2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

Fixture-Aware DDQN for Generalized Environment-Enabled Grasping

Eddie Sasagawa[1] and Changhyun Choi[2]

***Abstract*— This paper expands on the problem of grasping an object that can only be grasped by a single parallel gripper when a fixture (e.g., wall, heavy object) is harnessed. Preceding work that tackle this problem are limited in that the employed networks implicitly learn specific targets and fixtures to leverage. However, the notion of a usable fixture can vary in different environments, at times without any outwardly noticeable differences. In this paper, we propose a method to relax this limitation and further handle environments where the fixture location is unknown. The problem is formulated as visual affordance learning in a partially observable setting. We present a self-supervised reinforcement learning algorithm, Fixture-Aware Double Deep Q-Network (FA-DDQN), that processes the scene observation to 1) identify the target object based on a reference image, 2) distinguish possible fixtures based on interaction with the environment, and finally 3) fuse the information to generate a visual affordance map to guide the robot to successful Slide-to-Wall grasps. We demonstrate our proposed solution in simulation and in real robot experiments to show that in addition to achieving higher success than baselines, it also performs zero-shot generalization to novel scenes with unseen object configurations.**

I. INTRODUCTION

Grasping an unwieldy or irregularly-shaped object requires additional appendages or intelligent use of the surroundings. For example, in order to pick up a parcel or a book that is wider than our hand, we might use both hands or brace the parcel against a wall or heavier object (i.e., fixtures) to reach underneath and scoop it up. For a single parallel gripper that lacks a helping hand, only the latter option is available. Grasping by first pushing against the wall—referred to in [1], [2] as the Slide-to-Wall task (see Fig. 1b)—presents an interesting problem in robotic manipulation as it requires planning over a multi-step horizon while also understanding the elements in the scene to identify fixtures.

Robots that rely only on visual perception (e.g., RGB or RGB-D observations) must infer object intrinsic properties (e.g., mass, inertia, and friction) from visual cues such as size, color, and shape. As we explore later in Section II, prior works in object grasping tend to either reduce problem complexity and allow implicit learning of certain objects (e.g., targets are only a specific shape or color), or train their proposed networks on many different objects to achieve generality. However, when the Slide-to-Wall task is extended from walls to arbitrary fixture objects, such simplifications are not ideal because the approach must account for relative object properties (i.e., fixture only needs to be better anchored than the target object). For example, a robot attempting to grasp a light parcel may only require a nearby stack of magazines to leverage against, or some cans. Therefore, the Slide-to-Wall task can be considered an exercise in resourcefulness where the original tool provided—the parallel gripper in this case—is insufficient in isolation and flexible application of the environment is needed for success. Hence it is also in this same spirit that we investigate methods that promote online determination of 'usability'.

(a) (b)

Fig. 1: Slide-to-Wall task with fixture ambiguity. (a) There are two visually identical candidate fixtures in the workspace (i.e., white boxes), only one of which is heavy enough to leverage. (b) The target object can only be grasped if the robot pushes it against the correct fixture (i.e., the box on the right).

The relative object properties that determine what objects in the scene might be leveraged as fixtures can change across different environments in the real world. A box that was filled in one environment and thus usable as a fixture could also be empty in another, violating prior knowledge of fixtures. A given environment may even have both types of boxes at the same time, as shown in Fig. 1a, leading to ambiguity. Many models, such as the Target-Oriented Deep Q-Network (TO-DQN) [1], learn to adapt to the object property distribution during training. Such models that learn an implicit relationship become unable to adapt to the change in object properties and require fine-tuning with substantial data. Presenting both types of boxes (i.e., empty and full) during training induces difficulty, as it is unreasonable to expect to reliably infer fixture likelihood from only visual observations. Indeed, attempting to learn a general relationship between external object properties and intrinsic properties can be detrimental due to the large variety and be prone to over-fitting.

The Slide-to-Wall task with uncertain fixtures presents a challenging task in the robotic manipulation domain. First,

*This work was in part supported by the National Science Foundation Award 2143730 and MnDRIVE Initiative on Robotics, Sensors, and Advanced Manufacturing.

[1]The author is with the Minnesota Robotics Institute, Minneapolis, MN 55455, USA. arnev006@umn.edu

[2]The author is with the Department of Electrical and Computer Engineering, University of Minnesota Twin Cities, Minneapolis, MN 55455, USA. cchoi@umn.edu

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9982182

Authorized licensed use limited to: University of Minnesota. Downloaded on January 27,2023 at 19:42:51 UTC from IEEE Xplore. Restrictions apply.

it is unique from other well explored object grasping applications (e.g., pick-and-place) in that the target and fixture objects are not directly observable and require additional information outside of the given visual inputs to infer. The task also requires the ability to reason about multi-object interaction—the interplay between the target object, the wall, and other objects (i.e., clutter) in the scene. Furthermore, because the target object must be pushed often multiple times to the wall before it can be grasped, the task has a longer horizon to contend with, increasing the complexity of decision making.

In this work, we present an end-to-end reinforcement learning approach that accomplishes Slide-to-Wall tasks by processing visual input and identifying likely fixtures that can be leveraged. As briefly discussed above, since it is restrictive to train a network to infer fixture likelihood from only visual observations, we propose to track object fixture likelihood external to the learned model and provide a likelihood map as an input channel. As for selecting the target object, we propose to allow the network to choose by providing yet another input channel for target similarity (see Section IV for details). This is in contrast to [1] where the target object is selected and used as an valid action mask, limiting exploration of non-targeted objects.

In short, the main contribution of this work is an improvement to the end-to-end vision-to-action pipeline for grasping an object in the Slide-to-Wall task, and consists of:

- A new Fixture-Aware Double Deep Q-Network (FA-DDQN) that incorporates a target similarity channel and a fixture belief channel to generate a visual affordance map for a shovel-and-grasp action primitive.
- A fixture classification pipeline to suitably track fixture likelihoods of different objects and inform the fixture belief channel input.

II. RELATED WORK

1) Leveraging Environment for Grasping: While the majority of work in object manipulation centers around simplified geometries that can be easily picked up by a single parallel gripper [3], [4], there exists a much less explored branch that seeks to leverage the environment in object grasping. Eppner and Brock proposed a grasp planner in [5] that is based on traversing a manipulation graph from one environmental constraint to the next. More recently, Liang et al. achieve wall-enabled grasping using their presented Target-Oriented DQN (TO-DQN) [1]. TO-DQN, which is used as a baseline for this work, forces the output Q-map (i.e., visual affordance map) to be non-zero only on the target object. While this enables TO-DQN to learn multi-step planning from a sparse reward (+1 for successful grasp and 0 otherwise), it is often over-fitted to implicitly identify the target and wall objects, limiting generalization to new objects and environments. In contrast, the method described in this paper explicitly leaves beliefs of the target and fixtures as an input to the network so as to prevent implicit over-fitting to objects observed in training.

2) Pushing and Grasping: A related area of object manipulation, multi-step tasks that require a combination of prehensile and non-prehensile motion primitives has received increased attention in recent years. Zeng et al. [6] train a pair of DenseNet-121 networks [7] to simultaneously predict the Q-value of pushing and grasping actions to grasp a target object in spite of adversarial clutter. [8] proposes a goal re-labeling scheme to overcome the reward delay problem in push-supported grasping. [9] trains a Generative Residual Convolutional Network (GR-ConvNet) [10] for each motion primitive (Push, Pick, and Place) to perform multi-step tasks such as clutter removal and cube stacking. [11] trains a motion critic that selects a primitive to arrive at grasping a target object even if it is initially occluded by clutter. In these works, when a target object is required for the task (e.g., grasping in clutter), the networks have been implicitly trained to recognize the target object according to some visual property (e.g., a distinct color). Other works also account for more natural target object specification. Danielczuk et al. [12] present a heuristics-based mechanical search to find and retrieve a target object identified using a Siamese Network [13] from a high clutter bin. We maintain generalizability to other target objects by following similar reference-based target recognition ideas, namely using a Siamese network with a cosine angle output.

While the aforementioned works primarily apply model-free reinforcement learning (RL) approaches, there also exist model-based RL approaches to object manipulation that seek to use machine learning to learn complex dynamics. The learned dynamics are then used in standard planning and search algorithms to generate a plausible action policy [14], [15], [16]. The work in [17] builds an Interaction Network using encoded object representations to push objects to target positions. However, a major drawback of such approaches is that learning accurate dynamics models is extremely challenging and the performance is sub-optimal if the learned models are inaccurate or biased which is often the case in novel settings.

III. PROBLEM FORMULATION

We formulate this problem as a Partially-Observable Markov Decision Process (POMDP) since the task requires a sequence of pushing actions to move the target object to a fixture in order to grasp it, while dealing with uncertainty in location of the target and fixture objects. A POMDP is characterized by the tuple $\langle S, A, T, R, Z \rangle$, where S defines the state of the workspace, A refers to the action space available to the robot, T is the state transition from s_t to s_{t+1} as a result of $a_t \in A$, R is the reward function, and Z refers to the observation space. In this work, we reduce the POMDP to a belief-state MDP [19] and solve the MDP problem using the current belief state in place of the true state. That is, the robot updates its belief b_t at time t given an observation $z_t \in Z$ of the current state $s_t \in S$. It then computes and executes an action a_t according to some policy $\pi(b_t)$, after which the environment transitions to some new state s_{t+1} and the robot receives an immediate reward

Authorized licensed use limited to: University of Minnesota. Downloaded on January 27,2023 at 19:42:51 UTC from IEEE Xplore. Restrictions apply.

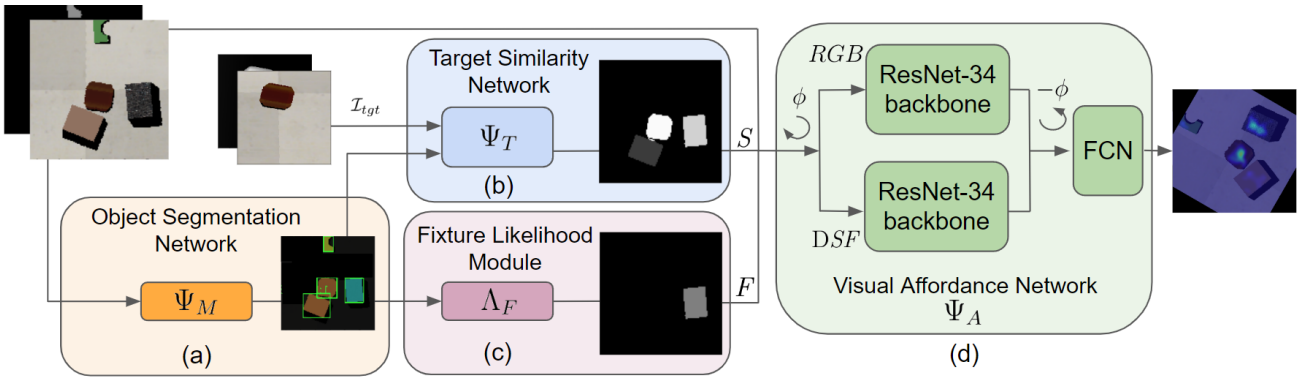


Fig. 2: FA-DDQN Overview. (a) Objects are segmented via Unseen Clustering Network [18]. (b) A dense target similarity map is computed by comparing each object segmentation to a reference image via Siamese Network with cosine angle output. (c) Object segmentation masks are matched across iteration pairs t and $t+1$, and the action-conditioned motion between them is used to generate a fixture likelihood map. (d) Concatenated RGB-D-S-F images are fed into the final DDQN network (through a rotation angle ϕ) to generate the rotation-conditioned, pixel-wise visual affordance map.

$r_t = R(s_t, a_t, s_{t+1})$. Therefore, the objective in this task is to learn an optimal policy π^* that maximizes the expected total return $R_t = \Sigma_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})$, where $\gamma \in [0, 1)$ is the discount factor.

For this particular task, observations of the state s_t arrive as RGB-D images of the workspace from a camera with known parameters. These images are converted to an orthographic view for use by the algorithm. Following [1], we adopt the use of the hybrid Shovel-and-Grasp (SaG) motion primitive to parameterize the action space as

$$a_t = (x, y, \phi) \tag{1}$$

where x, y denote the Cartesian coordinates of the action location in the workspace, and ϕ denotes the z-axis rotation angle. The remaining parameterizations (z, roll, pitch) are fixed hyper-parameters. The SaG primitive pushes for a fixed distance d with open grippers before closing the grippers and lifting vertically. By combining the pushing and grasping motions into a single primitive, we alleviate the need to learn over multiple action types, reducing the action space and easing the learning process.

For purposes of this work, we assume that the target object is always an object that cannot be grasped without the use of a supporting fixture object. That is, if the SaG primitive is performed on the target object without a fixture on the other side, the primitive reduces to a push action. To this end, we require that the workspace always contains at least one fixture object $\mathcal{F}$ in addition to the target object $\mathcal{O}$. The fixture objects do not conform to any particular form and can be visually diverse across episodes. The implication is that $\mathcal{F}$ cannot be inferred directly from the RGB-D images, and must be determined via interactions in the environment, thus leading to the POMDP formulation. Clutter objects $\mathcal{D}$, which are non-target objects that are also not fixtures, may also be present in the workspace image s_t and add complexity to the scene.

IV. METHOD

To solve the fixture-enabled grasping task, we propose Fixture-Aware Double Deep Q-Network (FA-DDQN), which consists of target identification and fixture prediction embedded within a Q-learning framework. Fig. 2 presents a visualization of our pipeline.

A. Object Segmentation

Our approach first identifies and segments all objects in the workspace using the RGB-D image $\mathcal{I}_t$. To do this, we opt to use the Unseen Clustering Network (UCN) proposed by Xiang et al. [18]. This is in contrast to the current standard of leveraging variants of Mask R-CNN [20], such as the depth-only variant [21] and the gray-depth-depth variant [1]. In addition to the strong results presented in [18], we empirically find that UCN segments objects in our environments more robustly than the gray-depth-depth Mask R-CNN of [1]. Instead of segmenting on the top-down image, we also perform the segmentation *before* transforming the image $\mathcal{I}_t$ into its orthographic projection, and simply transform the output masks and bounding boxes alongside the RGB-D pixels. This allows the segmentation network to operate on the pixel distribution directly from the camera viewpoint instead of the orthographic projection (which includes application of smoothing filters) and alleviates the need for fine-tuning and eases sim-to-real transfer.

B. Fixture-Aware Double Deep Q-Network

The action a_t is denoted with $a_t^{x,y,\phi}$ to describe the instantiated form with corresponding pixels in the workspace image and discretized rotation values (we use increments of $\Delta\phi = 5°$). With this definition, we employ a Deep Q-Network to generate the corresponding dense pixel-wise affordance map. That is, we train a network Ψ_A to learn to predict the action-value (Q-value) function as

$$Q^{\pi}(s, a|\theta) = \mathbf{E}_{s,a,\pi}\left[\Sigma_{t=1}^{\infty} \gamma^t R_t\right] \tag{2}$$

Authorized licensed use limited to: University of Minnesota. Downloaded on January 27,2023 at 19:42:51 UTC from IEEE Xplore. Restrictions apply.

where θ is the learned weights of the DQN. At each training step, the network parameters are updated according to

$$\theta_{t+1} \leftarrow \theta_t - \alpha \left(y_t^Q - Q(s_t, a_t|\theta_t) \right) \nabla_\theta Q(s_t, a_t|\theta_t) \quad (3)$$

with α as the learning rate and y_t^Q as the update target value. Inspired by [1], we observe that there is no value for the policy to act at unoccupied points in the workspace. Therefore we apply invalid action masking and constrain the update target value so that

$$y_t = \begin{cases} 0 & \text{if } a^{x,y,\phi} \notin \mathcal{G} \\ Q(s_t, a_t|\theta_t) & \text{if } a^{x,y,\phi} \in \mathcal{G} \text{ and } a^{x,y,\phi} \neq a_t \\ \hat{y}_t & \text{otherwise} \end{cases} \quad (4)$$

where $\mathcal{G}$ is defined to be the superposition of all object masks detected by the object segmentation network Ψ_M and represents occupied pixels. $\hat{y}_t$ is given by

$$\hat{y}_t = \begin{cases} r_t & \text{if } s_{t+1} \text{ is terminal} \\ r_t + \gamma Q(s_{t+1}, \arg\max_a Q(s_{t+1}, a|\theta)|\theta^-) & \text{otherwise} \end{cases} \quad (5)$$

where θ^- denotes the target DQN that is updated at a slower rate than the primary DQN under training. Note that we utilize the Double DQN formulation for the maximum expected Q-value [22] in order to reduce the well-known overestimation issue of vanilla DQN.

In addition, we note that our constraint on the update target value differs from [1] wherein the authors restrict the update target value by only the target object mask. By using $\mathcal{G}$ instead of only the target object mask, we allow our network to potentially act on other objects in the scene (e.g., to determine if it is a fixture), while still preserving the benefit of focusing training only on actions that will explicitly change the environment. By setting y_t to zero in unoccupied pixels, we also provide feedback to the back-propagation and directly teach the network to suppress background activations (though only gently through a small scale factor much less than one).

Since the target and fixture objects are hidden states, providing only the visual inputs (RGB-D images $\mathcal{I}_t$) as is typically done in works like [23], [4] is insufficient and leads to implicit learning of those objects. Instead we augment the inputs with a target similarity channel and a fixture probability channel to form a six-channel input (RGB-D-S-F) into the proposed FA-DDQN, thus offloading the inference of target and fixture objects from the DQN.

C. Target Similarity Channel

We build upon the target-agnostic identification method of [1] which relies on a Siamese Network [13] to compute the similarity of an object image to a given anchor image $\mathcal{I}_{tgt}$. We refer to this network in this paper with Ψ_T. Although the typical approach directly ranks the similarity scores and selects the best match as the target [1], [24], we propose instead to form another image channel—the target similarity channel S—to provide as additional input to FA-DQN. Much like the original Siamese network of [13], our implementation uses cosine angle as the distance metric, computed between the L2-normalized embedding vectors of a pair of images. By using cosine angle instead of the standard euclidean distance (L1- or L2-based) to measure similarity, we arrive at an arguably more natural formulation of likelihood that is suitable for ingestion by a neural network. This formulation is given by:

$$similarity(\mathcal{I}_1, \mathcal{I}_2) = max(\Psi_T(\mathcal{I}_1) \cdot \Psi_T(\mathcal{I}_2), 0) \quad (6)$$

where $\cdot$ indicates the dot product operation, and magnitude normalization to one has been assumed as an output of Ψ_T. While the cosine similarity function produces values between 1 (identical objects) and -1 (polar opposites), we clip all negative values to zero to produce a target similarity score $similarity(\mathcal{I}_1, \mathcal{I}_2) \in [0, 1]$. Thus, target similarity is naturally scored based on the projection to the reference object. As an added benefit, this similarity score is directly generalizable to output vectors of any length without modification since the cosine angle is a reduction from an N-dimensional space to a 2-dimensional space.

The target similarity channel relaxes the need to hand-select a fixed best target or implicitly teach the network on a fixed target, while leaving the choice up to the RL network to focus on any particular action. Furthermore, the use of the Siamese network enables zero-shot generalization to new target objects by simply swapping the anchor image used.

D. Fixture Likelihood Channel

To relieve the FA-DDQN from having to implicitly learn fixture objects, we provide an additional channel containing the current belief map of pixels likely to belong to the set of fixtures $\mathcal{F}$. Although the fixture object location is a hidden state and not directly observable from the input RGB-D image, its suitability for leveraging against can be identified quickly via physical interaction. If an object moves after the manipulator applies a force on it, it is reasonable to expect that it cannot be used as a fixture. On the other hand, an object that remains in place even after an interaction is likely a fixture object. While direct interaction by the robot is important, we must also account for secondary effects due to object-environment interactions. Finally, we propose to begin each episode with a neutral belief of fixture likelihood. That is, we make no assumptions about fixture likelihood and initialize each object in the scene to $l_o^- = 0.5$, where $l_o \in [0, 1]$ indicates the fixture likelihood of an object o (0 corresponds to a movable object and 1 corresponds to a fixture).

To determine the motion of an object, correspondence between object masks across subsequent time steps must first be identified. This correspondence is found by solving the linear sum assignment problem using the pairwise distances between mask centers of objects segmented from subsequent iterations t and $t+1$ [25]. Once masks belonging to the same object in t and in $t+1$ have been identified, the algorithm described in Algorithm 1 is applied to track and formulate the fixture likelihood map F_{t+1} that will be provided to FA-DDQN in iteration $t+1$. While simple, this fixture likelihood

Authorized licensed use limited to: University of Minnesota. Downloaded on January 27,2023 at 19:42:51 UTC from IEEE Xplore. Restrictions apply.

module (referred to herein also as Λ_F) sufficiently captures the results of the interactions with the scene to inform FA-DDQN.

E. Visual Affordance Learning

Finally, S and F are passed to the FA-DDQN visual affordance network along with the RGB-D image $\mathcal{I}$ to produce a Q-map for a given rotation. The $6 \times W \times H$ six-channel input image is first pre-rotated by the evaluated rotation angle ϕ before passing it through a ResNet backbone. In particular, we use two ResNet-34 [26] trunks, one for the RGB channels and the second to process the depth, S, and F channels. The shovel-angle-conditioned feature maps are then added together to blend the embeddings from both trunks, rotated back by $-\phi$, and finally passed through Fully Convolutional Networks (FCNs) [27] to produce the output visual affordance map $Q(s_t, a_t^{x,y,\phi}|\theta_t)$. By evaluating the DQN portion for varying rotations $\phi \in \{0°, 5°, 10°, ..., 90°\}$, we build a discretized affordance table of size $19 \times W_a \times H_a$ where W_a and H_a denote the width and height of the action space coordinates. The optimal action according to the current policy π is selected via

$$a_t^{x,y,\phi} = \underset{x,y,\phi}{\arg\max}\, Q(s_t, a_t^{x,y,\phi}|\theta_\pi). \tag{7}$$

As will be detailed in a following section, this policy will be trained in an ϵ-greedy fashion.

Algorithm 1 Fixture Likelihood Module (Λ_F)

Require: $\{o_t,\, o_{t+1}\}\ \forall\ o \in \mathcal{O}$ ▷ Matched object masks
Require: $l_o^-\ \forall\ o \in \mathcal{O}$ ▷ Prior fixture likelihoods
  $o_t^\pi \leftarrow$ object directly acted on
  for $o_t,\, o_{t+1}$ in $\mathcal{O}$ **do**
    if o_t is o_t^π **then** ▷ Object was acted on
      if $||o_{t+1} - o_t||_2 < \delta$ **then**
        $l_o \leftarrow min(l_o^- + 0.25, 1.0)$
      end if
    else ▷ Handle secondary interactions
      if $was\_in\_way(o_t, o_t^\pi)$ and $||o_{t+1} - o_t||_2 < \delta$ **then**
        $l_o \leftarrow min(l_o^- + 0.25, 1.0)$
      end if
    end if
    if $||o_{t+1} - o_t||_2 \geq \delta$ **then**
      $l_o \leftarrow 0$ ▷ Object moved
    end if
  end for

V. EXPERIMENTS

For training and evaluation in simulation environments, we leverage a UR5 manipulator rendered in CoppeliaSim [28] v4.1.0 with a parallel gripper attachment. The networks are constructed using PyTorch [29] v1.9.0.

A. Training

1) Object Segmentation Network: As previously discussed in Section IV-A, we feed in the RGB-D workspace image to UCN prior to orthographic projection to retrieve instance segmentation masks. No fine-tuning was found to be necessary and the pre-trained weights accompanying [18] are used.

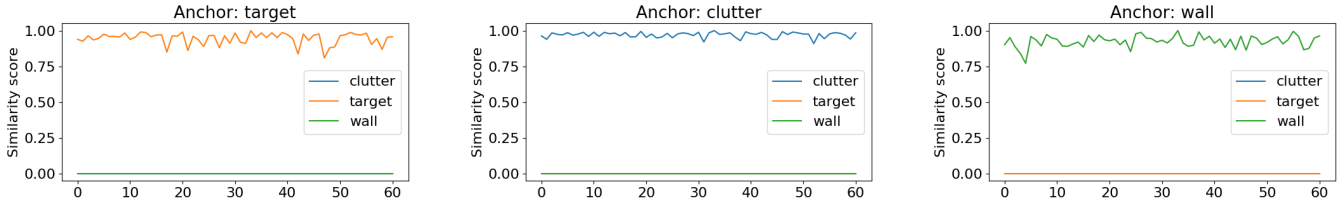
2) Target Similarity Network: To train the target similarity network Ψ_T, 180 RGB-D images were collected of single, randomly positioned fixtures, targets, and clutter objects in simulation. Triplets of anchor, positive, and negative samples were then pulled from the captured dataset to train using Triplet Loss [30], with cosine angle (not truncated at 0) as the distance metric. The network is thus trained to maximize the embedded vector differences between classes. For this task, we chose to train the Siamese network to distinguish between three broad classes (target, fixture, and clutter objects) so that the network remains flexible enough to recognize an object at any pose. The resulting similarity likelihood values between objects of same and different classes are shown in Fig. 3.

3) FA-DDQN: After training the preceding networks, they are integrated into the FA-DDQN architecture in order to support training of the final visual affordance network Ψ_A. Since we aim to learn a pixel-wise visual affordance map of a sparse reward problem, a fixed-step learning curriculum is adopted. To wit, the learning process is split into three phases: 1) single fixture only, 2) two fixtures, 3) two fixture objects but one is un-anchored and unusable. By introducing this multi-phase curriculum, we enable progressive fine-tuning of FA-DDQN from pushing towards a known fixture to handling multiple potential fixture objects. This also increases the density of successful grasps for experience replay to sample from, helping to combat the sparse reward problem. During the entire training process, however, we ensure the presence of one target block, one fixture (to ensure the episode has a success trajectory), and one clutter object.

An episode is defined as a series of sense-compute-act iterations (i.e., one full execution of the FA-DDQN pipeline) that terminates and resets when (a) the target object has been successfully grasped (lifted out of the workspace and remains in the gripper), (b) the target object is pushed out of the workspace, or (c) K iterations have passed (we use $K = 8$). Training on-policy takes place using an ϵ-greedy policy. That is, at every iteration there is a probability of ϵ to select a random action, with ϵ decaying at a rate of 0.997 after the first 100 iterations, and a discount factor of $\gamma = 0.95$. The slow ϵ decay rate is selected to allow for some exploration even in the later phases of training. Training is executed for 4000 iterations. In this work, we leverage Prioritized Experience Replay (PER) [31], using stochastic proportional-based prioritization with importance-sampling, in replay buffer sizes of 10 samples.

As with any RL paradigm, the reward function provides an important signal for learning an affordance predictor. Here, the reward function rewards the RL agent only for terminating in a successful grasp or changing the fixture likelihood of an object. The change in fixture likelihood happens if

Authorized licensed use limited to: University of Minnesota. Downloaded on January 27,2023 at 19:42:51 UTC from IEEE Xplore. Restrictions apply.



(a) Target objects (b) Clutter objects (c) Fixtures

Fig. 3: Cosine-based Siamese Network similarity scores on validation data. All non-matching classes are successfully suppressed to 0, while retaining a high score for objects in the same class.

the manipulator interacts with an object and the resulting predicted likelihood ($\hat{l}'^{t}_{o,a^{x,y,\phi}}$) is sufficiently different from the prediction prior to the interaction ($\hat{l}^{t}_{o,a^{x,y,\phi}}$). This is mathematically represented as:

$$r_t = \begin{cases} +1 & \text{if grasp success} \\ +0.25 & \text{if } |\hat{l}'^{t}_{o,a^{x,y,\phi}} - \hat{l}^{t}_{o,a^{x,y,\phi}}| > 0.2 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

B. Simulation Evaluation

We begin evaluation of FA-DDQN by empirically exploring its trained performance in simulation. The simulation environment is identical to the training environment except that the numbers of clutter objects uniformly vary between 0 and 2 across the workspace. To obtain statistical measures, we run 4 trials of 50 episodes and report the results in Table I. In these trials, the fixture object is sampled from the same object type distribution that the model has been trained on. For comparisons, we include similar evaluations of TO-DQN and ablations of FA-DDQN:

- *Random*: A baseline that randomly samples an action position from the target object mask, as determined by the object with the highest target similarity value ($\arg\max_{o \in \mathcal{O}} S$). The action direction is also uniformly sampled. This baseline serves as a validation to quantify the minimum level of performance expected by examined algorithms.
- *TO-DQN*: Re-trained in the same simulation environment as FA-DDQN, using the same training parameters that were indicated in [1].
- *FA-DQN*: This ablation uses the vanilla DQN update target y_t instead of the Double DQN form.
- *FA-DDQN-no-curriculum*: This ablation of FA-DDQN does not use the curriculum for training.
- *FA-DDQN-zero*: This ablation forces both the fixture likelihood and target similarity channels to be zero, to verify that the added channels do indeed provide information to the network.
- *FA-DDQN*: Our proposed end-to-end RL pipeline.

We also introduce another experiment wherein the aforementioned models are tested against an unseen fixture object to evaluate the generalizability of FA-DQN to new configurations. To be precise, a cylindrical wall is used as the only fixture in all 50 episodes in this experiment and the trained network must be able to utilize this information to continue grasping the target object. This is depicted in Fig. 4, and we include these results also in Table I.

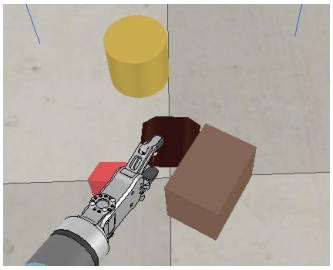


Fig. 4: Sample episode from evaluations with novel fixture. In these evaluations, a cylinder is always introduced as the fixture at random positions in the workspace.

TABLE I: Test Evaluation Comparisons. Simulated trials with known fixtures (4 trials of 50 episodes) and novel fixture objects (1 trial of 50 episodes)

	Average Grasp Success (%)	
Model	Known Fixture	Novel Fixture
Random	42.0 ± 9.4	31.5 ± 3.0
TO-DQN [1]	64.0 ± 3.5	41.0 ± 7.0
FA-DQN	65.7 ± 6.7	48.0 ± 7.2
FA-DDQN-no-curriculum	58.2 ± 8.4	43.5 ± 8.9
FA-DDQN-zero	54.0 ± 2.0	35.0 ± 7.0
FA-DDQN (ours)	**75.0 ± 6.4**	**53.0 ± 7.7**

It can be seen in Table I that although TO-DQN performs better than the random baseline and manages to push the target object towards fixtures (or at least what it believes to be so), it is still outperformed by our proposed FA-DDQN. With the additional channels of information, FA-DDQN is better able to grasp objects while leveraging fixtures, especially if the fixture is novel and not encountered in training. We note that all models encounter decreased performance in the novel object case, which is in part due to the more challenging nature of a cylindrical wall (less likely to get a stable grasp). This indeed supports the proposal that FA-DDQN resists implicit learning of fixtures and supports better generalization to new configurations.

C. Real-world Testing

We also examine the sim-to-real generalization of FA-DDQN by deploying the trained network to control a Franka

Authorized licensed use limited to: University of Minnesota. Downloaded on January 27,2023 at 19:42:51 UTC from IEEE Xplore. Restrictions apply.

(a) One fixture only (b) One fixture, one decoy (c) One cylinder as fixture (d) Novel target (e) Novel target w/ decoy

Fig. 5: Sim-to-real test scenes. Setups for real experiments, where (d) and (e) utilize a novel target.

TABLE II: Real-robot Results. Results from the real-robot experiments described by Fig. 5 where case A is one fixture only, case B is one fixture and one decoy, and case C is one cylinder (10 episodes for each case).

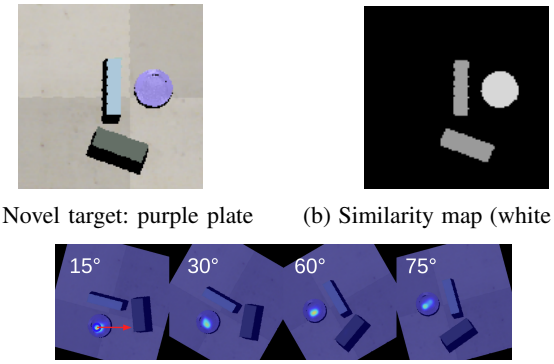
	Grasp Success (%)		
Target Object	Case A	Case B	Case C
Known target (black parcel)	80.0	70.0	50.0
Novel target (cat bowl)	80.0	50.0	-

Emika Panda robot manipulator that has been outfitted with a parallel-gripper end effector. To this end, several test scenes are constructed with varying clutter objects and fixture objects. We evaluate on a scene with only one fixture (Case A, Fig. 5a) for 10 trials, followed by 10 trials of a second scene with two visually identical boxes but only one of them being heavy enough for leveraging (Case B, Fig. 5b). We also perform experiments with a cylindrical fixture (Case C, Fig. 5c) akin to the simulated experiments. While executing FA-DDQN on the real robot, we restrict the robot from carrying out its intended motion if the action target lies on the fixture. This is done to protect the robot, and otherwise allow the robot to execute actions on the rest of the scene.

In these real experiments, we achieve successful grasp rates as reported in Table II. By comparing these results to Table I, it is apparent that the real world evaluations are on par with simulation results. We propose that this real-world generalizability is enabled by the additional input beliefs that also de-emphasize the RGB channels, which are notorious for poor extension from simulation to the real world. Hence, FA-DDQN is able to operate in the real-world without additional fine-tuning (though it may boost performance to some extent).

D. Novel Target Objects

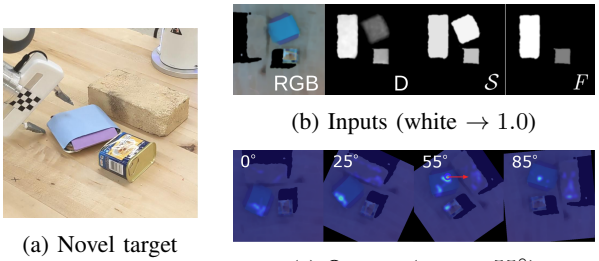
In the preceding section, we validated the usability of novel fixtures. In this section, we validate the use of novel target objects that were not seen in training. In simulation, we switch the target object with a purple plate and verify that the actions correctly focus on the new plate (after the reference image has been updated). Fig. 6 shows the input scene with the plate as the target. Note that the plate correctly registers as the object with the highest target similarity in Fig. 6b. The resulting Q-maps (Fig. 6c) also show that the high Q-values on the center of the plate, while pointing in the direction of the fixture.



(a) Novel target: purple plate (b) Similarity map (white → 1)

(c) Q-maps at various angles activating on plate

Fig. 6: Zero-shot generalization to novel target. A plate is used as the target object in this experiment. Red arrow in (c) indicates the chosen action direction.



(a) Novel target

(b) Inputs (white → 1.0)

(c) Q-maps (max at 55°)

Fig. 7: Demonstration with novel target. Grasping a novel target unseen in training (a), with input images (b) and resulting Q-maps in (c).

In addition, we also repeat several real-robot experiments with two novel target objects to further demonstrate zero-shot generalization to new targets via the Target Similarity channel. A sample of this is shown in Fig. 7a, wherein the robot successfully targeted the blue target object. The second novel target object for real-world experiments is the cat bowl shown in Fig. 5d. We perform the same set of experiments of grasping against a single wall and a single wall and decoy, and report the results in Table II.

VI. DISCUSSION AND FUTURE WORK

In this work, we introduced a new network architecture, Fixture-Aware DDQN, to tackle the challenging fixture-assisted grasping problem. A key component of FA-DDQN is the explicit handling of likely target and fixture objects as inputs to the Double Deep Q-Network, instead of relying on

Authorized licensed use limited to: University of Minnesota. Downloaded on January 27,2023 at 19:42:51 UTC from IEEE Xplore. Restrictions apply.

implicit learning of the task space. By handling the target and fixture objects outside of the network, we improve both flexibility and performance over the existing baselines.

A major benefit of the Shovel-and-Grasp (SaG) action primitive is that it allows the task to be reduced to finding the optimal shovel direction and location, simplifying the action space and complexity. However, one limitation encountered in a few corner cases is the lack of grasp stability when there is a certain distance between the target object and the fixture. Due to the fixed pushing distance in SaG, most of the action is expended on pushing the target object before the object contacts the fixture. Then the target object is only partially scooped up in the gripper, resulting in a weak grasp attempt. One future avenue to handling this limitation (which is a limitation of any push action) is to make push distance a part of the action space, so that the agent can select how far to push before completing the shovel primitive.

Learning to leverage objects in the environment to aid in a grasping task is a complex skill that requires object-environment interaction consideration and uncertainty handling over a longer horizon than traditional grasping tasks. Because of that, progress on the fixture-enabled grasping task lends to real-world application and helps bring robots closer to mimicking the resourcefulness of manipulation by humans. With that in mind, we believe that our work presented here, FA-DDQN, is the next step in that progress.

REFERENCES

[1] H. Liang, X. Lou, and C. Choi, "Knowledge induced deep q-network for a slide-to-wall object grasping," *arXiv preprint arXiv:1910.03781*, pp. 1–7, 2019.

[2] C. Eppner, R. Deimel, J. Alvarez-Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1021–1038, 2015.

[3] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," *arXiv preprint arXiv:2010.14406*, 2020.

[4] J. Zhang, W. Zhang, R. Song, L. Ma, and Y. Li, "Grasp for stacking via deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2543–2549, IEEE, 2020.

[5] C. Eppner and O. Brock, "Planning grasp strategies that exploit environmental constraints," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4947–4952, IEEE, 2015.

[6] 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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4238–4245, IEEE, 2018.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[8] K. Xu, H. Yu, Q. Lai, Y. Wang, and R. Xiong, "Efficient learning of goal-oriented push-grasping synergy in clutter," *arXiv preprint arXiv:2103.05405*, 2021.

[9] S. Kumra, S. Josh, and F. Sahin, "Learning robotic manipulation tasks through visual planning," *arXiv preprint arXiv:2103.01434*, 2021.

[10] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9626–9633, IEEE, 2020.

[11] Y. Yang, H. Liang, and C. Choi, "A deep learning approach to grasping the invisible," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2232–2239, 2020.

[12] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martín-Martín, A. Garg, S. Savarese, and K. Goldberg, "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 1614–1621, IEEE, 2019.

[13] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, pp. 669–688, 1993.

[14] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani, "Object-centric forward modeling for model predictive control," in *Conference on Robot Learning*, pp. 100–109, PMLR, 2020.

[15] B. Huang, S. D. Han, A. Boularias, and J. Yu, "Dipn: Deep interaction prediction network with application to clutter removal," *arXiv preprint arXiv:2011.04692*, 2020.

[16] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[17] J. Wang, C. Hu, Y. Wang, and Y. Zhu, "Dynamics learning with object-centric interaction networks for robot manipulation," *IEEE Access*, vol. 9, pp. 68277–68288, 2021.

[18] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, "Learning rgb-d feature embeddings for unseen object instance segmentation," *arXiv preprint arXiv:2007.15157*, 2020.

[19] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[21] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7283–7290, IEEE, 2019.

[22] H. Hasselt, "Double q-learning," *Advances in neural information processing systems*, vol. 23, 2010.

[23] 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.

[24] X. Lou, Y. Yang, and C. Choi, "Collision-aware target-driven object grasping in constrained environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6364–6370, IEEE, 2021.

[25] D. F. Crouse, "On implementing 2d rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[28] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. www.coppeliarobotics.com.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[30] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking.," *Journal of Machine Learning Research*, vol. 11, no. 3, 2010.

[31] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

Authorized licensed use limited to: University of Minnesota. Downloaded on January 27,2023 at 19:42:51 UTC from IEEE Xplore. Restrictions apply.