Variable Impedance Skill Learning for Contact-Rich Manipulation

Quantao Yang, Alexander Dürr, *Member, IEEE*, Elin Anna Topp, Johannes A. Stork, and Todor Stoyanov

***Abstract*—Contact-rich manipulation tasks remain a hard problem in robotics that requires interaction with unstructured environments. Reinforcement Learning (RL) is one potential solution to such problems, as it has been successfully demonstrated on complex continuous control tasks. Nevertheless, current state-of-the-art methods require policy training in simulation to prevent undesired behavior and later domain transfer even for simple skills involving contact. In this paper, we address the problem of learning contact-rich manipulation policies by extending an existing skill-based RL framework with a variable impedance action space. Our method leverages a small set of suboptimal demonstration trajectories and learns from both position, 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 deployed directly on the real robot, without resorting to learning in simulation.**

***Index Terms*—Machine learning for robot control, reinforcement learning, variable impedance control.**

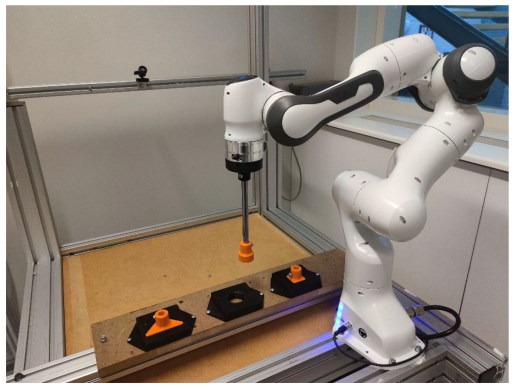


Fig. 1. Experimental setup for three peg-in-hole tasks of different shapes: circular, square and triangular. Variable impedance actions enable reinforcement learning to train directly on a real Franka Panda arm without pre-training in simulation.

I. Introduction

When it comes to learning how to solve a challenging real world task with a robot, we typically face a contact-rich manipulation or assembly problem. Despite their widespread relevance, tasks requiring controlling a robot in contact with the environment still pose a challenge to autonomous manipulation. Reinforcement Learning (RL) has been a promising framework to automatically learn these tasks, yet contact-rich behaviors are hard learning problems, because current state-of-the-art methods require a vast amount of interaction experience and are generally not safe to learn directly on a physical robot. For instance, learning a bin-picking task can already require large-scale data collection with an array of robots to collect the necessary dataset [1]. Moreover, contact-rich manipulation requires fine movements to correct deviations that may not be perceivable with a camera, and therefore solely relying on sensors integrated into the robot arm as feedback. In previous work, this problem setting required expert coded control, reinforcement learning in simulation and transfer to reality [2] or methods that require a large amount of time, resources and human feedback.

Traditional kinematic control in joint position or joint velocity space is ill-suited for contact tasks. When a segment of the robot is blocked due to interaction by an unforeseen obstacle, a kinematic control scheme would try to correct this disturbance by generating ever higher motor torques, resulting in potential damage to the robot and environment. Impedance control is one alternative that allows interaction with the environment by modeling the robot as a mass-spring-damper system. As a downside, if the stiffness of the virtual springs is too low, the desired trajectory will not be followed accurately, while if the stiffness is too high the robot can still end up generating excessive interaction forces.

An RL agent attempting to solve a contact-rich task via a direct joint position or joint torque interface would need to emulate an impedance-like behavior. However, 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

Manuscript received 24 February 2022; accepted 19 June 2022. Date of publication 30 June 2022; date of current version 11 July 2022. This letter was recommended for publication by Associate Editor F. Abu-Dakka and Editor J. Kober upon evaluation of the reviewers' comments. This work was supported by Knut and Alice Wallenberg Foundation through Wallenberg AI, Autonomous Systems, and Software Program (WASP). *(Corresponding author: Quantao Yang.)*

Quantao Yang and Johannes A. Stork are with the Autonomous Mobile Manipulation Lab, Center for Applied Autonomous Sensor Systems (AASS), Örebro University, 702 81 Örebro, Sweden (e-mail: quantao.yang@oru.se; johannesandreas.stork@oru.se).

Alexander Dürr and Elin Anna Topp are with the Department of Computer Science, Faculty of Engineering (LTH), Lund University, 221 00 Lund, Sweden (e-mail: alexander.durr@cs.lth.se; elin_anna.topp@cs.lth.se).

Todor Stoyanov is with the Autonomous Mobile Manipulation Lab, Center for Applied Autonomous Sensor Systems, Örebro University, 702 81 Örebro, Sweden, and also with the Department of Computing and Software, McMaster University, Hamilton, ON L8S 4L8, Canada (e-mail: todor.stoyanov@oru.se).

Digital Object Identifier 10.1109/LRA.2022.3187276

alignment. Considering a policy with a task involving multiple steps, different stiffnesses for each step can be required.

In this paper we present a variable impedance skill learning framework that performs well on challenging contact-rich tasks that were previously not considered, failed on real robots [3] or require simulation to reality transfer. 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. The types of movements, manipulations and contacts appear in basic everyday tasks as well as typical industrial assembly.

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 by incorporating variable impedance into the action space of this RL framework; (2) we evaluate our method on a number of peg-in-hole task variants with a Franka Panda arm and demonstrate improved generalization of the learned policies. By learning skills in a variable impedance control action space our method can be deployed directly on the real robot, without resorting to learning in simulation and a subsequent policy transfer.

II. Related Work

A. Variable Impedance Control

Conventional robot learning action space is only concerned with positional information in joint space or Cartesian space. However, impedance controller has made it possible to apply RL to contact-rich tasks [4]. Recently, variable impedance control and learning in complex interaction scenarios has gained much interest [5]–[7].

Variable impedance control in end-effector space (VICES), that is, control with a time-dependent stiffness profile, has been explored as an action space for reinforcement learning [2]. Although it is shown that VICES benefits RL policies transfer across robot models in simulation or simulation-to-reality, training policies directly on the real robot is still challenging. In comparison, we validate that our approach is applicable on the real robot directly. In [8] and [9], Buchli *et al.* achieved variable impedance control for practical high degree-of-freedom robotic tasks with an RL algorithm, PI^2 (Policy Improvement with Path Integrals), which requires no tuning of algorithmic parameters besides the exploration noise. However, their method used the joint space impedance that limited policy transferability. Based on PI^2, [10] draws parallels to human behavior, showing for unpredictable perturbations impedance is increased and for predictable perturbations a feed-forward policy is learned to offset disturbances. In contrast to these approaches, we combine variable impedance actions in Cartesian space with skill prior RL to improve the generalization ability of the policy.

A recent work [11] proposes to learn both the variable impedance policy and reward function using an inverse RL method. However, the policy outputs either the impedance gain or the feedback force and the action space does not contain positional information. In our preliminary work [12] we 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. In this letter we extend and validate our prior approach [12].

B. Domain Transfer

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 [13] and dynamics randomization [14], [15]. However, for 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. This reduces the ability to solve tasks requiring high accuracy. [16] investigated the effect of domain randomization on contact-rich, real-robot applications, but it is based on rigid position controller. In contrast, the combination of variable impedance actions and training skill priors learned from demonstration [3] allows us to learn a solution policy directly on the real robot.

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. Offline RL approaches [17], [18] can make use of offline data sets but require reward annotation for future tasks which can be challenging. To improve on this issue, we looked at transferring skills between tasks [19] without reward annotation. The two main options are the extraction of sub-policies that can be invoked [20], [21], and learning an embedding for skills in a continuous space with latent variable models [22], [23]. With the second option, the learning can be improved by narrowing down the latent search space by learning behavior priors [24]. We base our work on the idea of skill priors [3] — 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 data set to enhance learning efficiency.

Unlike [3] 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 directly on the target physical system and requires a fraction of the interaction samples needed by the state-of-the-art uninformed model-free approaches.

III. Approach

We consider an agent that acts according to a policy $\pi_\theta(a|s)$ which maps states $s \in \mathcal{S}$ to a probability distribution over the actions $a \in \mathcal{A}$. The agent is trained based on a reward signal $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and aims to maximize the expected return:

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

where T is the episode horizon, τ is the state-action trajectory induced by π_θ and $\gamma^t \in (0, 1]$ is the discount rate at time t.

A. Cartesian Impedance Control

To implement contact-rich tasks, we use a Cartesian impedance controller [25]. In Cartesian impedance control, the robot end-effector dynamics are modelled as a mass-spring-damper system:

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

where $\boldsymbol{F}_a$ 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 [26]. 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 Section III-C.

B. Reinforcement Learning With Skill Priors

We adapt the Skill Prior RL (SPiRL) [3] framework to solve 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) [27] 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 [3], 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 [28] we maximize the evidence lower bound (ELBO):

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

where β is a hyperparameter used to tune the regularization term.

The skill prior RL framework is illustrated in Fig. 2. A policy $\pi_\theta(z|s_t)$ over the latent action space is trained to output embeddings that are decoded into real action sequences by the pre-trained decoder $p(\boldsymbol{a}|z)$. We use Soft Actor-Critic (SAC) [29] to maximize the RL return plus the policy's entropy term:

$$G(\theta) = \mathop{\mathbb{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1}) + \alpha\mathcal{H}(\pi_\theta(a_t|s_t))\right], \quad (4)$$

where α is the weight for the entropy term. In our case the policy learns in the embedding variable space, producing a latent action $z \in \boldsymbol{\mathcal{Z}}$. The entropy term is defined as the negative Kullback-Leibler (KL) divergence between the policy $\pi_\theta(z_t|s_t)$ and learned skill prior $p_{\boldsymbol{a}}(z_t|s_t)$:

$$\mathcal{H}(\pi_\theta(z_t|s_t)) \propto -D_{\mathrm{KL}}(\pi_\theta(z_t|s_t), p_{\boldsymbol{a}}(z_t|s_t)). \quad (5)$$

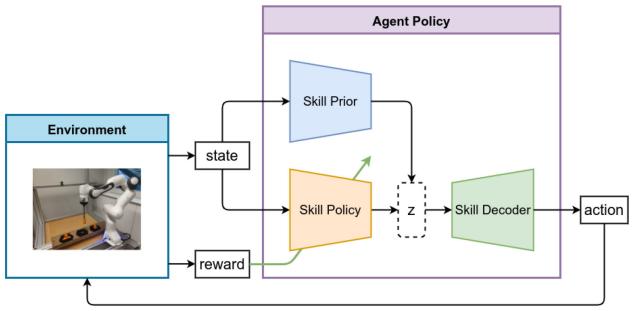


Fig. 2. Skill prior RL framework: once the skill prior and the skill decoder blocks in the diagram are learned, a skill policy is trained using RL to generate an embedded action z that can be decoded into a sequence of real robot action commands.

Algorithm 1: Learn Variable Impedance Actions

Collect demonstrated trajectories τ_i
Train the skill prior model $p(z|s_t)$ and skill decoder $p(\boldsymbol{a}|z)$ with trajectories
Initialize the policy $\pi_\theta(z_t|s_t)$ and replay buffer $\mathcal{B}$
for *each episode=1,M* **do**
  Select high-level action $z_t \sim \pi_\theta(z_t|s_t)$
  Decode and execute impedance action sequence $\boldsymbol{a} = \{a_t, \cdots, a_{t+H-1}\}$, receive the reward r_t
  Store transition tuple $(s_t,\ z_t,\ r_t,\ s_{t+1})$ in $\mathcal{B}$
  Update the policy π_θ to maximize the return in (4) by SAC
end
return the trained policy $\pi_\theta(z_t|s_t)$

Generally we follow the skill prior RL (SPiRL) algorithm described in [3] that maximizes the expected return along with the policy's entropy term that penalizes divergence from the action prior $p_{\boldsymbol{a}}(z|s_t)$. We add a 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.

C. Learning Variable Impedance Actions

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 end-effector. The concept of variable impedance control was firstly proposed in [30]. Considering the flexibility and safety of variable impedance, we propose to let the RL agent predict variable impedance actions for SPiRL (VIA-SPiRL) when the robot performs contact-rich tasks.

To train skill prior RL on the real robot directly, the system stiffness term $\boldsymbol{K}$ in (2) is incorporated into the agent action. According to [2], 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 coefficient for the redundant robot is set as a constant value. We choose to fix the damping and inertia as it is easier to make the real robot system more stable with fixed damping parameters.

The VIA-SPiRL is summarized in Algorithm 1. 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. Different from the work in [3], we normalize the action sequence before the encoder mapping and denormalized back to the command action sequence after decoding. We apply the action normalization 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.

IV. Evaluation

We evaluate our method on an instance of a contact-rich task and conducted peg-in-hole insertion experiments to evaluate the adaptation ability of our method using the Franka Panda arm as shown in Fig. 1. We also evaluate on several variations of the task in order to demonstrate generalization. Some are variations in the shapes of the workpieces, and some are variations on the pose of the workpiece relative to the manipulator base. We validated that our VIA-SPiRL method can train the real robot directly without training in simulation and the learned policy can be adapted to new contact-rich skills.

In our experiments we use a Cartesian impedance controller for a Frank Panda arm and Pytorch for RL training. The Robot Operating System (ROS) is used for communication between the robot controller and RL agent. Our implementation is available at https://github.com/yquantao/learning_impedance_actions.

A. State and Action Spaces

To solve contact-rich tasks we incorporate contact force between the end-effector and the environment into the RL state s_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 $\theta_{\mathbf{z}}$ in the $\mathbf{z}$ direction, and (c) the environment contact force $F_{ext} \in \mathbb{R}^3$.

While a full Cartesian action space is possible, as for example in [2], we note that in some cases the task 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 $\mathbf{xy}$ plane. Therefore, we ignore rotation around the $\mathbf{x}$ and $\mathbf{y}$ axes of the end-effector frame and the corresponding variable stiffness components. Our 8-dimensional action space is thus composed of:

- end-effector translations $\boldsymbol{x} \in \mathbb{R}^3$ in Cartesian space,
- rotational angle $\theta_{\mathbf{z}} \in \mathbb{R}$ around the $\mathbf{z}$ 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 fix the damping coefficients to equal $2\sqrt{k}$, which we find allows for stable behavior.

We simplify the end-effector rotation matrix by only considering the rotational angle $\theta_{\mathbf{z}}$ as we found the parameter to be particularly relevant when adapting the learned policy in some of our evaluations. In order to improve learning stability for the skill decoder $p(\boldsymbol{a}|z)$ it is beneficial to have a bounded and balanced output space for the action $\boldsymbol{a}$. While the position and orientation components are at a similar scale, the stiffness coefficients k could easily dominate the loss function. We alleviate this problem by normalizing k by rescaling it with the $L2$ norm of the standard deviation observed in the demonstration set. During training of the RL algorithm the decoded actions are then scaled back to the original space, prior to execution on the robot.

B. Experimental Setup

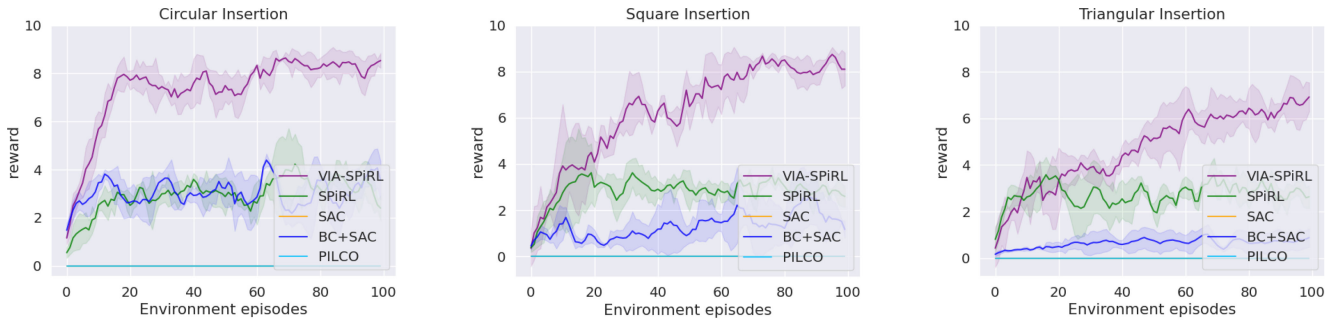
We implement contact-rich tasks with three different shapes of pegs and workpieces including circular, square and triangular. To train the skill prior in advance, we collected 200 trajectories for each insertion task using a finite state machine (FSM) that divides each trajectory into three phases: downward reaching, spiral motion alignment and insertion. During executing these motions, we record the corresponding Cartesian poses and stiffness. For simplicity, we use fixed stiffness during each motion phase in one specific trajectory. We then collect data by sampling values for this parametrized insertion controller and executing trajectories on the robot. We use a terminal flag value to indicate the success or failure of each trajectory.

Insertion trajectories were collected for each peg-in-hole task with the real robot in advance and were used to train the skill prior and embedding space. The purpose of spiral alignment is to search for the target hole in a large area. In our case the Archimedean spiral motion is defined in polar coordinates (r_p, φ):

$$r_p = b_0 + b\varphi, \tag{6}$$

where b_0 describes how far away from the origin the spiral should start and b represents the distance between each turn of the spiral. We set $b_0 = 0$ and randomize b when collecting training trajectories. When we collect the dataset for square and triangular peg-in-hole tasks, we use a $\sin$ function to control the rotation $\theta_{\mathbf{z}}$ along the $\mathbf{z}$ axis during spiral motion.

For training the skill embedding variable, the encoder and decoder of the VAE model are implemented as a long short-term memory (LSTM) of 128 hidden units to generate sequential robot trajectories appropriately. The latent variable z is embedded as the Gaussian posterior distribution and the skill prior is represented as a 5-layer fully-connected network. Adam optimizer is used to optimize the neural network model. We tuned some hyperparameters in our experiments and chose the regularization weight β in (3) as 5e-5 and the learning rate as 1e-3. We use SPiRL [3] implementation to predict the latent action with discount factor 0.99 and batch size 128.



(a) (b) (c)

Fig. 3. Reinforcement learning curves for all methods on peg-in-hole tasks, only VIA-SPiRL succeeds in learning the insertion policy. The plots show the average episode reward and standard deviation over three seeds: (a) results for circular peg insertion task; (b) results for square peg insertion task; (c) results for triangular peg insertion task.

TABLE I
COMPARISON OF DIFFERENT METHODS

Method	Access to Demonstrations	Variable Impedance	Online Learning
SAC	✗	✓	✓
BC+SAC	✓	✓	✓
PILCO	✗	✓	✓
SPiRL	✓	✗	✓
FSM	✗	✓	✗
VIA-SPiRL	✓	✓	✓

TABLE II
CIRCULAR PEG-IN-HOLE SUCCESS RATE

Method	1cm	2cm	3cm
PILCO	0.00	0.00	0.00
SAC	0.00	0.00	0.00
BC+SAC	0.35	0.33	0.22
SPiRL with max impedance	0.10	0.00	0.00
SPiRL with min impedance	0.34	0.27	0.18
SPiRL with average impedance	0.62	0.50	0.49
FSM	0.75	0.74	0.55
VIA-SPiRL	**0.96**	**0.92**	**0.90**

We compare our method against some RL baselines. We find that none of state-of-the-art RL methods, such as SAC [29], 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. Therefore, we choose several baseline methods to compare with: (1) Soft Actor-Critic (SAC), (2) Behavioral Cloning finetuned with SAC (BC+SAC), (3) Probabilistic Inference for Learning Control (PILCO) [31], (4) Skill Prior RL (SPiRL) with fixed impedance, and (5) Finite State Machine (FSM). We list all methods evaluated in this paper in Table I and compare them in terms of access to demonstration data, use of variable impedance and online learning.

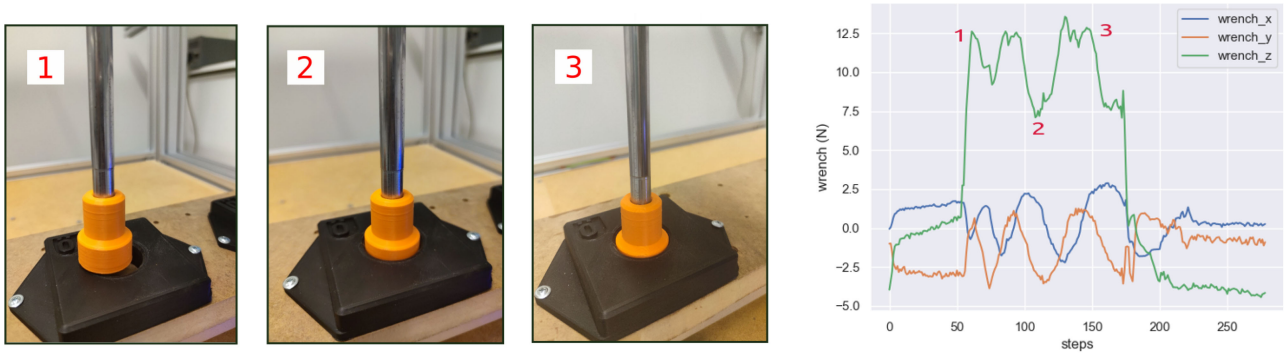
C. Learning Variable Impedance for Peg-in-Hole Tasks

We collect 200 demonstrated trajectories to train the skill prior for each peg-in-hole task and subsequently use the learned skill prior to train SAC on the real Franka Panda arm. Each insertion experiment is trained for 100 episodes and each episode consists of 300 actions sent to the variable impedance Cartesian controller at a rate of 5 Hz. Therefore, a total of 30 K online transitions are collected on the real robot which results in about two hours per experimental evaluation. We train 3 times for each case and plot the learning curves of all methods on peg-in-hole tasks in Fig. 3.

The training rewards for the experiment of inserting a circular peg are shown in Fig. 3(a). Only VIA-SPiRL succeeds in learning the insertion policy. SPiRL with fixed stiffness (500 N/m) and BC+SAC accumulate rewards occasionally, while SAC and PILCO fail to learn the policy. We observe similar learning results for the square and triangular peg-in-hole tasks shown in Fig. 3(b) and 3(c) respectively, although it takes more trials for the policy to converge on the triangular insertion task.

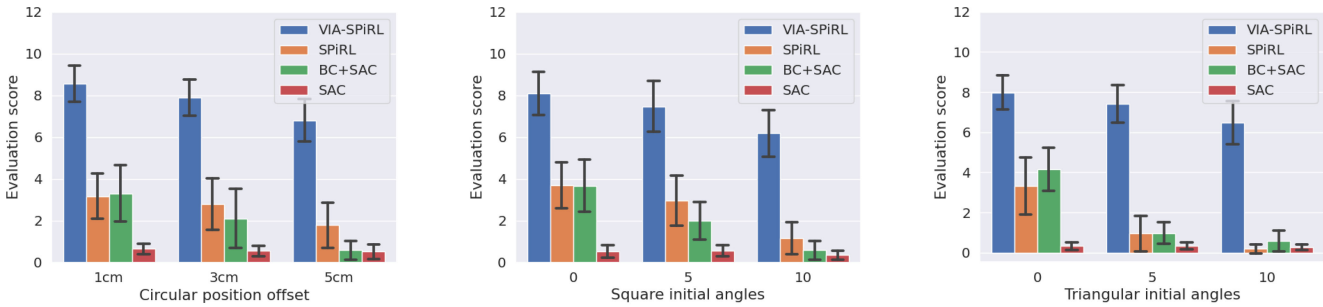
Fig. 4(a) demonstrates a sequence of a successful insertion in evaluation for a circular peg-in-hole that consists of downward reaching, spiral alignment and insertion. An example contact wrench of three axes during a successful insertion is depicted in Fig. 4(b). When the end-effector touches the workpiece, the contact wrench for $\mathbf{z}$ axis increases to around $12.5N$ and fluctuates round this value during the spiral motion until the robot finds the target hole. The contact force in the $\mathbf{z}$ direction increases again when the end-effector touches the bottom of the hole and decreases when it moves upwards. In Fig. 4(b) number labels indicate the wrench of touching the workpiece, inserting and touching the board.

We evaluate the circular peg-in-hole average success rate of using different methods. We test 100 insertion trials after training for each position offset and the results are shown in Table II. To test SPiRL with fixed impedance, we choose three different stiffness values from our demonstrated dataset: the maximum $1000N/m$, the minimum $100N/m$ and the average $500N/m$. The baseline SPiRL shows poor performance when the action is assigned high stiffness or low stiffness. With medium value, the policy shows better success rates, but still not comparable with our VIA-SPiRL. Due to the poor performance of SPiRL with high and low stiffness, we compare our method with SPiRL



(a) (b)

Fig. 4. An example of successful circular peg-in-hole insertion in evaluation: (a) the insertion consists of downward reaching, spiral alignment and insertion labeled by the numbers 1, 2 and 3 respectively; (b) the wrench for $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ axes during inserting, number labels indicate the wrench of touching the workpiece, inserting and touching the board.



(a) (b) (c)

Fig. 5. Evaluation results to test contact force for all methods. The scoring mechanism attributes higher scores to actions not exceeding the contact force threshold $30N$. Evaluation scores show for different: (a) target positions in the circular peg insertion tasks; and inital angle offsets in the square (b) and tiangular (c) insertion tasks.

using the average impedance ($500N/m$) in other experiments. As expected, PILCO and SAC do not succeed in learning a successful insertion policy. We stipulate that the main reason for that failure is that neither of these methods has access to prior demonstrations and must therefore explore with random actions, which naturally have a low probability of success. In addition, both of these methods would be affected by the inherent over-parametrization of the variable impedance action space. That is, multiple choices of a Cartesian pose and Cartesian impedance map to the same joint torques on the robot, which presents a challenge when learning the expected consequences of actions. In contrast, the methods that utilize demonstrations initialize in the vicinity of locally optimal variable impedance actions, which make learning more tractable. Nevertheless, the naive use of demonstrations (behavior cloning followed by SAC) still does not succeed in solving the task within our training horizons, testifying to the benefit of skill priors in guiding learning.

The skill prior RL action is extended by adding variable impedance in end-effector Cartesian space including three translational stiffness and one rotational stiffness of $\theta_{\mathbf{z}}$ direction. We plot the variable stiffness for a successful insertion example in Fig. 6. When the robot touches the workpiece at step 50 in this trial, the policy reduces the contact stiffness values, especially for the translational directions. During inserting, the robot adjusts the stiffness value according to the observed state. Before

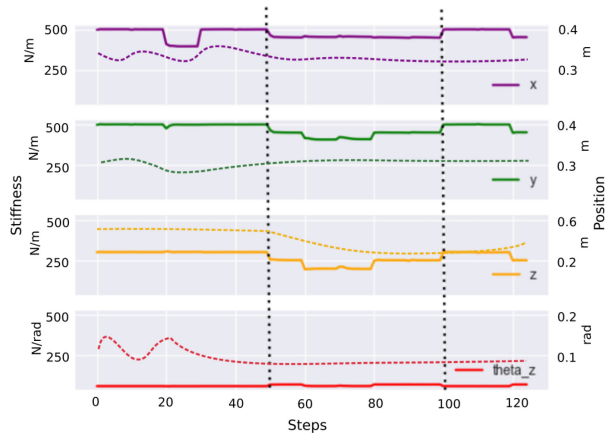


Fig. 6. Variable stiffness and Cartesian position command in a successful contact-rich insertion task are shown by solid and dotted lines respectively. The black dotted lines at step 50 and 100 indicate that the end-effector touches the workpiece and the robot moves the peg out of the hole respectively.

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 without simulation to reality domain transfer.

Following [29], we tune the entropy weight α in (4) automatically during training. The tuning procedure in circular, square

TABLE III
PEG-IN-HOLE SUCCESS RATE FOR DIFFERENT POSITION OFFSETS AND INITIAL ANGLES

	Circular					Square					Triangular				
	1cm	3cm	5cm	5°	10°	1cm	3cm	5cm	5°	10°	1cm	3cm	5cm	5°	10°
VIA-SPiRL	**0.96**	**0.90**	**0.77**	**0.97**	**0.95**	**0.92**	**0.88**	**0.68**	**0.90**	**0.89**	**0.87**	**0.89**	**0.60**	**0.79**	**0.72**
SPiRL	0.62	0.49	0.26	0.45	0.47	0.31	0.29	0.28	0.31	0.22	0.25	0.09	0.03	0.15	0.12
FSM	0.75	0.55	0.05	0.92	0.95	0.60	0.15	0.00	0.55	0.48	0.46	0.05	0.00	0.26	0.21

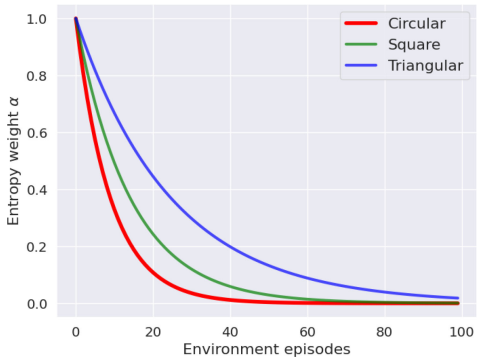


Fig. 7. Comparison of tuning entropy weight α during training policy for circular, square and triangular insertion tasks. In all our VIA-SPiRL experiments α converges to 0 smoothly.

TABLE IV
SUCCESS RATE WHEN ADAPTING POLICIES

From \ To	Circular	Square	Triangular
Circular	N/A	0.7	0.4
Square	0.8	N/A	0.6
Triangular	0.8	0.7	N/A

and triangular peg-in-hole experiments is shown in Fig. 7. In all our VIA-SPiRL experiments, the entropy weight α decreases and converges to 0 smoothly, which means that the entropy term accounts less for the total RL return in (4) when the policy succeeds in learning the peg-in-hole skill.

D. Safe Learning in Contact

During evaluation tests we monitor the contact force between the end-effector and the environment. We also assess the performance of the policy according to the contact force for all peg-in-hole insertion trials. We give a constant penalty -0.1 for each evaluation trial if the contact force in the $\mathbf{z}$ direction exceeds the threshold value. We choose $30N$ as the threshold value because the contact force above it might trigger the robot's emergency braking system. We set the starting highest score as 10 since we have 100 evaluation trials. We score the circular insertion policy for different position offsets and the square and triangular insertion policy for different initial angles. The evaluation results are shown in Fig. 5. In all testing scenarios, our VIA-SPiRL achieves better performance in terms of the contact force. SPiRL with fixed impedance and BC+SAC show similar performance, while SAC performs worst due to it easily triggering the robot braking system.

In all experiments for our method, no emergency stop 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. As comparison, the baseline SPiRL using high stiffness value easily exceed the contact force threshold and can not finish insertion tasks. When the stiffness is too low or average value, SPiRL also shows worse performance compared with our VIA-SPiRL method. Meanwhile, the end-effector got stuck and emergency stop was triggered in two triangular peg-in-hole trials, because the system does not adjust the interaction stiffness automatically as our method does.

E. Evaluation of Policy Generalization

We evaluate the performance of learned policies by testing different position offsets and initial rotation angles for target workpieces. For each evaluation case, we test 100 insertion trials and the success rates are shown in Table III. Similar to the results of circular peg-in-hole task, the policies using variable impedance actions outperform SPiRL with fixed impedance and FSM in all testing scenarios. We observe that the RL policy performs better when inserting the peg if the offset is small, but the success rate decreases sharply to 0.77, 0.68 and 0.60 for 5 cm offset in circular, square and triangular insertion tasks respectively. Meanwhile, for square and triangular insertion tasks, the success rates also depend on the initial angles between the target workpiece and the peg. The results indicate that our VIA-SPiRL method can generalize well to different target positions and initial angles.

In the last experiment, we adapt the learned RL policy from one peg-in-hole task to insert into holes of two other shapes. For this experiment, we ignore the influence of the position offsets and rotation angles and train each policy for further 50 episodes. We test 10 insertion trials for each adapting case after training and the success rates are shown in Table IV. The RL policies learned in square and triangular peg insertion experiments can generalize well in the easier circular peg-in-hole task. As comparison, the learned RL policy in the circular peg experiment shows inferior performance. We infer that it is because the dynamics of the square and triangular insertion tasks are more complex, while the rotational angle $\theta_{\mathbf{z}}$ is not utilized by the optimal policy in the circular insertion task.

V. CONCLUSION

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 evaluate 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 deployed on the real robot without simulation to reality domain transfer and the learned policy can be adapted to different position offsets and initial angles.

One limitation of our VIA-SPiRL method is that the learned policy from one specific peg-in-hole task can not be applied to another shape peg-in-hole insertion task. This might be because the embedding space has not seen the dynamics of a new insertion task from the demonstrated dataset. Although the skill prior in our work has achieved promising results on the real robot, there is still potential improvement for the reconstruction of the RL policy action. Therefore, we plan to investigate the RL policy adaptation in more dynamic conditions in our future work.

REFERENCES

[1] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4-5, pp. 421–436, 2018.

[2] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, IROS, 2019, pp. 1010–1017.

[3] K. Pertsch, Y. Lee, and J. J. Lim, "Accelerating reinforcement learning with learned skill priors," in *Proc. Conf. Robot Learn.*, 2020, pp. 188–204.

[4] F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Force-based learning of variable impedance skills for robotic manipulation," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots*, 2018, pp. 1–9.

[5] F. J. Abu-Dakka, B. Nemec, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, and A. Ude, "Adaptation of manipulation skills in physical contact with the environment to reference force profiles," *Auton. Robots*, vol. 39, no. 2, pp. 199–217, 2015.

[6] M. Oikawa, T. Kusakabe, K. Kutsuzawa, S. Sakaino, and T. Tsuji, "Reinforcement learning for robotic assembly using non-diagonal stiffness matrix," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2737–2744, Apr. 2021.

[7] M. Bogdanovic, M. Khadiv, and L. Righetti, "Learning variable impedance control for contact sensitive tasks," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6129–6136, Oct. 2020.

[8] J. Buchli, E. Theodorou, F. Stulp, and S. Schaal, "Variable impedance control a reinforcement learning approach," in *Robotics: Science and Systems VI*, vol. 153, Cambridge, MA, USA: MIT Press, 2011.

[9] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 820–833, 2011.

[10] F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E. A. Theodorou, and S. Schaal, "Model-free reinforcement learning of impedance control in stochastic environments," *IEEE Trans. Auton. Mental Develop.*, vol. 4, no. 4, pp. 330–341, Dec. 2012.

[11] X. Zhang, L. Sun, Z. Kuang, and M. Tomizuka, "Learning variable impedance control via inverse reinforcement learning for force-related tasks," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2225–2232, Apr. 2021.

[12] Q. Yang, A. Dürr, E. A. Topp, J. A. Stork, and T. Stoyanov, "Learning impedance actions for safe reinforcement learning in contact-rich tasks," in *Proc. Workshop Deployable Decis. Mak. Embodied Syst.*, 2021.

[13] F. Golemo, A. A. Taiga, A. Courville, and P.-Y. Oudeyer, "Sim-to-real transfer with neural-augmented robot simulation," in *Proc. Conf. Robot Learn.*, 2018, pp. 817–828.

[14] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2018, pp. 3803–3810.

[15] O. M. Andrychowicz *et al.*, "Learning dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[16] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, and K. Harada, "Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach," *Appl. Sci.*, vol. 10, no. 19, 2020, Art. no. 6923.

[17] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2052–2062.

[18] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy Q-learning via bootstrapping error reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11784–11794.

[19] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, 2009.

[20] A. Mandlekar *et al.*, "IRIS: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data," in *Proc. IEEE Int. Conf. Robot. Automat.*, ICRA, 2020, pp. 4414–4420.

[21] Y. Lee, S.-H. Sun, S. Somasundaram, E. S. Hu, and J. J. Lim, "Composing complex skills by learning transition policies," in *Proc. Int. Conf. Learn. Representations*, 2018.

[22] C. Lynch *et al.*, "Learning latent plans from play," in *Proc. Conf. Robot Learn.*, 2020, pp. 1113–1132.

[23] K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, and M. Riedmiller, "Learning an embedding space for transferable robot skills," in *Proc. Int. Conf. Learn. Representations*, 2018.

[24] N. Siegel *et al.*, "Keep doing what worked: Behavior modelling priors for offline reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2019.

[25] N. Hogan, "Impedance control: An approach to manipulation," in *Proc. Amer. Control Conf.*, 1984, pp. 304–313.

[26] A. Albu-Schaffer and G. Hirzinger, "Cartesian impedance control techniques for torque controlled light-weight robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2002, pp. 657–663.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014, *arXiv:1312.6114*.

[28] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1278–1286.

[29] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[30] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1995, pp. 3097–3102.

[31] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 465–472.