Work in Progress: Integrating Human Preference and Human Feedback for Environmentally Adaptable Robotic Learning

Yuxuan Li
University of Alberta
Edmonton, Canada
yuxuan.li@ualberta.ca

Qinglin Liu
University of Science and Technology of China
Hefei, China
qlliu@mail.ustc.edu.cn

Nan Lin
Fuxi Robotics in NetEase
Hangzhou, China
linnan04@corp.netease.com

Matthew E. Taylor
University of Alberta
Edmonton, Canada
matthew.e.taylor@ualberta.ca

ABSTRACT

Researchers have witnessed successful applications of reinforcement learning in robotics, and even better performance is achieved if expert demonstration is introduced. However, it is considered difficult to acquire expert demonstration. On the contrary, human preference and human feedback are easy to collect but challenging to exploit, especially for high-dimensional tasks. Applying binary signals like human feedback to such control tasks remains challenging. In this paper, we explore human feedback as a more simple form of human knowledge and exploit its ability to shape robotic behaviours. Additionally, human preference is introduced for system identification in a complicated manipulation task for improved parameter estimation. Alongside a hierarchical structure that comprises force control methods, we implement a brand-new control framework that can adapt to different environments and easily integrate non-expert human knowledge. Experiments in simulated environments are conducted to further verify its performance.

KEYWORDS

Reinforcement Learning, Robotics, Human In Loop Learning, Compliant Control, Imitation Learning, System Identification

1 INTRODUCTION

Reinforcement learning, as an emerging new method with promising performance in many control problems, can be generally summarised as the optimisation process of a control policy to maximise the accumulated rewards defined in a Markov Decision Process (MDP). Many variations of RL exist for different scenarios. Examples are POMDP [21] and Hierarchical RL [16].

Involving human knowledge, as one direction in reinforcement learning, has recently drawn researcher's attention recently. Compared with traditional reinforcement learning, it is challenging in several cases, such as no explicit reward function, huge observation space, low sample efficiency, safety, etc.. Taking expert demonstration, for instance, learning from demonstration methods [2, 24] is proven to boost learning efficiency. In addition to exploiting the full expert demonstration, researchers also proposed methods using human feedback [11]. Instead of providing expert actions, numerical values representing human assessment towards certain agent behaviours are collected from even non-expert participants. Similar cases of applying human preferences in the learning process [8] also show interesting results. However, acquiring a simpler form of human knowledge is a double-edged sword. Being simpler and easier means it contains less information to guide the agent policy, and such methods are challenged by large observation space, like robotic control tasks. Rare cases have been explored in this direction. One possible approach to reduce the observation space and the search space for policies is to design hierarchical control methods. Hierarchical RL [13, 16] focuses on the idea of dividing and conquering. By establishing a layer-structured policy, where each part is designed to solve a sub-task, the original task is solved by each sub-policy combined. Similar idea can be found in Options framework [26] and Option Critic [3] where options as sub-polices are learned with TD learning.

Apart from above learning-based control methods, another important method in robots is force control. Force control can effectively improve the stability and safety of this situation. Passive control and active control are two crucial categories of force control. For example, in passive control, we embed spring-actuated joints into robots to exploit its inherent elasticity, therefore to minimise contact forces. Unlike passive dynamic systems, applying appropriate forces on the robot joint using active control is critical for robots to perform tasks in complex environments. Proportional-derivative (PD) control is a common approach to compute control forces to track the kinematic state of a joint trajectory, but it aims to achieve a precise position. Compliance control is usually favoured for its ability to achieve a dynamic relationship between the manipulator and the environment based on contact forces [18]. For instance, as two popular active force control methods, impedance control and admittance control ensure the task being undertaken while dynamically compensating the position according to operational space or configuration space. In impedance control, the controller is a mechanical impedance, and consequently, the controlled plant is treated as an admittance, while in admittance control, the plant is position-controlled and behaves as a mechanical impedance. The parameters of these force control methods can be dynamically adjusted, like changing the overall stiffness, adapting the arm to an external environment, and adjusting the gain of the position control system for compliance [1, 25]. Our framework uses admittance control for compliance to complete relatively complex tasks and ensure the safety of the environment and the robot.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2023), Cruz, Hayes, Wang, Yates (eds.), May 9-10, 2023, Online, https://ala2023.github.io/. 2023.

In this paper, we follow a similar idea of hierarchical control methods by adding an abstract layer of the robotic control tasks. This allows us to use human feedback to influence the robotic behaviour indirectly by planning goals in the abstract layer controller, and therefore to reduce the amount of data required. Furthermore, for the lower level control, a goal-conditioned and parameter-conditioned policy is applied, combined with admittance control to reach environment adaptability. Lastly, a system identification model is added for parameter estimation which also inputs human preferences for better precision.

To summarise, our contributions are as follows:

(1) We propose a simple yet effective hierarchical control framework allowing human feedback and human preference to shape rather complicated, high-dimensional control policy behaviours.
(2) We propose a predefined high-level MDP as an abstract layer to exploit feedback provided by non-expert humans in robotic tasks, thereby simplifying the problem and strengthening the utility of human feedback.
(3) We propose a new loss function that allows learning-based system identification models to take human preferences for further fine-tuning and show human participation can improve performance with experiments in simulated environments.

2 RELATED WORK

In this paper, integration of human feedback and human preference into the control loop is the main topic. Hence, human knowledge in learning process and hierarchical control are the two important fields where previous related work can be found. Furthermore, we will also look into parameter estimation as it can improve control.

2.1 Human knowledge in shaping behaviours

Having prior knowledge provided by an expert is more effective and efficient than searching for a solution from scratch. Expert human knowledge has been exploited in multiple ways to boost learning speed, like behaviour cloning [22], teacher-student learning framework [5, 32], and inverse reinforcement learning [4]. However, these methods are challenged in certain scenarios, such as the fact that expert knowledge is difficult to acquire. Similar situations may be encountered due to equipment limitations in precision, high cost, lack of accurate knowledge of the environment, etc.

To gather human knowledge more easily, using human feedback has been proposed. For example, Know et al. proposed TAMER [11], where simple scalar human feedback signals are treated as an estimation of reward, hence a reward model is established. Macglashan et al. choose to view human feedback differently, as an estimation of the advantage value [17], where COACH is proposed. These methods are tested in several environments to show satisfactory performance and later works even push the boundary to a new level by going deep in neural networks, like Deep TAMER [29] and Deep COACH [2], where image-based input is used. Other forms of feedback are also explored. Comparison-based human preference serves as an interesting example. Lee et al. [14] show video clips to human participants to collect human preferences and thus succeed in several robotic manipulation and control tasks.

However, feedback-based methods are sometimes confronted with the fact that they require a huge amount of human feedback, if no further information is provided, and even fail in a very complicated environment, such as high-dimensional robotic manipulation tasks. Hence, little work has been done in applying human feedback to shape robotics behaviours in manipulation tasks, especially in an unstructured environment with dynamic multi-contacts. One close example can be seen in a case study [12], where Knox et al. applied the TAMER framework in robot navigation with pretrained policies.

2.2 Parameter estimation

System identification is a procedure where we manage to establish an understanding of the environmental dynamics model based on observed data. Thus, applying system identification could contribute to control methods that require a dynamics model. One of the special cases is that we want to estimate unknown parameters of dynamics. Data driven methods are also established for reinforcement learning related scenarios. Chebotar et al. proposed SimOpt framework [7] to bridge the difference between the simulator and the real world through Relative Entropy Policy Search [20] based parameter searching. Similar practice could be found in Grounded Simulation Learning [9], where Evolutionary Strategies is applied in dynamics model parameter estimation and a humanoid robot learnt to walk. Zhu et al. innovated [31] the parameter search process by introducing Greedy Entropy Search to reduce the search space. Yu et al. proposed online system identification [30] by pre-training a system identification model and verified its ability against domain randomisation. Yet many state-of-the-art methods are proposed, most methods view the problem as pure dynamics fitting or parameter searching, which is computationally costly. Involving human knowledge into parameter estimation is a new direction that might be promising in boosting performance.

2.3 Hierarchical Force Control

Performing interactive tasks while maintaining a range of manipulator contact forces in complex environments is challenging but necessary. To achieve the above relationship is to design a controller like a spring and a damper act as elastic actuators [19]. Maintaining contact force also needs to adapt to uncertain external environmental parameters. A vision servo-based adaptive controller for motion and force tracking [6] and combining fuzzy logic with traditional sliding mode control [23] are proposed to deal with this problem. Hierarchical control methods are widely used to effectively reduce the search space and complexity of tasks and improve adaptability in an unstructured environment. For example, Jiang et al. proposed a hierarchical control system for soft arms based on the Jacobian model and Q-learning. Using inherent passive compliance of the arm could allow it to perform tasks like a human [10]. Lee et al. proposed a two-level control architecture based on deep reinforcement learning to minimise the interaction forces and the control torques for imitation [15].

3 PRELIMINARIES

In this paper, we consider the control problem as a traditional Markov Decision Process (MDP), which is denoted by a quintuple

denoted by $M = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma\}$, where $\mathcal{S}$ denotes the agent's state space, $\mathcal{A}$ is the agent's action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the environmental dynamics transition probability, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the function that gives an immediate reward, γ is a discount factor. Furthermore, a goal-conditioned version of MDP is defined as a sextuple $M^g = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma, \mathcal{G}\}$, where $\mathcal{G}$ is the goal space and other notations remain the same as MDP. Alongside a MDP and a goal-conditioned MDP, we consider a problem that comes with an inherent hierarchical structure. Specifically, it could be treated as two problems. One is the abstract problem $M_{abstract}$ and the other is the actual problem M^g_{actual}. Expectedly, an abstract problem is a high-level abstraction of the task that avoids dealing with too many details and shrinks the state space. By solving the abstract problem, we shall find a high-level planning route specified by g, which shall be used as input for the policy of the actual problem (e.g. a specific manipulation task), hence $\mathcal{A}_{abstract} = \mathcal{G}_{actual}$.

In accordance to a problem with hierarchical structure, we consider a hierarchical control policy π_H, which maps from current state s_t to an action a_t. The hierarchical control policy consists of a $\pi^{abstract}$ and a π^{actual}, which represents a high-level policy for the abstract problem and a low-level policy for the actual problem. The high-level policy $\pi^{abstract} : S_{abstract} \rightarrow \mathcal{G}_{actual}$, maps the observation of the abstract problem to a goal. Thus the low level policy is $\pi^{abstract} : \mathcal{S}_{actual} \times \mathcal{G} \rightarrow \mathcal{A}_{actual}$.

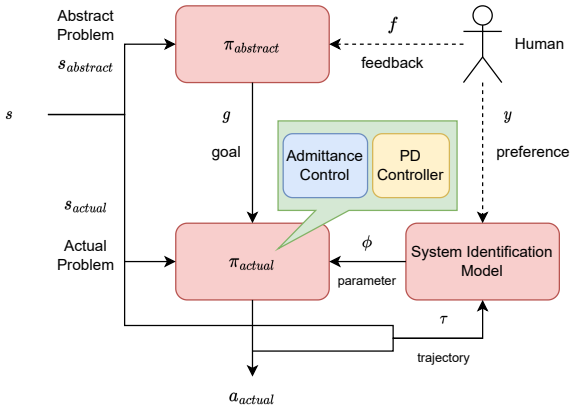


Figure 1: The diagram illustrates the overview of the proposed framework integrating human feedback (denoted by f) for the abstract problem and human preference (denoted by y) for system identification model.

4 PRELIMINARIES AND METHODOLOGY

4.1 Applying human feedback on an abstract problem

Human feedback is usually considered as vague scalar values that describe human's judgement of the agent's behaviour. Similar to COACH [29] and TAMER [11], human feedback is defined as $f \in \{-1, 0, 1\}$, where -1 shows human discourages certain behaviour, 0 means human is indifferent and 1 suggests human encourages the behaviour. Rare cases of its application in a complicated high-dimensional manipulation task have been researched, given the fact that it may require significantly large amounts of feedback. In this work, we explore the potential of human feedback in shaping behaviour in a high-dimensional task by applying human feedback to an abstract problem, hence reducing the policy search space. Similar to COACH, we interpret human feedback as an estimation of advantage and update the policy by:

$$\mathcal{L} = \alpha \nabla \log(\pi_{abstract}(a_t|s_t)) \cdot f_t, \tag{1}$$

where f_t is human feedback, π is a differentiable policy and α is the learning rate.

4.2 Human Involved Parameter Estimation

For the actual problem, we assume that there are unknown parameters ϕ that describe the environmental dynamics, which could influence the performance of low level policy π^{actual}_{ϕ}. Furthermore, estimating parameters will also boost its performance when transferred to a new environment. In order to estimate ϕ, we pre-train a system identification model $SI : \tau \rightarrow Dist(\phi)$, which maps from a trajectory slice $\tau : (s_{t-h}, a_{t-h}, s_{t-h+1}, a_{t-h+1} ... s_{t-1}, a_{t-1})$ of length h, to the distribution of ϕ. The pre-training is practiced on a dataset $\mathcal{D} = \{(\tau_i, \phi_i)|i = 1, ..., N\}$ following the Alg. 1 and a mean square loss.

Algorithm 1: Train System Identification Model

Data: System identification model SI parameterised by ξ, trajectory dataset $\mathcal{D}$, learning rate α, trajectory slice length h, batch size l

```
// Training
for i ← 0, 1, 2, ... do
    Sample batch {(τ_j, φ_j)|j = 1, ..., l} from D
    Update system identification model SI by MSE Loss
end
```

With a pre-trained system identification model SI, during policy execution, we may further involve human participant into this process and ask the human participant its preference $y_{\phi_0,\phi_1}(\phi) \in \{-1, 1\}$ over $\pi^{actual}_{\phi_0}$ and $\pi^{actual}_{\phi_1}$. After that, based on the human preference, SI is updated by human preference loss as Eqn. 2

$$\begin{aligned}\mathcal{L}_{\mathcal{HP}} = -[&log(Dist_{\phi}(\phi_1))(Dist_{\phi}(\phi_1) + p \cdot y_{\phi_1,\phi_2}(\phi_1)) + \\ &log(Dist_{\phi}(\phi_2))(Dist_{\phi}(\phi_2) + \cdot y_{\phi_1,\phi_2}(\phi_2)))],\end{aligned} \tag{2}$$

where p is the encourage percentage controlling the confidence in the human preference. Inherently, this is a weighted cross entropy loss, where human preferred parameter candidate will be encouraged and the other ones are discouraged. If we take one step further to ask a human participant to give it preference over N video clips, i.e., $y_{\phi_0,\phi_1,...\phi_N}(\phi) \in \{-\frac{1}{N-1}, 1\}$, the loss could be accordingly rewritten as Eqn. 3:

$$\mathcal{L}_{\mathcal{HP}} = -\Sigma_{i=1}^{N} log(Dist_{\phi}(\phi_i))(\Sigma_{i=1}^{N-1} Dist_{\phi}(\phi_i) + p \cdot y_{\phi_0,...,\phi_{N-1}}(\phi_i)) \tag{3}$$

Notably, human preference comes with inevitable error and sometimes we don't need to estimate the parameter to full precision. Therefore, estimating parameters by discretisation is also an

Algorithm 2: Fine-tune System Identification Model by Human Preference

Data: System identification model SI parameterised by ξ, learning rate α, encourage percentage p, trajectory slice length h, candidate parameter set Φ

```
// Finetune
for i ← 0, 1, 2, ... do
    Execute policy with φ_0, φ_1 sampled from Φ
    Sample trajectory slice τ
    Asking for human preference y
    Update system identification model SI by Eqn. 2 or Eqn. 3
end
```

attractive option. In that case, we need to have a finite candidate parameter set $\Phi = \{\phi_0, \phi_1, ..., \phi_N\}$ and the System Identification Model is accordingly set to predict a discrete distribution.

4.3 Low Level Controller Design

For our low level controller, we implemented a force control based controller and later extended it to a learning-based version. While the robotic arm is an open-chain articulation with the fixed root, its configuration can be expressed by its joint positions $q \in \mathbb{R}^n$, joint velocities $\dot{q} \in \mathbb{R}^n$ and joint accelerations $\ddot{q} \in \mathbb{R}^n$ in generalized coordinates. The equations of motion describe the dynamic system as follows:

$$M(q)\ddot{q} + c(q, \dot{q}) = \tau + \tau_{ext} \tag{4}$$

where $M(q)$ is the mass matrix and $c(q, \dot{q})$ is Coriolis and gravitational forces. τ_{ext} is the sum of external forces that includes contact force and other external perturbations. To deduce the PD controller, the control force τ is calculated using a stable proportional derivative (SPD) formulation [27]:

$$\tau = -k_p(q^n + \dot{q}\Delta t - \bar{q}^{n+1}) - k_d(\dot{q}^n + \ddot{q}^n\Delta t) \tag{5}$$

where both k_p and k_d are diagonal matrices that indicate the gains and damping coefficients. SPD computes the control forces using the next time step state q^{n+1}, which can be expanded as $q^n + \Delta t\dot{q}^n$ via Taylor series, so as $\dot{q}^n$. The acceleration can be written as:

$$\ddot{q}^n = (M + k_d\Delta t)^{-1}(-c - k_p(q^n + \dot{q}^n\Delta t - \bar{q}^{n+1}) - k_d\dot{q}^n + \tau_{ext}) \tag{6}$$

Then use the explicit Euler method to integrate to the next time step.

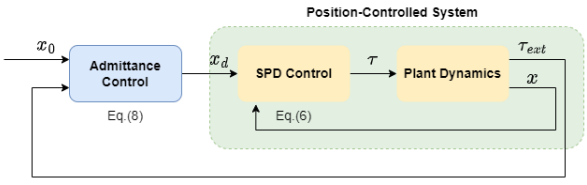


Figure 2: Our admittance control based implementation.

The mathematical expression of a single degree-of-freedom system in which a mass interacts with an environment is defined as $m\ddot{x} = \tau + \tau_{ext}$, where m and x are the inertia and displacement of the mass, respectively. The Admittance Control is to design the control force τ that will establish a given relationship between τ_{ext} and a desired trajectory x_0 and a desired position x_d. Typically, a linear second-order relationship of the form

$$M_d(\ddot{x}_d - \ddot{x}_0) + D_d(\dot{x}_d - \dot{x}_0) + K_d(x_d - x_0) = \tau_{ext} \tag{7}$$

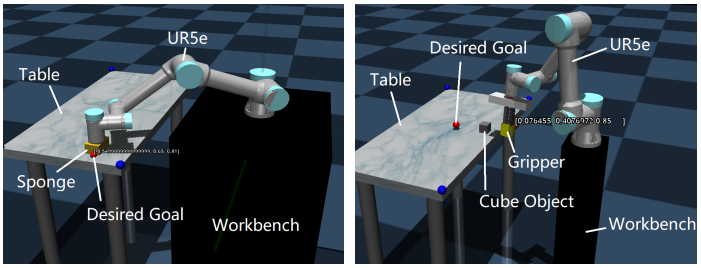
is considered, where the positive constants M_d, D_d, and K_d represent the desired inertia, damping, and stiffness,respectively. In Admittance Control, the plant is position-controlled and can be implemented using the SPD controller (5) mentioned above as shown in Fig.2. Furthermore, for better future adaptation with fine-tuning, instead of using fixed hand-coded force control methods, we practice imitation learning to learn a goal-conditioned and parameter-conditioned neural network based policy $\pi^{actual}(\bar{q}|s, \phi)$, together with admittance control to ensure force within a safe range, following Alg. 3.

Algorithm 3: Learning $\pi^{abstract}$

Data: Position controlled system P, Goal space $\mathbb{G}$, Environment candidate parameter set $\lesssim$, differentiable $\pi^{abstract}$, replay buffer R, batch size N

```
// Imitation Learning
for i ← 0, 1, 2, ... do
    Randomly sample goal g and environmental parameter φ
    Set simulator with environmental parameter φ
    Collect trajectory τ using P under goal g
    Pushing τ into R
    Update π^abstract by MSE loss, with N-sized batch sampled from R
end
```

5 EXPERIMENTS



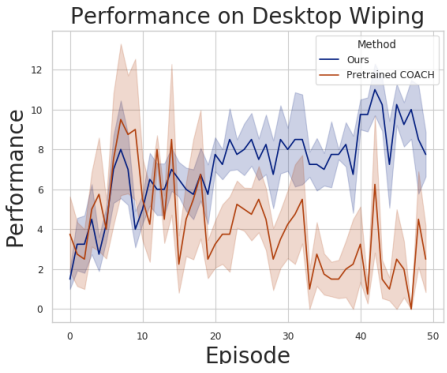
(a) (b)

Figure 3: (a) Desk wiping environment. The robot arm is trying to move the cleaning sponge to the desired next goal(red dot). (b) Pushing environment. The robot arm with a gripper is trying to push the cube object to the desired goal(red dot).

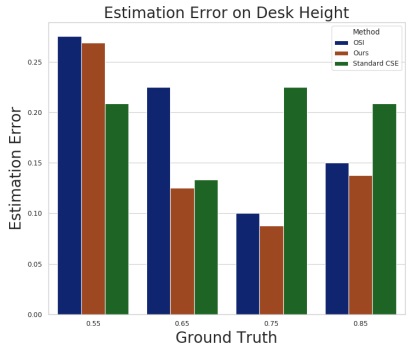
5.1 Experiment Setting

Our experiments are conducted in the Mujoco [28] simulator, featuring a UR5e robot arm with a piece of cleaning sponge attached

as the end effector, shown in Fig 3(a). The task is to control the robot to clean the desktop within a desired range of contact force and distance to the desktop, i.e., while we want to keep contact between the sponge and the desktop, we also want the robot to apply a proper force to the sponge to wipe as human beings, within safety range, thus a proper estimation of desktop height is required. The second task, shown in Fig. 3(b), is aimed to push the object to the desired position. The initial position and the target position of the object are randomly set within the reach of the robot arm on the table. We want to keep the object as close to the target location as possible. The threshold is set to 20mm, which is less than 50mm in size. The main parameter we want to estimate is the object mass, to ensure we could tweak the force controller and successfully move the object.



(a)



(b)

Figure 4: (a) High level policy performance over episodes comparison with Pretrained COACH. (b) Parameter estimation error of Online System Identification, our method, and our method but with standard cross entropy loss with different ground truth desk heights.

5.2 High Level Planner Evaluation

In this subsection, we intend to show a prototype combining feedback to solve abstract problems as to report its validness of migrating it for goal planning. In the desktop wiping setting, the abstract problem is defined as a grid world, where we want to control the agent to move and traverse all the grids,i.e., successfully wipe the whole desktop. Accordingly, the actual problem would be how to control the robot arm to a desired position, within a safe range of force applied to the sponge. Noticeably, there exists a certain linear coordinate transformation between the grid world to the actual world coordinate system on the desktop. Furthermore, the grid world may not strictly align with the actual world, especially when the π_{actual} fails to reach the goal. In that case, a reset of the grid world state is required.

The performance over different episodes of an agent learning to traverse the grid world guided by human feedback is shown in Fig. 4 (a). While directly applying feedback to a policy that's pretrained with limited wiping demonstration trajectories (Pretrained COACH) is shown that human feedback cannot properly guide the policy, our methods established a pretrained goal-conditioned low level policy and thus successfully learnt to traverse the grid on the abstract problem and the policy $\pi_{abstract}$ can successfully plan trajectory points and finish the task. Furthermore, without a proper hierarchical structure, the Pretrained COACH even shows worse performance overtime, suggesting that binary human feedback directly applied to the policy can be even misleading.

5.3 Parameter Estimation Evaluation

In this subsection, we estimate the effectiveness of integrating human preferences into parameter estimation. Serving as a baseline, we compare our algorithm to Online System Identification [30], where desktop height in the wiping task and object mass in the pushing task are to be estimated. Both models consist of four fully connected hidden layers with size of 256, 128, 64, 32, with hyperbolic tangent as its activation function. The input history length is set to 3 and the encourage percentage is 0.25. As illustrated in Fig. 1, the human participant will be asked to watch two video clips, each illustrating π_{actual} parameterised by different candidate paramtres and then be asked to provide their preference based on the performance. It could be seen from Fig. 4 (b) that the system identification gives a slightly more precise prediction after taking human preferences.

Furthermore, due to the limitation of human reaction speed, collecting real feedback is rather slow, and we also practised experiments using a faked human (programme-simulated) with different levels of wrong preferences, from perfectly correct, which means the percentage of wrong preference is 0, to perfectly wrong, giving all possible wrong preferences. The experimental results are shown in Fig. 5. The x-axis of the figure indicates how many steps of preferences judgment have passed, and the y-axis indicates the error between the estimated parameters and the real environment. For both environments, it is not only observed that preference input helps to reduce the prediction error over time; Also, with perfect preferences, the prediction error can be generally reduced over tens of steps and with perfectly wrong preferences, and vice versa.

5.4 Ablation Study

In this section, we practice ablation study to verify the effectiveness of our algorithm design. Recall Eqn. 2, instead of using Cross Entropy Loss like Christian et al. [8], we exploit human preferences by a soft cross entropy loss. This is due to the fact that preferences

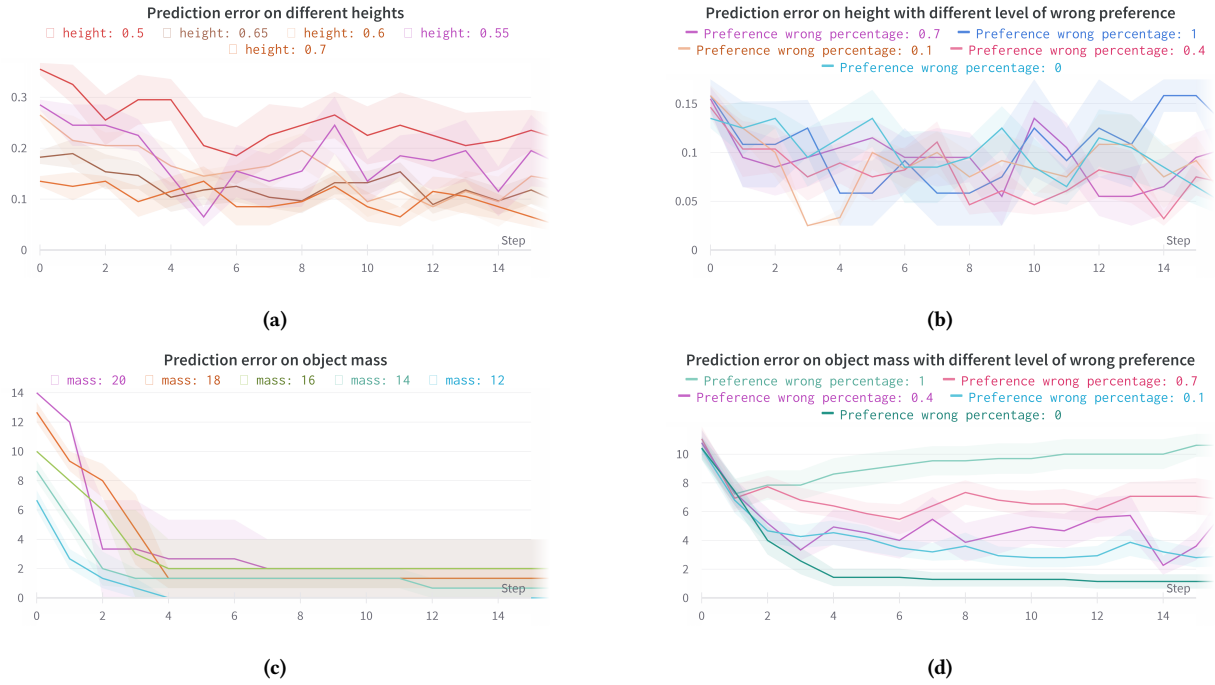


Figure 5: (a)(c) Prediction Error over time with perfect faked human preference on desk height/object mass and we can see the preference helps to reduce the prediction error over time. (b)(d) Prediction error on desk height/object mass, with different levels of wrong preference. It is observed the better preference provided, the better prediction precision achieved.

are given to sampled parameter candidates while there are still other possible parameter candidates. The soft cross entropy loss is adopted to illustrate the idea that a preferred parameter candidate may not be a "positive class" or the right prediction but only a signal to shape its distribution. The results of the ablation study using standard cross entropy loss are illustrated in Fig. 4(b). It is shown that generally, the standard CSE would gives worse results in parameter estimation.

We also conducted experiments to illustrate the benefits of using admittance control. We use two different low-level controllers including the SPD Controller and Admittance to manipulate the robotic arm directly. We let the robotic arm start from the same initial position and execute the same random trajectory on the table. At the same time, the data of the torque sensor installed at the end are collected. In the Mujoco simulation environment, the height of the table surface is 0.81m. We try to set the desired height of the trajectory to 0.72m, 0.74m, 0.76m, 0.78m and 0.8m. The mean and standard deviation of the z-axis force during robotic arm execution has been shown in Fig.6. We can see that the force exerted by the end effector on the table can be very large, which may cause damage to the robotic arm or the table in the real world, admittance control is necessary.

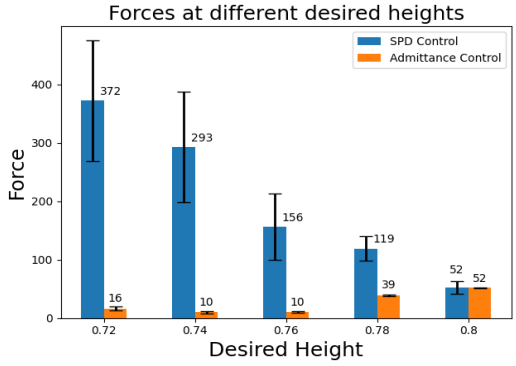


Figure 6: Admittance control gives smaller contact forces compared to SPD controller.

6 CONCLUSION AND FUTURE WORK

In this paper, we are challenged to learn from simple human feedback for complicated high-dimensional tasks, and integrate human preference into system identification for better control. With initial results showing the potential of involving human participants into the control process, the problem remains open to be explored as much previous work mainly focusing on full expert demonstration. As one of the possible future directions, we may further improve the framework by symbolic planning to the abstract policy to give more environmental adaptability. Lastly, experiments in real-world robotics and larger-scale user study might further reveal the potential of this work.

REFERENCES

[1] Carlos Aguilar-Ibanez, Javier Moreno-Valenzuela, Octavio García-Alarcón, Mizraim Martinez-Lopez, José Ángel Acosta, and Miguel S Suarez-Castanon. 2021. PI-type controllers and Σ−Δ modulation for saturated DC-DC buck power converters. *IEEE Access* 9 (2021), 20346–20357.

[2] Dilip Arumugam, Jun Ki Lee, Sophie Saskin, and Michael L Littman. 2019. Deep reinforcement learning from policy-dependent human feedback. *arXiv preprint arXiv:1902.04257* (2019).

[3] Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[4] Abdeslam Boularias, Jens Kober, and Jan Peters. 2011. Relative entropy inverse reinforcement learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 182–189.

[5] Andres Campero, Roberta Raileanu, Heinrich Küttler, Joshua B Tenenbaum, Tim Rocktäschel, and Edward Grefenstette. 2020. Learning with amigo: Adversarially motivated intrinsic goals. *arXiv preprint arXiv:2006.12122* (2020).

[6] Chien Chern Cheah, Saing Paul Hou, Yu Zhao, and Jean-Jacques E Slotine. 2009. Adaptive vision and force tracking control for robots with constraint uncertainty. *IEEE/ASME Transactions on Mechatronics* 15, 3 (2009), 389–399.

[7] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. 2019. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 8973–8979.

[8] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).

[9] Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone. 2013. Humanoid robots learning to walk faster: From the real world to simulation and back. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. 39–46.

[10] Hao Jiang, Zhanchi Wang, Yusong Jin, Xiaotong Chen, Peijin Li, Yinghao Gan, Sen Lin, and Xiaoping Chen. 2021. Hierarchical control of soft manipulators towards unstructured interactions. *The International Journal of Robotics Research* 40, 1 (2021), 411–434.

[11] W Bradley Knox and Peter Stone. 2009. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the fifth international conference on Knowledge capture*. 9–16.

[12] W Bradley Knox, Peter Stone, and Cynthia Breazeal. 2013. Training a robot via human feedback: A case study. In *International Conference on Social Robotics*. Springer, 460–470.

[13] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems* 29 (2016).

[14] Kimin Lee, Laura Smith, and Pieter Abbeel. 2021. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091* (2021).

[15] Seunghwan Lee, Phil Sik Chang, and Jehee Lee. 2022. Deep Compliant Control. *Trans. of ASME Journal of Dynamic System, Measurement, and Control* (2022).

[16] Nan Lin, Yuxuan Li, Keke Tang, Yujun Zhu, Xiayu Zhang, Ruolin Wang, Jianmin Ji, Xiaoping Chen, and Xinming Zhang. 2022. Manipulation Planning From Demonstration Via Goal-Conditioned Prior Action Primitive Decomposition and Alignment. *IEEE Robotics and Automation Letters* 7, 2 (2022), 1387–1394.

[17] James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. 2017. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning*. PMLR, 2285–2294.

[18] Hogan Neville. 1985. Impedance Control: An Approach to Manipulation: Part Ī III. *Trans. of ASME Journal of Dynamic System, Measurement, and Control* 107 (1985), 1.

[19] Christian Ott, Ranjan Mukherjee, and Yoshihiko Nakamura. 2010. Unified impedance and admittance control. In *2010 IEEE international conference on robotics and automation*. IEEE, 554–561.

[20] Jan Peters, Katharina Mulling, and Yasemin Altun. 2010. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

[21] Joelle Pineau and Geoffrey J Gordon. 2007. POMDP planning for robust robot control. In *Robotics Research*. Springer, 69–82.

[22] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. 2017. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087* (2017).

[23] Abbas Karamali Ravandi, Esmaeel Khanmirza, and Kamran Daneshjou. 2018. Hybrid force/position control of robotic arms manipulating in uncertain environments based on adaptive fuzzy sliding mode control. *Applied Soft Computing* 70 (2018), 864–874.

[24] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 627–635.

[25] Ramón Silva-Ortigoza, Eduardo Hernández-Márquez, Alfredo Roldán-Caballero, Salvador Tavera-Mosqueda, Magdalena Marciano-Melchor, Jose Rafael Garcia-Sanchez, Victor Manuel Hernández-Guzmán, and Gilberto Silva-Ortigoza. 2021. Sensorless Tracking Control for a "Full-Bridge Buck Inverter–DC Motor" System: Passivity and Flatness-Based Design. *IEEE Access* 9 (2021), 132191–132204.

[26] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

[27] Jie Tan, Karen Liu, and Greg Turk. 2011. Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications* 31, 4 (2011), 34–44.

[28] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033. https://doi.org/10.1109/IROS.2012.6386109

[29] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. 2018. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[30] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. 2017. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453* (2017).

[31] Shaojun Zhu, Andrew Kimmel, Kostas E Bekris, and Abdeslam Boularias. 2017. Fast model identification via physics engines for data-efficient policy search. *arXiv preprint arXiv:1710.08893* (2017).

[32] Matthieu Zimmer, Paolo Viappiani, and Paul Weng. 2014. Teacher-student framework: a reinforcement learning approach. In *AAMAS Workshop Autonomous Robots and Multirobot Systems*.