# Learn to Grasp via Intention Discovery and its Application to Challenging Clutter

Chao Zhao, Chunli Jiang, Junhao Cai, Hongyu Yu, Michael Yu Wang, and Qifeng Chen

Abstract—Humans excel in grasping objects through diverse and robust policies, many of which are so probabilistically rare that exploration-based learning methods hardly observe and learn. Inspired by the human learning process, we propose a method to extract and exploit latent intents from demonstrations, and then learn diverse and robust grasping policies through selfexploration. The resulting policy can grasp challenging objects in various environments with an off-the-shelf parallel gripper. The key component is a learned intention estimator, which maps gripper pose and visual sensory to a set of sub-intents covering important phases of the grasping movement. Subintents can be used to build an intrinsic reward to guide policy learning. The learned policy demonstrates remarkable zero-shot generalization from simulation to the real world while retaining its robustness against states that have never been encountered during training, novel objects such as protractors and user manuals, and environments such as the cluttered conveyor.

Index Terms—Grasping, Dexterous Manipulation, Reinforcement Learning, Imitation Learning, Learning from Demonstrations

# I. INTRODUCTION

Grasping is a fundamental maneuver in many tasks, and grasping a particular object may require a dedicated policy. For example, consider a common grasping scenario where the robot needs to grasp the credit card with a parallel gripper, as shown in Fig. 1. Grasping a credit card object is challenging because the card is so thin that a successful grasp policy may require the gripper to interact with the object and utilize external surfaces to aid manipulation. Although developing flexible and robust policies for grasping diverse objects is a breeze for humans, the current state of the art in robotics is still far from such a capability.

Recent studies have focused on autonomous grasping policy discovery. This area is dominantly driven by model-free reinforcement learning (RL), which obtains grasping policies by self-exploration [1], [2]. However, an important issue with exploration-based methods is that some grasping policies are

C. Zhao, C. Jiang, J. Cai, H. Yu, and Q. Chen are with The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong {czhaobb, cjiangab, jcaiaq}@connect.ust.hk and {hongyuyu, cqf}@ust.hk. J. Cai, H. Yu, and M. Wang are also with HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen. M. Wang is with Monash University michael.y.wang@monash.edu

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

1

Fig. 1: A parallel gripper with the learned policy picking objects using vision for sensing. The time-lapse image shows the actions of the gripper as it interacts with a credit card to pick it up. The red arrow shows the card's motion. The depth image on the bottom-left shows the visual observation.

probabilistically rare, which results in the discovered grasping policies having a similar pattern (i.e., approaching the object and closing the fingers). In this regard, imitation learning offers a way to learn robot skills by mimicking the expert behaviors in demonstrations [3], [4]. However, existing methods only attempt to match the expert action sequences [5], ignoring the understanding of high-level goal planning in the demonstration. As a result, the learned policy cannot be transferred to scenes absent from the demos, limiting the generalization ability.

Therefore, exploiting dexterous grasping strategies from human demos while retaining the ability to explore and adapt to novel scenarios autonomously remains an open problem. This motivates us to propose a method inspired by the human learning process to address this challenge.

Evidence from neuroscience suggests that when humans learn a skill or children learn from others, they selectively focus on the underlying intents of an actor's behavior rather than learning atomic actions [6]. Then, learning is facilitated by following the intents and self-practice. Inspired by this intuitive introspection, we propose a framework to mimic this process to learn grasping, as shown in Fig. 2. At its core, policy learning is based on a principled solution to incorporate the intrinsic reward from intents into RL training. The key component is an intention estimator that predicts probability distributions of a set of intents. The intents are the temporal abstraction of the important phases in the grasping trajectories (e.g., go to a position, rotate, close gripper) compared with detailed movements. The RL agent leverages the foresight afforded by the intention estimator to guide policy learning. Meanwhile, the agent is able to learn policies purely by self-

Manuscript received: September, 5, 2022; Revised November, 15, 2022; Accepted December, 4, 2022.

This paper was recommended for publication by Editor Hong Liu upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone, HZQB-KCZYB-2020083.

exploration when the intention estimator meets novel scenes. Thus, the proposed approach combines the best of both worlds: the diversity of policies provided by demonstration and the adaptability and generalizability brought by self-exploration.

The primary contribution of this paper is the proposed method for learning dexterous grasping policies that have the ability to: 1) grasp objects in broad categories, such as credit cards, Go stones, and soda cans, with only an offthe-shelf parallel gripper; 2) grasp in scenes that are never encountered in demonstrations, such as a cluttered table; 3) learn grasping policies that are unable to be obtained purely by self-exploration; and 4) only use easy, readily available sensors such as the depth camera. While some of these features have been individually demonstrated, we are unaware of published studies that tackle all four.

# II. RELATED WORK

Grasping is a fundamental problem in robotics; it enables further in-hand manipulation and interaction with the environment. Conventional analytic methods model physical processes between the object and the gripper and use modelbased planning to output grasping policies [7]. However, the complexity of physical analysis typically assumes known physical properties to make analysis or planning tractable, which are hard to obtain in practice. Meanwhile, objects are often occluded in cluttered scenes, which makes analyzing feasible grasps challenging. Some works have explored using pre-grasp manipulation, such as sliding [8], to create graspable poses of objects. For example, [9] grasp an object by pushing the object against a support surface and lift the object by pivoting. However, such methods need prior knowledge of the environment and objects' physical properties.

Learning-based methods have recently emerged as alternatives to robotic grasping [10], [11], as they can detect grasps from visual features rather than explicitly using prior knowledge of objects. For example, some grasp synthesis methods [12], [13] use neural networks to accept visual observation as input and output pose estimates of feasible grasps. On the other hand, [14] defines each pixel as a topdown grasp primitive rather than predicting a grasp pose and evaluates each grasp quality through a fully-convolutional neural network. [15] extends this method with an adjustable finger and a model-based primitive to produce an effective grasping system. While predefined primitives can improve data efficiency, they also limit the diversity of policies. Another line of work uses model-free RL algorithms to acquire the grasping policy autonomously through self-exploration [1], [16], [17]. However, the grasping policy is often hard to explore, particularly as the degrees of freedom increase [18]. Some studies [19], [20] introduce clustering-based intrinsic rewards to accelerate RL learning but cannot obtain policies beyond self-exploration capabilities.

For learning dexterous policies, imitation learning is a common approach [3], [21], [22]. The well-known imitation learning method includes behavior cloning, which realizes a mapping between robot states and actions from human demonstration. However, the application of imitation learning to grasping has largely been confined to the quality of expert demonstrations, and collecting demonstration data is often expensive and time-consuming [21]. Moreover, the common issue with these methods is that they are hard to generalize to unseen objects or environments that are not included in demonstrations due to distributional shifts and compounding errors [5],[23]. Other works focus on designing end-effectors to grasp challenging objects instead of focusing on the grasping policy. The end effector can be designed by humans or discovered through learning algorithms [24]. However, end effectors with complicated designs often only apply to specific object types, reducing the robot's versatility and increasing the system's complexity.

Compared with the abovementioned studies, our presented approach substantially improves the diversity of graspable objects and the grasping policies. Rather than imitating atomic actions, our method extracts the latent intents from demos and utilizes them in policy learning, incorporated with selfexploration. The entire learning is completed in simulation without expensive demo collection in field conditions and is consistently effective in zero-shot transfer to the real world.

# III. METHOD

In this section, we describe the proposed method for learning the grasping policy. Our method consists of three phases, as illustrated in Fig. 2.

First, an intent estimator is trained with simulated grasps to learn a mapping between the state in a grasp demonstration and intents (see Sec. III-A). Grasps are generated in the simulation using the three provided grasp types (see Fig. 3). The intention estimator captures the environment and robot information using a network and outputs a set of probabilities representing the distance between the given state and intents.

In the second phase, the grasping policy is trained with RL (see Sec. III-B). During training, we exploit two kinds of rewards: a task reward and an intrinsic reward. The task reward is sparse and given when the robot successfully grasps. The intrinsic reward is from the intention estimator and guides the RL agent when the agent approaches an intent. Chronologically, the latter appeared intention is both achievable and closer to the solution than the former. Therefore, providing positive rewards can facilitate robot learning after each intent is fulfilled. Such construction compensates for the inability to discover interesting policies with random exploration.

Last, we transfer and deploy the learned policy to the physical robot. Our training in simulation only uses rigid objects with simple geometry, such as the cube and cylinder (see Sec. III-C). Yet when deployed on a real robot, the robot successfully handles broad object categories (protractors, Go stones, etc.) and environments (cluttered table and conveyor) with only an off-the-shelf parallel gripper.

# A. Learning an Intention Estimator

In the first phase of learning, we aim to learn an intention partitioning strategy with a neural network, as shown in Fig. 3A. The input is a given state from simulated grasp demos, and



Fig. 2: System Overview. A: We generate a set of simulated grasps to learn an intention estimator. The state s in a grasp includes the depth image and gripper pose. They are processed separately using a Conv encoder for the former and an MLP encoder for the latter. Then, the concatenation of two vectors is fed through the subsequent FC layers to predict probabilities. B: We train our policy with PPO. The RL agent receives the observed state  $s_t$  and predicts the action at time step t. The robot executes and switches to the next state t + 1. The intention estimator discerns the intent of the given state t + 1, and the RL agent then receives a task reward from the simulation and an intrinsic reward from the intention estimator. C: To transfer to the real world, the policy network alone is used to control the robot. The wrist-mounted camera provides the depth image, and the gripper pose is from the robot's proprioception.

the output is a family of probability distributions indicating how likely the current state is to be divided into each intent.

Intent Segmentation and Data Collection: Considering a grasp demonstration  $S = (s_1, s_2, \ldots, s_l)$  represented by a sequence of states s = (I, h), where I is the camera observation of the environment, and h is the gripper pose. The state  $s \in S$  in a grasp demo can be naturally segmented into n intents, denoted as  $K = (k_0, k_1, \ldots, k_n)$ , according to the timing order and similarity. The index of k indicates the timing order of intents, and an intent  $k_{t+1}$  can only be reached after completing former intent  $k_t$ . The h consists of  $(x, y, z, \alpha, \beta, \gamma, \psi)$ , where  $(x, y, z, \alpha, \beta, \gamma) \in SE(3)$  is the 6D gripper pose, and  $\psi$  is one hot vector representing the opening and closing of the gripper.

We now give a formal definition of the k-intent segmentation problem. If k = l, each gripper pose corresponds to a segment. Otherwise, despite the fact that humans can manually label segments, the following segmentation algorithm can be used to reduce labor costs. Let  $T = (T_0, T_1, ...)$  denote the set of all possible ways of segmentation for a sequence S. The sequence S of length l contains n non-overlapping contiguous sub-sequences, denoted as  $T_i = (\tau_1, \tau_2, ..., \tau_n)$ . Each state s in segment  $\tau_i$  belongs to the intent  $k_i$ . We denote the dissimilarity in a segment  $\tau_i$  as  $e_i$ , then the error of segmentation  $T_i$  is calculated as  $E_p = \sum_{i=0}^n e_i$ . Thus, we define the optimal segmentation as to find the minimize  $E_p$  in T:

$$T_{\text{opt}}(S,n) = \arg\min_{T_i \in T} E_p(S,T_i).$$
(1)

The  $T_{opt}(S, n)$  can be found by the dynamic-programming (DP) algorithm [25], and the main recurrence of the DP is

$$E_p \left( T_{\text{opt}} \left( S[1 \dots l], n \right) \right) = \{ E_p \left( T_{\text{opt}} \left( S[1 \dots j], n-1 \right) \right) + E_p \left( T_{\text{opt}} \left( S[j+1, \dots, l], 1 \right) \right) \},$$
(2)

where  $S[1, \ldots, j]$  denotes the sub-sequence of S that contains states in positions from 1 to j. The function of the recurrence is to divide the segmentation problem into subproblems and combine their solutions to form the final segmentation. The dissimilarity  $e_i$  in a segment is the sum of the dissimilarities  $\Lambda$  between states, calculated by the following formula:

$$e_{i} = \sum_{s_{v}, s_{w} \in \begin{pmatrix} s \in \tau_{i} \\ 2 \end{pmatrix}} |\Lambda_{s_{v}, s_{w}}|, \qquad (3)$$



Fig. 3: Data collection for learning the intention estimator. (a) Three demonstrated grasps; (b) Examples of grasps augmented based on the three demos in (a). Left: changes in object positions. Middle: changes in the object orientations. Right: changes in object sizes.

$$\begin{aligned} |\Lambda_{s_v,s_w}| &= (|x_v - x_w| + |y_v - y_w| + |z_v - z_w|) \\ &+ (|\alpha_v - \alpha_w| + |\beta_v - \beta_w| + |\gamma_v - \gamma_w|) \\ &+ \mu(|\psi_v - \psi_w|), \end{aligned}$$
(4)

where  $\lambda$  and  $\mu$  are the hyper-parameters to adjust the influence of the gripper orientation and finger condition (i.e., open/close) change. The distance of the orientation is the relative difference of Euler angle changes and is normalized. After segmentation, each state s in a demo is assigned to an intent  $k_i = (1, 2, ..., n)$ as supervision signals.

To learn an intention estimator, a set of grasp demonstrations needs to be collected and segmented using the above algorithm. We generate grasps in simulation by augmenting three humanencoded grasps (see Fig. 2(a)), using invariant and equivariant principles. Consider an encoded grasp S for an object o with a pose  $o_p$ . A new grasp S' can be augmented by the following procedures. First, we apply a set of transformations to the object pose  $o_p$ , including changing object positions and orientations. Then the new grasp S' is transferred from S via homogeneous transformation by calculating the SE(3) matrix between  $o_p$ and  $o'_n$ , as shown in Fig. 2(b). We also randomize the aspect ratio of objects at each new grasp generation. Although such augmentation of grasps leads to some imperfect grasps, these imperfect demos do not affect policy learning. We further analyze the influence of imperfect grasp demonstrations in Sec. III-D and Fig. 4(c).

**Intention Estimator Learning:** The goal of the intent estimator is to map the similarity between the given state and each intent. Operationally, we form it as a classification problem and use a neural network p = f(s) to learn this mapping, where the given state *s* contains a depth image *I* of the environment and gripper pose *h*. The network processes grasp pose and visual observation in separate channels, and the output features are combined to feed into a feed-forward pipeline to calculate probabilities that the given state belongs to different intents. More precisely, the depth image *I* and gripper pose *h* are processed with a convolutional (Conv) encoder and a multilayer perceptron (MLP) encoder, respectively. Then the features are combined using concatenation operation and fed into three subsequent fully connected (FC) layers with 256, 256, and 128 neurons. The Conv encoder consists of one 1x1

convolutional layer followed by a global average pooling. The MLP encoder consists of one FC layer. We use the following cross-entropy loss to train the network, as shown in Eq. 5:

$$L = \frac{1}{N} \sum_{i} L_{i} = -\frac{1}{N} \sum_{i} \sum_{c=1}^{n} \{k_{i} = c\} \log(p_{ic}), \quad (5)$$

where c = (1, 2, 3, ..., n) is the class index of intents and  $k_i$  is the intent class label of the given state.

#### B. Policy Learning with Intention Estimator

After we train an intention estimator that can discern the intent of the given state, we distill an intrinsic reward from its prediction. The intrinsic reward allows the robot to follow the intent during policy learning in RL and is detailed below.

**Problem Formulation:** We formulate the picking problem as a Markov Decision Process (MDP). The MDP is defined by a state-space S, action space A, a function of reward  $R(s_t, s_{t+1})$ , and the transition probability  $P(s_{t+1}|s_t, a_t)$ . At time step t, a robot agent to pick objects observes the state  $s_t$  and predicts an action  $a_t$  based on current policy  $\pi(a_t|s_t)$ . The rewards from the environment and intention estimator are provided to the agent afterward and then transition to a new state  $s_{t+1}$ . RL aims to learn an optimal policy  $\pi$  that selects actions that maximize its cumulative reward.

**Rewards with Intents:** The output probabilities from the intention estimator are used to design a reward function r' as follows:

$$r'_{t} = p\{(k_{i} = t) | (s_{t+1})\},$$
(6)

where  $s_{t+1}$  is the state at time step t+1 and  $p\{(k_i = t)|(s_{t+1})\}$ is a predicted probability between 0 and 1 representing the similarity between the current state and the  $t^{th}$  intention  $k_t$ . The  $k_i = t$  represents that we bundle time step t with intent index  $k_i$  to encourage the agent to follow the intents during training. Note the proposed method does not strictly limit agents to follow intents. In order to grasp in finite steps, the episode length is fixed to the number of intents.

Meanwhile, a task reward  $r_{task}$  is given at the end of an episode, 10 for grasping one object successfully and 0 for otherwise. Thus the full reward function is defined as  $r_{total} = r_{task} + r'$ . When the RL agent meets scenes that are never encountered in demonstrations, though the intrinsic reward r' from the intention estimator is almost zero, the agent can still explore on its own and obtain the task reward  $r_{task}$  to learn the grasping policy in novel scenes.

**Policy Architecture:** The policy is trained with Proximal Policy Optimization (PPO) in simulation. PPO requires training a value network that forecasts the discounted sum of future rewards from the current state and a policy network that maps a current state to actions. The policy and value networks share the same state input, as shown in Fig. 3. The state is defined as  $s_t := (I_t, h_t)$ , where  $I_t$  is a depth image with a resolution of 120×120 from the camera, and  $h_t$  is the gripper pose at time step t including the position, orientation, and closure status of the gripper. The policy and value networks share the same front-end network.  $I_t$  is sequentially processed by three convolutional layers with kernel sizes of  $8 \times 8$ ,  $4 \times 4$ ,

and  $3 \times 3$ , and  $h_t$  is processed by one FC layer with eight neurons. Then, the concatenation of two extracted features is fed through the subsequent two FC layers with 64 neurons and split into two output layers: one for predicting the action and another for estimating the value. As a wrist-mounted camera on a real robot might capture things outside the workspace after acting  $a_t$ , to reduce the sim-to-real gap, we only update  $h_t$  and always use the initial depth observation  $I_0$  as the part of state  $s_t$  instead of updating the depth observation over time.

Actions: In our environment, each policy action  $a_t$  includes a gripper pose displacement and a vector to control the gripper closure. The gripper pose displacement is the difference between the initial pose and the desired one, encoded as  $(x'_t, y'_t, z'_t, \alpha'_t, \beta'_t, \gamma'_t)$ , where  $(x'_t, y'_t, z'_t)$  is the relative displacement and  $(\alpha'_t, \beta'_t, \gamma'_t)$  are the rotations of the gripper about its x-, y- and z-axes. The one-hot vector to control the gripper closure is denoted as  $\psi'_t$ . We discretize each action's coordinate according to the workspace. In addition, the episode will be terminated if  $\psi_t$  is true. If terminated, the robot returns to its initial pose, receives new observations, and executes the next grasp, which provides a certain degree of ability for handling uncertainty and imperfect executions. For example, if objects slip from the hand during the last grasp trail, the robot can try again when it receives new observations after resetting to its initial pose.

### C. Training Details

We train the policy in the Pybullet simulator[26]. The training process consists of two stages. First, we learn the intention estimator from grasp demos, and then the RL agent explores and learns the grasping policy with the aid of the intrinsic reward constructed by the intention estimator. To train the intention estimator, we generate 10000 grasps in the simulation for each provided grasp type using the method described in Sec. III-A. Each encoded grasps have four poses. We use n = 3 as the number of intents. Intuitively, when the gripper closes, it often represents a shift of intention, and thus we use  $\mu = 5$  to increase the influence of such activities in the dissimilarity calculation. A total of 30000 grasps were used to train the intention network with cross-entropy loss. The Adam optimizer was employed, starting with a learning rate of 0.001. One hundred epochs are performed, and the learning rate is halved every ten epochs during training.

During the RL policy learning phase, a pool of 64 robots generates training episodes by downloading the current policy parameters every 10 epochs from the optimizer. In each environment, random objects were placed in the workspace with random poses. Only cuboids, cylinders, and their variants with different aspect ratios are used during training, as shown in Fig. 4(a). The robot then collects the episodes in the simulation, during which the simulator automatically determines the task reward, and the estimator provides the intrinsic reward. If the workspace is empty or an object is dropped, the environment will be reset, at which point objects with random poses will fall into the workspace again. Finally, the resulting episodes are sent back to the optimizer. The Adam optimizer with a learning rate of  $10^{-4}$  is used. We also deploy domain randomization



Fig. 4: (a): Examples of environments. The first three rows are similar environments to the demo, the clutter of the last row is not in the demo (b): Success rate curve of our policy training. (c): Simulation results with different element choices of our method (Ours) and behavior cloning (BC).

to make the learned policy robust to a range of real-world conditions. Fig. 4(b) shows the learning curve for the final model training.

## D. Simulation Results

After training, the policy network alone is deployed to the robot in both simulation and the real world. In simulation experiments, we set up the following environments: a) Scenes in demonstrations (Similar Scene): environment constructed with a single object but in new configurations, including object friction and mass. b) Scenes not in demonstrations (Unseen Scene): a cluttered scene with multiple objects, where grasping policies can be found by self-exploration. Fig. 4(c) summarizes the result tested on similar and unseen scenes in simulation. The learned policy from the final model (denoted as *Ours*) is able to grasp the object with success rates of 98.3% in the similar scene and 91.2% in the unseen scene (row 3 in Fig. 4(c)). In contrast, removing the intent estimator from RL policy learning (denoted as Ours - w/o intent), the success rates are considerably lower (row 5 vs. row 3 in Fig. 4(c) because pure RL exploration cannot find a successful grasping policy for thin objects such as cards. Meanwhile, the policy directly learned by behavior cloning (denoted as BC) using the same demonstrations performs better than Ours-w/o intent but lower than *Ours*. This validates our hypothesis that learning from intent (row3 in Fig. 4(c)) can help the agent learn complex and better policies while retaining its ability to explore unseen scenarios beyond directly cloning policies (row 1 in Fig. 4(c)) or exploring entirely on its own (row 5 in Fig. 4(c)). Moreover, the method of behavior cloning achieves poor performance in the unseen scene. In comparison, our model generalizes well to the novel scene.

We also investigate the impact of demonstration quality on policy learning. A total of 12.3% of demos for training intent estimators fail. We remove these imperfect demonstrations and use the remaining perfect demos to learn the policy using

behavior cloning and our method (row 2 and row 4 in Fig. 4(c)). We observe that by using perfect demonstrations, the success rate of behavior cloning increases by more than 5% compared to using imperfect demonstrations (row 2 vs. row 1 in Fig. 4(c)). In contrast, our method does not rely on the quality of the demonstration. It achieves comparable performance with the one learned with perfect demonstrations (row 3 vs. row 4 in Fig. 4(c)). Because when the intent estimator's guidance is biased due to imperfect demonstrations, the RL agent can revise the policy through self-exploration, illustrating the superiority of learning from intents rather than direct cloning atomic actions. Such ability also helps agents to learn in unseen scenes that the agent seamlessly switches to self-exploration when meeting novel scenes, allowing agents to obtain policies in scenes that are not demonstrated. Real-world experiment results are presented in Sec. IV.

# **IV. REAL-WORLD EXPERIMENTS**

We executed a set of experiments to evaluate our system in the real world. The code of the presented work is available https://robotll.github.io/LearnfromIntents/

# A. Hardware Setting



Fig. 5: Our hardware setting for real-world experiments.

As shown in Fig. 5, we deployed the learned policy on an off-the-shelf robotic grasping platform, including a 6-DOF robot arm equipped with a robotiq140 parallel gripper and an Intel L515 depth camera.

# B. Real Robot Experiment

In this section, we quantitatively evaluated our picking system and other state-of-the-art methods with two protocols. In the first protocol, which we refer to as "isolated object grasping", the robot attempted to grasp a single object lying in the workspace. We also used a second protocol where the robot cleaned a pile of mixed objects randomly dumped into the workspace. This test was more challenging as the robot had to avoid collisions with other objects while grasping. We used two metrics for evaluation: successful picks per attempt (Success Rate) and picks per hour (PPH). A successful grasp is grasping only one object and not pushing any other objects out of the workspace. Tab. I summarizes the results of the learned policy in the real world.

We first examined the performance of our policy with the first protocol (col. 1-8 in Tab. I) on the table environment. Our method (Ours) obtained success rates of over 90% for dominos, tubes, cans, and cosmetics. For the most challenging objects, including cards, user manuals, protractors, and Go stones, our method achieved success rates of over 80% for cards and over 60% for the other objects. In contrast, the state-of-the-art 6-DoF grasp synthesis method (VPN) [13] and the learning-based planar grasping method (Planar) [14] could not successfully grasp cards, user manuals, and protractors. Notably, when testing dominos, tubes, and Go stones using the VPN baseline, we manually select top-down grasp poses; otherwise, the VPN method cannot detect a feasible grasp for these objects. Meanwhile, the behavior cloning (BC) method performed worse with all test objects.

We then evaluated our learned policy on the cluttered table populated by multiple objects (column 9 in Tab. I). Our method stably obtained a success rate of 82% in the challenging dense clutter. This level of performance is beyond other baselines. Also, the success rates of behavior cloning dropped below 15% on mixed objects due to the inherent compounding error and distribution shift. From Tab. I, the protractor and the Go stone are the most challenging to grasp among test objects. We hypothesize that this happened because protractors and Go stone have complex geometries and dynamics different from the training objects, increasing the difficulty of generalization. The other methods also perform less effectively.

At last, to emphasize the generalization ability of our learned policy and the value of using an off-the-shelf parallel gripper alone to grasp objects in broad categories, we also focused on comparing the presented approach with other methods in a conveyor environment common in industry (see Fig. 6A). The belt on the conveyor has higher friction than the table environment and is elastic. In addition, the significant variation of material properties over the surface adds extra noise to the depth camera. Overall, our method reported in the third row still achieved a higher grasp success rate (except for the tube) and PPH in all conditions. The tube has a lower success rate as it has a different non-centrosymmetric shape, making it easier to grasp from the head rather than the object's center. In contrast, all training objects are centrosymmetric and do not have such a characteristic.

The learned policy manifests a dexterous behavior, as shown in Fig. 6. The robot approaches the protractor and continues interacting to reach a state that is feasible to grasp (see Fig. 6C). This distinguishes the presented approach from other exploration-based methods, which confine the policy to a format of approaching the object with a certain pose, closing the finger, and avoiding interaction with objects (see Fig. 6D and 6E). We can also observe that the behavior cloning method performs poorly due to the distribution shift issue, further showing the significance of learning from intentions. Note also that the policy learned by our method is more robust and not tied to particular objects. Fig. 6B shows the learned policy responding to different poses of the same object. The

Credit Card Domino Tube Soda Can Dense Clutter User manual Protractor Go Stone Cosmetic Jars ENV Method SR PPH SR PPH SR PPH SR PPH SR PPH SR PPH PPH SR PPH SR PPH SR Table VPN [13] 0% 0% 0% 98% 101 98% 101 22% 23 80% 82 84% 87 38% 39 -238 Planar [14] 0% 0% 0% 96% 346 92% 331 64% 230 90% 324 82% 295 66% BC 50% 178 18% 90% 320 341 12% 46% 163 64 84% 298 52% 185 96% 86% 305 43 Ours 82% 291 76% 270 64% 227 98% 348 94% 334 68% 241 96% 341 92% 327 82% 291 84% VPN [13] 0% 98% 22% 23 82% 84 52% 54 0% 0% 101 98% 101 87 Convevo BC 38% 135 34% 121 12% 43 90% 320 84% 298 40% 142 92% 327 86% 305 10% 36 Ours 80% 284 68% 241 60% 213 98% 348 92% 327 64% 227 98% 348 90% 320 78% 277

TABLE I: Results of experiments in the real world.

\* SR stands for Success rate. \*\* Dense Clutter: Mixed objects on the cluttered table or the conveyor belt

 A: Convey
 Image: Amage: Am

Fig. 6: A: our method (Ours) grasps from a cluttered conveyor; Successful grasp. B: our method (Ours) responds to a soda can with different poses; Successful grasps with robust grasping behavior. C: our method (Ours) grasps a protractor; Successfully grasp. D: top-down grasp (Planar [14]) cannot pick the credit card. The card slips out of the fingertip. E: 6-DoF grasp synthesis method (VPN [13]) fails to grasp the user manual due to the collision.

TABLE II: Analysis of generalization

Method	Use extra objects?		Protractor		Go stone		Tube	
	Phase A*	Phase B <sup>**</sup>	Sim	Real	Sim	Real	Sim	Real
Ours	No	No	67%	64%	72%	68%	95%	94%
Ours-extra	No	Yes	97%	84%	98%	88%	98%	98%
Ours-w/o intent	-***	Yes	0%	0%	93%	84%	98%	98%
* Intention estimator learning stage. ** Policy learning stage. *** Phase A excluded.								

policy identifies purely from observations and adopts different strategies. Such behavior is not specified during training in any way and is discovered by itself. Our training environment features only simple rigid objects, with no complex geometry or compliance, such as protractors and user manuals. Nevertheless, the learned policy successfully meets the diversity of real-world conditions encountered at deployment.

### C. Further analysis of generalization

In this section, we investigate 1) the effect on the success rate of adding novel objects, which perform relatively poorly during real-world testing, into the policy training phase; 2) how well the intention estimator, trained on only cube and cylinder, generalizes to different objects. For the first question, we add models of the protractor, Go stone, and tube to the policy learning phase (i.e., Phase B in Fig. 3) and train with

our proposed method (denoted as Ours-extra). Qualitative real-world and simulation results (row 2 vs. row 1) show that the success rates of these objects can be improved by adding their models to training. For the second question, we learn the policy with extra objects and without using the intention estimator, denoted as Ours-w/o intent in Tab. II, and compare its performance with Ours-extra. The results (row 3 vs. row 2) show that the intention estimator successfully generalizes to objects that differ from those used in the intention estimator training phase. Ours-extra achieves an over 90% success rate for grasping the protractor, while Ours-w/o intent, which only purely relies on self-exploration, cannot grasp the protractor successfully. Meanwhile, Oursextra achieves higher performance for the Go stone than Oursw/o intent. Both results show the successful generalization of the intention estimator to different novel objects. The results also show that without the guidance of the intention estimator, RL agents' self-exploration cannot discover a successful policy to grasp challenging thin objects (e.g., protractor).

#### V. DISCUSSION AND FUTURE WORK

Unlike other state-of-the-art methods, our approach mimics the human learning process, which abstracts and learns intent from demonstrated grasps, and then develops grasping policies through self-exploration. Despite the challenging objects, our method achieves up to 82% success in the dense clutter. While a set of demoed grasps need to be collected, all grasps are collected automatically in the simulation based on three humanencoded grasps. This minimizes the workload on humans. On the other hand, our approach leverages the intent as a reward during RL policy training without imitating detailed motion. Hence, it can learn to react to environments and scene settings not included in the demos.

We see several limitations and opportunities for future research. First, our result describes a far wider range of objects, which achieves substantial improvements over other approaches. Future research could extend the present work to include grasping deformable objects. Another hint is that we hypothesize that diverse environments and demos could extend the presented work to long-horizon tasks since the proposed methodology is generic concerning the tasks. Finally, the presented work relies on human-encoded grasps to learn complex policies that self-exploration cannot discover. This is a significant advantage in that some grasping policies are hard to discover with pure exploration. Nevertheless, humans can easily learn behavior from videos or descriptions in books instead of human-encoded movement. A major opportunity for future studies will be to extend the proposed work to develop a method that can directly learn grasping policies from video or language descriptions.

#### REFERENCES

- D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*, pp. 651–673, PMLR, 2018.
- [2] C. Bodnar, A. Li, K. Hausman, P. Pastor, and M. Kalakrishnan, "Quantile qt-opt for risk-aware vision-based robotic grasping," in *Robotics: Science* and Systems XVI, Virtual Event / Corvalis, Oregon, USA, July 12-16, 2020 (M. Toussaint, A. Bicchi, and T. Hermans, eds.), 2020.
- [3] J. S. Dyrstad, E. Ruud Øye, A. Stahl, and J. Reidar Mathiassen, "Teaching a robot to grasp real fish by imitation learning from a human supervisor in virtual reality," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7185–7192, 2018.
- [4] E. Johns, "Coarse-to-fine imitation learning: Robot manipulation from a single demonstration," in 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 4613–4619, 2021.
- [5] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in 5th Annual Conference on Robot Learning, 2021.
- [6] M. Meyer, H. M. Endedijk, and S. Hunnius, "Intention to imitate: top-down effects on 4-year-olds' neural processing of others' actions," *Developmental cognitive neuroscience*, vol. 45, p. 100851, 2020.
- [7] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 1, pp. 348–353 vol.1, 2000.
- [8] K. Hang, A. S. Morgan, and A. M. Dollar, "Pre-grasp sliding manipulation of thin objects using soft, compliant, or underactuated hands," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 662–669, 2019.
- [9] Z. Sun, K. Yuan, W. Hu, C. Yang, and Z. Li, "Learning pregrasp manipulation of objects from ungraspable poses," in 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 9917– 9923, IEEE, 2020.

- [10] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, *et al.*, "Deep learning approaches to grasp synthesis: A review," *arXiv preprint arXiv:2207.02556*, 2022.
- [11] J. Kerr, L. Fu, H. Huang, J. Ichnowski, M. Tancik, Y. Avigal, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping," in 6th Annual Conference on Robot Learning, 2022.
- [12] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.
- [13] J. Cai, J. Cen, H. Wang, and M. Y. Wang, "Real-time collision-free grasp pose detection with geometry-aware refinement using high-resolution volume," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1888– 1895, 2022.
- [14] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in 2018 IEEE international conference on robotics and automation (ICRA), pp. 3750–3757, IEEE, 2018.
- [15] C. Zhao, Z. Tong, J. Rojas, and J. Seo, "Learning to pick by digging: Data-driven dig-grasping for binpicking from clutter," in 2022 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2022.
- [16] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.
- [17] H.-G. Cao, W. Zeng, and I.-C. Wu, "Reinforcement learning for picking cluttered general objects with dense object descriptors," in 2022 International Conference on Robotics and Automation (ICRA), pp. 6358–6364, IEEE, 2022.
- [18] A. Morel, Y. Kunimoto, A. Coninx, and S. Doncieux, "Automatic acquisition of a repertoire of diverse grasping trajectories through behavior shaping and novelty search," in 2022 International Conference on Robotics and Automation (ICRA), pp. 755–761, 2022.
- [19] A. Dereventsov, R. Vatsavai, and C. G. Webster, "On the unreasonable efficiency of state space clustering in personalization tasks," in 2021 International Conference on Data Mining Workshops (ICDMW), pp. 742– 749, IEEE, 2021.
- [20] T. Mandel, Y.-E. Liu, E. Brunskill, and Z. Popovic, "Efficient bayesian clustering for reinforcement learning," in *IJCAI*, pp. 1830–1838, 2016.
- [21] S. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4978–4985, 2020.
- [22] M. Hamaya, F. von Drigalski, T. Matsubara, K. Tanaka, R. Lee, C. Nakashima, Y. Shibata, and Y. Ijiri, "Learning soft robotic assembly strategies from successful and failed demonstrations," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8309–8315, 2020.
- [23] J. Yang, J. Zhang, C. Settle, A. Rai, R. Antonova, and J. Bohg, "Learning periodic tasks from human demonstrations," in 2022 International Conference on Robotics and Automation (ICRA), pp. 8658–8665, 2022.
- [24] M. Kodnongbua, I. Good, Y. Lou, J. Lipton, and A. Schulz, "Computational design of passive grippers," ACM Transactions on Graphics (TOG), vol. 41, no. 4, pp. 2–12, 2022.
- [25] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Communications of the ACM*, vol. 4, no. 6, p. 284, 1961.
- [26] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning." http://pybullet.org, 2016–2021.