Evaluating Guided Policy Search for Human-Robot Handovers

Alap Kshirsagar[1], Guy Hoffman[1†], Armin Biess[2†]

Abstract—**We evaluate the potential of Guided Policy Search (GPS), a model-based reinforcement learning (RL) method, to train a robot controller for human-robot object handovers. Handovers are a key competency for collaborative robots and GPS could be a promising approach for this task, as it is data efficient and does not require prior knowledge of the robot and environment dynamics. However, existing uses of GPS did not consider important aspects of human-robot handovers, namely large spatial variations in reach locations, moving targets, and generalizing over mass changes induced by the object being handed over. In this work, we formulate the reach phase of handovers as an RL problem and then train a collaborative robot arm in a simulation environment. Our results indicate that GPS is limited in the spatial generalizability over variations in the target location, but that this issue can be mitigated with the addition of local controllers trained over target locations in the high error regions. Moreover, learned policies generalize well over a large range of end-effector masses. Moving targets can be reached with comparable errors using a global policy trained on static targets, but this results in inefficient, high-torque, trajectories. Training on moving targets improves trajectories, but results in worse worst-case performance. Initial results suggest that lower-dimensional state representations are beneficial for GPS performance in handovers.**

Index Terms—**Physical Human-Robot Interaction, Reinforcement Learning, Manipulation Planning**

I. Introduction

In this work, we develop and evaluate a robot controller that uses Guided Policy Search (GPS) to perform reaching motions for object handovers. Handovers are a core competency for collaborative and assistive robots working with humans, for example, in collaborative assembly, surgical assistance, household chores and elder care. A handover consists of three phases: reach, transfer and retreat [1]. We focus on the reach phase of a handover, in which both actors extend their arms towards the handover location. While researchers have proposed a number of offline [2]–[7] and online [1], [8]–[23] controllers for the reach phase, these methods rely on accurate models of the robot's dynamics and/or of the human kinematics. Recently researchers have suggested GPS [24]–[26], a reinforcement learning (RL) algorithm, with promising success in a number of autonomous tasks [27]–[29]. Some variants of GPS, like the one we use in this work [29], do not require prior knowledge of the robot or environment dynamics.

While GPS has been demonstrated on a number of autonomous manipulation and navigation tasks, it has not been tested in a physical human-robot collaborative task such as a handover. Examples of successfully learnt manipulation tasks with GPS include stacking small blocks, assembling toys, inserting rings on wooden pegs, screwing bottle caps, inserting shapes into sorting cubes and opening doors [27]–[29]. Common to all of these GPS applications are fixed targets, small variations in the test locations, and fixed robot dynamics. The task of object handovers has important characteristics that deviate from previous work: First, it requires a robot to plan its motion towards a moving target, i.e., the human's hand. Second, given the unpredictability of human behavior, the training and test target trajectories could be very different. Finally, due to different objects that are handed over, the robot dynamics are not fixed.

In this work, we evaluate GPS for handovers, and tackle previously unanswered questions such as: How does GPS perform if the training and test conditions are spatially far apart? How does GPS perform when reaching for an unpredictable moving target? How does GPS perform in the case of changes in the robot's end-effector mass?

To do so, we formulate the reach phase of a handover as an RL problem and investigate the performance of GPS for the scenarios listed above. We find that the global policy learnt with GPS does not perform well for target test locations spatially too distant from the target training locations but that this can be addressed with the addition of local controllers which are trained over target locations in the high error regions. In that case, the learnt global policy can also handle moving targets with comparable errors, albeit with highly inefficient trajectories. Training on moving targets improves the trajectories, but results in higher worst-case errors. Finally, we find that a learnt global policy adapts well to changes in robot dynamics due to changes in the robot's end-effector mass. In an exploratory evaluation of different state representations, we find that a low dimensional state representation may be more suitable for GPS-trained handover controllers.

There are important features of human-robot handovers that

Manuscript received: October, 14, 2020; Revised February, 3, 2021; Accepted February, 28, 2021.

This paper was recommended for publication by Editor Gentiane Venture upon evaluation of the Associate Editor and Reviewers' comments.

A. Kshirsagar was supported by the Ben-Gurion University of the Negev. Prof. Hoffman was supported by a Mills Family Faculty Fellowship. Dr. Biess was supported in part by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive (ABC) Robotics Initiative, the Marcus Endowment Fund both at Ben-Gurion University of the Negev and the Israel Science Foundation (grant no. 1627/17).

[1]Alap Kshirsagar (Corresponding Author, ak2458@cornell.edu) and Prof. Guy Hoffman (hoffman@cornell.edu) are with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, USA.

[2]Dr. Armin Biess (Corresponding Author, abiess@bgu.ac.il) is with the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er Sheva, Israel

[†]Prof. Hoffman and Dr. Biess contributed equally to this work.

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

we do not address in this work, such as human adaptation to the robot's movement, human safety, or motion legibility. This work also does not use human participants, but uses simulation to study aspects of handovers that have not been addressed in prior work. The main contribution of this work is empirical (evaluating GPS in unexplored scenarios) and model-related (comparing state representations). Our results provide new insights into the advantages and limitations of GPS, and lay the foundation for designing appropriate training regimens for learning human-robot interaction (HRI) controllers with GPS.

II. Related Work

In this section, we provide a brief review of existing controllers for the reach phase of human-robot handovers, and prior work related to GPS.

A. Human-Robot Handover Reach Phase Controllers

Several controllers have been proposed for the reach phase of human-robot handovers, operating either offline or online. Offline controllers [2]–[7] compute the robot's motion plan before the start of the reach phase and do not update it during the reach phase. Offline controllers require the human to adapt to the robot's motion and hence are not desirable, especially in situations where the human is distracted or occupied with other tasks. Our proposed controller is an online controller which constantly updates the robot's motion plan during the reach phase and takes into account the observed behavior of the human.

The simplest online controllers for the reach phase of handovers take a visual servoing approach, i.e., driving the robot towards the human hand [8]–[10]. This controller updates the robot's motion plan continuously by generating velocities proportional to the error between the human hand's position and the robot gripper's position. Some researchers have used other velocity profiles or motion planners to drive the robot towards the human hand or the predicted handover location. For example, Pan et al. [13] used Bézier curves to generate smooth minimum-jerk trajectories; Scimmi et al. [14] used a smooth predefined velocity profile; Kshirsagar et al. [1] used automated synthesis from formal specifications. Similar to our controller, these controllers do not produce a human-like motion. Some online controllers have used movement primitives such as Dynamic Movement Primitives (DMPs) [15], Probabilistic Movement Primitives (ProMPs) [16] and triadic interaction meshes (IMs) [17] to imitate the demonstrated human reaching motions in handovers. Other approaches have used dynamical systems [20], look-up tables [21], or neural networks [22], [23] to encode the demonstrations and generate robot motion in the reach phase. Some researchers have used reinforcement learning techniques to learn online controllers for the reach phase from human feedback [18], [19].

Existing reach phase controllers require known robot dynamics, which may be difficult to obtain for custom built robots and for commercial robots with proprietary claims. Robot dynamics may also change due to the varied and possibly unknown mass of the object to be handed over. System identification techniques can be used to build dynamics models but require large training data especially for building global models. In contrast, GPS is data efficient as it builds local models of the system and uses a combination of locally optimal controllers and a global policy trained using the local controllers via supervised learning.

B. Guided Policy Search

Initial variants of the GPS algorithm [24]–[26] required known dynamics of the system. For optimizing trajectories of systems with unknown dynamics, Levine and Abbeel [27] extended the constrained GPS algorithm of Levine and Koltun [26] with iterative refitting of locally linear dynamics models. They showed that this method requires less samples than model-free methods and does not need to learn global models, which are difficult to learn for complex systems. They evaluated their method on simulated robotic manipulation tasks such as peg insertion, and locomotion tasks such as swimming and walking. Levine et al. [28] extended the evaluation of the algorithm through a variety of experiments on a real robotic platform for tasks such as stacking lego blocks, assembling toys, inserting a shoe tree, inserting rings on wooden pegs and screwing bottle caps.

Levine et al. [29] proposed an end-to-end approach to learn policies that map raw image observations directly to robot joint torques. They used the constrained GPS algorithm and formulated it as an instance of Bregman-Alternating Direction Method of Multipliers (BADMM). They tested this method on tasks that require close coordination between vision and control such as inserting shapes into a sorting cube, screwing a bottle cap, placing hanger on a bar and inserting hammer underneath a nail. Zhang et al. [30] augmented the original GPS algorithm with a model predictive control (MPC) scheme to generate training data without catastrophic failures. They showed that this algorithm was comparable to the original GPS algorithm in the absence of model errors and outperformed the GPS algorithm when model errors were introduced. Chebotar et al. [31] augmented GPS with a model-free local optimizer based on path integral (PI) stochastic optimal control, instead of iLQR, to generate local controllers. Also, unlike GPS algorithms of Levine and Koltun, which used the local controllers to generate training data, Chebotar et al. generated training samples by running the global policy on new sets of task instances in each iteration. This method performed better than iLQR-based GPS on tasks with intermittent and variable contacts and discontinuous cost functions.

While researchers have tested GPS algorithms on a variety of locomotion and autonomous manipulation tasks, to the best of our knowledge, there is no work that evaluated GPS for tasks with large variations in target locations, moving targets and changes in robot dynamics, as are typical in HRI scenarios such as handovers. Also, none of the prior works on GPS have evaluated the sensitivity of GPS to the system's state-space representation. We seek to address this gap in this work by evaluating a robot controller that uses GPS for the reach phase of human-robot object handovers. A large body of work has studied transfer learning [32] and domain adaptation [33] where the training and testing conditions belong to different tasks/distributions. However, in our work the training and testing conditions belong to the same task and are drawn from the same distribution. Therefore, our problem statement is different from transfer learning or domain adaptation.

III. Policy Search Formulation of Handovers

We start by briefly describing the GPS algorithm and then formalize the reach phase of a handover task as a reinforcement learning problem. To do so, we have to specify the state/action space, as well as a cost/reward function in the form of a differentiable function over the system states and control inputs.

A. Guided Policy Search Algorithm

The goal of policy search algorithms is to find a policy $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$ that minimizes the expected cost $E_{\pi_\theta}[\sum_{t=1}^{T} l(\mathbf{x}_t, \mathbf{u}_t)]$ of executing a task. Here θ denotes the policy parameters, for example weights of a neural network, $\mathbf{u}_t$ is the control input at time t, $\mathbf{x}_t$ is the state of the system at time t, and $l(\mathbf{x}_t, \mathbf{u}_t)$ is the cost associated with the task at time t. Directly solving this minimization problem through reinforcement learning requires large amounts of training data and is susceptible to local minima. Guided policy search algorithms overcome these issues through the use of guiding distributions or "local" controllers $p_i(\mathbf{u}_t|\mathbf{x}_t)$ to train the "global" policy $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$ through supervised learning. The local controllers can be trained via trajectory optimization methods such as iLQR. Thus GPS poses the expected cost minimization problem as a constrained problem given by

$$\min_{p,\theta} E_p[\sum_{t=1}^{T} l(\mathbf{x}_t, \mathbf{u}_t)] \quad \text{s.t.} \quad p(\mathbf{u}_t|\mathbf{x}_t) = \pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\ \forall t, \quad (1)$$

where $p(\mathbf{u}_t|\mathbf{x}_t)$ is a mixture of guiding distributions $p_i(\mathbf{u}_t|\mathbf{x}_t)$. The expectation is taken with respect to $p(\tau) = p(\mathbf{x}_1)\prod_{t=1}^{T} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)p(\mathbf{u}_t|\mathbf{x}_t)$, where $\tau = \{\mathbf{x}_1, \mathbf{u}_1, ..., \mathbf{x}_T, \mathbf{u}_T\}$ is a trajectory and $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$ is the dynamics model of the system. As described in Section II-B, some variants of GPS algorithms require known dynamics models while others iteratively learn locally linear dynamics models from the training data.

In this work, we use the Bregman-Alternating Direction Method of Multipliers (BADMM) GPS algorithm proposed by Levine et al. [29] which does not require prior knowledge of the robot dynamics. In this algorithm, the local controllers $p_i(\mathbf{u}_t|\mathbf{x}_t)$ and the dynamics $p_i(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$, $\forall i \in [1, 2, ..., N]$ where N is the number of local controllers, are represented with time-varying Linear Gaussians:

$$p_i(\mathbf{u}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{K}_{t,i}\mathbf{x}_{t,i} + \mathbf{k}_{t,i}, \mathbf{C}_{t,i}), \quad (2)$$

$$p_i(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(f_{\mathbf{x}t,i}\mathbf{x}_t + f_{\mathbf{u}t,i}\mathbf{u}_t + f_{ct,i}, \mathbf{F}_{t,i}). \quad (3)$$

The linear Gaussian controllers and dynamics can be efficiently learned with a small number of samples. A different set of controller and dynamics parameters are fitted for each training target trajectory (in our case: the human's reaching motion). But a single global policy is supervised by all of the local controllers, making it generalizable to different test target trajectories.

Levine et al. [29] suggest modifying the constraint in Eq. 1 by multiplying with $p(\mathbf{x}_t)$ and applying it to expected action, to make the constraint tractable:

$$\min_{p,\theta} E_p[\sum_{t=1}^{T} l(\mathbf{x}_t, \mathbf{u}_t)] \\ \text{s.t. } E_{p(\mathbf{x}_t,\mathbf{u}_t)}[\mathbf{u}_t] = E_{p(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)}[\mathbf{u}_t]\ \forall t. \quad (4)$$

The GPS algorithm alternates between generating optimal trajectories for each local controller with iLQR and training a global policy supervised by the local controllers. The global policy is also used to improve the local controllers, such that the local controllers stay close to the global policy. GPS thus alternates minimization of θ and p as follows:

$$\theta \leftarrow \arg\min_\theta \sum_{t=1}^{T} E_{p(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)}[\mathbf{u}_t^T \lambda_{\mu t}] \\ + \nu_t E_{p(\mathbf{x}_t)}[D_{KL}(p(\mathbf{u}_t|\mathbf{x}_t)||\pi_\theta(\mathbf{u}_t|\mathbf{x}_t))], \quad (5)$$

$$p \leftarrow \arg\min_p \sum_{t=1}^{T} E_{p(\mathbf{x}_t,\mathbf{u}_t)}[l(\mathbf{x}_t, \mathbf{u}_t) - \mathbf{u}_t^T \lambda_{\mu t}] \\ + \nu_t E_{p(\mathbf{x}_t)}[D_{KL}(\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)||p(\mathbf{u}_t|\mathbf{x}_t))], \quad (6)$$

$$\lambda_{\mu t} \leftarrow \lambda_{\mu t} + \alpha\nu_t(E_{p(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)}[\mathbf{u}_t] \\ - E_{p(\mathbf{x}_t)p(\mathbf{u}_t|\mathbf{x}_t)}[\mathbf{u}_t]), \quad (7)$$

where $\lambda_{\mu t}$ is the Lagrange multiplier on the expected action at time t, ν_t is the weight of the Kullback–Leibler divergence term that serves to keep $p(\mathbf{u}_t|\mathbf{x}_t)$ close to $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$. For a detailed description of the GPS algorithm, see [29].

B. System State Representation

Any reinforcement learning method is sensitive to its state representation, and in this work, we explored three alternatives for the system state $\mathbf{x}_t$. The first one is the `FULL` state, which might be available in a laboratory setup supported by a motion tracking system. In this representation, the state consists of the robot joint angles θ_r, the robot joint velocities $\dot{\theta}_r$, the human arm joint angles θ_h, the human arm joint velocities $\dot{\theta}_h$, the positions and velocities of three points on the object $(\mathbf{p}_o, \dot{\mathbf{p}}_o)$, the human hand $(\mathbf{p}_h, \dot{\mathbf{p}}_h)$ and the robot gripper $(\mathbf{p}_r, \dot{\mathbf{p}}_r)$, and the robot gripper's width $g_r \in [0, g_{open}]$ (0 for fully closed, g_{open} for fully open). The positions are measured in an inertial frame fixed to the base of the robot. A state is thus given by

$$\mathbf{x}_t = [\theta_r, \dot{\theta}_r, \theta_h, \dot{\theta}_h, \mathbf{p}_o, \mathbf{p}_h, \mathbf{p}_r, \dot{\mathbf{p}}_o, \dot{\mathbf{p}}_h, \dot{\mathbf{p}}_r, g_r]_t. \quad (8)$$

As the human's joint angles are difficult to measure for a robot outside a laboratory, we also explore a `REDUCED` state representation, which excludes the human arm joint angles and joint velocities:

$$\mathbf{x}_t = [\theta_r, \dot{\theta}_r, \mathbf{p}_o, \mathbf{p}_h, \mathbf{p}_r, \dot{\mathbf{p}}_o, \dot{\mathbf{p}}_h, \dot{\mathbf{p}}_r, g_r]_t. \quad (9)$$

Given the possible large variation in human position, we also explore a third option, which includes the human hand and the object poses in the robot end-effector frame instead of an inertial frame fixed to the base of the robot. This RELATIVE representation corresponds to the configuration in which a camera is attached to the robot end-effector:

$$\mathbf{x}_t = [\theta_r, \dot{\theta}_r, \mathbf{p}_o^r, \mathbf{p}_h^r, \dot{\mathbf{p}}_o^r, \dot{\mathbf{p}}_h^r, g_r]_t. \tag{10}$$

In all of the three alternatives, the robot's control input $\mathbf{u}_t = [\tau, f_g]_t$ consists of the robot joint torques τ and the force applied by the gripper's actuator f_g, constrained by $\mathbf{u}_{\mathbf{min}} \leq \mathbf{u}_t \leq \mathbf{u}_{\mathbf{max}}$.

We use the REDUCED state representation in the majority of the results below. We conclude with an exploratory comparison with the two other state representations.

We use torques as control inputs instead of velocities or positions to take into account the dynamics of the robot. This eliminates the need to tune low-level position/velocity controllers. Also, position or velocity controllers might exert large impact forces on the human while trying to move the robot with commanded position or velocity. Therefore torque controllers are preferred over position or velocity controllers for human-safe robot behavior.

C. Cost Function

In the reach phase of handovers, the robot should move its gripper towards the human hand. We represent this behavior with a cost function given by

$$c_{reach} = ||\mathbf{p}_r - \mathbf{p}_h||^2 + \ln(||\mathbf{p}_r - \mathbf{p}_h||^2 + \alpha_{reach}). \tag{11}$$

The first term of this cost function penalizes robot positions far away from the human hand, while the second term encourages precise placement due to its concave shape, as described in [28]. Thus this cost function encourages the robot to reach towards the human hand quickly and precisely. The parameter α_{reach} determines the penalty in the vicinity of the target. Similar to [28], we set $\alpha_{reach} = 1e-5$ in the evaluations described in the next section.

IV. Evaluation

We evaluate the performance of the global policy learnt with GPS for large variations in target locations, moving targets, and changes in robot dynamics. To do so, we train a collaborative robot to perform handovers over repeated trials in a simulation environment with different training regimens, and test on different target trajectories. We measure the performance of the global policy in terms of the error between the end-effector's position and the human hand's position.

A. Implementation

We build upon the BADMM-GPS implementation by Finn et al. [34]. The collaborative robot in the handover task is simulated in MuJoCo [35] (Multi-Joint dynamics with Contact). Fig. 1 shows the MuJoCo simulation environment built for this study. The robot on the left is a Franka-Emika Panda with 7 degrees-of-freedom (DoFs), equipped with a two fingered gripper. In the remaining text we call this robot the "learner". The environment also includes a pseudo-robot arm with two DoFs and a mass rigidly attached to its end-effector. In the remaining text we call this robot the "trainer" or the "tester" depending on whether it is used to train the global policy or to test the learnt global policy. This robot stands in for the human and "teaches" the learner to perform handover reaching motions in simulation.

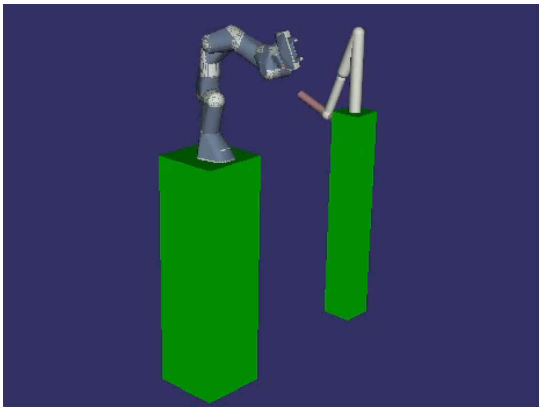


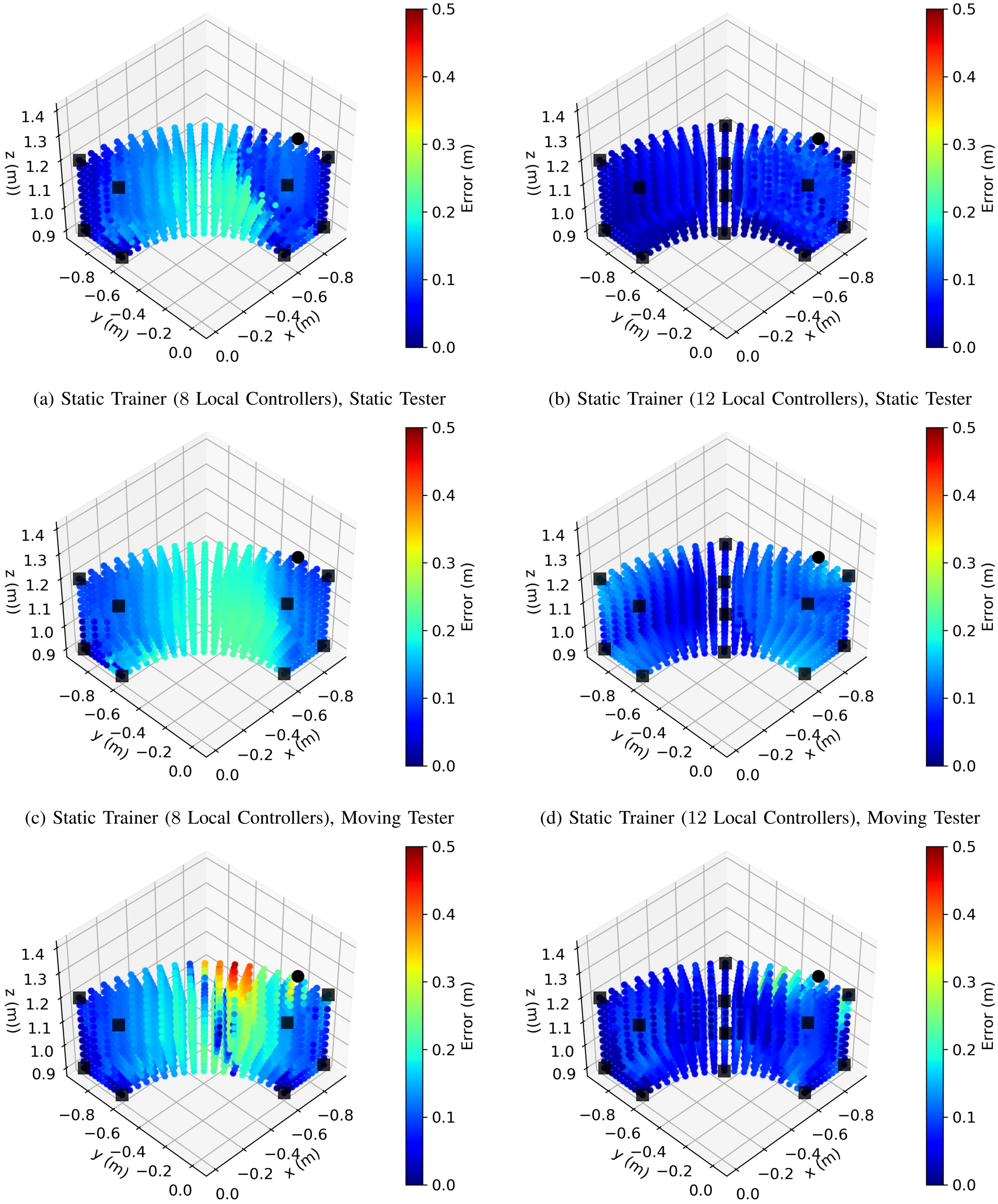
Fig. 1: MuJoCo simulation environment. We train a Panda robot arm (left) to perform handover reaching motions by simulating the reach phase with another robot (right), standing in for the human.

B. Simulation Results

The first research question that we investigate is the spatial generalizability of the learnt global policy, i.e., how does the global policy perform for large spatial differences between training and test locations. To answer this question, we test the learnt global policy at different locations of a static tester on a semi-hemispherical shell around the learner robot, which represents the workspace of the robot. For each angle in $5\,\mathrm{deg}$ increments, we test on a grid of 11×11 targets, resulting in 2299 test locations. We initially train the global policy with eight local controllers for target locations at the corners of the workspace. Each trial runs for 2 seconds, both the learner and the trainer/tester start moving at the same time, and the global policy is improved over 12 trials. The test performance is measured as the mean error between the learner's gripper position and the tester's hand position over the last 0.5 seconds of each trial.

Fig. 2a shows the performance of the learnt global policy, The training locations are marked with black squares and the learner's gripper's initial position with a black circle. Fig. 3 (left) shows the mean, range, and standard deviation of the error. The mean testing error (128 mm) is more than 6 times the mean training error (20 mm). The error increases up to 241 mm as the spatial distance between the training and the testing target locations increases. This issue can be somewhat addressed by adding four additional local controllers trained with target locations in the plane dividing the workspace (Fig. 2b). For a global policy trained with these 12 local controllers, the mean and standard deviation of the testing error is reduced to 69 ± 32mm.

Next, we investigate how GPS performs when the target is moving. First, we used the same global policy shown in Fig. 2a



(a) Static Trainer (8 Local Controllers), Static Tester

(b) Static Trainer (12 Local Controllers), Static Tester

(c) Static Trainer (8 Local Controllers), Moving Tester

(d) Static Trainer (12 Local Controllers), Moving Tester

(e) Moving Trainer (8 Local Controllers), Moving Tester

(f) Moving Trainer (12 Local Controllers), Moving Tester

Fig. 2: Global policy evaluation for different types of trainers and testers. In the ‘static’ case, the trainer/tester stays in a fixed configuration. In the ‘moving’ case, the trainer/tester moves with a human-like trajectory and reaches the locations given by colored dots. Thus each point corresponds to the final position of the tester’s gripper in a trial, and the black square markers correspond to the training target locations. The black round marker corresponds to the learner robot’s gripper’s starting position. Error between the learner’s gripper position and the tester’s gripper position is averaged over the last 0.5 seconds of each trial. We find that: 1) Error increases as the target location is shifted away from the training locations (all figures). 2) Increasing the number of training locations reduces the error (left column vs right column). 3) Error is higher if the trainer is static and the tester is moving (first row vs second row). 4) Error is more sensitive to location and performs worse in the worst case if the global policy is trained on a moving target (bottom row).

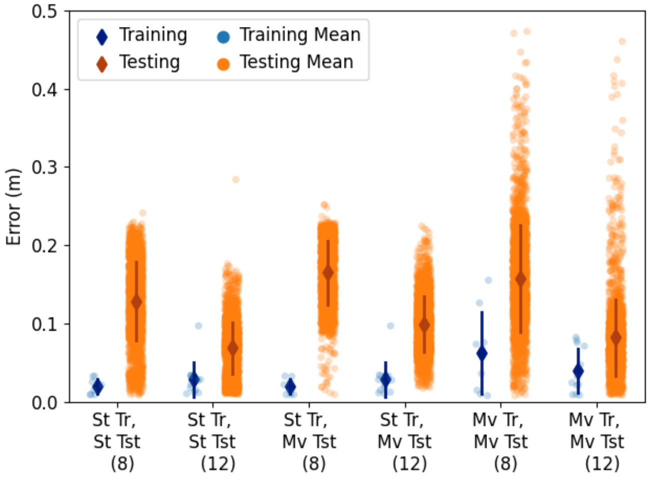


Fig. 3: Distributions of training and testing performance for each target scenario. Each point is the mean error between the learner's gripper position and the tester's hand position over the last 0.5 seconds of a trial. Error bars show one standard deviation around the mean of each distribution.

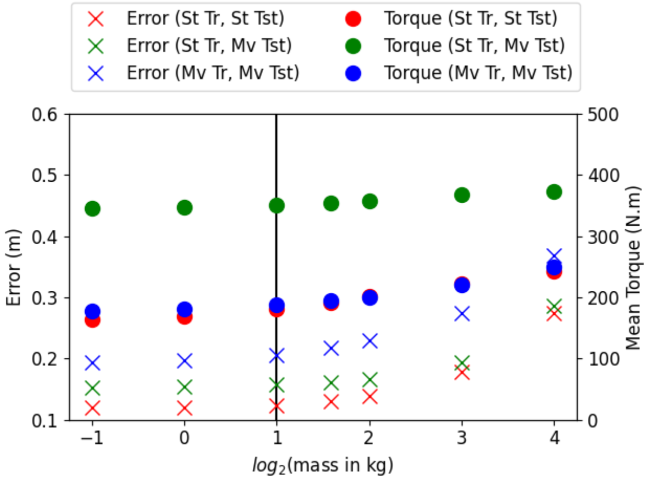


Fig. 4: Mean error and torque for different robot end-effector masses (0.5–16kg). The vertical line represents the baseline mass of the robot end-effector used during training (2kg). Each error marker corresponds to the mean error of all testing locations on a semi-hemispherical shell around the robot as shown in Fig. 2. Torque markers show the mean of the norm of torques applied by the robot, averaged over the same testing locations. The error remains fairly constant over a wide range of the robot end-effector mass (up to 4kg), and the global policy produces total torques proportional to the changes in the robot mass. Torques are significantly higher for the 'static train, moving test' scenario, indicating highly inefficient trajectories.

(static training) but instead of a static tester we simulate the tester to execute a human-like trajectory in joint space [36], given by

$$\theta_{h,i} = \frac{a(\theta_{f,i} - \theta_{0,i})}{b + e^{\frac{-ct}{t_f}}} + \theta_{0,i} \quad \forall i \in 1, 2, \tag{12}$$

where $a = 9.05e^{-4}$, $b = 8.908e^{-4}$ and $c = 12.87$ are empirical coefficients determined by Rasch et al. [36] from human arm motion data. $\theta_{0,i}$ and $\theta_{f,i}$ are the initial and final values of the i^{th} joint angle, respectively, t_f is the movement duration, while $\theta_{h,1}$ and $\theta_{h,2}$ correspond to the shoulder and elbow joints, respectively. We set $\theta_{0,1} = \theta_{0,2} = 0$, $t_f = 1$, and use inverse kinematics to compute the final values of $\theta_{h,1}, \theta_{h,2}$ for a given Cartesian position of the tester's gripper. We vary the tester's trajectories such that its gripper's final position is on the same semi-hemispherical shell around the learner robot as before. The global policy's performance is again measured as the mean error between the learner's gripper position and the tester's gripper position over the last $0.5s$ of each trial. Since we set $t_f = 1$ in Eq. 12, this error is calculated after the tester has reached the final position.

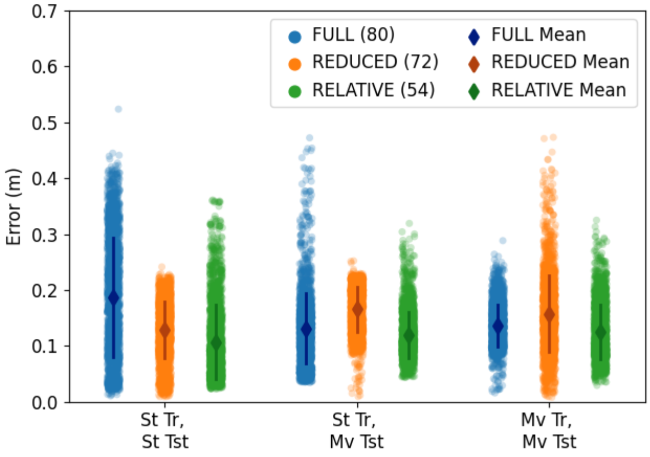


Fig. 5: Distributions of testing performance for different state representations described in Section III-B. In each case the global policy is trained with 8 local controllers. The number in parentheses represents the number of state variables in each representation. Each point is the mean error between the learner's gripper position and the tester's hand position over the last 0.5 seconds of a trial. Error bars show one standard deviation around the mean of each distribution.

Fig. 2c shows the results for the global policy trained with 8 local controllers; Fig. 3 (middle) shows the mean, range, and standard deviation of the error. The performance is worse with a mean testing error of 165 mm for a moving target, 28.9% higher than the mean testing error for a static target, but the range of error is comparable. Few target locations result in low errors. For the global policy trained with 12 local controllers (Fig. 2d), the mean testing error is 99 mm for a moving target, 43.5% higher than the mean testing error for a static target. The range, again, is comparable, with more target locations, as compared to the global policy trained with 8 local controllers, having low errors.

That said, the trajectories generated by these "Static Trainer, Moving Tester" trials are highly inefficient. The video attachment shows examples of the resulting circumvent reach trajectories, and Fig. 4 (center line) shows that the mean torque i.e. the L^2 norm of the robot's joint torques averaged over all test points and time-steps, is almost double over the trajectory.

A possible way to address this issue is to train the controller with a moving target, also executing a human-like trajectory in the joint space, as described in Eq 12. Fig. 2e and Fig. 2f show the performance of the global policy for various final positions of the tester's gripper, defined as in previous trials. Fig. 3 (right) shows error distributions.

For the global policy trained with a moving trainer and 8 local controllers (Fig. 2e), the mean testing error is 157mm, and thus does not provide a meaningful improvement. Moreover, the variance over target location is high, and the worst-case error is 473 mm, 87.7% higher than the maximum error for the static trainer condition (252 mm). In fact, the GPS process does not converge to a low training error, which is more than 3x that of the static training results. For the global policy trained with a moving trainer and 12 local controllers (Fig. 2f), the mean testing error is reduced to 82 mm, 17.2% lower than the mean testing error for the static trainer condition. But the variance of the performance remains high, with a 461 mm worst case performance. That said, an inspection of the generated trajectories and torques shows that this approach results in more efficient trajectories and torques similar to those achieved with static targets.

The third research question that we address is how the global policy performs under changes in robot end-effector's mass. To investigate this question, we train the robot with a baseline end-effector mass of 2kg and evaluate the performance of the global policy for different robot end-effector masses, ranging from ~0.5kg to ~16kg. Fig. 4 shows the mean error between the learner's gripper position and the tester's gripper position for different robot end-effector masses. We find that the mean error across the same testing locations as shown in Fig. 2 remains largely unaffected between $0.5-4$kg, but the error increases if the end-effector's mass is increased beyond this limit. Fig. 4 also shows means of the norm of torques applied by the seven joints of the robot for different robot masses. We find that the mean increases with increase in the robot end-effector's mass, except when the robot is trained on static targets but tested on moving targets, where the torques are always high. We also investigated the effect of changing the total mass of the robot, and found that for a baseline mass of 18.5kg the error remained fairly constant up to 100kg.

In section III-B, we proposed different possible state representations. Fig. 5 shows the performance of the global policy trained with 8 local controllers, across all three state models. For policies trained on static targets, the `REDUCED` state representation has the lowest variance (best generalization), but this does not hold for policies trained on moving targets. Overall, a global policy trained with the lowest-dimensional `RELATIVE` state representation (54 dimensions) has a better average performance than the other state representations. This suggests that lower-dimensional state models may be more appropriate for GPS-trained handover controllers.

V. Discussion and Conclusion

We evaluate the feasibility of GPS as a learning method for human-robot handovers. We use a variant of the GPS algorithm that does not require prior knowledge of the robot dynamics, and instead, learns locally linear dynamics models from the training data [29]. Previously, GPS was used for tasks in which the environment was static and the variations in target locations were small. To successfully complete a handover, however, the robot must cope with a dynamic environment including unpredictable human motion in a wide range of target locations holding objects of different mass. Our study thus contributes to the design of control policies for human-robot handover tasks by providing a detailed analysis of GPS in terms of three of these requirements: moving targets, large variations in target location, and a changing end-effector mass.

When evaluating static reach targets only, we find that the performance of the GPS-learned global policy does not generalize well to spatial variations in target locations, and its performance worsens significantly (Fig. 2a). The performance of the global policy can be improved by training it with more local controllers (Fig. 2a vs Fig. 2b). The additional local controllers should be trained with target locations distributed in the regions with high testing errors.

When evaluating the global policy with a moving target which was simulated to mimic human reaching motions, the performance of the global policy decreases on average, but can still achieve reasonable error performance, especially in areas near the training locations (Fig. 2a vs Fig. 2c). Similar to the static case, the generalizability of the performance of the global policy can be improved by training it with more local controllers (Fig. 2e vs Fig. 2f). However, a global policy trained with static targets results in highly inefficient trajectories for moving targets, which are not only high-torque, but would be confusing to a human confronted with them. The obvious solution of training the global policy with moving targets is a double-edged sword. It is successful in reducing the mean error and results in more legible and low-torque efficient trajectories, but at the cost of a more high-variance (unreliable) global policy with significantly larger worst-case errors. Further research is required to strike the best balance of trajectory shape, efficiency, and reach error.

In a handover task, the robot end-effector's mass could be different in the training and testing scenarios due to different objects being handed over. We found that the trained global policy adapts well to a range of changes in the robot end-effector's mass. The robot is able to reach the target locations with similar accuracy even with large variations in the end-effector's mass, but only up to a limit as shown in Fig. 4. This adaptability could be because our cost function (Eq. 11) results in a global policy which is similar to a proportional visual servoing controller. This controllers adapts to changes in robot mass by applying control inputs proportional to the error between the desired position and the current position. Another possible explanation for the invariance of the error under changes of robot mass could be that changes in the robot's mass do not have a large effect on the robot's trajectory in state-space, and hence, on the performance of the global policy. Contrarily, shifting the target location in the Cartesian space away from the training locations also shifts the robot's trajectory away from the explored region of the robot's state-space, and thus worsens the global policy's performance.

In contrast to prior works on GPS, we also present an exploratory study of the effect of different state representations on the performance of GPS. We found that removing the human's joint angles and velocities from the state representation, and expressing the human hand's position and velocity in a reference frame attached to the robot gripper, improved the performance of the trained global policy. This suggests that a low dimensional state-space would be more suitable for GPS, even though it contains less information about the task dynamics.

This work presents initial steps toward using GPS for human-robot handovers. We did not consider other important aspects of handovers, such as the human adaptation to the robot's motion, their proactivity, the legibility of the robot's movement, and so forth. Our studies were also conducted in simulation with a robot arm standing in for the human, generating the variability and movement of the handover target location. While this allows for highly controlled empirical conditions, their application in a real-world context is limited. In future work, we plan to test GPS on a physical robot for object handovers with human participants. Still, the current study contributes to our understanding of the possibilities and limits of GPS with respect to important aspects of human-robot collaboration.

REFERENCES

[1] A. Kshirsagar, H. Kress-Gazit, and G. Hoffman, "Specifying and synthesizing human-robot handovers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5930–5936.

[2] M. Cakmak, S. Srinivasa, M. Lee, S. Kiesler, and J. Forlizzi, "Using spatial and temporal contrast for fluent robot-human hand-overs," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2011, pp. 489–496.

[3] A. Moon, D. Troniak, B. Gleeson, M. Pan, M. Zheng, B. Blumer, K. MacLean, and E. Croft, "Meet me where I'm gazing: How shared attention gaze affects human-robot handover timing," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2014, pp. 334–341.

[4] E. Sisbot and R. Alami, "A human-aware manipulation planner," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1045–1057, 2012.

[5] L. Peternel, W. Kim, J. Babic, and A. Ajoudani, "Towards ergonomic control of human-robot co-manipulation and handover," in *IEEE-RAS International Conference on Humanoid Robotics*, 2017, pp. 55–60.

[6] M. Cakmak, S. Srinivasa, M. Lee, J. Forlizzi, and S. Kiesler, "Human preferences for robot-human hand-over configurations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1986–1993.

[7] R. Rasch, S. Wachsmuth, and M. König, "An evaluation of robot-to-human handover configurations for commercial robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7588–7595.

[8] V. Micelli, K. Strabala, and S. Srinivasa, "Perception and control challenges for effective human-robot handoffs," in *Robotics: Science and Systems (RSS) Workshop on RGB-D Cameras*, 2011.

[9] M. Bdiwi, A. Kolker, J. Suchỳ, and A. Winkler, "Automated assistance robot system for transferring model-free objects from/to human hand using vision/force control," in *International Conference on Social Robotics*, 2013, pp. 40–53.

[10] M. Pan, E. Croft, and G. Niemeyer, "Exploration of geometry and forces occurring within human-to-robot handovers," in *IEEE Haptics Symposium*, 2018, pp. 327–333.

[11] W. He, D. Sidobre, and R. Zhao, "A Reactive Trajectory Controller for Object Manipulation in Human Robot Interaction," in *International Conference on Informatics in Control, Automation and Robotics*, 2013.

[12] A. Fishman, C. Paxton, W. Yang, N. Ratliff, and D. Fox, "Trajectory optimization for coordinated human-robot collaboration," *arXiv preprint arXiv:1910.04339*, 2019.

[13] M. Pan, E. Knoop, M. Bächer, and G. Niemeyer, "Fast handovers with a robot character: Small sensorimotor delays improve perceived qualities," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6735–6741.

[14] L. Scimmi, M. Melchiorre, S. Mauro, and S. Pastorelli, "Experimental real-time setup for vision driven hand-over with a collaborative robot," in *International Conference on Control, Automation and Diagnosis (ICCAD)*, 2019, pp. 1–5.

[15] M. Prada, A. Remazeilles, A. Koene, and S. Endo, "Implementation and experimental validation of Dynamic Movement Primitives for object handover," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 2146–2153.

[16] G. Maeda, M. Ewerton, R. Lioutikov, H. Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 527–534.

[17] D. Vogt, S. Stepputtis, B. Jung, and H. Amor, "One-shot learning of human–robot handovers with triadic interaction meshes," *Autonomous Robots*, vol. 42, no. 5, pp. 1053–1065, 2018.

[18] A. Kupcsik, D. Hsu, and W. Lee, "Learning dynamic robot-to-human object handover from human feedback," *Robotics Research*, vol. 1, pp. 161–176, 2017.

[19] F. Riccio, R. Capobianco, and D. Nardi, "Learning human-robot handovers through π-STAM: Policy improvement with spatio-temporal affordance maps," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 857–863.

[20] J. Medina, F. Duvallet, M. Karnam, and A. Billard, "A human-inspired controller for fluid human-robot handovers," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 324–331.

[21] K. Yamane, M. Revfi, and T. Asfour, "Synthesizing object receiving motions of humanoid robots with human motion database," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 1629–1636.

[22] X. Zhao, S. Chumkamon, S. Duan, J. Rojas, and J. Pan, "Collaborative human-robot motion generation using LSTM-RNN," in *IEEE-RAS International Conference on Humanoid Robots*, 2018, pp. 1–9.

[23] W. Yang, C. Paxton, M. Cakmak, and D. Fox, "Human grasp classification for reactive human-to-robot handovers," *arXiv preprint arXiv:2003.06000*, 2020.

[24] S. Levine and V. Koltun, "Guided policy search," in *International Conference on Machine Learning*, 2013, pp. 1–9.

[25] ——, "Variational policy search via trajectory optimization," in *Advances in neural information processing systems*, 2013, pp. 207–215.

[26] ——, "Learning complex neural network policies with trajectory optimization," in *International Conference on Machine Learning*, 2014, p. II–829–II–837.

[27] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," in *Advances in Neural Information Processing Systems*, 2014, pp. 1071–1079.

[28] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 26–30.

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

[30] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 528–535.

[31] Y. Chebotar, M. Kalakrishnan, A. Yahya, A. Li, S. Schaal, and S. Levine, "Path integral guided policy search," in *IEEE international conference on robotics and automation (ICRA)*, 2017, pp. 3381–3388.

[32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[33] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[34] C. Finn, M. Zhang, J. Fu, W. Montgomery, X. Yu Tan, Z. McCarthy, B. Stadie, E. Scharff, and S. Levine, "Guided policy search code implementation," Software available from rll.berkeley.edu/gps (2020/06/19).

[35] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[36] R. Rasch, S. Wachsmuth, and M. Konig, "A joint motion model for human-like robot-human handover," in *IEEE-RAS International Conference on Humanoid Robots*, 2018, pp. 180–187.