invariant visual features that are transferable across different domains. In our problem setup, the simulated basic objects constitute the source domain, and our goal is to transfer the learned model to a target domain that is

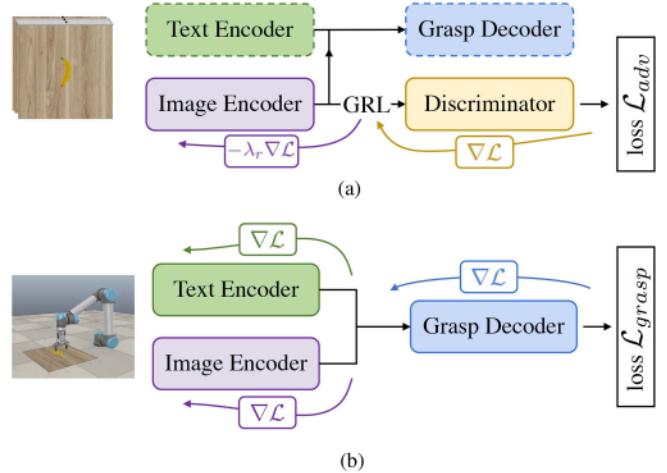


Fig. 6. *Overview of the proposed adaptation schemes*. In adversarial adaptation (top), the adversarial loss \mathcal{L}_{adv} regulates the image encoder using augmented data of unlabeled images to learn domain-invariant features for the decoder. The dotted boxes indicate no gradient flows corresponding to \mathcal{L}_{adv} (there are gradients from $\mathcal{L}_{\text{train}}$ though). One-grasp adaptation updates the end-to-end model through $\mathcal{L}_{\text{grasp}}$ using augmented data from one grasp trial. (a) Adversarial adaptation. (b) One-grasp adaptation.

prone to domain shifts. As shown in Fig. 6(a), adversarial adaptation regularizes the weights of the image encoder ϕ_v by enforcing a two-player game similar to the generative adversarial network (GAN) [49]. A domain classifier (i.e., discriminator) learns to distinguish between two domains, while the image encoder learns to fool the domain classifier by learning domain-invariant features. To achieve adversarial training, we connect the encoder and the discriminator via a gradient reversal layer (GRL) [50] that has reverse forward and back-propagation schemes. The GRL \mathcal{R} is an identity mapping during forward-propagation but reverses the sign of the gradients during back-propagation

$$\mathcal{R}(x) = x \quad (7)$$

$$\frac{d\mathcal{R}}{dx} = -\lambda_r I \quad (8)$$

where I is an identity matrix, and λ_r is a positive constant.

During adaptation, images from the two different domains are fed into the image encoder. The domain classifier f_d takes as input the encoded features, and is trained to predict which domain the feature is from, by maximizing the binary cross-entropy \mathcal{H}_d . Using the source domain dataset $\mathcal{D}_s = \{v_s, t_s, y_s\}^{N_s}$ ($\{\text{image, text, label}\}$ collected in Algorithm 1) and assuming a target domain dataset $\mathcal{D}_t = \{v_t\}^{N_t}$, the model is updated to be optimal on the training loss $\mathcal{L}_{\text{train}}$ (6) under the regularization of the adversarial loss \mathcal{L}_{adv}

$$\mathcal{L}_{\text{adv}} = - \sum_{v_i \in \mathcal{D}_s \cup \mathcal{D}_t} \mathcal{H}_d(f_d(\mathcal{R}(\phi_{v, \text{vec}}(v_i))), d_i) \quad (9)$$

$$\mathcal{L}_{\text{advadp}} = \mathcal{L}_{\text{train}}(v_s, t_s, y_s) + \mathcal{L}_{\text{adv}}(v_t) \quad (10)$$

where $\mathcal{L}_{\text{advadp}}$ is the overall loss for adversarial adaptation, $d_i \in \{0, 1\}$ are the binary domain label for each input v_i . Note that the reversal layer \mathcal{R} is augmented between the classifier f_d and the encoder $\phi_{v, \text{vec}}$ and updates them in reversal directions.

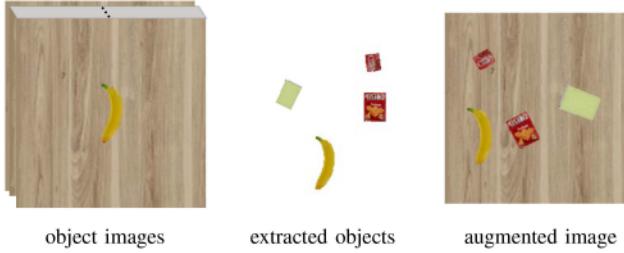


Fig. 7. *Object-level data augmentation*. We extract objects from the images of a single object, apply the augmentation methods (e.g., shifting, rotation), and assemble them on a background to generate diverse adaptation data for adversarial adaptation.

The target dataset \mathcal{D}_t for the new domain is the prerequisite to performing adversarial adaptation. As shown in Fig. 7, we propose a object-level augmentation (*ObjectAug*) approach instead of collecting data from a vast number of configurations. Since the grasping label is not required in \mathcal{D}_t , synthetic generation of a large image dataset would be more efficient. We begin by collecting RGB-D images of all conceivable objects in the target domain. Using the object mask acquired from the depth channel, we extract each object individually and randomize them with the single-object augmentation methods commonly used (e.g., scaling, flipping, and rotation). To generate an augmented RGB-D image, the augmented objects are randomly sampled and shifted before being overlaid with a background image. We also perform IoU threshold verification to avoid dense overlapping. To simulate varying conditions of target domains, we can apply the visual jitter technique discussed in Section VI-A to obtain more diverse data. Given the ease with which unlabeled images may be acquired (e.g., the internet, image collection), generating a target dataset that is of the same magnitude as the source dataset is rather efficient.

B. One-Grasp Adaptation

By learning domain-invariant features, the adversarial adaptation technique in Section V-A improves model generalization using unlabeled images of the target domain. However, the adversarial loss uses unlabeled images to only update the image encoder and leave the text encoder and the grasping affordance decoder unadapted. When deploying in a new domain, end-to-end model fine-tuning is often necessary, but this comes at the cost of a large dataset covering all potential testing object configurations. To further adapt to novel objects and new scenes in a data-efficient manner, we present a one-grasp adaptation scheme [see Fig. 6(b)] that only requires one successful grasp of a novel object. The inductive bias of object attributes in Section IV-B is the key to adaptation in this limited-data regime. If similar objects are enforced closer in the embedding space, the adaptation distance for a novel object is likely to be shorter [51].

The proposed one-grasp adaptation method improves the model performance on a novel target object at the cost of only one grasp. The adaptation data are collected with the following one-grasp data augmentation (*OneGraspAug*) procedure. We place the object solely in the workspace and run the generic model to collect one successful grasp. The setting

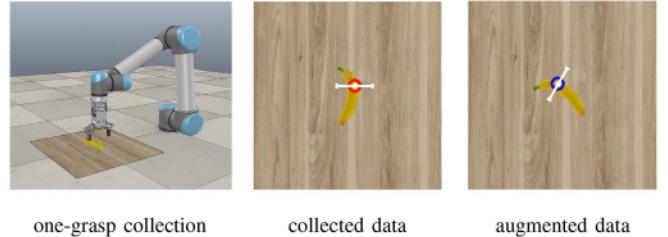


Fig. 8. *One-grasp data augmentation*. We place the target object solely on the workspace and collect one successful grasp. To enrich the distribution of grasping angles, we rotate the collected data and synthesize a few samples for one-grasp adaptation. One rotation example of the augmented data is visualized.

of a sole object facilitates grasping and avoids combinatorial object arrangements. Because convolutional neural networks are not rotation-invariant by design, we also augment the grasp data by rotating with various orientations to achieve rotation-invariance [52], [53], i.e., the ability to recognize and grasp an object regardless of its orientation. As shown in Fig. 8, we rotate the collected image and action execution to have rotated versions of the collected data. In the adaptation stage, we add the category name of the object as an additional token to the query text, e.g., “apple, red sphere” for the testing object apple. The token embedding of the object name is initialized properly to keep the embedding vector of the query text unchanged. The addition of the object name allows for a more specific grasping instruction and distinguishing from similar objects via adaptation. By optimizing over motion loss $\mathcal{L}_{\text{grasp}}$ in (1), we jointly fine-tune the recognition and grasping of our model for unknown objects and scenes. As delineated in Section VII-C, the adapted model outputs higher affordances on the target objects that are not seen and grasped before.

VI. SYSTEM IMPLEMENTATION

A. Simulation Environment

We use CoppeliaSim [54] to build our simulation environment. The simulation setup includes a UR5 robot arm and an RG2 gripper, with Bullet Physics 2.83 for dynamics and CoppeliaSim’s internal inverse kinematics module for robot motion planning. We simulate a statically mounted overhead 3-D camera in the environment from which perception data is captured. The camera renders RGB-D images with a 640×480 pixels resolution using OpenGL. We use various basic and novel objects for training and testing in the simulation. As shown in Fig. 4, the basic objects consist of 36 different 3-D toy blocks, whose shapes and colors are randomly chosen during experiments. We collect 34 different 3-D household objects from the YCB dataset [55] or the 3-D mesh library SketchUp [56], as shown in Fig. 5(a). To produce more diverse simulation configurations, the simulation environment supports several domain randomization techniques, including background randomization, color jitter (i.e., randomly changing the brightness, contrast, and saturation of the color channel), and depth jitter (i.e., adding Gaussian noise to the depth channel).

B. Real-Robot System

Fig. 1(a) shows our real-world setup that includes a Franka Emika Panda robot arm with a FESTO DHAS soft gripper and a hand-mounted Intel RealSense D415 camera overlooking a tabletop scene. We use the soft fingers because they are more suitable for grasping the objects in our experiments and are similarly compliant to the RG2 fingers. For perception data, RGB-D images of resolution 640×480 are captured from the RGB-D camera, statically mounted on the robot arm. The camera is localized with respect to the robot base using the automatic calibration procedure of ViSP [57], during which the camera tracks the location of a checkerboard pattern taped on the table. For a given pose, the robot follows the corresponding trajectory generated with MoveIt [58] in open-loop. The entire system is implemented under the robot operating system framework and runs on a PC workstation with an Intel i7-8700 CPU and an NVIDIA 1080Ti GPU. Objects vary throughout tests, with a collection of 20+ different household objects being used to test model generalization to novel objects, as shown in Fig. 5(b).

VII. EXPERIMENTS

We propose training with simulated basic objects first to have a generic model and then adapting it to novel objects and real-world scenes. In the experiments, we first analyze the structured metric space of our generic model and show the consistency between attention and grasping maps. Next, we evaluate the instance grasping performance of the generic model and show its modest generalization even before adaptation. Then, we adapt the model using the proposed adversarial and one-grasp adaptation methods and test the grasping models after adaptation. Finally, we run a series of ablation studies to investigate the two adaptation methods. The goals of the experiments are four-fold:

- 1) to show the effectiveness of multimodal attribute learning for instance robotic grasping,
- 2) to evaluate our attribute-based grasping system in both simulated and real-world settings,
- 3) to evaluate the proposed adversarial and one-grasp adaptation methods, and
- 4) to show the importance of the proposed data augmentation methods for grasping adaptation.

A. Multimodal Attention Analysis

By embedding workspace images and query text into a joint metric space, the multimodal encoder (ϕ_v and ϕ_t), supervised by metric loss $\mathcal{L}_{\text{attr}}$ and motion loss $\mathcal{L}_{\text{grasp}}$, learns attending to text-correlated visual features. We visualize what our model “sees” by computing the dot product of text vector φ_t with each pixel of the visual matrix $\varphi_{v,\text{spa}}$. This computation obtains an attention heatmap over the image, which refers to the similarity between the query text and each pixel’s receptive field [see Fig. 9(a)]. We quantitate the attention of our model and report its attention localization performance in Table I (see *Ours-Attention*). *Evaluation metrics*: An attention localization is considered correct only if the maximum value in the attention heatmap lies on the target object.

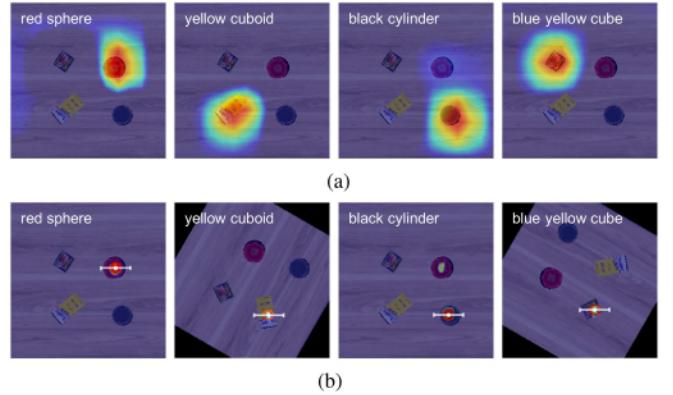


Fig. 9. *Visualization of attention and grasping*. (a) shows attention of our encoder, and (b) shows heatmaps of grasping affordances for different targets (described by the query text). The maps of attention and grasping are consistent even for the novel objects. (a) Encoder’s attention. (b) Grasping affordances.

TABLE I
TARGET RECOGNITION ACCURACY (%)

Method	sim basic	sim novel	real novel
ClassIndis	100.0	56.3	35.7
EncoderIndis	95.8	71.5	57.1
NoMetric	93.4	70.4	54.0
Ours-Attention	95.3	74.5	70.2
AttrID	96.8	79.1	70.6
Generic (Ours)	100.0	79.7	71.8

The bold entities are typically used to emphasize the best results across the methods.

Ours-Attention (in Table I) performs target localization at a 74.5% accuracy on simulated novel objects and a 70.2% accuracy on real-world objects, without any localization supervision provided. In summary, our multimodal embeddings demonstrate a consistent pattern across object categories and scenes. Though the localization results are not directly used for grasping, the consistent embeddings facilitate learning, generalization, and adaptation of our grasping model, as shown in Fig. 9 and discussed in the following subsections.

B. Generic Instance Grasping

We compare the instance grasping performance of our generic model with the following baselines.

- 1) *Indiscriminate* is an indiscriminate grasping version of our approach and composed of a visual spatial encoder $\phi_{v,\text{spa}}$ and a grasping affordances decoder ϕ_g . We collect a dataset of binary indiscriminate grasping labels and train *Indiscriminate* using $\mathcal{L}_{\text{grasp}}$ in (1).
- 2) *ClassIndis* extends *Indiscriminate* with an attributes classifier that is trained to predict color and shape attributes on cropped object images. We filter the grasping maps from *Indiscriminate* using the mask of a target recognized by the classifier.
- 3) *EncoderIndis* is similar to [21] and is another extension of *Indiscriminate*, which leverages a multimodal encoder ($\phi_{v,\text{vec}}$ and ϕ_t in Section IV-A) for text template matching.

TABLE II
INSTANCE GRASPING SUCCESS RATE (%)

Method	sim basic	sim novel	real novel
Indiscriminate	27.2	22.8	13.5
ClassIndis	91.6	51.0	32.9
EncoderIndis	89.1	63.9	52.0
NoMetric	90.5	62.7	50.0
AttrID	90.8	68.7	60.0
Generic (Ours)	98.4	72.1	63.1

The bold entities are typically used to emphasize the best results across the methods.

The encoder is trained using $\mathcal{L}_{\text{attr}}$ in (5) to evaluate the similarity between each cropped object image and query text. During training, we also include attributes classification as an axillary task.

- 4) *AttrID* is for an ablation study of our text encoder ϕ_t . The only difference between *AttrID* and the proposed method is that *AttrID* takes the attribute shape and color ID one-hot encoding as the system input, but our generic model uses Word2Vec continuous bag-of-words (CBOW) model to convert the texts into vector inputs. During training, we use both the motion loss and the metric loss to update the model.
- 5) *NoMetric* is for an ablation study of multimodal metric loss. We simply remove the metric loss on the basis of our approach during its training.

Evaluation metrics: These methods have different target recognition schemes: *ClassIndis* and *EncoderIndis* recognize a target by classification and text template matching, respectively; *NoMetric*, *AttrID* and *Ours* are end-to-end. We report their target recognition performance (in addition to instance grasping performance, as in the next paragraph). A target localization is correct only if the predicted grasping location lies on the target object. The instance grasping success rate is defined as $\frac{\# \text{ of successful grasps on correct target}}{\# \text{ of total grasps}}$. In each testing scene, we only execute grasping once.

We evaluate the methods on both simulated basic (sim basic) and simulated novel (sim novel) objects in simulation, where there are 1200 test cases for the basic objects (see Fig. 4) and 3400 test cases for the 34 novel objects (see Fig. 5(a), mostly from the YCB dataset [55]). We assume the objects are placed right-side up to be stable while their 4-D pose (3-D position and a yaw angle) can vary arbitrarily. For each testing object, we prechoose a query text that best describes its color and/or shape. In each test case, four objects are randomly sampled and placed in the workspace, except avoiding any two objects with the same attributes. The robot is required to grasp the target queried by an attribute text. We report the results of target recognition in Table I and the results of instance grasping in Table II.

Overall, our approach outperforms the baselines remarkably (in both recognition and grasping) and achieves a 98.4% grasping success rate on the simulated basic objects and an 72.1% grasping success rate on the simulated novel objects. *ClassIndis* extends *Indiscriminate* that is well trained in target-agnostic tasks and performs well on the basic objects, but the attributes classifier generalizes poorly. *EncoderIndis* utilizes a more generalizable recognition module and performs better on the novel

objects. However, *EncoderIndis* fails to reach optimality because its separately trained recognition and grasping modules have different training objectives from instance grasping. Despite training the recognition and grasping modules simultaneously, *AttrID* using sparse attribute one-hot IDs as a substitute for text inputs yields lower recognition accuracy and grasping success rate compared to *Ours*. As an ablation study, the performance gap between *NoMetric* and *Ours* shows the effectiveness of multimodal metric loss, which supervises the joint latent space to produce consistent embeddings, as discussed in Section VII-A. Our approach successfully learns object attributes that generalize well to novel objects, as shown in Fig. 9.

We further evaluate our approach and the baselines on the real robot before any adaptation (see Tables I and II). Fig. 5(b) shows 21 testing objects of various colors and shapes used in our real-robot experiments. The robot is tasked to grasp the target within a combination of six objects placed on the table. We use the same 21 object combinations that are randomly generated and repeat each combination twice, resulting in a total of 252 grasping trials for each method. Overall, the grasping performance of all the methods decreases due to the domain gap. However, our approach shows the best generalization and achieves a 63.1% grasping success rate, before adaptation, in the real-world scenes.

C. Adapted Instance Grasping

The generic model in Section VII-B infers the object closest to the query text as the target. Overall, our generic model demonstrates good generalization despite the gaps in the testing scenes. Specifically, these gaps are 1) RGB values of the testing objects deviate from training ranges, 2) some testing objects are multicolored, 3) shape and size differences between the testing objects and the training objects, and 4) depth noises in the real world causing imperfect object shapes. To account for the gaps, we further adapt our generic model to increase instance recognition and grasping performance.

We first collect one successful grasp of a solely placed target object and then augment the collected data by rotating with additional $N - 1$ angles, as discussed in Section V-B and shown in Fig. 8. The compared methods that are adapted with the same adaptation data are as follows.

- 1) *ClassIndis* updates its attributes classifier for a better recognition accuracy on the adaptation data.
- 2) *EncoderIndis* minimizes the latent distance between cropped target images and query text to improve text template matching.
- 3) *NoMetric* takes as input images and text, and minimizes motion loss on the adaptation data.
- 4) *One-Grasp* is our prior work [1] which updates the encoder-decoder in an end-to-end manner.
- 5) *AttrID-One-Grasp* uses the same adaptation method and data as *One-Grasp* baseline on *AttrID* from Section VII-B.

We also update *Indiscriminative* grasping in *ClassIndis* and *EncoderIndis*. We keep the experimental setup the same with Section VII-B and evaluate the instance grasping performance of the adapted models. We report the adapted instance grasping

TABLE III
ADAPTED INSTANCE GRASPING (%)

Method	sim novel	real novel
ClassIndis	56.8	37.3
EncoderIndis	72.0	60.3
NoMetric	68.1	53.6
One-Grasp ([1])	83.7	76.6
AttrID-One-Grasp	79.7	73.0
Adversarial+One-Grasp (Ours)	86.0	81.7

The bold entities are typically used to emphasize the best results across the methods.

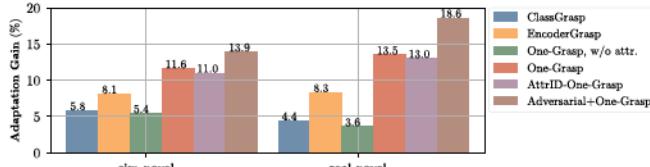


Fig. 10. *Adaptation gains* of instance grasping made by different adaptation methods. The plot shows the effectiveness of our *One-Grasp* and *Adversarial+One-Grasp* adaptation approach, which achieve significant adaptation gains.

success rate in Table III and the adaptation gains in Fig. 10. The attributes classifier in *ClassIndis* suffers from insufficient adaptation data, limiting its target recognition and adaptation performance. While *EncoderIndis* minimizes embedding distances in its latent space and shows better performance, it is still worse than *One-Grasp*. By fine-tuning over the structured metric space, *One-Grasp* updates the end-to-end model and improves target recognition and grasping jointly. At the cost of minimal data collection, *One-Grasp* achieves an 83.7% grasping success rate on the simulated novel objects and an 76.6% grasping success rate on the real objects, which shows the significant adaptation gains. On the contrary, the unstructured latent space in *NoMetric* limits its adaptation, demonstrating the importance of attributes learning for grasping affordances learning. The difference in adaptation performance between *NoMetric* and *One-Grasp* demonstrates the significance of the regulated feature space in adaptation. Furthermore, the substantial adaptation gain observed in *AttrID* validates the applicability of our adaptation method with sparse feature inputs.

As an improvement, we propose applying *Adversarial* adaptation on the image encoder to learn domain-invariant features before *One-Grasp* adaptation. The image encoder's domain invariancy results in superior transferring performance for *Adversarial+One-Grasp*: an instance grasping success rate of 86.0% in the domain of sim novel and 81.7% in the domain of real-world novel (real novel). The qualitative results in Fig. 11 suggest the efficacy of the two adaptation methods: 1) both *Adversarial* adaptation and *One-Grasp* adaptation increase the recognition and grasping performance of the models, and 2) *Adversarial* adaptation minimizes grasping noises around nontarget objects (by reducing domain feature changes), while *One-Grasp* adaptation can rectify recognition and grasping errors through end-to-end updates. The more compact and centered contour in Fig. 11(b) could be explained by the hypothesis that domain-invariant features improve the output

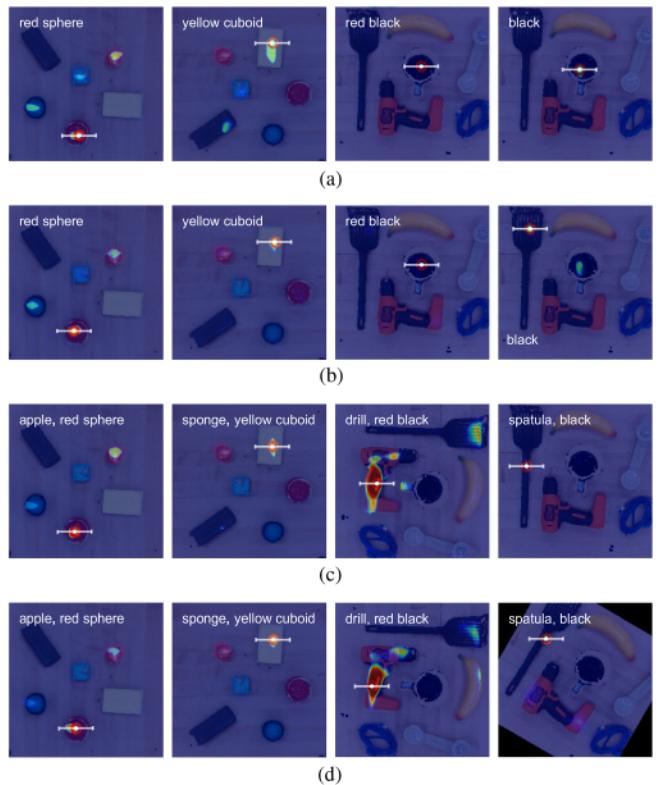


Fig. 11. *Visualization of grasping maps before and after adaptation*. (a) shows the grasping affordances from the generic model trained only with simulated basic objects, and (b) to (d) show the affordances from the adapted models after *Adversarial*, *One-Grasp*, and *Adversarial+One-Grasp* adaptation, respectively. Our adaptation methods, which require only a limited amount of adaptation data, effectively enhance model performance. (a) *Generic* model before adaptation. (b) *Adversarial* adaptation. (c) *One-Grasp* adaptation. (d) *Adversarial+One-Grasp* adaptation.

consistency of the encoder across domains. The corrected target recognition in Fig. 11(c), on the other hand, is attributed to the *One-Grasp* adaptation which effectively shifts the affordances from irrelevant objects to the target object through the end-to-end model updates. Another noteworthy finding is that the combined adaptation *Adversarial+One-Grasp* appears to benefit from both *Adversarial* adaptation and *One-Grasp* adaptation, as their focuses are complementary.

D. Comparison With a Foundation Model

In addition to the instance grasping experiments, we perform another quantitative evaluation of the adapted model on sim novel objects. For comparison, we utilize the CLIP model [59], a recently prevailing multimodal (text and image) foundation model. CLIP aligns language and image features through training with millions of text-image pairs and is widely acclaimed for its robust generalization capability across various objects. To evaluate the performance of CLIP, we segment and crop randomly placed objects in the workspace. For each object crop and its attribute text description, the CLIP model calculates a matching score, which is then used to construct the confusion matrix shown in Fig. 12(a). The pairwise matching scores reflect the similarity between the object crops and the text inputs. As

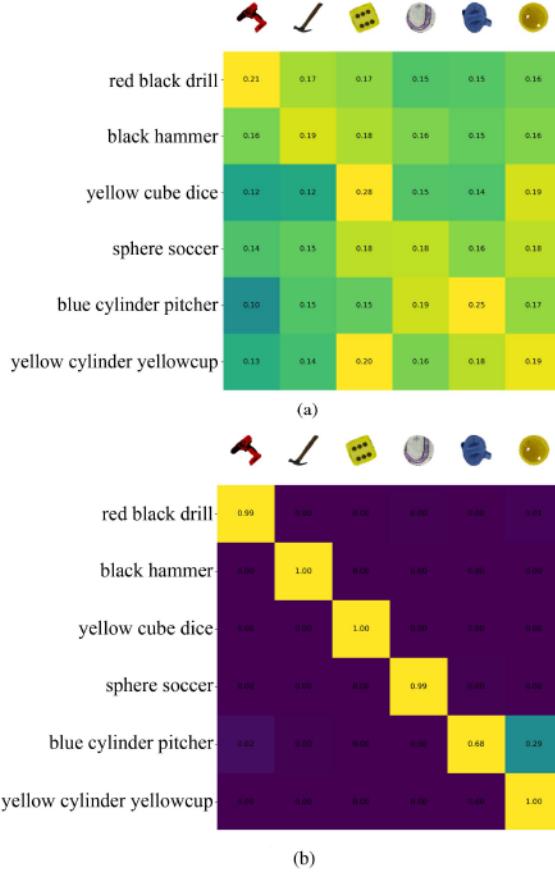


Fig. 12. *Visualization of confusion matrices.* The comparison between the two confusion matrices shows the effectiveness of our adaptation method trained with a minimum amount of data. (a) CLIP matching scores. (b) *Adversarial+One-Grasp* scores.

for our *Adversarial+One-Grasp* model, we compute another confusion matrix shown in Fig. 12(b) by collecting the highest affordance value, under each target attribute description, from affordance maps Q_g within each object’s segmentation mask. While CLIP matching uses cropped images of single objects, *Adversarial+One-Grasp* is tested using workspace images of multiple objects, which is a more natural but also more challenging setting. Fig. 12 shows that *Adversarial+One-Grasp* achieves more accurate grounding of the correct target objects. In contrast, the matching scores produced by the CLIP model often lack discrimination, leading to some misclassifications (e.g., “sphere soccer” and “yellow cylinder yellowcup”). This comparison highlights the limitations of zero-shot generalization in a foundation model and showcases the effectiveness of our adaptation method.

E. Ablative Analysis of Adaptation

The proposed adaptation approaches efficiently improve the instance grasping performance of our model. As discussed in Section VII-C, the finding that the approaches have complementary adaptation focuses leads to one of the major features: the two adaptation methods can be employed individually or in combination, depending on the availability of adaptation data.

To investigate their independent and combinative performance, we conduct an ablation study to compare the grasping models as follows.

- 1) *Generic* is the baseline generic model obtained in Section VII-B before any adaptation.
- 2) *Adversarial* adapts the generic model to learn domain-invariant features using a large augmented data, as discussed in Section V-A.
- 3) *One-Grasp* adapts the generic model using one grasping trial of the target object, as discussed in Section V-B.
- 4) *Adversarial+One-Grasp* applies the two adaptation approaches on the generic model sequentially to improve the recognition and grasping.

As shown in Fig. 14, the adaptation methods are compared in four testing environments with an increasing extent of domain shifts: 1) sim basic jitter—simulated basic objects with visual jitter (see Section VI-A) applied on color and depth channels as well as background, 2) sim novel—simulated novel objects, 3) sim novel jitter—simulated novel objects with visual jitter, and 4) real novel—real novel objects. As the training environment uses simulated basic objects, the testing environments include the domain shifts caused by novel objects, visual jitter, and real-world noises.

We report the experimental results of the instance grasping success rate in Fig. 13. Overall, all adaptation methods improve the grasping performance across the testing environments. It is not surprising that adaptation is likely to be more effective (i.e., leading to more performance increments) if the domain shifts are severer. For example, the adaptation gain in the environment of sim basic jitter is less than 4%, while the real environment witnesses an adaptation gain of over 18%. Moreover, when encountering complex novel objects that are more challenging (e.g., drill) than the basic training objects (e.g., red cuboid), the *One-Grasp* method provides more adaptation power than the *Adversarial* method. In *Adversarial* adaptation, we use unlabeled data from the target domain to update the image encoder, while the text encoder and grasping decoder remain unadapted. As a result, the *Adversarial* adapted model is more prone to encountering difficulties with challenging novel objects (e.g., drill and spatula). On the other hand, *One-Grasp* adaptation adapts the entire model and demonstrates better performance on these challenging objects, but it requires additional labeled data (at least one grasp) of the objects. Another observation is that we can combine the two adaptation methods to achieve an even higher adaptation performance. Through various tests, the combined method *Adversarial+One-Grasp* adaptation consistently shows the best performance in all the testing environments. This suggests that the two adaptation methods adapt our model complementarily for accumulative adaptation gains. In practice, we can choose a configuration of adaptation methods based on the availability of the adaptation data.

F. Data Augmentation for Adaptation

The quality of adaptation data is critical for grasping adaptation. The two data augmentation methods, *ObjectAug* and *OneGraspAug*, are proposed for the two adaptation methods,

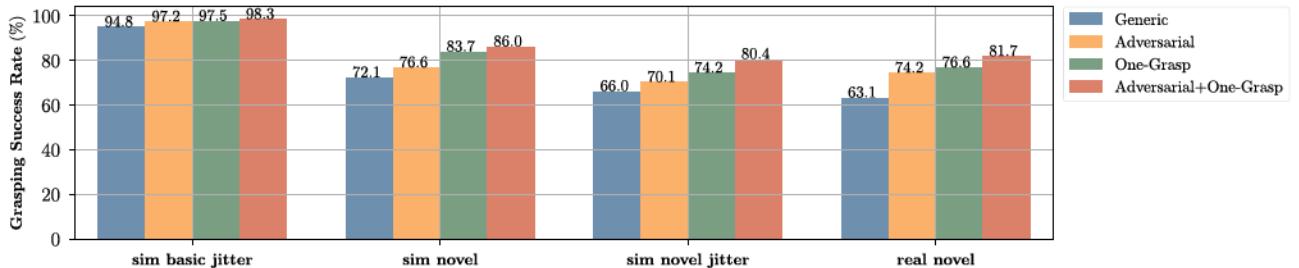


Fig. 13. *Adaptation performance in the designed testing environments.* The instance grasping success rate of four approaches in the four testing environments. The plot shows the effectiveness of the proposed two data-efficient adaptation methods, *Adversarial* and *One-Grasp* adaption, which consistently improve the grasping performance. It is worth noting the additive performance of the two methods, and the combined *Adversarial+One-Grasp* adaption performs the best in all the testing cases.



Fig. 14. *Example images* of the designed testing environments for ablative analysis of adaptation. The environments are created to test the adaption performance of the grasping models as the degree of domain shift increases.

TABLE IV
ABLATIONS FOR OBJECT-LEVEL AUGMENTATION (%)

Method	sim novel	real novel
Objects (w/o overlay and single-object aug.)	73.8	70.0
ObjectOverlay (w/o single-object aug.)	76.1	72.6
ObjectAug (Ours)	76.6	74.2

The bold entities are typically used to emphasize the best results across the methods.

TABLE V
ABLATIONS FOR ONE-GRASP AUGMENTATION (%)

Method	sim novel	real novel
OneGrasp (w/o repetition and rotation)	81.5	72.6
OneGraspRpt (w/o rotation)	81.7	73.8
OneGraspAug (Ours)	83.7	76.6

The bold entities are typically used to emphasize the best results across the methods.

respectively. We execute the ablation studies in Tables IV and V to examine the augmentation methods. The results of adapted instance grasping are presented in the tables. In the ablations for object-level augmentation, the compared approaches are

- 1) *Objects* uses the raw data (images of single objects) as the adaptation data.
- 2) *ObjectOverlay* overlays the randomly sampled objects on the background image to synthesize a large dataset covering possible object combinations and locations.
- 3) *ObjectAug* is our object-level data augmentation method discussed in Section V-A, where much richer object configurations (i.e., orientations and scales) are covered in the synthesized dataset.

For the above augmentation methods, we keep the dataset size constant and use each augmentation data in (9), accordingly.

Even though the data is unlabeled, the adaptation data quality has a direct impact on grasping performance, as seen in Table IV. The performance difference between *Objects* and *ObjectAug*, for example, is up to 4% on real novel objects, despite the fact that they nominally contain the identical objects. This finding demonstrates that the suggested object-level data augmentation successfully reduces domain shift by supplying rich unlabeled data.

In the ablations for one-grasp augmentation, the comparable approaches include

- 1) *OneGrasp* uses the raw data of one successful grasp trial, including an RGB-D image and the corresponding grasping action.
- 2) *OneGraspRpt* simply repeats the one-grasp data N times without rotating the data, where $N = 6$ is the angle discretion parameter.
- 3) *OneGraspAug* is our one-grasp data augmentation method discussed in Section V-B, where we augment the one-grasp data by rotating for N orientations to enrich possible orientations of objects and robot grasping.

As shown in Table V, *OneGraspAug* outperforms the compared methods by over 4% grasping success rate on real robot experiments, which demonstrates how angle-augmented data can be used to make the grasping model rotation-invariant for object recognition and grasping.

VIII. CONCLUSION

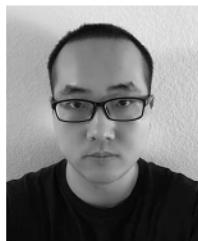
In this work, we presented a novel attribute-based robotic grasping system. An end-to-end architecture was proposed to learn object attributes and manipulation jointly. Workspace images and query text were encoded into a joint metric space, which was further supervised by object persistence before and after grasping. Our model was self-supervised in a simulation only using basic objects but showed good generalization. To further adapt to novel objects and real-world scenes, we proposed two data-efficient adaptation methods, adversarial adaptation and one-grasp adaptation, which only require unlabeled object images or one grasp trial. Our grasping system achieved an 86.0% instance grasping success rate in simulation and an 81.7% instance grasping success rate in the real world, both on unknown objects. Our approach outperformed the other compared methods by large margins.

We showed that incorporating object attributes in robotic grasping improves the performance of the deep learning grasping model. To the best of our knowledge, this is the first work to explore object attributes for the generalization and adaptation of deep learning robotic grasping models. Our long-term goal is to further improve the effectiveness and robustness of our model by pre-training it with objects of richer attributes. Another possible avenue for future work is to study object attributes under partial observation, such as the shape of an object in dense clutter. It would be of interest to explore how to achieve more robust attribute perception, potentially using shape completion algorithms.

REFERENCES

- [1] Y. Yang, Y. Liu, H. Liang, X. Lou, and C. Choi, "Attribute-based robotic grasping with one-grasp adaptation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6357–6363.
- [2] K. Fang, Y. Bai, S. Hinterstoisser, S. Savarese, and M. Kalakrishnan, "Multi-task domain adaptation for deep learning of instance grasping from simulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 3516–3523.
- [3] Y. Yang, H. Liang, and C. Choi, "A deep learning approach to grasping the invisible," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2232–2239, Apr. 2020.
- [4] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, "Covariate shift and local learning by distribution matching," *Dataset Shift Mach. Learn.*, MIT Press, pp. 131–160, 2008.
- [5] S. Ben-David et al., "Analysis of representations for domain adaptation," *Adv. Neural Inf. Process. Syst.*, vol. 19, pp. 137–144, 2007.
- [6] A. Zeng, "Learning visual affordances for robotic manipulation," Ph.D. dissertation, Princeton Univ., 2019.
- [7] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3 d object grasp synthesis algorithms," *Robot. Auton. Syst.*, vol. 60, no. 3, pp. 326–336, 2012.
- [8] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, Apr. 2014.
- [9] X. Lou, Y. Yang, and C. Choi, "Learning to generate 6-DoF grasp poses with reachability awareness," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1532–1538.
- [10] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4238–4245.
- [11] H. Yu and C. Choi, "Self-supervised interactive object segmentation through a singulation-and-grasping approach," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 621–637.
- [12] M. Danielczuk et al., "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 1614–1621.
- [13] E. Jang, C. Devin, V. Vanhoucke, and S. Levine, "Grasp2vec: Learning object representations from self-supervised grasping," in *Proc. Conf. Robot Learn.*, 2018, pp. 99–112.
- [14] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping," in *Proc. Conf. Robot Learn.*, 2017, pp. 119–132.
- [15] J. Cai, X. Tao, H. Cheng, and Z. Zhang, "Ccan: Constraint co-attention network for instance grasping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 8353–8359.
- [16] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1778–1785.
- [17] Y. Zhong and A. K. Jain, "Object localization using color, texture and shape," *Pattern Recognit.*, vol. 33, no. 4, pp. 671–684, 2000.
- [18] Y. Sun, L. Bo, and D. Fox, "Attribute based object identification," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 2096–2103.
- [19] T. Hermans, J. M. Rehg, and A. Bobick, "Affordance prediction via learned object attributes," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA): Workshop Semantic Perception, Mapping, Exploration*. Citeseer, 2011, pp. 181–184.
- [20] S. Pirk, M. Khansari, Y. Bai, C. Lynch, and P. Sermanet, "Online learning of object representations by appearance space feature alignment," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 10473–10479.
- [21] V. Cohen, B. Burchfiel, T. Nguyen, N. Gopalan, S. Tellex, and G. Konidaris, "Grounding language attributes to objects using bayesian eigenobjects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1187–1194.
- [22] H. Ahn, S. Choi, N. Kim, G. Cha, and S. Oh, "Interactive text2pickup networks for natural language-based human–robot collaboration," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3308–3315, Oct. 2018.
- [23] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," in *Proc. Robot. Sci. Syst.*, 2017.
- [24] 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 *Proc. IEEE/RSJ Int. Conf. Intell. robots Syst.*, 2017, pp. 23–30.
- [25] J. Tobin et al., "Domain randomization and generative models for robotic grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3482–3489.
- [26] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," in *Proc. Conf. Robot Learn.*, 2020, pp. 1162–1176.
- [27] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [28] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.
- [29] Y. Chen, J. Jiang, R. Lei, Y. Bekiroglu, F. Chen, and M. Li, "Deep grasp adaptation through domain transfer," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 10268–10274.
- [30] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.
- [31] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2. Lille, 2015.
- [32] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.
- [33] S. Mottian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6670–6680.
- [34] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto, "A baseline for few-shot image classification," in *Proc. Int. Conf. Learn. Representations*, 2019.
- [35] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 787–798.
- [36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
- [37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [38] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proc. Assoc. Comput. Linguistics*, 2015, pp. 1681–1691.
- [39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
- [40] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
- [41] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov, "Gated-attention architectures for task-oriented language grounding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2819–2826.
- [42] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
- [43] H. Liang, X. Lou, Y. Yang, and C. Choi, "Learning visual affordances with target-orientated deep Q-network to grasp objects by harnessing environmental fixtures," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 2562–2568.
- [44] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
- [45] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

- [46] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.
- [47] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
- [48] M. Andrychowicz et al., "Hindsight experience replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5048–5058.
- [49] I. Goodfellow et al., "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 139–144, 2014.
- [50] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [51] I. H. Laradji and R. Babanezhad, "M-adda: Unsupervised domain adaptation with deep metric learning," in *Domain Adaptation for Visual Understanding*. Berlin, Germany: Springer, 2020, pp. 17–31.
- [52] M. Jaderberg et al., "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
- [53] F. Quiroga, F. Ronchetti, L. Lanzarini, and A. F. Bariviera, "Revisiting data augmentation for rotational invariance in convolutional neural networks," in *Proc. Int. Conf. Modelling Simul. Manage. Sci.*, Springer, 2018, pp. 127–141.
- [54] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1321–1326.
- [55] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Proc. Int. Conf. Adv. Robot.*, 2015, pp. 510–517.
- [56] A. Choppa, *Introduction to Google Sketchup*. Hoboken, NJ, USA: Wiley, 2012.
- [57] É. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: A generic software platform with a wide class of robot control skills," *IEEE Robot. Automat. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.
- [58] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 18–19, 2012.
- [59] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.



Yang Yang received the B.Eng. degree in energy and power engineering from Huazhong University of Science and Technology, Wuhan, China, in 2015, and the Ph.D. degree in computer science from University of Minnesota, Minneapolis, MN, USA, in 2022.

He joined Meta shortly before graduation, and he is currently a Senior Research Scientist with Meta AI, working on ultra-scale ranking and foundation models. His research interests include recommendation and monetization AI, embodied AI, and AGI.

Dr. Yang is the recipient of the ICRA 2022 Outstanding Student Paper Award and the UMN 2021 UMII-MnDRIVE Graduate Assistantship.



Houjian Yu received the B.Eng. degree in electrical engineering from North China Electric Power University, Beijing, China, in 2018, and the M.S. degree in electrical and computer engineering from University of California San Diego, CA, USA, in 2020. He is currently working toward the Ph.D. degree in computer engineering with the University of Minnesota, Minneapolis, MN, USA.

His research interests are in robotics and machine learning.



Xibai Lou received the B.Eng. and M.S. degrees in electrical engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2014 and 2015, respectively, and the Ph.D. degree in electrical engineering from University of Minnesota, Minneapolis, MN, USA, in 2023.

He is currently an Applied Scientist with Amazon Robotics, Seattle, USA, focusing on deep learning in 3-D robot vision for warehouse automation. His research interests include robot manipulation, perception, and deep learning-based algorithms.

Dr. Lou is the recipient of the ICRA 2022 Outstanding Student Paper Award and the UMN 2022 UMII-MnDRIVE Graduate Assistantship.



Yuanhao Liu received the B.Eng. degree in electrical information engineering from the Chinese University of Hong Kong, in 2019, and the M.S. degree in electrical and computer engineering from University of Minnesota, Minneapolis, MN, USA, in 2020.

After graduation, he worked with Huawei Technologies and currently works as a Machine Learning Engineer for a robotics startup in Shanghai, China. His research interests are robotics and machine learning.



Changhyun Choi received the Ph.D. degree in robotics from Georgia Institute of Technology.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Minnesota. Before joining the UMN, he was a Postdoctoral Associate with the Computer Science & Artificial Intelligence Lab, Massachusetts Institute of Technology. His research interests are in visual perception for robotic manipulation, with a focus on deep learning for object grasping and assembly manipulation, soft manipulation, object pose estimation, visual tracking, and active perception.

Dr. Choi was the recipient of the NSF CAREER Award, Sony Research Award, Russell J. Penrose Excellence in Teaching Award, and ICRA 2022 Outstanding Student Paper Award.