PSRO-Based Robust Handover Policy Under Adversarial Perturbations

Boxiang (William) Fu
Robotics Institute, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
boxiangf@andrew.cmu.edu

Abstract

We formulate robust human–robot handover as a two-player zero-sum game between a manipulator agent and an adversarial perturbation generator. Policy Space Response Oracles (PSRO) is used to iteratively compute best-responses and Nash meta-strategies over their respective policy spaces. Baseline PSRO failed to improve the agent due to repeated re-initialization and insufficient training. To address this, we introduce pre-training to initialize the agent using a checkpoint that has already learned the nominal handover task. The resulting meta-strategy achieved much better results and increased the handover success rate from 42% to 61% under no disturbances and from 29% to 58% under adversarial sampling. These results motivate the use of PSRO as a promising game-theoretic framework for learning robust manipulation policies for safe human-robot interaction tasks.

1 Introduction

With the advancement of embodied artificial intelligence and humanoid robotics, collaboration between humans and robots has become increasingly central to next-generation human-robot interactive systems (Goodrich and Schultz [2007], Dahiya et al. [2022]). Nowhere is this more evident than in the task of physical handover, where an object must be safely and reliably transferred between a human and a robot (Kshirsagar et al. [2021], Duan et al. [2024]). An accurate and robust handover policy is foundational for a wide spectrum of real-world applications — including collaborative assembly, surgical and rehabilitation assistance, warehouse operations, household support, and general-purpose service robotics.

Despite its importance, modern robotic manipulation policies often lack perturbation robustness, performing well under nominal conditions but failing under external disturbances, human motion unpredictability, or environmental variability. In this paper, we investigate a game-theoretic framework to robustify the robot-human handover manipulation policy. We formulate the manipulator-environment interaction as a two-player zero-sum game and employ Policy Space Response Oracles (PSRO) (Lanctot et al. [2017]) to iteratively find best-responses for both players. The learned distribution of best-response meta-strategies constitutes an approximate Nash equilibrium for the two-player game (McMahan et al. [2003]) and enables the manipulator agent to more robustly adapt to adversarial perturbations.

2 Related Work

Previous approaches to robustifying robotic manipulation often rely on rule-based safety mechanisms, such as the Franka Emika Watchman package (Franka Emika GmbH [2022]), which stops the robot when the force sensor detects a collision or disturbance. Beyond such rule-based systems, a large

Final course project for 15-888 Computational Game Solving (Fall 2025).

Algorithm 1 Robust Handover Policy Under Adversarial Perturbations

Require: Initial policy sets for the agent and adversary

$$\Pi : \{\Pi_a, \Pi_v\}$$

Compute expected utilities as empirical payoff matrix

$$U^{\Pi} \text{ for each joint policy } \pi : \{\pi_a, \pi_v\} \in \Pi$$

Compute meta-Nash equilibrium σ_a and σ_v over policy sets (Π_a, Π_v)
for PSRO iteration in $\{1, 2, \ldots\}$ **do**
    for many iterations N_{π_a} **do**
        Sample the adversary policy $\pi_v \sim \sigma_v$
        Train π'_a with trajectories against the fixed adversary π_v
    end for
    $\Pi_a \leftarrow \Pi_a \cup \{\pi'_a\}$
    for many iterations N_{π_v} **do**
        Sample the agent policy $\pi_a \sim \sigma_a$
        Train the adversary policy π'_v with trajectories
    end for
    $\Pi_v \leftarrow \Pi_v \cup \{\pi'_v\}$
    Compute entries in U^{Π} from Π from rollouts
    Compute new meta strategies σ_a and σ_v from U^{Π}
end for
return Current meta-Nash equilibrium on whole population σ_a and σ_v

body of work in robust reinforcement learning (RL) seeks to make policies resilient to adversarial perturbations of states, actions, or dynamics. Strategies include regularization-based techniques (Oikarinen et al. [2021], Zhang et al. [2020], Shen et al. [2020]), gradient-based adversarial attacks (Vinitsky et al. [2020], Franzmeyer et al. [2024]), alternating training methods (Zhang et al. [2021], Sun et al. [2023]), adversarial skill learning using SAC (Jian et al. [2021]), and worst-case motivated methods (Liang et al. [2022]).

A more recent line of work for robust manipulation relies on modeling the agent-environment interaction as a two-player zero-sum game. Such games can be solved to epsilon-precision using CFR-based techniques such as tabular CFR (Zinkevich et al. [2007], Farina et al. [2019]), Deep CFR (Brown et al. [2018]), DREAM (Steinberger et al. [2020]), and ESCHER (McAleer et al. [2023]). For our setting, we employ the use of Policy-Space Response Oracles (PSRO) (Lanctot et al. [2017]) to iteratively add best-response policies for each player and compute approximate Nash equilibria over the resulting restricted game. This has been extended in the literature to include temporally coupled disturbances (Liang et al. [2024]).

3 Preliminaries

3.1 Methodology

To robustify the human-robot handover policy, we formulate the interaction as a two-player zero-sum game between a manipulator policy (agent) and an adversarial disturbance generator (adversary). The agent seeks to maximize the probability of successfully receiving an object from a human hand, while the adversary applies force, torque, and duration perturbation profiles to destabilize the transfer and minimize the success rate. Following the framework proposed by GRAD (Liang et al. [2024]), we use a PSRO approach (Lanctot et al. [2017]) to iteratively expand the population of agent and adversary policies through best-response training and recomputing Nash meta-strategies over their restricted policy spaces. The high level pseudo-code of our method is provided in Algorithm 1.

We initialize the algorithm using an initial set of agent and adversary policies. We estimate the payoff matrix empirically by performing rollouts of a pairwise agent-adversary policy and aggregating its success rate. At each PSRO iteration, the best-response agent policy is trained against the adversary meta-strategy, and the best-response adversary perturbation profile is trained against the agent meta-

strategy. We append the learned best-response strategies to their respective policy sets. The empirical payoff matrix is recomputed, and the updated Nash equilibrium of the expanded game defines new meta-strategies for both players.

3.2 Simulation

All environment interactions and rollouts are performed using Handover-Sim, an Isaac Gym–based simulation environment developed by NVIDIA Research (Chao et al. [2022]). The environment models the agent as a Franka Emika Panda manipulator with a 7 degree of freedom arm and 2 degree of freedom end-effector. The adversary is modeled as Gaussian perturbations to both the object and the hand that holds the object. Figure 1 shows a graphical depiction of the environment setup. To reduce the training process, we use a simplified scenario in which we train only on the cereal box object. This allowed us to save compute on training a generalist grasping policy and focus our attention on the robustness properties of our handover policy rather than general object recognition or grasp planning.

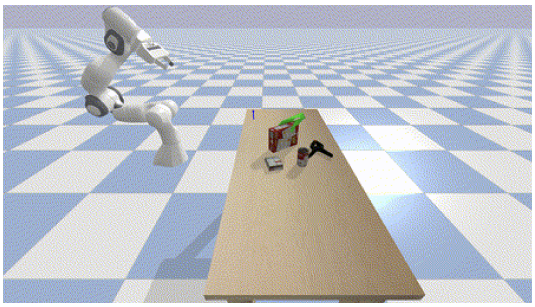

Figure 1: Handover-Sim environment setup

4 Implementation

4.1 Initialization

Algorithm 1 is initialized by constructing an initial restricted game in which both the agent and the adversary possess only a single action. The agent's initial policy is a manipulator policy snapshot trained for 40,000 epochs against a no disturbance adversary. The adversary's initial policy set contains only a zero-perturbation profile. Rollouts on this agent-adversary pair resulted in an expected handover success rate of 42%. Thus, we initialize the empirical payoff matrix U^{Π} to be a 1×1 matrix with the entry 0.42. Because both players have singleton action sets, their initial meta-strategies are to play their respective policies with probability 1.

We limit the number of outer loop PSRO iterations to 9 mainly due to computational constraints. At each PSRO iteration, a best-response policy is trained for both the agent and the adversary, appended to the policy set, and is evaluated against all opponent policies to update the empirical payoff matrix.

4.2 Agent (Manipulator) Policy

The agent's best-response policy is trained using a Deep Deterministic Policy Gradient (DDPG) reinforcement learning algorithm (Lillicrap et al. [2019]) and augmented using a Goal-Auxiliary Actor-Critic network (GA-DDPG) (Wang et al. [2021]). Practically, any neural network-based learning architecture could be used instead of GA-DDPG as a black-box optimizer. Observations consist of simulated depth and point cloud features from the simulator, and the agent outputs continuous end-effector pose commands. Training was carried out by simultaneously interacting with the environment and offline learning using a stored replay buffer. The learning pipeline is shown graphically in Figure 2 and proceeds as follows:

1. **State Representation**: Represent the state as a point cloud and action as an end-effector pose.

2. **Expert Label Generation**: Generate per-timestep expert action labels and final grasp goals using a classical planner.

3. **Actor–Critic Training**: Train actor and critic networks using policy gradients (DDPG) using both online interactions and offline stored replay buffer.

4. **Goal-Auxiliary Heads**: Add auxiliary prediction heads to both the actor and critic.

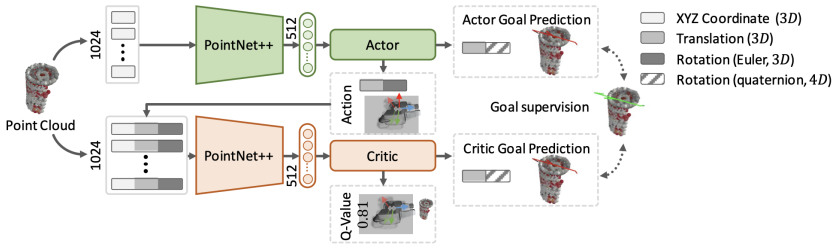

Figure 2: GA-DDPG training pipeline (from Wang et al. [2021])

Two variants of the policy training pipeline were implemented. The first (baseline) variant initializes the agent policy network from a uniform randomized policy for each PSRO iteration. The agent's best-response policy is then trained by interacting with the environment for 40,000 epochs. The second (pre-train) variant loads the agent's policy network using a baseline checkpoint that was trained against a no-disturbance adversary. This avoids having to re-learn the basic handover behavior for each PSRO epoch. The agent is then fine-tuned for an additional 20,000 environment interaction epochs against perturbations sampled from the adversary's current meta-strategy. Each newly trained best-response policy is appended to the agent's policy set at the end of training.

4.3 Adversary (Perturbation) Policy

The adversary policy is parameterized by a triplet $[\sigma_{\text{force}},\ \sigma_{\text{torque}},\ \sigma_{\text{duration}}]$ corresponding to the standard deviations of the Gaussian force (N), torque (Nm), and disturbance duration (s) applied to both the object and the end-effector. For example, a three-tuple of $[1.0,\ 1.0,\ 1.0]$ corresponds to a standard deviation Gaussian perturbation force of 1 Newton, a torque of 1 Newton-meter, and for 1 second applied to the object and the end-effector. The adversary is constrained by a budget of 6 "credits" that can be allocated across the three axes, with a maximum of 2.0 credits per dimension.

For each PSRO iteration, we optimize the perturbation profile using 15 iterations of Bayesian optimization. The optimized best-response policy is appended to the adversary's policy set at the end of training. The Bayesian optimization framework is represented graphically in Figure 3 and follows the four-step process shown below:

1. **Surrogate Model**: Construct a surrogate model that predicts performance (agent success rate) as a function of perturbation parameters.

2. **Acquisition Function**: Use an acquisition function to choose the next hyperparameter point to evaluate.

3. **Evaluate**: Evaluate the candidate by running multiple rollouts against the agent meta-strategy.

4. **Update**: Update the surrogate model with the new observation and repeat.

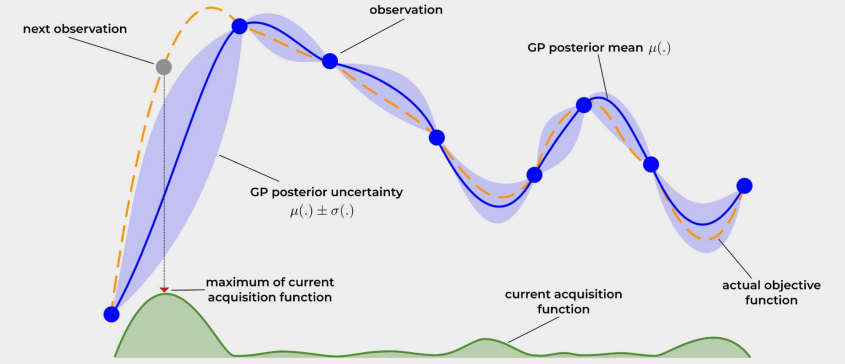

Figure 3: Bayesian optimization pipeline (from Al-Hafez [2021])

4.4 Meta-Strategy Computation

After adding new best-response policies for both the agent and adversary, we compute the empirical payoffs for each agent–adversary pair by performing 100 environment rollouts in the simulation. Each element in the expanded payoff matrix is populated according to

$$U_{ij}^{\Pi} = u(\pi_a^i, \pi_d^j),$$

where u is the success probability of the agent from the rollouts. The meta-strategy of both players is computed by solving the two-player zero-sum Nash equilibrium of this new restricted game. Linear programming is used to find the minimax equilibrium of the normal form game. The resulting Nash equilibrium meta-strategies determine how opponents are sampled in the next PSRO iteration.

4.5 Returns

After the PSRO iterations are complete, the algorithm returns the following objects:

1. A probability vector for the agent meta-strategy.
2. A vector for the agent containing the neural network for each agent policy.
3. A probability vector for the adversary meta-strategy.
4. A vector for the adversary containing the perturbations used for each adversary policy.

The final meta-strategy of the manipulator agent represents a robust ensemble over all learned policies, whereas the learned disturbance meta-strategy reveal which perturbations are most harmful to handover stability.

5 Training

Two variants of Algorithm 1 were trained. The baseline algorithm using the re-train agent policy was trained on a NVIDIA L4 GPU and 128 GB RAM on Google Cloud. The training process took approximately 1.5 days and had $360,000$ agent policy updates. The pre-train algorithm using the pretrained checkpoint agent policy was trained on a NVIDIA RTX 3050 GPU and 16 GB RAM on a personal laptop. The training process also took approximately 1.5 days and had $180,000$ agent policy updates. For both variants, the adversary had 135 parameter updates and a total of $42,000$ agent vs. adversary evaluations were performed.

6 Results

6.1 Baseline

The baseline policy yielded sub-optimal results. The final agent Nash meta-strategy is shown in Figure 4 and assigns probability 1 to the initial snapshot agent policy, with all subsequently added

agent policies receiving zero probability. This is an indication that the agent was not learning against the adversary in the PSRO iterations as it failed to produce improved strategies. The adversary's strategy was also highly concentrated and had only two disturbance policies remaining in its strategy set (Figure 5):

- Disturbance of $[0.62,\ 1.72,\ 0.97]$ with 0.33 probability.
- Disturbance of $[1.06,\ 1.74,\ 0.66]$ with 0.67 probability.

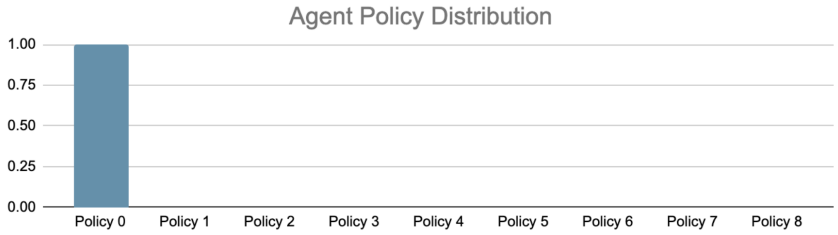

Figure 4: Baseline agent policy

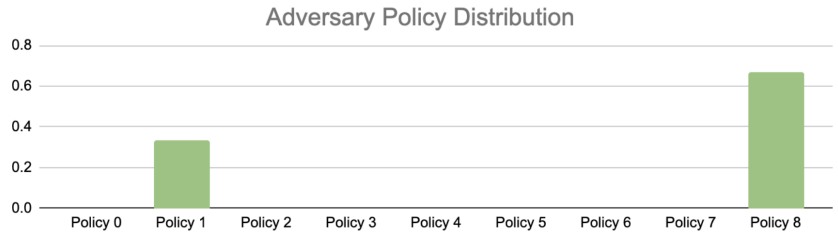

Figure 5: Baseline adversary policy

The agent-adversary interaction yielded a game value of 0.5 based on the empirical payoff matrix:

$$\begin{bmatrix} 0.6 & 0.5 & 0.6 & 0.6 & 0.6 & 0.6 & 0.7 & 0.8 & 0.5 \\ 0.1 & 0.4 & 0.2 & 0.2 & 0.5 & 0.1 & 0.1 & 0.3 & 0.2 \\ 0.4 & 0.2 & 0.4 & 0.3 & 0.3 & 0.5 & 0.5 & 0.2 & 0.5 \\ 0.1 & 0.2 & 0.1 & 0.5 & 0.1 & 0.1 & 0.1 & 0.3 & 0.4 \\ 0.1 & 0.3 & 0.2 & 0.1 & 0.3 & 0.3 & 0.3 & 0.2 & 0.1 \\ 0.4 & 0.3 & 0.4 & 0.4 & 0.2 & 0.3 & 0.4 & 0.2 & 0.5 \\ 0.5 & 0.3 & 0.4 & 0.6 & 0.7 & 0.6 & 0.7 & 0.7 & 0.6 \\ 0.1 & 0.6 & 0.3 & 0.1 & 0.4 & 0.2 & 0.3 & 0.2 & 0.2 \\ 0.4 & 0.5 & 0.4 & 0.4 & 0.4 & 0.4 & 0.5 & 0.3 & 0.3 \end{bmatrix}$$

Furthermore, the agent achieves a 34% success rate when the adversary randomly samples from its two disturbance policies in its policy set, compared to 42% under a no-perturbation adversary. This suggests that the adversarial strategies produced by PSRO behave as expected for the minimizing player in the two-player game. However, learning did not extend to increasing the performance of the agent. The agent's meta-strategy only included the initial snapshot policy and did not return a better mixed-strategy equilibrium. A hypothesis for this phenomenon could be that the baseline handover policy was chosen using the best policy from a set of successful learned policies. This meant that the baseline policy's performance could be very hard to beat under random re-initialization during each PSRO iteration. The hypothesis is further collaborated by the fact that Wang et al. [2021] trained the GA-DDPG policy using $150,000$ epochs, while we only trained up to $40,000$ epochs. This meant many re-initialized policies may not even learn the handover task, let alone robustify its policy against an adversary.

6.2 Pre-Train

Motivated by this shortcoming, we introduce a pre-training modification to the PSRO framework in which each new agent re-initialization no longer begins from a randomly initialized policy. Instead, every PSRO iteration initializes the agent from a baseline checkpoint that has already learned the nominal handover task under zero disturbances. We train on top of this policy to additionally learn the adversarial perturbation. By avoiding repeated restarts from re-initialization, we expect the agent to preserve the previously acquired task knowledge and allows the agent to focus its learning capacity on adapting to the adversarial perturbations.

This design change had a material impact on the meta-strategies of both players. The policy sets are now much more diverse, distributing probability across four learned policies for both the agent (Figure 6) and the adversary (Figure 7). The adversary's disturbance policies are as follows:

- Disturbance of $[0.00,\ 0.00,\ 0.00]$ with 0.3 probability.
- Disturbance of $[0.43,\ 0.71,\ 0.90]$ with 0.0286 probability.
- Disturbance of $[0.37,\ 1.86,\ 1.89]$ with 0.4143 probability.
- Disturbance of $[0.11,\ 1.66,\ 0.73]$ with 0.2571 probability.

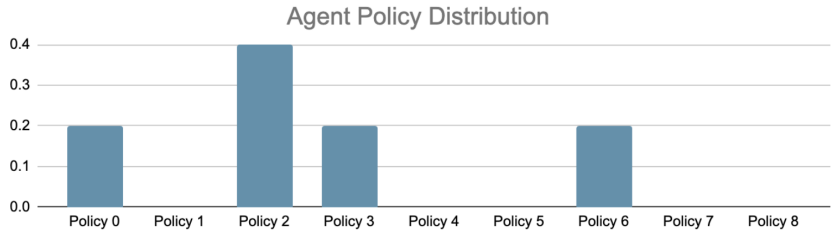

Figure 6: Pre-train agent policy

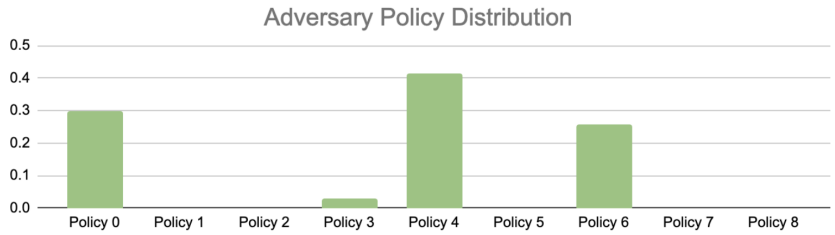

Figure 7: Pre-train adversary policy

This improved interplay between agent and adversary also results in a higher game value of 0.58, indicating that the meta-strategy handover success rate of the agent is higher compared to the baseline. The game's empirical payoff matrix is as follows:

$$\begin{bmatrix} 0.6 & 0.7 & 0.8 & 1.0 & 0.4 & 0.8 & 0.8 & 0.5 & 0.7 \\ 0.4 & 0.3 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.6 & 0.8 \\ 0.5 & 0.4 & 0.5 & 0.4 & 0.7 & 1.0 & 0.5 & 0.6 & 0.8 \\ 0.7 & 0.8 & 0.6 & 0.3 & 0.5 & 0.7 & 0.6 & 0.5 & 0.5 \\ 0.0 & 0.1 & 0.0 & 0.2 & 0.0 & 0.4 & 0.2 & 0.1 & 0.2 \\ 0.4 & 0.6 & 0.6 & 0.5 & 0.4 & 0.4 & 0.7 & 0.5 & 0.4 \\ 0.6 & 0.7 & 0.9 & 0.8 & 0.6 & 0.8 & 0.5 & 0.7 & 0.6 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 \\ 0.1 & 0.4 & 0.3 & 0.4 & 0.1 & 0.4 & 0.3 & 0.4 & 0.2 \end{bmatrix}$$

The diversified meta-strategy results in a much better agent performance compared to the baseline. Under no adversarial disturbances, the meta-strategy agent achieves a 61% handover success rate,

outperforming the 42% success rate of the baseline policy. Under adversarial sampling from the four learned disturbance policies, the meta-strategy still maintains a 58% success rate, double the baseline's 29%. This indicates that pre-training the agent policy instead of re-initializing from a uniform random policy does indeed produce improved robust handover policies under adversarial disturbances.

However, it is interesting to note that the no-disturbance policy $[0.00,\ 0.00,\ 0.00]$ remains in the adversary's policy set and is being played with 0.3 probability. The no-disturbance policy was originally removed from the adversary's meta-strategy in PSRO iteration 3, but was reintroduced in iteration 6. A possibility for this could be that the no-disturbance policy is being played to counteract the agent's policy 6. Referring to the seventh row of the game's empirical payoff matrix (corresponding to the agent's policy 6), the adversary's policy 0 and policy 6 result in a comparatively lower success rate for the agent. Incorporating the no-disturbance policy may be a result of the adversary's counter-action against the agent playing policy 6 with positive probability.

7 Future Work

Several directions for extending this work could be taken beyond the scope of this project. The first action to take is to deploy the learned agent meta-strategy on a real Franka Emika Panda manipulator and report on its performance metrics. We could also incorporate temporally coupled or human-like perturbations such as pushing, yanking, shaking, or dropping to mimic more realistic adversarial behavior. With more computational resources, we could also expand the training to more objects and scenes to test generalization beyond the cereal box task. To align the methodology closer to theory, we could add a dynamic stopping criterion for PSRO so that the algorithm terminates whenever the new best-responses are close to existing policies in the policy set. Finally, we could explore the use of imitation learning or behavior cloning to aggregate the set of meta-strategy policies into a single policy for deployment.

8 Conclusion

In this project, we formulate the manipulator-environment interaction as a two-player zero-sum game and employ the use of Policy Space Response Oracles (PSRO) (Lanctot et al. [2017]) to train robust handover policies under adversarial perturbations. The baseline PSRO implementation, which reinitialized the agent from scratch at every iteration, failed to improve agent performance due to the difficulty of surpassing a strong baseline handover policy within only 40,000 agent training epochs.

To address this, we introduced a pre-training modification in which each agent best-response is initialized from a checkpoint policy that has already learned the nominal handover task. This change allowed the agent and adversary to diversify their meta-strategies and improved empirical success rates for the agent. The resulting agent meta-strategy achieved a 61% success rate compared to 42% against a no disturbance adversary. Sampling the perturbations from the adversary's meta-strategy, the agent again achieved a 58% success rate compared to the baseline's 29%.

Overall, the use of PSRO allowed the manipulator agent to more robustly adapt to adversarial perturbations. The learned meta-strategy of the adversary also reveals which disturbances are most harmful to handover stability. We hope that this project motivates continued development of robust handover policies and inspires future work toward safer and more robust human-robot interactions using game-theoretic techniques.

References

Michael A. Goodrich and Alan C. Schultz. Human-robot interaction: a survey. *Found. Trends Hum.-Comput. Interact.*, 1(3):203–275, January 2007. ISSN 1551-3955. doi: 10.1561/1100000005. URL https://doi.org/10.1561/1100000005.

Abhinav Dahiya, Alexander M. Aroyo, Kerstin Dautenhahn, and Stephen L. Smith. A survey of multi-agent human-robot interaction systems, 2022. URL https://arxiv.org/abs/2212.05286.

Alap Kshirsagar, Guy Hoffman, and Armin Biess. Evaluating guided policy search for human-robot handovers. *IEEE Robotics and Automation Letters*, 6(2):3933–3940, 2021. doi: 10.1109/LRA.2021.3067299.

Haonan Duan, Yifan Yang, Daheng Li, and Peng Wang. Human–robot object handover: Recent progress and future direction. *Biomimetic Intelligence and Robotics*, 4(1):100145, 2024. ISSN 2667-3797. doi: https://doi.org/10.1016/j.birob.2024.100145. URL https://www.sciencedirect.com/science/article/pii/S2667379724000032.

Marc Lanctot, Vinícius Flores Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:9072400.

H. McMahan, Geoffrey Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. pages 536–543, 01 2003.

Franka Emika GmbH. *Watchman: Operating Instructions*. Munich, Germany, release version 1.0 edition, October 2022. Document No. 231010.

Tuomas Oikarinen, Wang Zhang, Alexandre Megretski, Luca Daniel, and Tsui-Wei Weng. Robust deep reinforcement learning through adversarial loss, 2021. URL https://arxiv.org/abs/2008.01976.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21024–21037. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f0eb6568ea114ba6e293f903c34d7488-Paper.pdf.

Qianli Shen, Yan Li, Haoming Jiang, Zhaoran Wang, and Tuo Zhao. Deep reinforcement learning with robust and smooth policy. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8707–8718. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/shen20b.html.

Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. Robust reinforcement learning using adversarial populations, 2020. URL https://arxiv.org/abs/2008.01825.

Tim Franzmeyer, Stephen McAleer, João F. Henriques, Jakob N. Foerster, Philip H. S. Torr, Adel Bibi, and Christian Schroeder de Witt. Illusory attacks: Information-theoretic detectability matters in adversarial attacks, 2024. URL https://arxiv.org/abs/2207.10170.

Huan Zhang, Hongge Chen, Duane S. Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. *ArXiv*, abs/2101.08452, 2021. URL https://api.semanticscholar.org/CorpusID:231662383.

Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. Who is the strongest enemy? towards optimal and efficient evasion attacks in deep rl, 2023. URL https://arxiv.org/abs/2106.05087.

Pingcheng Jian, Chao Yang, Di Guo, Huaping Liu, and Fuchun Sun. Adversarial skill learning for robust manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2555–2561, 2021. doi: 10.1109/ICRA48506.2021.9561379.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, and Furong Huang. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=y-E1htoQl-n.

Martin Zinkevich, Michael Bowling, Neil Burch, and Michael Johanson. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*, pages 1601–1608, 2007.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Regret circuits: Composability of regret minimizers, 2019. URL https://arxiv.org/abs/1811.02540.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:53183381.

Eric Steinberger, Adam Lerer, and Noam Brown. Dream: Deep regret minimization with advantage baselines and model-free learning, 2020. URL https://arxiv.org/abs/2006.10410.

Stephen Marcus McAleer, Gabriele Farina, Marc Lanctot, and Tuomas Sandholm. ESCHER: Eschewing importance sampling in games by computing a history value function to estimate regret. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=35QyoZv8cKO.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, Xiangyu Liu, Benjamin Eysenbach, Tuomas Sandholm, Furong Huang, and Stephen Marcus McAleer. Game-theoretic robust reinforcement learning handles temporally-coupled perturbations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wZWTHU7AsQ.

Yu-Wei Chao, Chris Paxton, Yu Xiang, Wei Yang, Balakumar Sundaralingam, Tao Chen, Adithyavairavan Murali, Maya Cakmak, and Dieter Fox. HandoverSim: A simulation framework and benchmark for human-to-robot object handovers. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019. URL https://arxiv.org/abs/1509.02971.

Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=jOSWHddP1fZ.

Firas Al-Hafez. Finding the optimal learning rate using bayesian optimization, May 12 2021. URL https://firasalhafez.com/2021/05/12/finding-the-optimal-learning-rate-using-bayesian-optimization/.