2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Working Backwards: Learning to Place by Picking

Oliver Limoyo[1,2], Abhisek Konar[1], Trevor Ablett[1,2], Jonathan Kelly[2], Francois R. Hogan[1], Gregory Dudek[1,3]

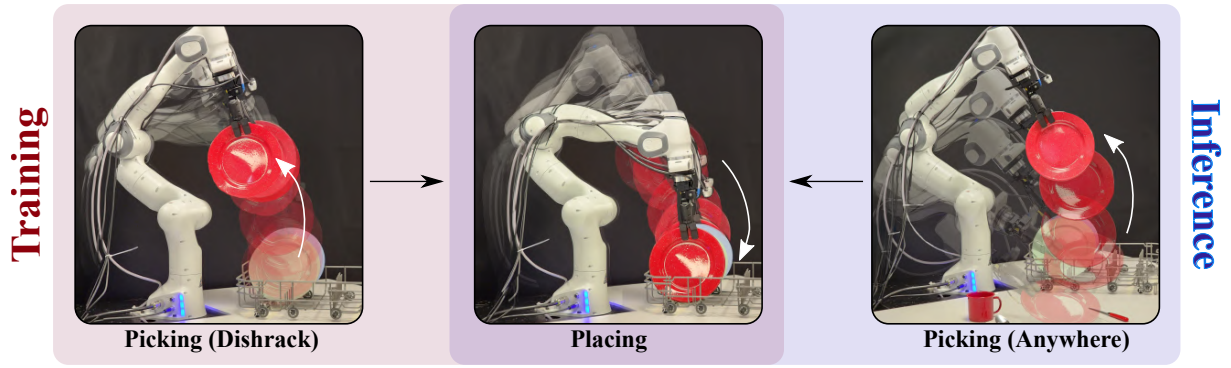


Fig. 1: An overview of *placing via picking* (PvP). *Left:* During training, we collect placing demonstrations by reversing the grasping trajectory with objects that are initially at their target locations (in the dish rack). *Right:* During inference, our learned vision-based policy can generalize to object placement scenarios where the objects are not at their target locations initially.

***Abstract*— We present *placing via picking* (PvP), a method to autonomously collect real-world demonstrations for a family of placing tasks in which objects must be manipulated to specific, contact-constrained locations. With PvP, we approach the collection of robotic object placement demonstrations by reversing the grasping process and exploiting the inherent symmetry of the pick and place problems. Specifically, we obtain placing demonstrations from a set of grasp sequences of objects initially located at their target placement locations. Our system can collect hundreds of demonstrations in contact-constrained environments without human intervention using two modules: compliant control for grasping and tactile re-grasping. We train a policy directly from visual observations through behavioural cloning, using the autonomously-collected demonstrations. By doing so, the policy can generalize to object placement scenarios outside of the training environment without privileged information (e.g., placing a plate picked up from a table). We validate our approach in home robot scenarios that include dishwasher loading and table setting. Our approach yields robotic placing policies that outperform policies trained with kinesthetic teaching, both in terms of success rate and data efficiency, while requiring no human supervision.**

I. Introduction

Imitation learning (IL) provides a scaleable, simple, and practical option to learn control policies from expert demonstrations [1], [2]. The general approach of training expressive models on large and diverse datasets has proven to be quite effective outside of robotics [3], [4]. There have been many attempts to transfer this paradigm to IL for robotic systems [5], [6], but similar successes have yet to be achieved. A major bottleneck for IL-based approaches in these data regimes is the human time and effort required to collect a large number of expert demonstrations in the physical world. Having said this, many recent methods attempt to streamline [7], [8] or automate [9]–[12] data collection. As highlighted by the authors of [11], the lack of robust and diverse autonomous data collection policies is a major limitation to scaling up IL. In this work, we are interested in investigating autonomous data collection for robotic object placement. As opposed to much prior research, we tackle the *full* placing problem, without restricting our task to simple, small objects placed on flat, horizontal surfaces with no environment contact constraints. We assert that object and environment contact constraints can be a valuable, natural guide for learning how to place, if handled properly.

We propose a novel approach named *placing via picking* (PvP) that automates the collection of expert demonstrations for a large subset of contact-constrained placing problems. We do so by taking advantage of a powerful grasp planner [13], tactile sensing, and compliant control. PvP is a self-supervised pipeline to autonomously collect expert placing demonstrations by leveraging the cyclical nature and inherent symmetry of the pick and place tasks. Given an environment with objects initially at their target locations, we alternate between picking (i.e., grasping and retrieving) and placing, by time-reversing the retrieval trajectory. We generate demonstration data for object placement in a self-supervised manner—the picking phase provides the training supervision for the placing task. While appearing deceptively easy at first, we highlight the crucial importance of two modules, compliant control for grasping (CCG) and tactile regrasping (TR). CCG and TR enable (1) robust and uninterrupted pick-and-place and (2) the use of environment contact-constraints as guides to inform object placement. Fig. 1 provides a visual summary of our approach. Our main contributions are

[1]Authors are with the Samsung AI Centre, Montréal, Québec H3A 3G4, Canada. Email: Email: {o.limoyo, t.ablett}@partner.samsung.com, {abhisek.k,f.hogan,greg.dudek}@samsung.com

[2]Authors are with the STARS Laboratory, University of Toronto Institute for Aerospace Studies, Toronto, Ontario M5S 1A4, Canada. Email: {*first-name.last-name*}@robotics.utias.utoronto.ca

[3]Gregory Dudek is with McGill University, Montréal, Québec H3A 089, Canada. Email: gregory.dudek@mail.mcgill.ca

1) PvP, a self-supervised data collection method for 6-DOF robotic object placements;
2) a compliant grasping and a tactile-driven regrasping strategy to achieve uninterrupted pick and place data collection in contact-constrained environments;
3) a language-specified grasp planning pipeline; and
4) real-world experimental validation of PvP using collected demonstrations to train a vision-based policy capable of placing multiple plates in a dish rack and of setting a table.

II. Related Work

In this section, we review prior work on automated robot data collection, discuss the general idea of "working backwards," and investigate existing robotic placement strategies.

A. Automatic Data Collection

A large number of existing policy learning approaches use prior policies to automate the data collection process. Examples include using trajectory optimization [14], [15] and, closer to our work, simple scripted pick-and-place policies [9]–[11] as forms of supervision. The authors of [9] use a scripted pick-and-place policy to bootstrap the collection of the initial dataset used for off-policy deep reinforcement learning with a real-world manipulator. In [11], the authors present AutoRT, a method that uses a combination of scripted, learned, and tele-operated data collection policies to efficiently collect a large amount of language-specified manipulation demonstration data. Using a combination of a large language model (LLM) and visual language models (VLMs), AutoRT observes scenes in the wild and comes up with plausible tasks. Once a task is defined and chosen, AutoRT samples a data collection policy to attempt to gather real world data for the task. In our work, we focus on developing a method for collecting demonstrations of placement with larger and more complex objects (i.e., objects that require a planner to grasp) in environments with contact constraints. Unlike prior work, we do not restrict the placing task to simple small objects on flat horizontal surfaces. The authors of [11] mention the lack of robust and diverse real-world autonomous collection policies as a limitation of AutoRT; our work is complementary to many of the previous frameworks, where PvP could be used as a prior policy.

Other works have explored automatic data generation by bootstrapping from a single human demonstration [16], [17]. In [16], a robot manipulation task is modelled as having two phases: a coarse approach trajectory phase towards a bottleneck pose, followed by a fine interaction trajectory phase. The single human demonstration provides the relative pose of the manipulator with respect to the task-relevant object, from which a self-supervised data collection procedure is formulated to train a bottleneck pose predictor. In our work, we also automatically collect and augment approach trajectory data. However, we do not require any human demonstrations since we focus on placing tasks only. By doing so, we can leverage the inherent symmetry of picking and placing for self-supervised data collection through the use of a combination of tactile sensing, compliant control, and an off-the-shelf grasp planner.

B. Working Backwards

The general idea of working backwards or time-reversal has been exploited in multiple contexts including generative modelling [18], learning visual predictive models for control [19]–[21], and reinforcement learning [22]. Our self-supervised data collection method, which extends the concept of working backwards to the problem of robotic placement, is most similar in spirit to [23]–[25]. These methods consider objects that are assumed to be in their desired place configuration during data collection. The manipulator can then pick an object and move back to the initial goal location to generate training data. However, we focus on placement as opposed to insertion. Insertion policies generally only reason about local interactions. On the other hand, placement involves reasoning over a larger scene with the potential for multiple place poses and more complex objects that require planners to grasp successfully. Furthermore, unlike prior work [23]–[25], our system does not require a human to guide the manipulator to the object and instead leverages tools from grasp planning. Our approach also resembles [26], where time-reversed trajectories are used to generate assembly data from disassembly data in a self-supervised manner. However, we focus on generating demonstrations for a closed-loop reactive policy that uses a 6-DOF pose-based action representation for placement. In contrast, the self-supervised data collection procedure introduced in [26] uses a 2D open loop pick-and-place action primitive.

C. Robotic Manipulator Placing

Pick and place is a fundamental problem in robotic manipulation. A significant amount of research has focused on the picking problem [27]–[29], while placing has been relatively less studied.

Previous approaches have explored the use of motion planning combined with place-specific objectives [30], [31]. For example, in [30], the authors demonstrate a hierarchical sampling-based motion planner that first finds poses which satisfy constraints (e.g., reachability) and then performs a local optimization for user-specified objectives (e.g., clearance). The place problem has also been studied from the perspective of geometric modelling [32], where the planar surfaces of a discretized object model are matched to planar patches in the environment. Importantly, 3D models of the objects being placed as well as the environment are assumed to be available ahead of time.

Closer to our approach, other works have trained end-to-end place policies using reinforcement learning [33] and imitation learning [34]. In [34], the authors demonstrate an inverse reinforcement learning technique capable of simultaneously learning a cost function and a control policy from human kinesthetic demonstrations of dish placing. The main contribution and novelty of our work is a data collection technique for robotic placing tasks that can be used in tandem with methods that require expert demonstrations.

III. Placing via Picking

We first present an overview of our self-supervised data collection method, PvP, in Section III-A, and then describe a method for noise augmentation during data collection in Section III-B. Finally, we provide details about the procedure used to train our IL policy in Section III-C.

A. Self-Supervised Data Collection

PvP consists of a cycle with four main phases: (1) grasp planning, (2) grasping, (3) retrieving, and (4) placing by reversing; these steps are visualized in Fig. 3. During training, the objects of interest are initially in their goal positions.

1) Grasp Planning: At the start of the data collection process, the manipulator moves to a pre-determined pose where the camera has a clear view of the objects of interest in the environment, as shown in Fig. 3a. We use the grasp planner Contact-GraspNet [13] to generate L candidate grasp poses $\{\mathbf{T}_{g,i}\}_{i=0}^{L}$, where $\mathbf{T}_{g,i} \in \mathrm{SE}(3)$ for all $i \in \{0, ..., L\}$. Given a set of text-based descriptions of the objects of interest (e.g., "plates," "green plate," "cup" or, more generally, "objects"), potentially from a user or high-level LLM planner, we use Grounding-Dino [35] to find the object bounding boxes. Finally, we run Segment Anything [36] on each of the previously-generated image crops from the bounding boxes to generate per-object masks. Contact-GraspNet then uses these masks to filter grasps that are not on objects of interest and keeps only a pruned set of grasps $\{\mathbf{T}_{g,i}\}_{i=0}^{K}$, where $K \leq L$. We visualize an example of this process in Fig. 4 for two different scenes. After pruning, we randomly select a single grasp pose, $\mathbf{T}_{\text{grasp}}$, from $\{\mathbf{T}_{g,i}\}_{i=0}^{K}$. To approach without collisions, we include a pregrasp pose $\mathbf{T}_{\text{pregrasp}}$ that is defined with a translational offset above the grasp pose.

2) Grasping: Given the computed pregrasp and grasp poses, we move to each pose sequentially by linearly and spherically linearly interpolating to produce a smooth trajectory. An example of a grasp sequence is shown in Fig. 3b. During grasping, two modules play a critical role in the robustness of PvP: compliant control for grasping (CCG) and tactile regrasping (TR).

We use a Cartesian impedance controller for manipulation. This choice allows us to set the level of compliance of the manipulator at various stages as needed. Before closing the gripper to grasp, we set the translational and rotational stiffness of the manipulator's controller to minimal values. We call this CCG. The high compliance of the robot arm allows the manipulator and gripper to comply with the contact constraints and minimize the applied forces against the environment. Larger contact-constrained objects generate unnecessary reaction forces when gripped without compliance. Large forces, in turn, trigger emergency stops and can damage the environment and the robot. Specific to PvP, large reaction forces also disrupt and move the object relative to the gripper, ultimately leading to inaccurate placements. On the other hand, when grasping with compliance, the environment contact constraints guide the manipulator towards a natural object placement pose.

Our method assumes that the grasps that we perform are *stable*, that is, the object does not move or slip significantly once grasped. In the real world, the accumulation of errors from various sources (e.g., camera calibration, controller, learned grasp planner, and noise from the RGB-D sensor and manipulator encoders) can result in grasps that are unstable (e.g., shallow grasps) and not ideal for PvP. To mitigate this issue, we use tactile sensing—specifically, a visuotactile Finger-STS [37] sensor—to preemptively detect if a grasp is stable and perform a regrasp if needed. The Finger-STS is capable of providing multimodal feedback (i.e., both tactile and visual observations). To this end, we design our TR module to detect the contact surface area and calculate the relative change in end effector (EE) pose required to recover a larger contact surface and, thereby, a more stable grasp. We detect regions of contact based on marker displacements and the RGB image from the Finger-STS. If partial contact is detected, we calculate the difference between the ideal and detected contact region and use this to calculate the commanded change in EE pose. We note that a simpler and more general random regrasp strategy could also be used. An example of this is shown in Fig. 5. We visualize contact regions for various household items in Fig. 6.

In Section IV-A, we experimentally investigate the effects of including these modules through an ablation study.

3) Retrieving: We begin the retrieval phase when a valid, stable grasp is detected. An image of the retrieval procedure is shown in Fig. 3c. We store the poses of the EE as well as the respective timestamps throughout the retrieval process. For our work, we measure joint encoder readings at a rate of 120 Hz. The manipulator is set to be compliant along the rotational axes of the EE only, when commanded to move to the pregrasp pose. Once at the pregrasp pose, we randomly sample a clearance pose $\mathbf{T}_{\text{clearance}}$ around a set fixed pose above the scene. That is, given a set fixed pose, we add random translations sampled from $\mathcal{N}(0, \sigma_{tr}^2)$ to recover a clearance pose. We use σ_{tr} = 2.5 cm during data collection.

At the end of the retrieval phase, we have an expert retrieval trajectory of length M from grasp pose to clearance pose via a pregrasp pose,

$$\tau_r^{\text{expert}} = ((\mathbf{T}_0, t_0), ..., (\mathbf{T}_M, t_M)), \tag{1}$$

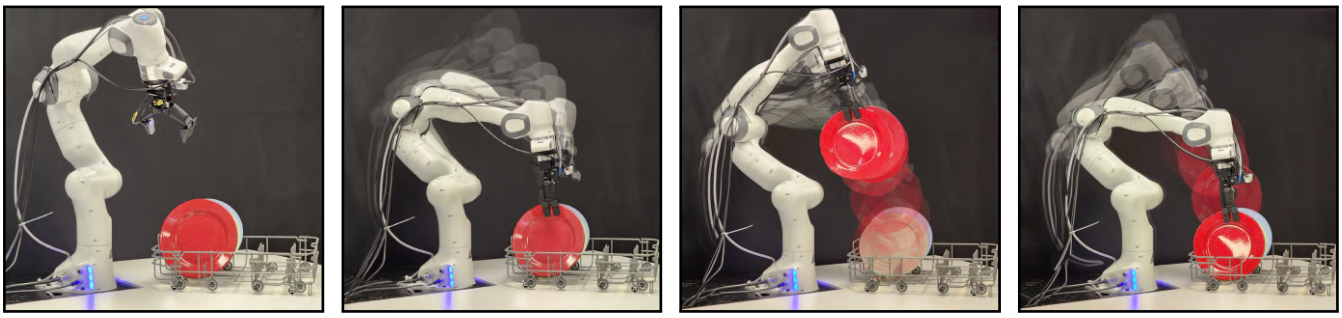
where $\mathbf{T}_0 = \mathbf{T}_{\text{grasp}}$, $\mathbf{T}_M = \mathbf{T}_{\text{clearance}}$ and $t_0, ..., t_M$ are the respective timestamps of the trajectory starting with t_0 = 0. To match the desired control frequency of our policy, we sample states Δt apart to generate a sparser trajectory $\bar{\tau}_r^{\text{expert}}$. We extract a total of $\bar{M}+1$ states, where $\bar{M} = \lceil \frac{t_M - t_0}{\Delta t} \rceil$, by finding the nearest pose (in terms of time) from the dense trajectory τ_r^{expert} at every Δt interval:

$$\bar{\tau}_r^{\text{expert}} = ((\bar{\mathbf{T}}_0, 0), (\bar{\mathbf{T}}_1, \Delta t), ..., (\bar{\mathbf{T}}_{\bar{M}}, \bar{M}\Delta t)). \tag{2}$$

In this work, our visual policies operate at 5 Hz, so we set $\Delta t = 0.20$ s.

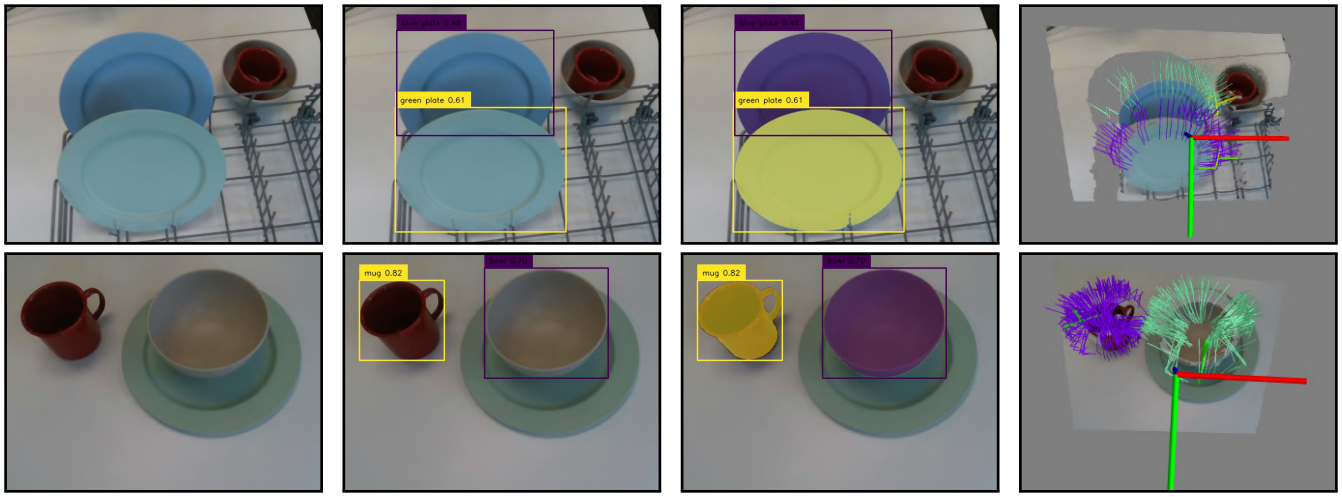
4) Placing by Reversing: We reverse the sparse retrieval trajectory $\bar{\tau}_r^{\text{expert}}$ to extract a desired place trajectory,

$$\bar{\tau}_p^{\text{expert}} = (\bar{\mathbf{T}}_{\bar{M}}, ..., \bar{\mathbf{T}}_0). \tag{3}$$



(a) Grasp planning. (b) Grasping. (c) Retrieving. (d) Placing by reversing.

Fig. 3: The four steps involved in PvP, our autonomous demonstration data collection process for placing. We (a) generate grasps with an off-the-shelf grasp planner [13]; (b) compliantly grasp the object to apply minimal forces to the environment while ensuring a stable grasp via tactile sensing; (c) retrieve the object with rotational compliance while storing the trajectory; and (d) generate placement demonstration data by rolling out the reversed grasp trajectories while storing the observations and actions.



(a) Raw image. (b) Text-based object detection. (c) Segmentation. (d) Pruned grasps.

Fig. 4: The steps involved in our language-driven grasp planning pipeline. We use (b) Grounding-Dino [35] for object bounding box detection based on text descriptions and (c) Segment Anything [36] on the cropped images to produce object specific masks. We then use (d) Contact-GraspNet [13] for grasp generation on only the segmented objects (i.e., masked areas). *Top*: using "green plate" and "blue plate" as the targets for data collection. *Bottom*: using "bowl" and "mug" as the targets for data collection.

An example of a place trajectory is shown in Fig. 3d. Additionally, we convert the global poses to relative poses in order to use state differences as expert actions:

$$\Delta \mathbf{T}_i = \bar{\mathbf{T}}_{\bar{M}-i}^{-1} \bar{\mathbf{T}}_{\bar{M}-i-1}, \tag{4}$$

where $i \in \{0, ..., \bar{M}-1\}$ and $\Delta \mathbf{T}_i \in \mathrm{SE}(3)$ is a relative change in EE pose. We define the expert action as $\mathbf{a}_i = (\Delta \mathbf{T}_i, a_{\text{gripper},i})$, where $a_{\text{gripper},i} \in \{0, 1\}$ is a binary variable describing the gripper command (open or close). To collect an episode of expert data, we perform a rollout of the downsampled place commands, $\{\mathbf{a}_i\}_{i=0}^{\bar{M}-1}$, and store the sequences of observations and actions as an expert trajectory.

We use RGB images from the wrist camera to form the robot's observation space $\mathbf{o}_i$. We downsample the images to a resolution of 128×128 and also stack the three previous RGB frames, which results in $\mathbf{o}_i \in \mathbb{R}^{128 \times 128 \times 12}$. During the place demonstration, the robot keeps its gripper closed (i.e., the robot uses the gripper command $a_{\text{gripper},i} = 1$ for all $i \in \{0, ..., \bar{M}-1\}$). Finally, at the end of the trajectory, we command the robot to keep the EE in its current pose and to open the gripper (i.e. $\Delta \mathbf{T}_i = \mathbf{I}$ and $a_{\text{gripper},i} = 0$) for N time steps. We use $N = 5$, which gives the robot an adequate amount of time to open the gripper and release the object. A single expert demonstration or trajectory is then defined as a set of training tuples

$$\tau^{\text{expert}} = \{(\mathbf{o}_t, \mathbf{a}_t, \mathbf{o}_{t+1})\}_{t=0}^{T}, \tag{5}$$

with a total length of $T = \bar{M} + N$.

B. Noise-Augmented Data Collection

To increase the coverage of our demonstrations and train more robust policies, we perturb the first 75% of the poses with noise. We take inspiration from previous works that have investigated the use of noise injection for both the policy observations [38], and actions [39]. We also take inspiration from robust control theory, where injecting isotropic Gaussian noise has been shown to achieve *persistent excitation* [40], a condition where the training data is informative enough to learn a model that is robust to compounding errors during deployment. We call this variant *noise-augmented*

data collection. In practice, we represent pose commands $\Delta\mathbf{T}$ as a combination of a translation vector $\mathbf{t} \in \mathbb{R}^3$ and a rotation vector $\boldsymbol{\theta} \in \mathbb{R}^3$. The rotation vector is defined as $\boldsymbol{\theta} = \theta\mathbf{e}$, where $\theta \in \mathbb{R}$ is the rotation angle and $\mathbf{e} \in \mathbb{R}^3$ is the rotation axis. We perturb the translation of the pose with

$$\mathbf{t}_{\text{perturbed}} = \mathbf{t} + \delta\mathbf{t}, \tag{6}$$

where $\delta\mathbf{t} \in \mathbb{R}^3$ is an isotropic Gaussian noise vector and each dimension of $\delta\mathbf{t}$ is sampled from $\mathcal{N}(0, \sigma_t)$. Similarly, we perturb the rotation with

$$\boldsymbol{\theta}_{\text{perturbed}} = (\theta + \delta\theta)(\mathbf{e} + \delta\mathbf{e}), \tag{7}$$

where $\delta\mathbf{e} \in \mathbb{R}^3$ is an isotropic Gaussian noise vector and each dimension of $\delta\mathbf{e}$ is sampled from $\mathcal{N}(0, \sigma_e)$, and $\delta\theta \in \mathbb{R}$ is sampled from $\mathcal{N}(0, \sigma_\theta)$. We perturb both the rotation axis and the rotation angle. We find $\sigma_t = 0.5$ cm, $\sigma_e = 0.5$ cm and $\sigma_\theta = 0.5^\circ$ to be reasonable values. We evaluate the effects of this augmentation on policy performance in Section IV-B

C. Policy Learning

We learn a policy $\pi_{\boldsymbol{\phi}}(\mathbf{a}_t \mid \mathbf{o}_t)$ parameterized by $\boldsymbol{\phi}$ using data generated by PvP. We use behavioural cloning (BC) [41] to train our policy with a likelihood-based loss or objective function. Given a dataset of N_{train} demonstration trajectories, our loss function is then:

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \sum_{t=0}^{T} -\log \pi_\phi(\mathbf{a}_t \mid \mathbf{o}_t). \tag{8}$$

The main backbone of our policy network consists of a convolutional neural network (CNN) based on the ResNet18 architecture [42], followed by a multilayer perceptron (MLP) layer that maps the CNN output into the parameters of the distribution of actions. In this work, we consider two representations for the action distribution $\pi_\phi(\mathbf{a}|\mathbf{o})$: a uni-modal Gaussian and a multimodal mixture of Gaussians. In both cases, during inference or control, we use a low-noise evaluation scheme, as done in [43] and [44], by setting the standard deviation of the Gaussian distributions to be an arbitrarily small value. Note that the unimodal Gaussian is then equivalent to a standard deterministic policy trained with a mean squared error (MSE) loss.

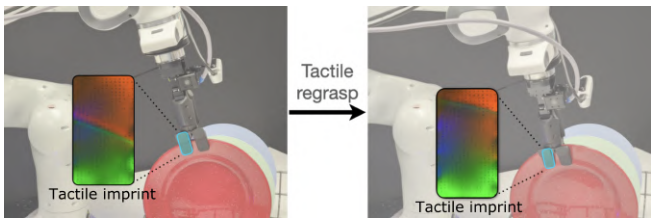
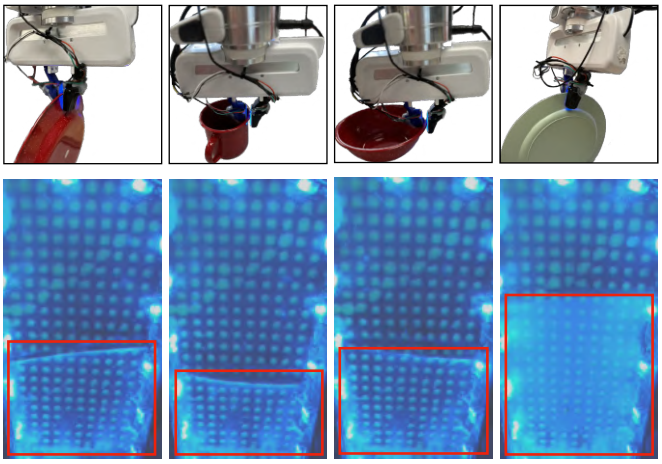


Fig. 5: Tactile images from before and after a tactile regrasp (TR). Left: the contact surface area of the plate fills half of the tactile image, indicating a shallow grasp. Right: after a regrasp, the contact surface area has increased, indicating a stable grasp.



(a) (b) (c) (d)

Fig. 6: Visualization of contact surface region in the tactile image of grasps of various objects: (a) a red metal plate, (b) a red metal mug, (c) a red metal bowl and (d) a green wheat straw plate.

IV. Experiments

We used PvP to collect data and train robotic placement policies for two separate tasks: dishrask loading and table setting, as shown in Fig. 7. In both cases, the place policy had to account for the weight and shape of each item, the initial grasp of the object, and the state of the current scene to decide where and how to place (e.g., place angle). In this section, we only present our quantitative results for the more challenging dishrack loading task as it better represents the type of task (with added contact constraints from the environment) that PvP was designed for. We study the robustness of PvP in Section IV-A, the effect of noise augmentation on the performance of the learned placing policy in Section IV-B, and the relative quality of policies trained with data from PvP and from kinesthetic teaching in Section IV-C.

A. PvP Data Collection Robustness

We require a robust data collection loop to collect a large number of place demonstrations autonomously with PvP. We studied the effects of CCG and TR, as introduced in Section III, on the robustness of PvP.

We observed two main causes of misplaced objects during data collection, initially discussed in Section III-A.2, First, the manipulator would stiffly grasp the object causing it to move and generate a large amount of force preload against the environment. During retrieval, significant object motion occurred at the moment when the object's contact force with the environment was released. This unexpected shift in the object pose lead to less accurate placements since the reversal procedure no longer resulted in correct alignment between the object and the environment. With PvP, we mitigate this failure mode with CCG. The second failure mode involved unstable grasps that lead to a large relative motion between the object and the gripper. With PvP, we mitigate unstable

(a) Dishrack placement task, which consists of placing multiple plates of varying physical properties in evenly spaced slots.

(b) Table placement task, which consists of placing a bowl on a plate and a cup on a coaster.

Fig. 7: Sequence of images from roll outs of place policies trained using data collected with PvP. The policies are able to place objects of varying properties in the scene using images from the wrist camera directly.

grasps using TR to preemptively detect unstable grasps and to regrasp if needed.

We conducted an ablation study on CCG and TR where we ran self-supervised data collection for a total of 128 episodes and recorded the number of failures. We visualize the results in Fig. 8. The naive approach without CCG and without TR failed a total of 15 times, which roughly translates to an average of nine episodes collected autonomously before a human intervention is needed. The number of failures dropped to three with the addition of CCG, which raises the average to 43 episodes collected before each intervention. Finally, with both CCG and TR, we were able to successfully collect 128 episodes without any human intervention.

B. Noise Augmentation Ablation

In Section III-B, we introduced a noise-augmented data collection variant of PvP. We investigated its contribution to the success rate of the policy. We collected two different datasets each containing 128 demonstrations, with and without added noise augmentation. We tested the effect of noise-augmented data on policies trained using two different action representations: a deterministic policy and a Gaussian mixture policy with five modes. We present the results in Table I.

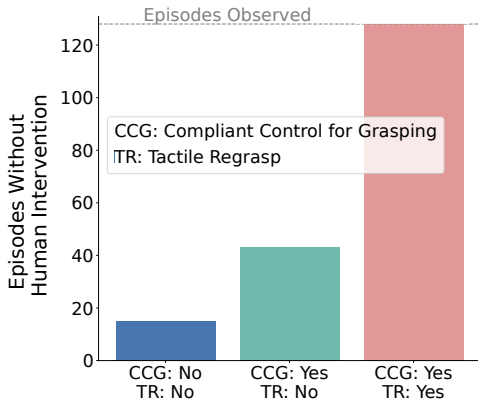


Fig. 8: Ablation study on compliant control for grasping (CCG) and tactile regrasping (TR). We measured the average number of episodes that could be collected autonomously based on the number of failures. Both CCG and TR play a crucial role in reducing the number of failures to zero, which allowed us to collect the necessary amount of data seamlessly.

For each row, we averaged the success rate over three models trained from scratch with three different seeds. We used 20 or 30 rollouts to calculate the success rate of each model. The numbers in the brackets denote variation of one standard deviation. We also report the total successes summed over the three models. For both action representations, we found that training on noise-augmented data improved the performance of the final policy significantly. The deterministic policy had an improvement of approximately 10% while the mixture policy had a larger improvement of 15%.

We hypothesize that adding noise further accentuates the multimodal nature of the data (i.e., multiple valid actions are available for roughly the same observation) and that a mixture representation for the actions allows the policy to better capture this by not having to average over multiple possible expert actions [45]. We also qualitatively observed that policies trained with noise-augmented datasets were better at re-adjusting the object with respect to the goal.

C. Comparison to Kinesthetic Teaching

We compared the performance of policies trained using demonstration data from PvP and from kinesthetic teaching. In particular, we investigated whether the quality of the demonstrations differ. To compare, we trained two separate policies, both sharing the architecture outlined in Section III-C, on two separate datasets: one collected with PvP and another collected with kinesthetic teaching [46]. We used a Gaussian mixture model as the action representation of both policies. For PvP, we used the noise-augmented variant since it was the best-performing model, as shown in Section IV-B.

TABLE I: Success rates for a model trained on a dataset collected with and without noise augmentation and using two different policy action representations. The noise-augmented data collection procedure improves the performance of both deterministic and Gaussian mixture policies.

Action Representation	Trained with Noise Aug. Data	Avg. Success Rate (↑)	Total Successes (↑)
Deterministic	No	71.67(5.93)	57/80
	Yes	81.11(5.50)	56/70
Gaussian Mixture	No	72.22(5.67)	50/70
	Yes	**87.78(3.93)**	**62/70**

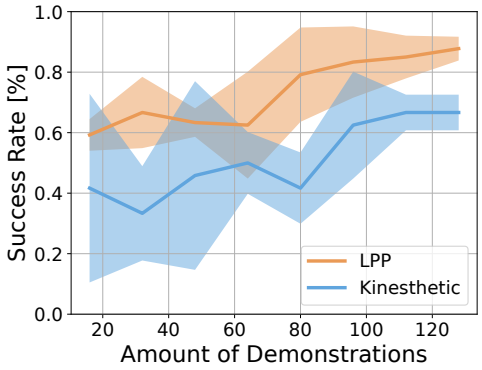


Fig. 9: Comparison of success rates of policies trained using a dataset collected by PvP and a traditional kinesthetic teaching approach. We tested models trained on an increasing amount of demonstration data. Policies trained on data collected by PvP outperform those trained on data collected from kinesthetic teaching for almost all dataset sizes. The shaded region consists of one standard deviation.

In Fig. 9, we visualize the success rates of both models trained with varying numbers of demonstrations. As done previously, we averaged the performance over three models with different seeds for each datapoint. We tested each model using eight rollouts. The models trained with data from PvP, collected without any human involvement, significantly outperform the models trained using kinesthetic data by about 20% for almost all numbers of demonstrations. We observed that policies trained with kinesthetic teaching often misplaced the plate (e.g., by not having the right place angle or not resting the plate in a stable manner with respect to the environment contact points) and struggled to open the gripper at the correct times. The latter of the two error modes did not appear in policies trained with PvP.

We hypothesize that the kinesthetic policy performed more poorly than the PvP policy because the overall quality of the human demonstrations was lower than the programmatic or machine-generated demonstrations produced by PvP. Previous work has studied the difficulties and challenges of training policies using human demonstrations when compared to programmatic or machine-generated demonstrations [44], [47], [48]. In particular, human demonstrations can be sub-optimal due to mistakes (e.g., large variations in trajectory length, unwanted movements, and remaining idle for some time to think). The authors of [44] note that the relative average trajectory length is a good proxy for the quality of the dataset. The average number of time steps for all demonstrations was 29.41 (2.22) for the PvP dataset and 41.66 (5.69) for the kinesthetic dataset. The numbers in the brackets denote variation of one standard deviation. Kinesthetic demonstrations were on average longer and had greater variance in length. In addition, PvP benefits from systematic noise-augmentation, as covered in Section III-B, which a human demonstrator cannot emulate consistently.

V. Limitations and Future Work

PvP relies on effective grasping (i.e., accurate grasp detections and stable grasps during retrieval). This makes this method susceptible to the typical challenges involved in grasping, namely dealing with noisy point cloud measurements, unreliable object segmentation, and unmodelled object dynamics (e.g. object slip). The quantity, quality, and diversity of the grasp poses found by the planner are partly determined by the viewpoint of the camera. In this work, we empirically selected a viewpoint that is adequate for our use case. It would be interesting to allow the manipulator to explore and find better grasp poses. Furthermore, while tactile sensing is used to improve the robustness of data collection, it could also be used as an extra modality for the policy. We have introduced a framework for data collection and tested it with two real-world tasks. We would like to test PvP on a larger set of tasks for a more exhaustive evaluation. Lastly, PvP is a language-driven data collection method, and it would be natural to integrate it into a high-level language based planner [11].

VI. Conclusion

In this paper, we presented PvP, a self-supervised data collection method to collect expert demonstrations for placement tasks. PvP leverages recent advancements in grasp planners, tactile sensing, compliant control, and the symmetry between picking and placing to autonomously collect placing demonstrations. PvP is able to robustly collect a large amount of data consistently without human intervention by making use of compliant manipulator control and regrasping based on tactile sensing. We showed that the quality of the demonstrations produced by PvP surpasses that of human demonstrations collected by kinesthetic teaching. PvP provides a promising and pragmatic direction to collect real world robotic placing demonstration data with minimal human labour.

References

[1] T. Ablett, Y. Zhai, and J. Kelly, "Seeing all the angles: Learning multiview manipulation policies for contact-rich tasks from demonstrations," in *IEEE/RSJ Intl. Conf. Intelligent Robots and Systems (IROS)*, pp. 7843–7850, 2021.

[2] 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 *Conf. Robot Learning (CORL)*, pp. 158–168, 2021.

[3] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, *et al.*, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023.

[4] OpenAI, "GPT-4 Technical Report," tech. rep., 2023.

[5] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conf. Robot Learning (CORL)*, pp. 991–1002, 2022.

[6] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters (RAL)*, pp. 1–8, 2023.

[7] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," in *arXiv preprint arXiv:2402.10329*, 2024.

[8] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Robotics: Science and Systems (RSS)*, 2023.

[9] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "QT-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conf. Robot Learning (CORL)*, pp. 651–673, 2018.

[10] D. Kalashnkov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman, "Mt-opt: Continuous multi-task robotic reinforcement learning at scale," *arXiv preprint arXiv:2104.08212*, 2021.

[11] M. Ahn, D. Dwibedi, C. Finn, M. G. Arenas, K. Gopalakrishnan, K. Hausman, B. Ichter, A. Irpan, N. Joshi, R. Julian, S. Kirmani, I. Leal, E. Lee, S. Levine, Y. Lu, I. Leal, S. Maddineni, K. Rao, D. Sadigh, P. Sanketi, P. Sermanet, Q. Vuong, S. Welker, F. Xia, T. Xiao, P. Xu, S. Xu, and Z. Xu, "AutoRT: Embodied foundation models for large scale orchestration of robotic agents," *arXiv preprint arXiv:2401.12963*, 2024.

[12] K. Bousmalis, G. Vezzani, D. Rao, C. M. Devin, A. X. Lee, M. B. Villalonga, T. Davchev, Y. Zhou, A. Gupta, A. Raju, *et al.*, "Robocat: A self-improving generalist agent for robotic manipulation," *Trans. Machine Learning Research*, 2023.

[13] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pp. 13438–13444, 2021.

[14] M. Dalal, A. Mandlekar, C. R. Garrett, A. Handa, R. Salakhutdinov, and D. Fox, "Imitating task and motion planning with visuomotor transformers," in *Conf. Robot Learning (CORL)*, pp. 2565–2593, 2023.

[15] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[16] E. Johns, "Coarse-to-fine imitation learning: Robot manipulation from a single demonstration," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pp. 4613–4619, 2021.

[17] X. Li, M. Baum, and O. Brock, "Augmentation enables one-shot generalization in learning from demonstration for contact-rich manipulation," in *IEEE/RSJ Intl. Conf. Intelligent Robots and Systems (IROS)*, pp. 3656–3663, 2023.

[18] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Intl. Conf. Machine Learning (ICML)*, pp. 2256–2265, 2015.

[19] S. Nair, M. Babaeizadeh, C. Finn, S. Levine, and V. Kumar, "Trass: Time reversal as self-supervision," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pp. 115–121, 2020.

[20] O. Limoyo, B. Chan, F. Marić, B. Wagstaff, A. R. Mahmood, and J. Kelly, "Heteroscedastic uncertainty for robust generative latent dynamics," *IEEE Robotics and Automation Letters (RAL)*, vol. 5, no. 4, pp. 6654–6661, 2020.

[21] O. Limoyo, T. Ablett, and J. Kelly, "Learning sequential latent variable models from multimodal time series data," in *Intelligent Autonomous Systems 17*, pp. 511–528, 2023.

[22] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in Neural Information Processing Systems Deep Learning Symposium (NeurIPS)*, pp. 5055–5065, 2017.

[23] L. Fu, H. Huang, L. Berscheid, H. Li, K. Goldberg, and S. Chitta, "Safe self-supervised learning in real of visuo-tactile feedback policies for industrial insertion," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pp. 10380–10386, 2023.

[24] O. Spector and D. Di Castro, "Insertionnet-a scalable solution for insertion," *IEEE Robotics and Automation Letters (RAL)*, vol. 6, no. 3, pp. 5509–5516, 2021.

[25] O. Spector, V. Tchuiev, and D. Di Castro, "InsertionNet 2.0: Minimal contact multi-step insertion using multimodal multiview sensory input," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pp. 6330–6336, 2022.

[26] K. Zakka, A. Zeng, J. Lee, and S. Song, "Form2fit: Learning shape priors for generalizable assembly from disassembly," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pp. 9404–9410, 2020.

[27] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pp. 348–353, 2000.

[28] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis — a survey," *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[29] 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," *Intl. J. Robotics Research*, vol. 41, no. 7, pp. 690–705, 2022.

[30] J. A. Haustein, K. Hang, J. Stork, and D. Kragic, "Object placement planning and optimization for robot manipulators," in *IEEE/RSJ Intl. Conf. Intelligent Robots and Systems (IROS)*, pp. 7417–7424, 2019.

[31] J. A. Haustein, S. Cruciani, R. Asif, K. Hang, and D. Kragic, "Placing objects with prior in-hand manipulation using dexterous manipulation graphs," in *IEEE-RAS Intl. Conf. Humanoid Robots (Humanoids)*, pp. 453–460, 2019.

[32] K. Harada, T. Tsuji, K. Nagata, N. Yamanobe, and H. Onda, "Validating an object placement planner for robotic pick-and-place tasks," *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1463–1477, 2014.

[33] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, "Tactile-rl for insertion: Generalization to objects of unknown geometry," *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pp. 6437–6443, 2021.

[34] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Intl. Conf. Machine Learning (ICML)*, pp. 49–58, 2016.

[35] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[36] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[37] F. R. Hogan, J.-F. Tremblay, B. H. Baghi, M. Jenkin, K. Siddiqi, and G. Dudek, "Finger-sts: Combined proximity and tactile sensing for robotic manipulation," *IEEE Robotics and Automation Letters (RAL)*, vol. 7, no. 4, pp. 10865–10872, 2022.

[38] P. Florence, L. Manuelli, and R. Tedrake, "Self-supervised correspondence in visuomotor policy learning," *IEEE Robotics and Automation Letters (RAL)*, vol. 5, no. 2, pp. 492–499, 2019.

[39] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg, "Dart: Noise injection for robust imitation learning," in *Conf. Robot Learning (CORL)*, pp. 143–156, 2017.

[40] M. Green and J. B. Moore, "Persistence of excitation in linear systems," *Systems and Control Letters*, vol. 7, no. 5, pp. 351–360, 1986.

[41] D. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems Deep Learning Symposium (NeurIPS)*, pp. 305–313, 1989.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[43] Z. Wang, A. Novikov, K. Zolna, J. S. Merel, J. T. Springenberg, S. E. Reed, B. Shahriari, N. Siegel, C. Gulcehre, N. Heess, *et al.*, "Critic regularized regression," in *Advances in Neural Information Processing Systems Deep Learning Symposium (NeurIPS)*, pp. 7768–7778, 2020.

[44] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Mart'in-Mart'in, "What matters in learning from offline human demonstrations for robot manipulation," in *Conf. Robot Learning (CORL)*, pp. 1678–1690, 2021.

[45] C. Bishop, "Mixture density networks," Tech. Rep. NCRG/94/004, Dept. of Computer Science and Applied Mathematics, Aston University, 1994.

[46] A. G. Billard, S. Calinon, and R. Dillmann, "Learning from humans," *Springer Handbook of Robotics*, pp. 1995–2014, 2016.

[47] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson, "Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning," in *Intl. Conf. Learning Representations (ICLR)*, 2019.

[48] M. Orsini, A. Raichuk, L. Hussenot, D. Vincent, R. Dadashi, S. Girgin, M. Geist, O. Bachem, O. Pietquin, and M. Andrychowicz, "What matters for adversarial imitation learning?," in *Advances in Neural Information Processing Systems Deep Learning Symposium (NeurIPS)*, pp. 14656–14668, 2021.