# Learning Impedance Actions for Safe Reinforcement Learning in Contact-Rich Tasks

**Quantao Yang**[1], **Alexander Dürr**[2], **Elin Anna Topp**[2], **Johannes A. Stork**[1], **Todor Stoyanov**[1]
[1]AMM Lab, Örebro University, Sweden
[2]Dept. of Computer Science, Lund University, Sweden
quantao.yang@oru.se

## Abstract

Reinforcement Learning (RL) has the potential of solving complex continuous control tasks, with direct applications to robotics. Nevertheless, current state-of-the-art methods are generally unsafe to learn directly on a physical robot as exploration by trial-and-error can cause harm to the real world systems. In this paper, we leverage a framework for learning latent action spaces for RL agents from demonstrated trajectories. We extend this framework by connecting it to a variable impedance Cartesian space controller, allowing us to learn contact-rich tasks safely and efficiently. Our method learns from trajectories that incorporate both positional, but also crucially impedance-space information. We evaluate our method on a number of peg-in-hole task variants with a Franka Panda arm and demonstrate that learning variable impedance actions for RL in Cartesian space can be safely deployed on the real robot directly, without resorting to learning in simulation and a subsequent policy transfer.

## 1 Introduction

Reinforcement Learning (RL) has been a promising framework to automatically fulfill complex continuous control tasks, yet contact-rich behaviors are hard learning problems, because current state-of-the-art methods are generally not safe to learn directly on a physical robot and require a vast amount of interaction experience. When it comes to learning how to solve a challenging real world task with a robot, trial-and-error learning is critically unsafe as random exploration can harm the environment or the robot itself. Despite their widespread relevance, tasks requiring controlling a robot in contact with the environment still pose a challenge to autonomous manipulation. In previous work, this problem has been solved by RL in simulation and transfer to reality [14], exploration with safety constraints [25] and learning from demonstration [17]. In order to guarantee safety in contact-rich tasks, we take advantage of variable impedance actions for RL to predict contact stiffness between the real robot and the environment.

When learning a policy through RL while using a simple controller, the policy would need to exhibit impedance behavior when coming in contact. This is a rather difficult behavior to learn for a RL algorithm because of the sudden change in dynamics. On the other hand, applying RL with a hand-tuned fixed stiffness impedance controller can solve the contact-rich task, but depending on the necessary softness of the contact, this can influence the behavior during free-space motion and alignment. Considering a policy with a task involving multiple steps, different stiffnesses for each step can be required.

In this paper we present an efficient RL framework that performs well on challenging contact-rich tasks that were previously not considered, failed on real robots [16] or require simulation to reality transfer [14]. We validate the successful application of our method to tasks that contain a combination of free-space motion, manipulation of constrained mechanisms and contact-rich manipulation.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

The main contributions of this paper are: (1) we leverage a framework for learning latent action spaces for RL agents from demonstrated trajectories and integrate it with a variable impedance Cartesian space controller; (2) our method extends the action space of this RL framework by incorporating variable impedance, allowing us to learn contact-rich manipulation tasks safely and efficiently; (3) we evaluate our method on a number of peg-in-hole task variants with a Franka Panda arm and demonstrate that learning variable impedance actions for RL in Cartesian space can be safely deployed on the real robot directly, without resorting to learning in simulation and a subsequent policy transfer.

## 2 Related Work

Compliant robot control allows for uncertanties when interacting through contact with the environment. Impedance control [9] presents an avenue for safe contact-rich manipulation. Variable impedance actuators, compliance and admittance control were summarized in [4, 24] with a focus on safe human-robot interaction and interaction through contact forces with the environment. This paper focuses on the learning aspect for variable impedance control and possible ways to approach this problem. Variable impedance control, that is, control with a time-dependent stiffness profile, has also been explored as an action space for reinforcement learning [14]. The resulting framework — Variable Impedance Control in End Effector Space (VICES) — was shown to enable faster and more easily transferrable learning of tasks involving contact, in comparison to action spaces relying on joint-space kinematic and dynamic control or fixed stiffness Cartesian space impedance control. Similar to [14], we also rely on variable stiffness impedance control, but allow for general task space control (e.g., relative to an arbitrary frame) and for fixing as constants parts of the task space and stiffness parameters.

A common approach to enabling RL on physical systems is to first train in a simulated version of the environment where safety and sample efficiency are not of critical importance. The learned policies can then be transferred to the real system via domain adaptation [1, 5] and dynamics randomization [3, 15, 21]. However, in domain adaptation an amount of real world samples are needed to update the simulation system to match the real one, while dynamics randomization requires a variety of simulated environments with randomized properties to train a model that can work across all of the environments. Another choice is to leverage safety constraints to restrict the RL exploration during trial-and-error learning [25]. Although this method can reduce collisions, it still requires learning in simulation and safety constraints need to be pre-defined in advance.

Instead of focusing on simulation to reality transfer and solving the hurdles that come with it, we focus on the possible ways to re-use previous real experience to learn in reality directly. To improve on this issue we investigate learning an embedding for skills in a continuous space with latent variable models [7, 13, 23]. The performance can be improved by narrowing down the latent search space by learning behavior priors [20]. We base our work on the idea of skill priors [16] — a framework for learning a low-dimensional embedding space for generating action sequences, along with a set of task-relevant prior distributions within that latent space. Learning prior and representation facilitates transfer of a learned skill to another task from a potentially large offline dataset to enhance learning efficiency. Unlike [16] which demonstrates the utility of the learned skill priors for learning long-horizon tasks in simulation, we concentrate on shorter but more complex and higher-dimensional tasks. Our approach is able to learn safely on the target physical system directly and consumes a fraction of the interaction samples considered in [16].

## 3 Approach

In this paper we consider an agent that acts according to a policy $\pi_\theta(a|s)$ which maps an action $a \in \mathcal{A}$ for each state $s \in \mathcal{S}$. The agent is trained based on a reward signal $r \in \mathcal{R}$ and aims to maximize the expected return:

$$G(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \right], \tag{1}$$

where $\tau$ is the state-action trajectory and $\gamma^t \in (0, 1]$ is the discount rate at time $t$. For a robot with $k$ joints, the observation vector $s_t$ is composed of (a) joint positions $\boldsymbol{q} \in \mathbb{R}^k$ and joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^k$, (b) end-effector position offset $\boldsymbol{e} \in \mathbb{R}^3$ and rotation $\theta_z$ in the $z$ direction with respect to the world frame, and (c) the environment contact force $F_{ext} \in \mathbb{R}^3$.

### 3.1 Cartesian Impedance Control

To implement contact-rich tasks, we use a Cartesian impedance controller [9]. In Cartesian impedance control, the robot end-effector dynamics are modelled as a mass-spring-damper system:

$$\boldsymbol{F}_a = \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}_d) + \boldsymbol{D}(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d) + \boldsymbol{M}(\ddot{\boldsymbol{x}} - \ddot{\boldsymbol{x}}_d), \tag{2}$$

where $\boldsymbol{F}_a \in \mathbb{R}^{6\times6}$ is the contact wrench with the environment, $\boldsymbol{x}$ and $\boldsymbol{x}_d$ are the current Cartesian pose and the desired pose of the robot end-effector. $\boldsymbol{K} \in \mathbb{R}^{6\times6}$, $\boldsymbol{D} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{M} \in \mathbb{R}^{6\times6}$ are the stiffness, damping and mass matrices of the system respectively.

Impedance control can be applied in Cartesian space to make the robot end-effector interact with the environment [2]. Standard impedance control uses constant or variable stiffness to command the system, but a pre-defined impedance behavior needs to be realized. We combine variable impedance control with the RL method by incorporating stiffness terms into the RL action space as described in Sec. 3.2.

### 3.2 Variable Impedance Action Space

In many tasks where the robot needs to interact physically with the environment, impedance control enables the manipulator to behave safely by balancing the contact stiffness and desired position of the task. The concept of variable impedance control was firstly proposed in [10]. Tsumugiwa et al. used a recursive least-square method to apply variable impedance control [22] by tuning the stiffness coefficient. Considering the flexibility and safety of variable impedance, we propose to let the RL agent predict the stiffness when the robot performs contact-rich tasks.

To train RL policy on the real robot directly, the system stiffness term $\boldsymbol{K}$ in equation (2) is incorporated into the agent action. According to [14], variable stiffness impedance control can enable the learned RL policy to adapt to the contacting environment while following the predicted Cartesian position for the robot end-effector. Therefore, we extend the policy action as the combination of end-effector pose $\boldsymbol{\xi} \in SE(3)$ in Cartesian space and variable stiffness matrix $\boldsymbol{K} \in \mathbb{R}^{6\times6}$. Stiffness matrix $\boldsymbol{K}$ contains 6-dimensional end-effector stiffness coefficients. One extra null-space stiffness coeffcient for the redundant robot is set as a constant value.

While a full Cartesian action space is possible, as for example in [14], we note that in some cases the task space may allow for a reduced action space. In our evaluations we consider a number of sample peg-in-hole insertion tasks wherein the end-effector is vertical to the $xy$ plane. Therefore, we ignore rotation around the $x$ and $y$ axes of the end-effector frame and the corresponding variable stiffness components by setting these components to desired constant values. Our 8-dimensional action space is thus composed of:

- end-effector translations $\boldsymbol{x} \in \mathbb{R}^3$ in Cartesian space,
- rotational angle $\theta_z \in \mathbb{R}$ around the $z$ axis,
- the diagonal coefficients $\boldsymbol{k} \in \mathbb{R}^4$ that determine the variable stiffness matrix $\boldsymbol{K}$ for the corresponding four Cartesian components.

We simplify the end-effector rotation matrix by only considering the rotational angle $\theta_z$ as we found the parameter to be particularly relevant when adapting the learned policy in some of our evaluations. To train the latent model, the action sequence is mapped to a posterior distribution $q(z|\boldsymbol{a})$ over embedding space by the skill encoder (Appendix A.1). The action normalization is applied because the scales of variable stiffness and position command values in our RL action space are different, which can lead to the stiffness component dominating the loss function.

## 4 Evaluation

We evaluated our method on an instance of a contact-rich task and conducted three peg-in-hole insertion experiments to evaluate the adaptation ability of our method using the Franka Panda arm.


(a) (b) (c)

Figure 1: Reinforcement learning curves for three experiments: (a) traing results for different target offset positions in circular peg insertion task; (b) comparison for learning curves from scratch in three different shape peg-in-hole experiments; (c) learning curves for different initial angles between the peg and the target workpiece when the trained policy in circular peg-in-hole task is adapted on the other two insertion experiments.

We also evaluate on several variations of the task in order to prove generalization. We validated that our skill prior RL method with variable impedance action space can train the real robot directly without training in simulation and the learned policy can be adapted to new contact-rich skills.

### 4.1 Peg-in-hole Learning from Scratch

We evaluated skill prior RL training on three peg-in-hole insertion tasks on the real robot Franka Panda directly, without resorting to pre-training in simulation. We collected 200 example trajectories to train the skill prior and subsequently used the learned skill prior to train SAC on the real Franka Panda arm. We trained 3 times for each case and plotted the mean and standard deviation for the episode reward in Figure 1.

The training rewards for the first experiment of inserting a circular peg are shown in Figure 1(a). The skill prior RL can generalize to different target positions. However, as the skill prior was trained with a fixed position, it takes more episodes to finish the first insertion task for different positions. Before the RL policy was trained on the real robot, we did not do any RL training in simulation, demonstrating that our skill prior RL using variable impedance in Cartesian space can be applied to contact-rich tasks safely without simulation to reality [26] domain transfer.

We compared the training results learned from scratch of three different shape pegs. For the training results shown in Figure 1(b), all target pegs are placed with 3cm position offset from the place where the circular hole was during dataset collection. We can see that the skill prior RL policy accumulated episode rewards successfully in circular peg-in-hole task, while the RL policy struggled when learning from scratch in the square and triangular peg insertion experiments. We speculate that this is due to the fact that we did not include any trajectory for square and triangular pegs in our insertion skill prior training dataset.

### 4.2 Adapting Learned Policy

In the last experiment, we adapted the learned RL policy from circular insertion task to insert into holes of two other shapes — square and triangular. For each experiment, we tested with two initial relative angles between the peg and the target workpiece. The learning curves are shown in Figure 1(c). The RL policy learned in circular peg insertion experiment can generalize well in similar peg-in-hole tasks if the initial angle is smaller than 5°. As comparison, when the initial angle is larger than 10°, the learned RL policy in the circular peg experiment failed to finish new square and triangular insertion tasks shown by green and red lines in Figure 1(c).

We count the emergency stop events and keep monitoring the contact force between the end-effector and the environment. In all experiments, zero emergency stop or excessive contact force event occurred because the skill prior guides learning variable impedance actions for the RL agent and the learned policy will adjust the contact stiffness when the end-effector interacts with the environment. In our experiments we tried to compare our method against some RL baselines. We found that none of state-of-the-art RL methods, such as Proximal Policy Optimization (PPO) [19] and SAC [6], can be applied directly to contact-rich manipulation tasks on a real robot without simulation to reality

transfer. Extending SAC by learning variable impedance actions also leads to collision or emergency stop when applied on the real robot.

## 5 Conclusion and Future Work

We have presented an approach that incorporates variable impedance in Cartesian space into the action space of a RL framework that learns latent embeddings from demonstrated trajectories. Our approach learns prior knowledge over the specific skill and a latent space that can be further decoded into real robot command sequences. We evaluated our method on three peg-in-hole insertion tasks with a Franka Panda arm and show that our skill prior RL using variable impedance in Cartesian space can be safely deployed on the real robot without simulation to reality domain transfer and it accelerates the adaptation of the learned policy over similar contact-rich skills. For future work, we intend to explore additional perception modalities and add visual information as one part of RL state space to improve the generalization ability and robustness of the policy.

## Acknowledgments and Disclosure of Funding

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

## References

[1] Anurag Ajay, Jiajun Wu, Nima Fazeli, Maria Bauza, Leslie P Kaelbling, Joshua B Tenenbaum, and Alberto Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3066–3073. IEEE, 2018.

[2] Alin Albu-Schaffer and Gerd Hirzinger. Cartesian impedance control techniques for torque controlled light-weight robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 657–663. IEEE, 2002.

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[4] Andrea Calanca, Riccardo Muradore, and Paolo Fiorini. A review of algorithms for compliant control of stiff and fixed-compliance robots. *IEEE/ASME transactions on mechatronics*, 21(2):613–624, 2015.

[5] Florian Golemo, Adrien Ali Taiga, Aaron Courville, and Pierre-Yves Oudeyer. Sim-to-real transfer with neural-augmented robot simulation. In *Conference on Robot Learning*, pages 817–828. PMLR, 2018.

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[7] Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Neville Hogan. Impedance control: An approach to manipulation. In *1984 American control conference*, pages 304–313. IEEE, 1984.

[10] Ryojun Ikeura and Hikaru Inooka. Variable impedance control of a robot for cooperation with a human. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 3, pages 3097–3102. IEEE, 1995.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

[13] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, pages 1113–1132. PMLR, 2020.

[14] Roberto Martín-Martín, Michelle A Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1010–1017. IEEE, 2019.

[15] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[16] Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on Robot Learning (CoRL)*, 2020.

[17] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:297–330, 2020.

[18] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[20] Noah Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In *International Conference on Learning Representations*, 2019.

[21] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[22] Toru Tsumugiwa, Ryuichi Yokogawa, and Kei Hara. Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 644–650. IEEE, 2002.

[23] William Whitney, Rajat Agarwal, Kyunghyun Cho, and Abhinav Gupta. Dynamics-aware embeddings. In *International Conference on Learning Representations*, 2019.

[24] Sebastian Wolf, Giorgio Grioli, Oliver Eiberger, Werner Friedl, Markus Grebenstein, Hannes Höppner, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, Manuel G Catalano, et al. Variable stiffness actuators: Review on design and components. *IEEE/ASME transactions on mechatronics*, 21(5):2418–2430, 2015.

[25] Quantao Yang, Johannes A Stork, and Todor Stoyanov. Null space based efficient reinforcement learning with hierarchical safety constraints. In *2021 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2021.

[26] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744. IEEE, 2020.

# A Appendix

## A.1 Reinforcement Learning with Skill Priors

We adapt the Skill Prior RL (SPiRL) [16] framework to solve robot contact-rich tasks (e.g. peg-in-hole) by learning jointly a latent representation of skills and the prior over this latent space. We use a modified variational autoencoder (VAE) [12] model to learn a low-dimensional skill latent space $\boldsymbol{\mathcal{Z}}$ from a dataset of pre-collected contact-rich trajectories. The VAE model consists of a skill encoder $q(z|\boldsymbol{a})$ that outputs the latent representation $z$ of a skill and a decoder $p(\boldsymbol{a}|z)$ that predicts a sequence of actions $\boldsymbol{a} = \{a_t, \cdots, a_{t+H-1}\}$ that the skill embedding $z$ represents, where $H \in \mathbb{N}^+$ is the action horizon. As described in [16], a skill prior model $p_{\boldsymbol{a}}(z|s_t)$ is used to generate a prior distribution over the latent space $\boldsymbol{\mathcal{Z}}$ based on the state $s_t$. This distribution serves as guidance for the policy to determine which skills are worth exploring. Following [18] we maximize the evidence lower bound (ELBO):

$$\log p(\boldsymbol{a}) \geq \mathbb{E}_q[\log p(\boldsymbol{a}|z) - \beta \left(\log q(z|\boldsymbol{a}) - \log p(z)\right)], \tag{3}$$

where $\beta$ is a hyperparameter used to tune the regularization term.

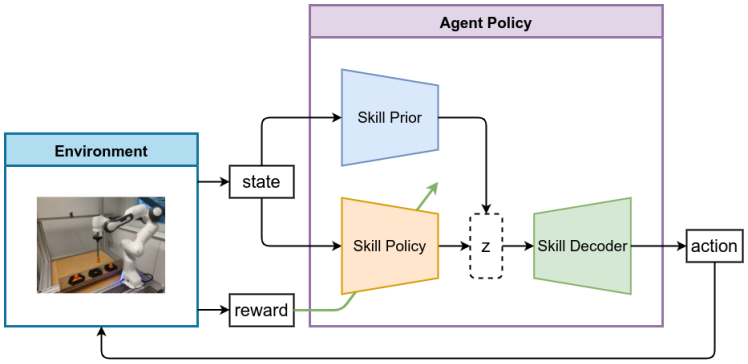


Figure 2: Skill prior RL framework: once the skill prior and the skill decoder block in the diagram are learned, a skill policy is trained using RL to generate embedding action $z$ that can be decoded into a sequence of real robot action commands.

We follow the skill prior RL (SPiRL) algorithm described in [16] that maximizes the expected return along with the policy's entropy term that penalizes divergence from action prior depicted in Figure 2. We add the normalization operation for the reconstructed action sequence due to the variable stiffness scale. As we train our skill policies directly on the real robot, we keep monitoring the contact force between the end-effector and the environment and reset the system if any constraint is violated. Such reset events are undesireable as they slow down learning and are potentially dangerous to the robot. Thus, one of the goals of our approach is to minimize the number of reset events that occur during training.

## A.2 Experimental Setup

We implemented contact-rich tasks with three different shapes of pegs and workpieces including circular, triangular and square. To train the skill prior in advance, we collected 200 insertion trajectories for the circular hole using a finite state machine that divides each trajectory into downward reaching, spiral motion alignment and insertion.

For training the skill embedding variable, the encoder and decoder of the VAE model are implemented as a long short-term memory (LSTM) [8] of 128 hidden units. The latent variable $z$ is embedded as the Gaussian posterior distribution of different dimensions. The skill prior is represented as a 5-layer fully-connected network. Adam [11] is used to optimize the neural network model. We tuned some hyperparameters in our experiments and chose the regularization weight $\beta$ in equation (3) as 5e-5 and the learning rate as 1e-3.

We use the skill prior SAC [16] implementation to predict the latent action with RL discount factor 0.99 and batch size 128. The reward function is simply defined as:

$$r = \begin{cases} 10, & \text{if } e_z < h_d \\ -d, & \text{otherwise} \end{cases} \tag{4}$$

where $e_z$ is the end-effector position in the $z$ direction, $h_d$ is the hole height and $d$ is end-effector distance to the hole.