Data-driven Grasping and Pre-grasp Manipulation Using Hierarchical Reinforcement Learning with Parameterized Action Primitives

Shih-Min Yang, Martin Magnusson, Johannes A. Stork, Todor Stoyanov

***Abstract*— In many real-world robotic grasping tasks, the target object is not directly graspable because all possible grasps are obstructed. In these cases, single-shot grasp planning will not work, and the object must first be manipulated into a configuration that allows for grasping. Our proposed method ED-PAP solves this problem by learning a sequence of actions that exploit constraints in the environment to change the object's pose. Concretely, we employ hierarchical reinforcement learning to distill controllers that apply a sequence of learned parameterized feedback-based action primitives. By learning a policy that decides on a sequence of manipulation actions, we can generate a complex manipulation behavior that exploits physical interaction between the object, the gripper, and the environment. Designing such a complex behavior analytically would be difficult. Our hierarchical policy model operates directly on depth perception data without the need for object detection or pose estimation. We demonstrate and evaluate our approach for a clutter-free table-top scenario where we manipulate a box-shaped object and use interactions with the environment to re-orient the object in a graspable configuration.**

I. INTRODUCTION

State-of-the-art systems for autonomous grasp acquisition [1], [2], [3] function well in moderately cluttered scenes but are fundamentally limited in assuming that objects are directly graspable — a situation that arises often in practice, for example, in cases when objects are tightly packed together, or placed in configurations that invalidate all feasible grasps (e.g., think of a book wider than the opening of the gripper, lying flat on a table). To address such practically relevant scenarios the robot arm needs to re-arrange objects in a non-prehensile manner, which poses unique challenges to perception, planning, and control.

Current non-prehensile object rearrangement methods use trial-and-error learning to deal with the stochastic and unpredictable nature of in-contact physical interactions [4], [5], [6], [7]. As black-box reinforcement learning involving contact dynamics has prohibitively high interaction sample complexity, a common solution is to employ manually designed parametric controllers dubbed *behavior primitives*. However, this has two disadvantages compared to the more general end-to-end approaches: first, it limits applicability to only tasks that can be solved by existing primitives; and second, it necessitates expert input in designing, implementing, and tuning the primitive controllers. While some progress has recently been made in alleviating the first shortcoming through e.g., the use of atomic actions to "stitch" together primitives [7], the need for expert input in primitive design still poses a major challenge.

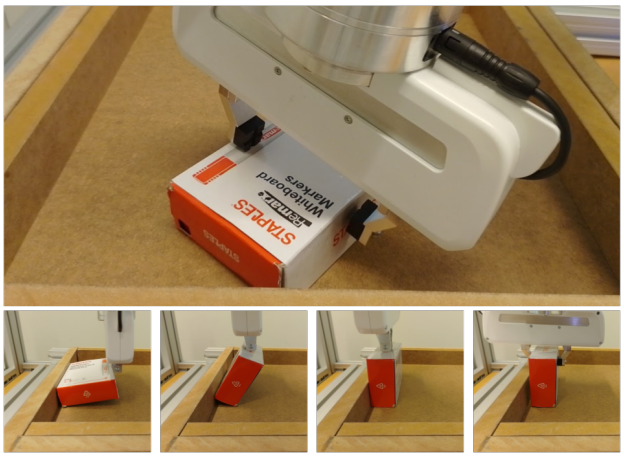

Fig. 1. *Top:* In the initial pose, all feasible grasps on the target object are obstructed by the environment. *Bottom (left to right):* We learn to use a wall to flip the object up and grasp it from the top.

Instead of relying on manually-defined behavior primitives or resorting to costly end-to-end reinforcement learning (RL), we take a middle ground and learn hierarchical control policies. This allows us to maintain the generality of full-scale RL while imposing strong inductive biases in terms of the decomposition of tasks to a number of parametric primitives with associated lower-dimensional state-action spaces. We solve a variation of the *occluded grasping* task [8] in which a robot arm equipped with a parallel jaw gripper needs to pick a flat object placed on a table-top (see Fig. 1). As the object is only graspable along approach directions that collide with the table, the task can only be solved via non-prehensile manipulation and interaction with the environment: the robot needs to slide the object, push it against one of the boundaries, pivot to flip and finally grasp it from the top. We train our approach in simulation through curriculum learning [9] and apply Automatic Domain Randomization [10] to enable zero-shot transfer to the real world.

Our contributions are thus: first, we propose an approach for learning hierarchical manipulation policies that allow for imposing intuitive inductive biases without relying on expert controller design; and second, we instantiate the proposed framework to the task of picking objects from non-graspable configurations and demonstrate that our approach is able to efficiently solve the problem for a variety of object instances.

The authors are with the Center for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. Corresponding author: shih-min.yang@oru.se.

*This work has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101017274 (DARKO), and was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

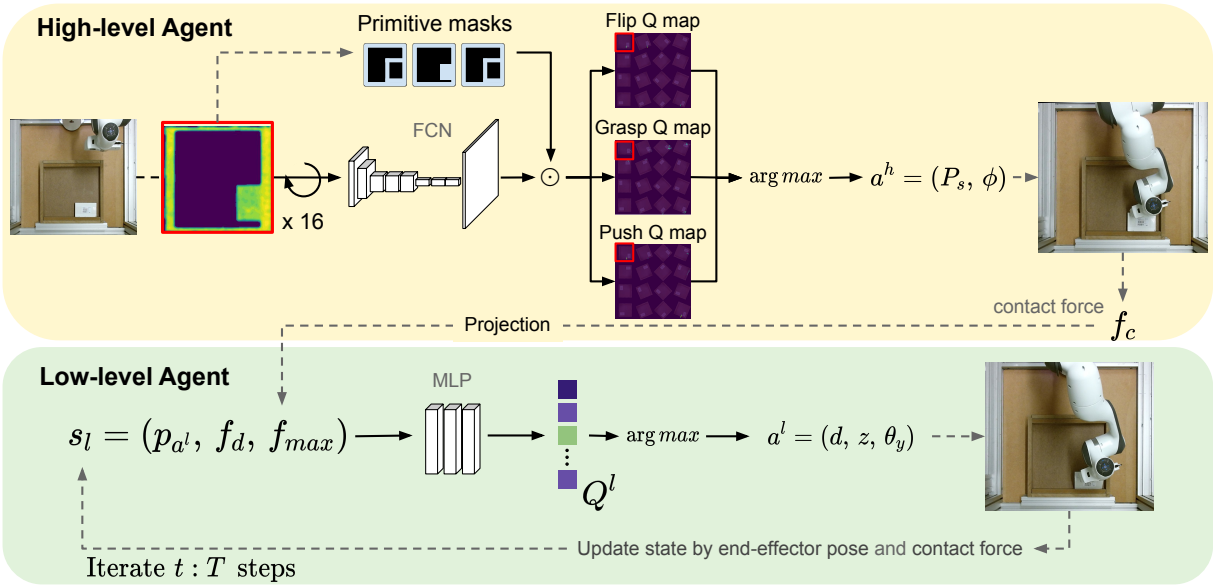

Fig. 2. **Overview**. Our proposed ED-PAP method includes a high-level agent and a low-level agent. The high-level agent takes a height map as input to an FCN model and outputs pixel-wise maps of Q values, with each pixel corresponding to a starting pose and a primitive. The low-level agent records the starting pose given by the high-level agent as the base frame of reference and combines the current end-effector pose and contact force as the state to estimate a series of actions for accomplishing sub-task by a DQN model.

II. RELATED WORK

A. Primitive-based robotic manipulation

Reinforcement learning for robotic manipulation poses a significant challenge due to the difficulty of effectively exploring in a high-dimensional continuous action space, which can lead to inefficient learning. To address these problems, early works [4], [5], [11], [12], [13], [6] have explored reinforcement learning for manipulation using pre-defined primitives to avoid exploring a high-dimensional continuous action space. Recent work [7] applies hierarchical reinforcement learning [14] for separating the primitive and the estimation of its parameters to improve performance. Although these works demonstrate significant results, they all rely heavily on pre-defined behavioral primitives. Manually designed complex behavioral primitives often requires human expertise and takes a significant amount of time and effort. In this paper, we learn the behavior of a contact-rich primitive for flipping the flat object by hierarchical reinforcement learning without manually designing it.

B. Extrinsic dexterity for grasping

To enhance the ability of a robot hand for robotic manipulation, extrinsic dexterity [15] is a type of skill that exploits external resources to assist manipulation. Early works [16], [17], [18] have explored environmental constraints exploitation such as slide-to-wall and slide-to-edge to assist grasping. Some recent works [19], [20] learn a policy to grasp flat objects based on visual information. However, these works rely on simple visual servoing to initiate grasping, assume the object position is given, or need a specific gripper design. In our work, we use the standard Franka Emika gripper for grasping flat objects based on visual information without a given object position or grasp pose. The work by Zhoua and Held [8] is closely related to ours, as it also targets grasping objects placed in unfavorable configurations through reinforcement learning. They propose to learn a policy to achieve a given target grasp configuration that is initially occluded by the table. However, the approach presented has several limitations: the target object needs to be placed close to the wall, a target grasp configuration needs to be available, and the object pose has to be tracked throughout the interaction. In contrast, we combine different primitives to overcome the assumption of object-to-wall proximity and demonstrate our approach is able to efficiently solve the problem without access to a target grasp or object pose estimate.

III. METHOD

We address the *occluded grasping* problem using hierarchical DQN [14]. A high-level agent selects pose-parametrized *primitives*, while low-level agents operate in primitive-specific state-action spaces to accomplish sub-tasks. We use 3 primitives: a *flip primitive* that uses contact with the environment to pivot an object; a *top-down grasp primitive* that picks directly graspable objects, and a *push primitive* that achieves in-plane object motion. We employ a low-level DQN agent to learn the complex *flip primitive* and design the other two manually. This allows us to achieve good performance while keeping the system simple.

A. Manipulation with Parameterized Primitives

An overview of our method is shown in Fig 2. Given an input depth image of the workspace, the goal of our high-level policy is to pick a starting pose and choose an appropriate primitive to apply. We use a Fully Convolutional Network (FCN) [21] as our high-level policy model, inspired by Ren et al. [5]. The action space for the high-level model is formulated as a tuple (P_s, ϕ), where $\phi \in \{\phi_f, \phi_g, \phi_p\}$

is a discrete choice between the flip primitive ϕ_f, the top-down grasp primitive ϕ_g, or the push primitive ϕ_p; while $P_s = (x, y, \theta_i)$ encodes an end-effector pose corresponding to a translation to the (x, y)-th pixel of the depth image and a rotation of $\theta_i = 2\pi i/K$ rad around the z-axis.

We pre-process the depth image by transforming it to a robot-centric coordinate frame, passing it through a noise smoothing filter, and projecting it to a height map. We then rotate the height map K times, concatenate the results, and pass them to the FCN. The FCN outputs a batch of Q maps, each corresponding to a possible primitive choice (3 in this paper) for each rotated height map. Each pixel in the Q map corresponds to a starting pose P_s and a primitive id ϕ. The optimal action is the pixel with the maximum Q-value, potentially within a masked region of interest.

We train the high-level agent using a sparse reward. Successful actions with the flip primitive and the top-down grasp primitive are rewarded by $r_t^{H_f} = 1$ and $r_t^{H_g} = 1$ respectfully. The reward of successful push primitive actions is set to a value of $r_t^{H_p} = 0.2$ on success and $r_t^{H_p} = 0.1$ on a change in the workspace configuration. The flip primitive is considered successfully executed if it results in the object being reoriented vertically. The top-down grasp primitive succeeds when the target object is acquired between the gripper fingers. Finally, the push primitive is successful if the object is moved in the proximity of a workspace wall.

During training, we employ the ϵ-greedy strategy to explore the different actions. To reduce the region that the agent needs to explore, we adopt the masking approach from Ren et al. [5]. The idea is that the agent only has to explore pixels nearby the target object. Thus, we calculate the mask from the height map and decide the affordance region for each primitive based on the pixel values.

B. Learning behavior for a contact-rich primitive

A key distinguishing feature of our approach is that we do not rely solely on expert-devised behavior primitives, but rather learn low-level action policies in conjunction with the high-level policy from the previous section. In this paper we train only one such low-level policy — for the flipping primitive; though extensions to multiple learnable low-level policies are in principle possible.

The low-level primitives encode motion policies that can be learned via black-box RL with a state-action space based on the end-effector pose. While this would be a general approach, it is likely to require a large amount of interaction experience (high-dimensional state and action spaces) and poses challenges for learning in conjunction with the high-level policy. We avoid these issues by encoding a strong inductive bias in devising the state and action spaces of the low-level policy. We formulate the action space as the end-effector velocity in a 3D task space composed of the distance d, height z, and angle θ_y relative to the primitive starting pose. The possible actions are then discrete steps along $d \in \{0, a_d\}$ to move forward; $z \in \{0, a_z\}$ to move up; and the $\theta_y \in \{-r_y, 0, r_y\}$ to rotate along the y-axis. We formulate the state as $s_l = (p_{a^l}, f_d, f_{max})$, where $p_{a^l} \in \mathbb{R}^3$

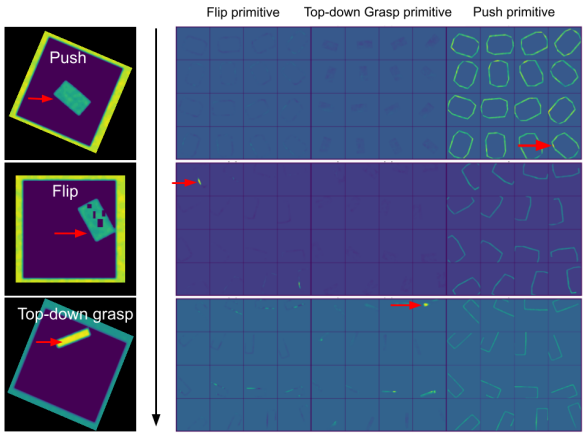

Fig. 3. An example sequence of picking up a flat object (in simulation). The left column shows the current observation and the right column contains the estimated Q maps for each of the three primitives across 16 discrete orientations.

is the current end-effector task-space pose; f_d is the contact force along the d-axis; and f_{max} is the maximum of the current contact force.

To learn the low-level action policy, we could in a straightforward manner define the reward sparsely on successful flips. However, as sparse rewards result in less efficient learning [22], we choose to instead design a reward function that considers the contact force and end-effector position:

$$r_\tau = r_t^{H_f} + r_\tau^L \tag{1}$$

$$r_t^H = \begin{cases} 1 & \text{, if flip success} \\ 0 & \text{, otherwise} \end{cases} \tag{2}$$

$$r_\tau^L = \begin{cases} \min(\sigma, \frac{z_\tau \sigma}{w}) & \text{, if } f_c > 0 \\ -1 & \text{, if } f_c > f_{limit} \\ 0 & \text{, otherwise} \end{cases} \tag{3}$$

where f_c is the current contact force, f_{limit} is the maximum safety contact force, z_τ is the current end-effector height, and σ and w are hyper-parameters that normalize the z_τ and limit the upper bound of the reward. Based on this reward function, we encourage the agent not only to flip the object up but also to pivot it while maintaining contact and avoid applying too much contact force. We find that without the penalty for applying excessive contact force, the robot may trigger emergency stops in the real world, making the transfer of policies learned in simulation more challenging.

C. Curriculum learning and domain randomization

Training the high-level and low-level models simultaneously can be difficult, as it is hard to determine whether a failure is due to the high-level agent or the primitive behavior. To address this, we train the two models separately, following a curriculum learning [9] approach. We train the low-level agent first, devising progressively more complex interaction scenarios and only include the high-level model once the low-level policy can successfully flip objects.

To adapt our method to the domain shift between simulation and the real world, we employ Automatic Domain

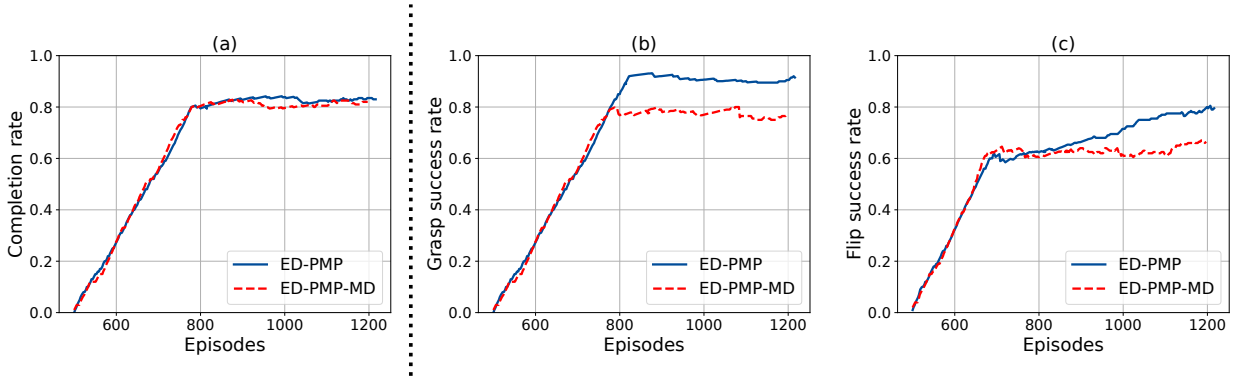

Fig. 4. Testing curve of success rate versus training transitions of the high-level model in simulation. (a) Full-task success rate (successfully picking the object with 10 or less primitives). (b) The success rate for the grasp primitive. (c) The success rate for the flip primitive. Success rates of the primitives are computed over the last 100 attempts.

Randomization (ADR) [10] during simulation training. ADR randomly samples the object size, friction, and mass in each episode, making our method more robust to variations in real-world objects. We also add Gaussian noise and randomly block a few regions in the simulated depth image to simulate the noise and reflections of the real-world height map.

IV. EXPERIMENTS

A. Experimental Setup

We address the *occluded grasping* task by flipping a large flat object and grasping from the side as shown in Fig. 1. To demonstrate our model's ability to learn extrinsic dexterity skills, we use a Franka Emika Panda arm with a 2-finger gripper which is not wide enough to grasp large flat objects by top-down grasping directly. For the environment setup, we put a 44.8×44.8 cm wooden box with four boundaries as the workspace (see Fig. 1) and mount a camera directly above it. An overhead depth image is captured by a Kinect v2 camera and transformed into a height map as input to the high-level model. To avoid the robot occluding the workspace, we move the robot to one corner before acquiring the depth map.

To evaluate our approach, we test our method in the real world with 5 box-shape objects with different weights, sizes, and friction. We compare our proposed method ED-PAP with an alternative instance (denoted ED-PAP-MD) where the learned flip primitive is replaced by a manually designed version, in order to demonstrate that the learned low-level policy performs better than the manually designed one.

We test our method in two setups: *close*, where the object is placed next to one random wall of the box; and *random* where the object is placed randomly near the center of the workspace. We use the completion rate as the evaluation metric and test 10 episodes for each object. An episode is considered successful if the robot picks up the object after applying up to 10 primitive actions.

B. Training in simulation

We use Isaac Sim to build the simulation for a variation of the *occluded grasping* task to train our model. To better realize how the robot makes a decision in the current state, we visualize the high-level model's Q maps in Fig 3. As the first action (Fig 3 top row), the robot moves the object against one of the four walls by rotating the gripper $-22.5°$ around the z-axis and applying the push primitive. After pushing (middle row), the robot executes the flip primitive to flip up the object. Finally, the robot rotates the gripper $67.5°$ around the z-axis and executes a top-down grasp primitive to pick up the object.

Our method is trained in simulation using flat objects of random friction, weight, and size. We randomly place the object in the workspace and evaluate whether the robot can pick it up, applying 10 primitives or less. Fig. 4 shows the learning curve of success rate versus training transitions. We measure the grasp success rate over the last 100 grasp attempts and use the same way to measure the flip success rate and full-task completion rate. In this experiment, we start to test the model after 5000 transitions. Although our method as well as ED-PAP-MD can both reach an 80% completion rate before 8000 transitions, the improvement of ED-PAP-MD slows down after 8000 and 7000 transitions. This result implies that the learned parameterized feedback-based action primitive can better adapt to diverse situations than the manually designed primitive.

C. Real-world experiments

The real-world experiments are still ongoing and not included in this paper. We are currently evaluating our method with real-world experiments, using zero-shot transfer from simulation to a real-world setup using five different box-type objects. For all objects, we evaluate the method both with the same type of randomized start configuration as is done in simulation, and a start configuration where the box is placed next to a wall.

V. CONCLUSION

We propose an approach for learning hierarchical manipulation policies that allow for imposing intuitive inductive biases without relying on expert controller design. We apply our framework to the task of picking objects from non-graspable configurations using a learnable wall-assisted flipping primitive and demonstrate that our approach is able to efficiently solve the problem for a variety of object instances. Notably, compared to the state-of-the-art, we do not require the object to be placed near a supporting wall.

REFERENCES

[1] D.-C. Hoang, J. A. Stork, and T. Stoyanov, "Context-aware grasp generation in cluttered scenes," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1492–1498.

[2] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *2018 IEEE International Conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5620–5627.

[3] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1Billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

[4] 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)*. IEEE, 2018, pp. 4238–4245.

[5] D. Ren, X. Ren, X. Wang, S. T. Digumarti, and G. Shi, "Fast-learning grasping and pre-grasping via clutter quantization and Q-map masking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3611–3618.

[6] M. Dalal, D. Pathak, and R. R. Salakhutdinov, "Accelerating robotic reinforcement learning via parameterized action primitives," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 847–21 859, 2021.

[7] S. Nasiriany, H. Liu, and Y. Zhu, "Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7477–7484.

[8] W. Zhou and D. Held, "Learning to grasp the ungraspable with emergent extrinsic dexterity," in *Conference on Robot Learning (CoRL)*, 2022.

[9] R. Portelas, C. Colas, L. Weng, K. Hofmann, and P.-Y. Oudeyer, "Automatic curriculum learning for deep RL: A short survey," *arXiv preprint arXiv:2003.04664*, 2020.

[10] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving Rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[11] Y. Yang, Z. Ni, M. Gao, J. Zhang, and D. Tao, "Collaborative pushing and grasping of tightly stacked objects via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 135–145, 2021.

[12] K. Xu, H. Yu, Q. Lai, Y. Wang, and R. Xiong, "Efficient learning of goal-oriented push-grasping synergy in clutter," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6337–6344, 2021.

[13] B. Tang, M. Corsaro, G. Konidaris, S. Nikolaidis, and S. Tellex, "Learning collaborative pushing and grasping policies in dense clutter," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6177–6184.

[14] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," *Advances in neural information processing systems*, vol. 29, 2016.

[15] N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand manipulation with external forces," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1578–1585.

[16] 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.

[17] C. Eppner and O. Brock, "Planning grasp strategies that exploit environmental constraints," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4947–4952.

[18] J. Bimbo, E. Turco, M. Ghazaei Ardakani, M. Pozzi, G. Salvietti, V. Bo, M. Malvezzi, and D. Prattichizzo, "Exploiting robot hand compliance and environmental constraints for edge grasps," *Frontiers in Robotics and AI*, vol. 6, p. 135, 2019.

[19] 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 *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2562–2568.

[20] 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)*. IEEE, 2020, pp. 9917–9923.

[21] 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*, 2015, pp. 3431–3440.

[22] Y. Hu, W. Wang, H. Jia, Y. Wang, Y. Chen, J. Hao, F. Wu, and C. Fan, "Learning to utilize shaping rewards: A new approach of reward shaping," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 931–15 941, 2020.