Variable Stiffness Control for Sequential Contact Tasks

MSc Thesis

Amin Berjaoui Tahmaz

TU Delft

Variable Stiffness Control for Sequential Contact Tasks

by

Amin Berjaoui Tahmaz

Supervisors: Jens Kober & Ravi Prakash
Department: Cognitive Robotics (CoR)
Faculty: Faculty of Mechanical Engineering (ME)

TU Delft

Variable Stiffness Control for Sequential Contact Tasks

Amin Berjaoui Tahmaz, Ravi Prakash, and Jens Kober

***Abstract*—This paper presents a hierarchical reinforcement learning framework for efficient robotic manipulation in sequential contact tasks. We leverage this hierarchical structure to sequentially execute behavior primitives with variable stiffness control capabilities for contact tasks. Our proposed approach relies on three key components: an action space enabling variable stiffness control, an adaptive stiffness controller for dynamic stiffness adjustments during primitive execution, and affordance coupling for efficient exploration while encouraging compliance. Through comprehensive training and evaluation, our framework learns efficient stiffness control capabilities and demonstrates improvements in learning efficiency, compositionality in primitive selection, and success rates compared to the state-of-the-art. The training environments include block lifting, door opening, object pushing, and surface cleaning. Real world evaluations further confirm the framework's sim2real capability. This work lays the foundation for more adaptive and versatile robotic manipulation systems, with potential applications in more complex contact-based tasks.**

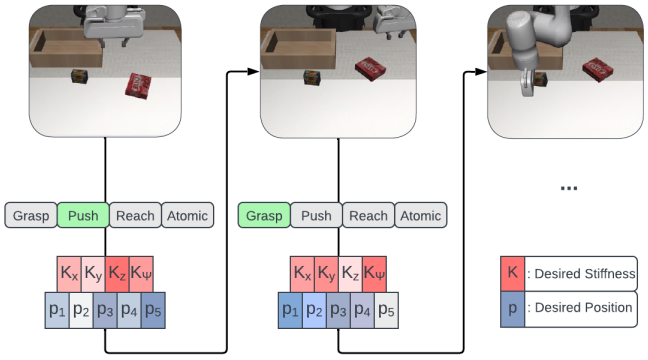

Fig. 1: The agent sequentially chooses a *behavior primitive* along with *controller parameters* to complete a task. At each step, it initiates a control loop with the desired parameters to execute the chosen behavior.

I. Introduction

FOR decades, the challenge of enabling robotic manipulators to solve complex long-horizon tasks has persisted. While existing research has made strides in addressing important aspects of long-horizon tasks, a critical gap remains in the context of contact-rich environments, highlighting a crucial area that requires further exploration and development. An example can be found in a common manipulation task: object sorting. A robot should be able to plan a series of precise actions over time while adjusting its positioning and applied forces to accommodate objects of varying shapes and sizes, while also taking the interaction environment into consideration. This paper focuses on the intersection of deep reinforcement learning (DRL) and adaptive stiffness control with the aim of addressing this longstanding challenge.

Prior work has extensively explored robotic manipulation in long-horizon applications. Conventional methods often use state machines [1][2] or symbolic reasoning [3][4] to learn action sequences for solving a task. However, these approaches explicitly design the decision-making sequence, which may introduce constraints that limit adaptability to different tasks and contribute to error accumulation throughout the task sequence. In response to these limitations, learning techniques such as hierarchical reinforcement learning (HRL) [5] have been employed, establishing themselves as a common approach for problems requiring sequential decision-making.

When deploying long-horizon frameworks in contact-rich environments, the integration of stiffness control becomes crucial for adapting to external forces and uncertainties during task execution. This adaptability ensures precision and stability in navigating contact-rich environments. However, despite a substantial body of research dedicated to variable stiffness control, current approaches are primarily tailored to short-horizon applications. These methods typically involve designing controllers that adjust end-point force in response to environmental forces [6], adapting impedance and damping parameters through learning techniques [7][8], and learning from a human demonstrator [9][10].

This paper aims to bridge the gap between sequential planning and adaptive stiffness control using a reinforcement learning framework. We design a framework that selects an action from a pre-defined library and outputs an initial estimate for controller parameters. During primitive execution, an adaptive controller is initiated to optimize the robot's stiffness, aiming for an balance between safety and performance. This design allows the robot to dynamically optimize stiffness parameters, enabling it to transition between high stiffness for precision tasks and increased compliance for enhanced adaptability. We present experiments conducted in both simulation and the real world, focusing on sequential tasks that deal with different contact challenges. Our results highlight notable advantages when compared to a state-of-the-art baseline.

The main contributions of this work include: (i) Designing a framework that can execute variable stiffness control across long-horizon tasks; (ii) Introducing a novel behavior affordance that concurrently optimizes for position and compliance; (iii) Evaluation of the learning efficiency, stiffness behavior, compositionality, and sim2real capability.

II. Related Works

A. Sequential Planning

Extensive work exists in the domain of task and motion planning (TAMP) encompassing a wide range of robotics applications. These methods range from the explicit design of decision-making frameworks to learned behavior sequences via machine learning.

Common approaches use *hierarchical task planning* to create high-level planners paired with a low-level controller. In the context of robot manipulation, this is particularly present in the form of finite state machines [1][11][12] or behavior trees [13][14] as high-level controllers. Similar approaches use symbolic reasoning [15][16][17] in which symbols are used to represent high-level tasks and constraints to guide the decision-making process. Despite the explainability offered by these methods, their pre-defined nature limits their ability to handle the inherent uncertainty and variability in real-world scenarios, which may lead to suboptimal performance. In contrast, our proposed framework simultaneously learns the high-level planner and optimizes the low-level controller parameters, making it better generalized and robust to uncertainty.

In recent years, learning approaches have been used to overcome those limitations. *Imitation learning (IL)* emerges as a prominent candidate for sequential planning by enabling the robot to learn a demonstrated behavior sequence. Most well-established approaches belong to behavior cloning methods, in which the robot replicates a demonstration sequence [18][19][20]. However, this makes the model overreliant on the demonstration sequence and significantly limits generalizability. To tackle this problem, imitation learning methods that can generalize learned sequences have been introduced [21][22][23]. Nevertheless, these methods are still greatly limited in their ability to generalize to new environments outside of a constrained setting. On the other hand, our framework tackles this problem by adapting the action sequence depending on the environment state. Furthermore, fitting on demonstration data leads to suboptimal performance due to human error.

To address these limitations, *Hierarchical Reinforcement Learning (HRL)* has garnered attention due to its capacity for long-horizon planning. State-of-the-art approaches include MAPLE [24], RAPS [25], and STAP [26]. All three methods train a hierarchical policy to choose and execute a primitive from a library of behavior primitives. Despite their ability to handle complex tasks and improve sample efficiency, a notable drawback is their reliance on static controllers. This greatly hinders performance particularly in contact tasks and may pose potential risks in real-world settings. Our method builds on these concepts and address these challenges by optimizing stiffness to maximize compliance without compromising task success.

B. Variable Stiffness Control

Existing methods for adapting the stiffness of an impedance controller primarily involve using task-specific impedance profiles. Common approaches include learning from demonstration methods to encode a stiffness profile, such as Dynamic Motion Primitives [27][28][29] or Gaussian Mixture Models [9][30]. Alternatively, some works rely on scheduling variable stiffness gains for different phases of a task [31][32][33]. Despite their ease of application, these methods suffer from a limited ability to generalize a given stiffness profile to different tasks while also being dependent on an expert demonstrator.

Reinforcement Learning (RL) has emerged as a promising learning method for learning stiffness profiles. There exists an abundance of methods that bootstrap the RL policy with initial stiffness demonstrations [34][35][36] to accelerate learning, which are then optimized for a given task. However, the issue of depending on an expert demonstrator remains unsolved.

Other RL approaches shift the focus to designing an appropriate action space. Using this approach, the agent samples impedance parameters as actions which are then used to adapt controller behavior. For applications requiring adaptive stiffness, an impedance action space has been implemented, in which the agent learns stiffness and damping parameters in joint space [37] and end-effector space [8]. Similar approaches used residual reinforcement learning solutions in which a policy outputs actions to support an existing controller in completing a task [1][38][39]. However, these methods fail in the context of long-horizon tasks due to their limited ability to capture sequential dependencies.

III. Preliminaries

In this section, we provide an overview of the fundamental concepts and terminologies surrounding our research. We also delve into the design decisions implemented in our framework.

Compliant Robot Control refers to a control strategy that allows robots to interact with their environment in an adaptive manner. In contrast to traditional rigid control which exerts fixed forces, compliant control enables a robot to adjust its force exertion in real-time based on environmental feedback. This feedback includes disturbances, contact forces, and uncertainties. In the context of Reinforcement Learning (RL), an agent can leverage compliant control to learn a policy that adapts its actions to maximize rewards, while also accommodating variations in the environment and adjusting to dynamic conditions.

In order to achieve variable stiffness control, a Cartesian impedance controller in the robot end-effector frame is used. We use a dynamical robot model defined as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau_u + \tau_{ext} \tag{1}$$

in which $M(q)$ is a symmetric, positive definite inertia matrix, $C(q,\dot{q})$ is the Coriolis matrix, $g(q)$ contains the gravity torques, τ_u represents the joint torques, and τ_{ext} is the external torques applied on the robot. Following the impedance control law [40], the operational space formulation becomes

$$\begin{aligned} \tau_u(t) &= J(q)^T F_u(t) \\ &= J(q)^T(-K(e(t) - D\dot{e}(t))) \end{aligned} \tag{2}$$

where $F_u(t)$ is the input wrench, $J(q)$ is the Jacobian matrix, $e(t)$ is the pose error, and $\dot{e(t)}$ is the velocity error. Additionally, K and D represent the stiffness and damping matrices, and both are positive, definite, symmetric matrices.

Reinforcement Learning aims to learn a policy π that maximizes the expected sum of rewards obtained from interactions with an environment. It is typically modeled as a Markov Decision Process (MDP) and is defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \rho, \gamma)$. In this context, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition probability function such that $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, $\mathcal{R}$ is the reward function such that $\mathcal{S} \times \mathcal{A} \to r$, ρ denotes the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor regulating the balance between immediate and long-term rewards.

The agent's objective is to find an optimal policy π^* that maximizes the expected return, defined by the sum of rewards $\mathcal{R}$ discounted by γ:

$$R^{\pi} = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$$

To estimate the value of state-action pairs (Q-values), the agent uses the Bellman equation:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[R^{\pi} | s_t, a_t]$$

Accordingly, the optimal policy (π^*) can be found by selecting actions that maximize the expected cumulative reward:

$$\pi^*(a|s) = \arg\max_a Q^*(s, a)$$

Sequential Manipulation Frameworks that use reinforcement learning generally follow a standardized structure. The current state-of-the-art framework is Manipulation Primitive-Augmented Reinforcement Learning (MAPLE), which leverages a hierarchical policy with a library of behavior primitives [25]. Drawing inspiration from this approach, our framework similarly employs hierarchical reinforcement learning with pre-defined behavior primitives. Additionally, we adopt a Soft Actor-Critic (SAC) architecture [41] due to its ability to output continuous values.

The policy structure consists of a high-level task policy $\pi_{primitive}$ and a low-level parameter policy π_{param}. Both policies receive an observation containing information regarding the state of the environment and the robot, with π_{param} additionally taking in the output of $\pi_{primitive}$. The high-level policy, implemented as a neural network, selects a primitive based on the observation. In contrast, the low-level policy has a separate neural network for each primitive and aims to predict the parameters for the chosen primitive. The framework is depicted in Figure 2.

We frame the sequential decision-making problem as a Parameterized Action MDP (PAMDP) [42]. At each time step, $\pi_{primitive}$ selects and executes a parameterized behavior primitive p_n from a library of primitives $\mathcal{L} = \{p_1, p_2, ..., p_n\}$. Each primitive is characterized by a function $f_n(s, x)$ in which s represents the current state of the robot while x represents the parameters outputted by π_{param}. This function initiates a closed-loop control sequence over a finite time horizon, whose length is determined by the number of *atomic actions* needed to execute the selected primitive. These atomic actions are essentially short motions that cannot be further subdivided.

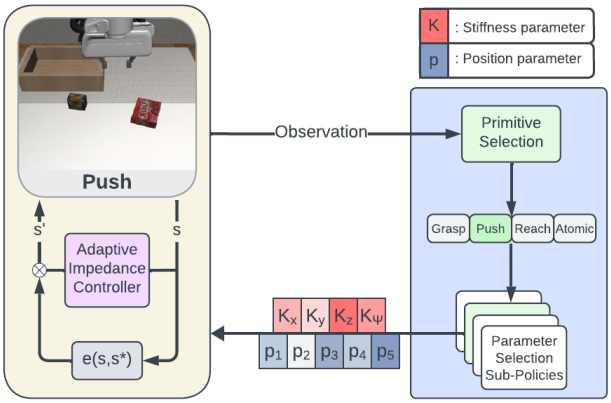

Fig. 2: An overview of the our framework highlighting the extended parameter space and an adaptive impedance controller

During primitive execution, the control loop aims to minimize the error between the current state, defined by s, and the target state, defined by the parameters x. For instance, the agent receives an observation and accordingly selects a grasping primitive. Subsequently, this primitive initiates a closed-loop control sequence to guide the end-effector toward specific coordinates (determined by x), then closes the gripper.

During learning, MAPLE incorporates *affordances* to improve exploration and incentivize desired behaviors, which is a common practice in the existing literature [43][44][45]. A typical affordance is position-based where executing a primitive around an object of interest leads to higher affordance rewards, promoting exploration in the proximity of relevant objects. While existing approaches have predominantly relied on position affordances, our work extends this to maximize compliance whenever possible (discussed in Section IV-B).

IV. Methodology

In the pursuit of enhancing the capabilities of robotic manipulation, we draw inspiration from the state-of-the-art frameworks and build upon components established in Section III. Our proposed framework is represented in Figure 2. We introduce three elements that allow us to achieve variable stiffness control for sequential contact:

- Extending the action space to allow for stiffness parameter selection (Section IV-A)
- Using an affordance that encourages compliance (Section IV-B)
- Introducing an adaptive stiffness controller during primitive execution (Section IV-C).

A. Extending the Action Space

As explained in Section III, primitives consist of closed-loop controllers that execute pre-defined behaviors. When a high-level policy selects a primitive to execute, the low-level policy determines the primitive parameters which specify a target state (or sequence of target states). To accommodate contact-rich environments, the target states need to be extended from

TABLE I: Description of primitives and their parameters

Primitive	Description	Parameters
Reach	Moves the end-effector to a target location	(x, y, z, K_x, K_y, K_z)
Grasp	Moves end-effector to grasp location then activates gripper	$(x, y, z, \psi, K_x, K_y, K_z, K_\psi)$
Push	Moves end-effector to a target location, then applies a displacement δ	$(x, y, z, \delta_x, \delta_y, \delta_z, K_x, K_y, K_z)$
Atomic	Apply atomic action	$(\delta_x, \delta_y, \delta_z, K_x, K_y, K_z)$
Gripper	Open/Close binary gripper	g

exclusively position-based parameters to also include variable stiffness.

We propose augmenting the primitive parameter action space with the Variable Impedance Control in End-Effector Space (VICES) action space [8]. VICES is an action space that gives the agent control over the impedance controller parameters by allowing it to sample impedance parameters as actions. Following the formulation in Equation (2), the parameter action space is now extended to contain (K_x, K_y, K_x) to allow for variable stiffness control along different coordinate axes, as well as K_ψ for handling orientation or angular variations. It is important to note that we set the damping matrix D to have a critical damping condition, with the goal of reducing the total number of controllable parameters and guarantee behavior stability. The available primitives and their parameters in the extended action space are documented in Table I.

Nevertheless, a limitation of this approach arises from the sequential nature of decision-making: once the policy triggers a behavior primitive, it is required to wait for the primitive to complete its execution before modifying the stiffness value again. On the other hand, using an action space with dynamically adapting stiffness parameters introduces a learning challenge. Therefore, the stiffness parameters predicted by the parameter policy will act as an initial stiffness prediction which will be further adjusted using an adaptive stiffness controller. This is explained in greater detail in Section IV-C.

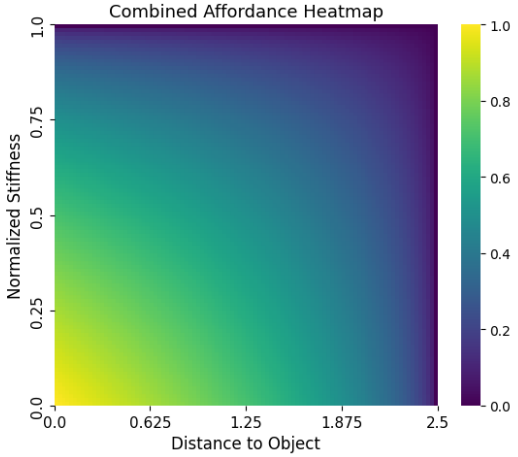

Fig. 3: Heatmap visualization of affordance coupling

B. Affordance Coupling - Combining Position and Stiffness Affordances

Despite the accelerated exploration offered by behavior primitives, our approach still encounters an exploration challenge. Accordingly, we incorporate *affordances* to incentivize the correct usage of these primitives and accelerate convergence. This means that when selecting primitive p with parameters x in a given state s, an affordance value $a(s, x; p) \in [0, 1]$ is added to the reward function. For example, executing a grasping primitive yields a higher affordance score when executed around graspable objects. This position affordance is modeled as

$$a_{\text{pos}}(s, x; p) = \max_{\kappa \in \mathcal{K}} (1 - \tanh(\max(||x_{\text{primitive}} - \kappa|| - \tau, 0))) \tag{3}$$

where $\mathcal{K}$ represents the set of object keypoints and $x_{\text{primitive}}$ is the chosen parameters for a primitive. An example can be pushing an object in which executing a pushing primitive with parameters near a pushable object's position would lead to a higher affordance score.

In the context of tasks that can benefit from variable stiffness control, these position-based affordances are insufficient since they focus exclusively on spatial information. To address this limitation, we propose *stiffness affordances* that seek to maximize compliance whenever possible. Accordingly, stiffness is only increased when it is necessary to meet task requirements. This stiffness affordance is modeled as

$$a_{\text{stiff}}(s, x; p) = 1 - \frac{K(s, x; p) - K_{\text{min}}}{K_{\text{max}} - K_{\text{min}}} \tag{4}$$

where $K(s, x; p)$ is the selected stiffness and $(K_{\text{min}}, K_{\text{max}})$ represent a pre-defined stiffness range in the action space. In practice, a_{stiff} increases linearly as stiffness decreases.

To effectively leverage both position and stiffness affordances, a geometric mean of both affordances is used to balance the two objectives. This approach leads to *affordance coupling*, which makes increments in one affordance have a more pronounced impact when the other affordance is also high. This affordance is visualized in Figure 3 and modeled as

$$a_{\text{combined}}(s, x; p) = \sqrt{a_{\text{pos}}(s, x; p) \cdot a_{\text{stiff}}(s, x; p)} \tag{5}$$

All in all, this coupling improves exploration efficiency and encourages the agent to select low stiffness parameters during the early stages of training. Furthermore, this method eliminates the necessity for careful reward weight tuning that is typically required when directly penalizing high stiffness values. Such tuning would otherwise need to be conducted for each new environment, potentially having a detrimental effect on learning performance [46].

It is also worth noting that the atomic and gripper release primitives always have an affordance score of 1 since they possess an inherent utility across different tasks.

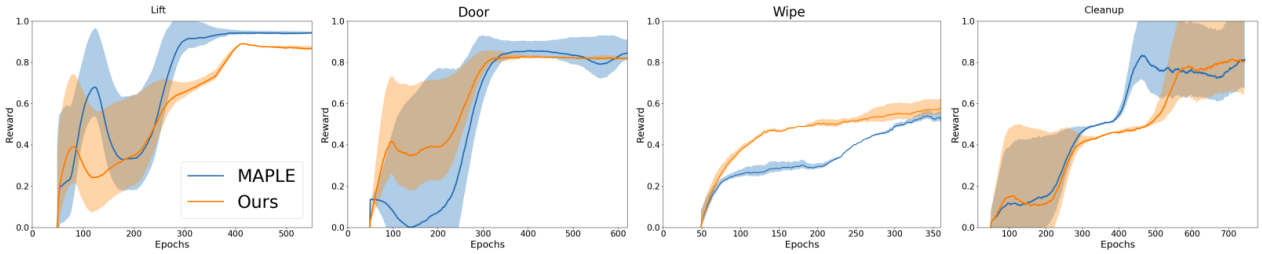

Fig. 4: Comparison of learning behavior and convergence time. A rolling mean with a window size of 50 is used to make the visualization clearer.

C. Adaptive Controller

After the policy selects a primitive and its parameters, the behavior is executed through a closed loop control scheme. Using the stiffness parameters outputted by the parameter policy as an initial estimate of the required stiffness to complete a given stage of the task, this stiffness is adapted in real-time using an adaptive stiffness controller.

1) Adaptive Controller: The adaptive controller used in this work draws inspiration from the Cartesian-Adaptive Force-Impedance Control (AFORCE) controller [47]. The controller aims to mimic the way humans adapt muscle stiffness when executing motions. This is done by adapting the stiffness in Equation (2) based on the output of

$$\dot{K}(t) = \beta|\epsilon(t)| - \gamma E \tag{6}$$

where $\epsilon(t)$ is the feedback error and E is the energy consumed by the robot joints. Tunable parameters β and γ scale these values to influence the stiffness behavior. Once the new stiffness matrix is calculated, the corresponding damping matrix satisfies a critical damping condition such that $D(t) = 2\sqrt{K(t)}$.

In practice, the controller calculates the stiffness at the next step by using β to scale the increase in stiffness proportional to the feedback error. Simultaneously, it reduces stiffness by scaling the current energy consumption E with γ. This process yields a net increase or decrease in the controller's stiffness, subsequently applied to to the controller in the next step. This behavior is visualized in Figure 5 in which an elliptical wiping motion is taking place.

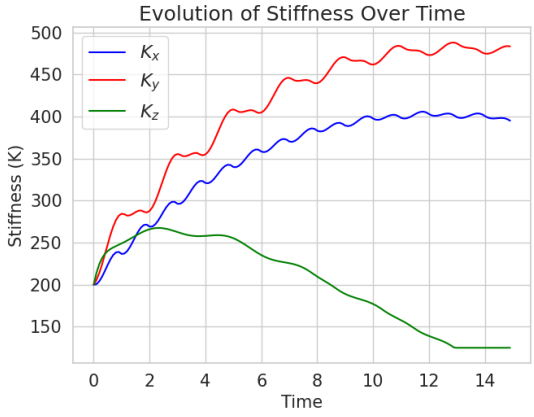

Fig. 5: An example of the adaptive stiffness behavior generated by the adaptive controller. This scenario is the robot following an elliptical wiping trajectory in simulation.

2) Acquiring Controller Parameters: The primitives in this work are simple linear motions from the current state to a target state. We perform a kinesthetic demonstration in which the end-effector is naturally move along a linear path from a random starting position towards a target state.

Following the method by Dou et al. [48], the impedance of a human arm is modeled as

$$F_h = K_h \Delta x \tag{7}$$

where F_h is the interaction force and K_h is the human arm stiffness. This force is then mapped to *normalized stiffness* using

$$K = \underline{K} + (\overline{K} - \underline{K}) \cdot \frac{(F_h - \underline{F}_h)}{(\overline{F}_h - \underline{F}_h)} \tag{8}$$

where $\overline{K}$ and $\underline{K}$ represent the upper and lower thresholds of the calculated stiffness while $\overline{F}_h$ and $\underline{F}_h$ represent the upper and lower thresholds of the interaction force.

Lastly, we find the values of β and γ by minimizing the Mean Squared Error (MSE) between the demonstration $\dot{K}(t)$ values and the values predicted by the Equation (6):

$$\min_{\beta,\gamma} \sum_{t} (\dot{K}(t) - (\beta|\epsilon(t)| - \gamma E))^2 \tag{9}$$

where the summation is over all the time points considered in the demonstration.

V. Experiments

In the experiments, our goal was to investigate the framework's learning efficiency, analyze its stiffness and force behavior, highlight patterns in primitive selection, and evaluate its performance in a real-world setting. This section is divided into four parts: Experimental Setup (V-A), Experimental Evaluation in Simulation (V-B), Experimental Evaluation in Real-World Scenarios (V-C), and Ablation Studies (V-D).

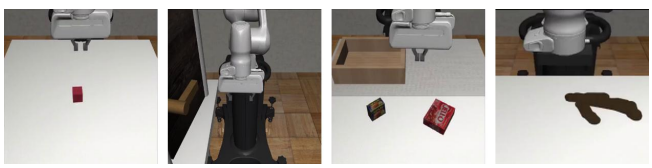

Fig. 6: Simulation Environments: Lift, Door, Cleanup, Wipe

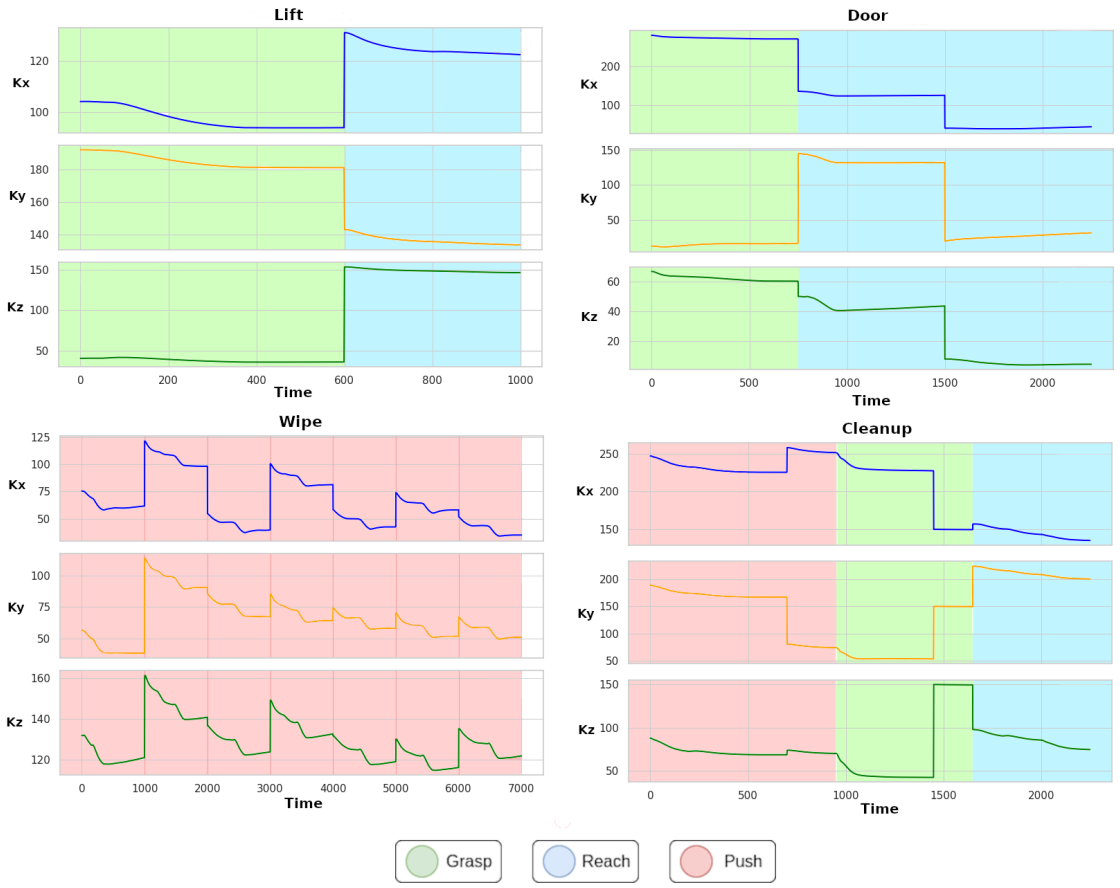

Fig. 7: Variable stiffness behavior demonstrating an emphasis on compliance and stiffness reduction

A. Experimental Setup

We evaluated our framework in four contact-rich environments: Lift, Door, Wipe, and Cleanup. These interactions include basic object manipulation in the Lift environment, continuous contact in the Door and Wipe environments, and a mix of contact and manipulation interactions in the Cleanup environment. The robot utilized for these experiments was a Franka Emika Panda, and the employed simulation framework was Robosuite [49].

We apply domain randomization to randomize table friction, table height, object positions, and initial end-effector position. This approach introduces variability into key parameters of the simulation environment to enhance the model's robustness and allows it to better generalize for real world experiments.

B. Experimental Evaluation - Simulation

We compare our proposed framework with the MAPLE baseline, both utilizing a Soft Actor-Critic architecture as detailed in Section III. As for the hyperparameters employed during training, they are documented in Appendix A. The chosen evaluation metrics are Learning Performance, Variable Stiffness Control, Compositionality, and Success Rate.

Evaluation Metrics. In *Learning Performance*, we examine learning convergence time to get an insight into the learning efficiency of the proposed framework. In *Variable Stiffness Control*, we assess our framework's ability to adapt its stiffness across diferent contexts, and the subsequent effect on the applied forces when interacting with the different environments. In *Compositionality*, we compare the compositional structure of the learned policies and quantify recurring patterns in primitive selection using a 'compositionality metric' [24]. Lastly, in *Success Rate*, we analyze the framework's ability to consistently achieve the desired task objectives across the different environments.

Evaluation Results - Learning Performance. We analyze the convergence times by referring to the learning curves found in Figure 4. Given that our approach and MAPLE use different affordances, then direct comparisons with MAPLE may not be appropriate since the reward functions are different. However, we can still assess convergence times, defined here as the time taken to learn a near-optimal policy for a given task.

Firstly, the number of epochs till convergence in the Door environment is approximately equal for both our approach and MAPLE's. Regarding the Lift and Cleanup tasks, it can be noted that the convergence time for MAPLE is slightly better than the one achieved our method. We hypothesize that the task's dependence on manipulation rather than contact led to this slower convergence time. More specifically, it is easier for MAPLE to learn the task since it has to deal with fewer primitive parameters as a result of extending the action space, as well as less constraints due to affordance coupling. In the Wiping task, it is evident that our approach converges much

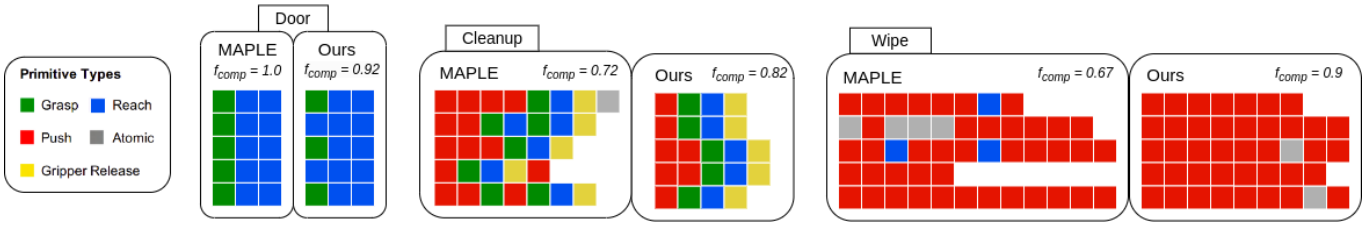

Fig. 8: Compositionality comparison showcasing the learned sequential behavior

faster than MAPLE. This can be attributed to our method's ability to leverage variable stiffness, allowing it to adapt its force behavior to match the wiping task requirements.

Evaluation Results - Variable Stiffness Control. We demonstrate samples of the variable stiffness behavior across the different environments in Figure 7. We also include a graph showing the average applied end-effector forces over a sample of 500 evaluation runs in Figure 9. These forces were acquired directly from the simulation environment.

In the Lift and Cleanup environments, both of which are tabletop settings, K_z is consistently maintained at a low level when interacting close to the table. These environments require relatively higher stiffness in the K_x and K_y values to ensure precise positioning of the end-effector for tasks like grasping objects and pushing them in the right direction.

In the Door environment, the stiffness value K_x is notably higher than K_y and K_z to provide stability during initial contact with the door. As the door handle is pushed downward, K_y gradually increases to maintain stable contact, and all stiffness values decrease when pulling the door open, as high accuracy and force control are not required.

In the Wipe environment, K_x and K_y are kept at relatively low values since the primary action involves contact with the table along the z-axis. Meanwhile, K_z maintains a relatively higher value while still exhibiting compliant behavior to exert enough force for effective stain removal without excessive interaction forces with the table.

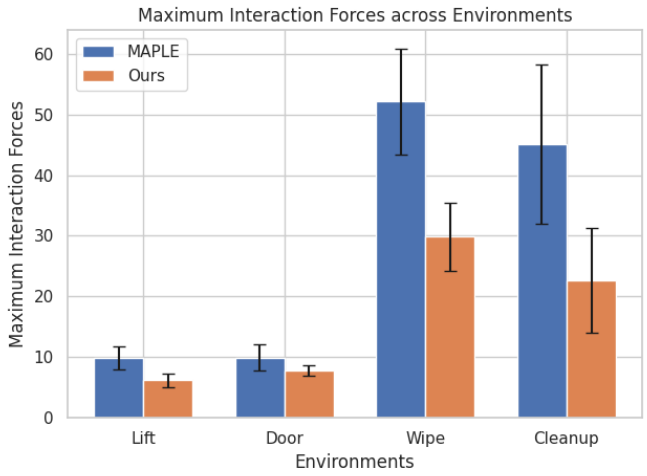

Fig. 9: Comparison of maximum interaction forces highlighting our framework's ability to finish the task while exerting less force

This decrease in stiffness also translates to less interaction forces across the different environments, which is demonstrated in Figure 9. It shows that our approach consistently exerts less force to accomplish the same tasks. Moreover, the standard deviation of applied force across these tasks is consistently lower than MAPLE, implying that our method is less sensitive to the randomization across task environments.

Evaluation Results - Compositionality. In order to quantify recurring patterns of primitive choices for solving a given task, a *compositionality metric* has been introduced by Nasiriany et al. [24]. This reflects the policy's ability to generate a well-defined and repeatable behavior sequences to complete a given task. In other words, a Lift task should consistently execute a *grasping primitive* followed by a *reaching primitive* to complete all randomized variations of this task. A detailed mathematical explanation regarding compositionality calculations can be found in the Appendix B.

The compositionality across the different environments is visualized in Figure 8. The Lift environment was excluded from this analysis as it shared the same compositionality score ($f_{comp} = 1$), consisting of a grasp and reach primitive sequence. In the Door environment, MAPLE appears to be more consistent in terms of primitive type selection, but our approach shares the same number of executed primitives. With regards to the change in primitive type, our approach successfully opens the door even without grasping the handle. This suggests that our method's variable stiffness may enable it to bypass the need to grasp the handle to establish stability while opening.

In the Cleanup environment, there is a notable reduction in the number of primitive executions to complete the task. This is attributed to a more robust ability to push on a tabletop environment, as well as higher precision when approaching an object for grasping. In the Wipe environment, our method executes primitives in a significantly more consistent manner as compared to MAPLE, implying a better understanding of the primitives needed to complete the task.

Evaluation Results - Success Rate. A comparison of the success rates between MAPLE and our method is shown in Table II. The results indicate that our approach achieves a comparable success rate across the Lift, Door, and Cleanup tasks. Notably, our method achieves *double* the success rate of MAPLE in the Wipe task, which can be attributed to the introduction of variable stiffness control.

TABLE II: Success Rates (%)

	Lift	Door	Wipe	Cleanup
MAPLE	100.0 ± 0.0	100.0 ± 0.0	42.0 ± 11.7	91.0 ± 5.8
Ours	100.0 ± 0.0	100.0 ± 0.0	86.0 ± 6.2	87.0 ± 6.1

C. Experimental Evaluation - Real World

This section discusses the experimental setup and evaluation of the Real World experiments.

Hardware Setup. An Intel RealSense D435i[1] was used to generate an RGB-D stream of the environment. These streams are later used to identify object poses. For the Lift and Cleanup experiments, we placed the camera on a tripod such that it is aligned with the tabletop. As for the Wipe environment, the camera was mounted on the robot end-effector to provide it with an accurate view of the stains.

Software Setup. ROS Noetic was used to interface between the cameras, trained model, and the robot. A RealSense ROS Wrapper[2] was used to extract the RGB-D stream from the camera. In turn, we used Deep Object Pose [50] to estimate the 6D pose of the objects in the environment. Further details regarding observation acquisition are provided in Appendix C.

Robot Control. The impedance controller used was the *human-friendly controller*[3] developed by Franzese et al. [51]. Since our model only outputs stiffness parameters and target positions, we used these parameters as input to the impedance controller. In turn, the controller acts as an interface with the robot to actuate its joints and reach the target position.

Success Rate. Each experiment was run 20 times with randomized object placements and end-effector starting positions. The model achieved a success rate of 90% on the Lift task, 80% on the Cleanup task, and 70% on the Wipe task.

D. Ablation Studies

We conduct ablation studies to measure the impact of the added components on the performance of our system. Specifically, we trained a model on the Wipe environment due to the extensive contact nature of the task. Accordingly, we investigate 3 cases, each of which omits components of the proposed framework. The results are visualized in Figure 10 and are representative of the overall performance across all environments.

- Case 1: Extended action space *with Adaptive Controller*
- Case 2: Extended action space *with Affordance Coupling*
- Case 3: Extended action space

Evaluation Results - Convergence Time. The convergence time results in Figure 10 clearly reflect that the extension of the action space with stiffness parameters is the greatest contributor to the accelerated learning. On the other hand, the exclusive use of an adaptive controller (Case 1) or affordance coupling (Case 2) leads to a notable deterioration in learning performance, as compared to the use of both (Ours).

[1]https://www.intelrealsense.com/depth-camera-d435i/
[2]https://github.com/IntelRealSense/realsense-ros
[3]https://github.com/franzesegiovanni/franka_human_friendly_controllers

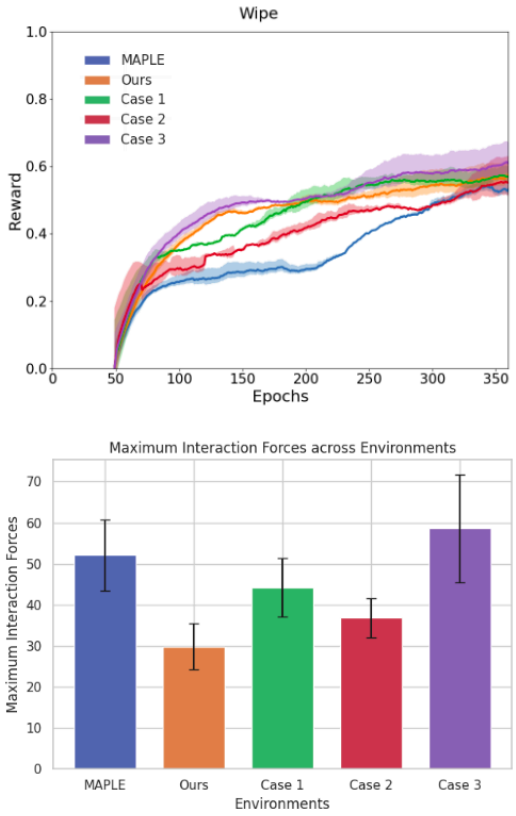

Fig. 10: Comparison of convergence time and maximum interaction forces across 3 ablation cases

Evaluation Results - Maximum Interaction Forces. Figure 10 presents the maximum interaction forces achieved through variable stiffness in different frameworks. Our proposed approach consistently minimizes these forces during environmental interactions. This is followed by the framework using only an adaptive controller (Case 1), where stiffness reduction takes place in a relatively narrower range. In turn, this leads to a relatively higher force exertion. Lastly, exclusive reliance on an extended action space yields the worst performance due to a lack of incentive to reduce stiffness, so the agent opts for high stiffnesses to ensure stain removal.

Evaluation Results - Stiffness Behavior. Upon removing affordance coupling from the framework (Case 1), the agent exhibits a dependence on high stiffness values, which are subsequently reduced using the adaptive controller. As for the case that employs affordance coupling but omits the adaptive controller (Case 2), the agent tends to select relatively low stiffness values, but the profile remains static. Additionally, the absence of corrective behavior leads the agent to attempt corrections during the execution of the next primitive rather than concurrently with the current one. Upon removing both affordance coupling and the adaptive controller (Case 3), the stiffness profile becomes static, and the agent tends to select high stiffness values due to a lack of incentive for reduction.

VI. Conclusion

This paper presents a hierarchical reinforcement learning framework aimed at enabling adaptive stiffness control in

sequential contact tasks. It utilizes a pre-defined library of behavior primitives and equips them with variable stiffness capabilities. This was done by incorporating two important elements in the framework: an expanded action space to allow the agent to modify its stiffness and an adaptive controller for dynamic stiffness modifications during primitive execution. During training, we introduce affordance coupling to combine position and stiffness affordances, which promotes efficient exploration while incentivizing compliance. The framework showcases notable results in learning efficiency, variable stiffness control, compositionality in primitive selection, and success rates when compared to MAPLE, a state-of-the-art framework in sequential planning. Furthermore, real-world evaluations validate the proposed approach's sim2real capability. Interesting directions for future research involve extending the framework with contact-specific primitives tailored for challenging tasks, such as screwing and flipping. Another area for exploration involves learning behavior primitives and impedance profiles through demonstrations rather than relying on pre-defined primitives.

ACKNOWLEDGMENT

I extend my sincere appreciation to my supervisors for their invaluable support and guidance. Your feedback have inspired me to surpass my boundaries and deepen my passion for robotics even further.

I would also like to express my deepest gratitude to my parents for their encouragement and trust throughout this journey. I am forever grateful for the love, patience, and understanding you have provided me. A heartfelt acknowledgment is reserved for my brother, whose presence has been a constant source of strength and inspiration. Thank you for consistently reminding me that actions speak louder than words.

Lastly, a special thanks goes out to my friends for showing me that a lifetime of cultural differences is not a sufficient barrier to building enduring connections.

REFERENCES

[1] A. Ranjbar, N. A. Vien, H. Ziesche, J. Boedecker, and G. Neumann, "Residual feedback learning for contact-rich manipulation tasks with uncertainty," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2383–2390.

[2] Q. Li, M. Meier, R. Haschke, H. Ritter, and B. Bolder, "Object dexterous manipulation in hand based on finite state machine," in *2012 IEEE International Conference on Mechatronics and Automation*. IEEE, 2012, pp. 1185–1190.

[3] S. Nguyen, O. Oguz, V. Hartmann, and M. Toussaint, "Self-supervised learning of scene-graph representations for robotic sequential manipulation planning," in *Conference on Robot Learning*. PMLR, 2021, pp. 2104–2119.

[4] Z. Zhao, Z. Zhou, M. Park, and Y. Zhao, "Sydebo: Symbolic-decision-embedded bilevel optimization for long-horizon manipulation in dynamic environments," *IEEE Access*, vol. 9, pp. 128 817–128 826, 2021.

[5] M. M. Botvinick, "Hierarchical reinforcement learning and decision making," *Current opinion in neurobiology*, vol. 22, no. 6, pp. 956–962, 2012.

[6] D. W. Franklin, G. Liaw, T. E. Milner, R. Osu, E. Burdet, and M. Kawato, "Endpoint stiffness of the arm is directionally tuned to instability in the environment," *Journal of Neuroscience*, vol. 27, no. 29, pp. 7705–7716, 2007.

[7] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5844–5850.

[8] 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 *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1010–1017.

[9] F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Force-based variable impedance learning for robotic manipulation," *Robotics and Autonomous Systems*, vol. 109, pp. 156–167, 2018.

[10] S. Dou, J. Xiao, W. Zhao, H. Yuan, and H. Liu, "A robot skill learning framework based on compliant movement primitives," *Journal of Intelligent & Robotic Systems*, vol. 104, no. 3, p. 53, 2022.

[11] I.-A. Gal, A.-C. Ciocîrlan, and M. Mărgăritescu, "State machine-based hybrid position/force control architecture for a waste management mobile robot with 5dof manipulator," *Applied Sciences*, vol. 11, no. 9, p. 4222, 2021.

[12] Y. Onishi and M. Sampei, "Priority-based state machine synthesis that relaxes behavior design of multi-arm manipulators in dynamic environments," *Advanced Robotics*, vol. 37, no. 5, pp. 395–405, 2023.

[13] K. French, S. Wu, T. Pan, Z. Zhou, and O. C. Jenkins, "Learning behavior trees from demonstration," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7791–7797.

[14] F. Rovida, B. Grossmann, and V. Krüger, "Extended behavior trees for quick definition of flexible robotic tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6793–6800.

[15] S. Cheng and D. Xu, "Guided skill learning and abstraction for long-horizon manipulation," *arXiv preprint arXiv:2210.12631*, 2022.

[16] C. Agia, T. Migimatsu, J. Wu, and J. Bohg, "Taps: Task-agnostic policy sequencing," *arXiv preprint arXiv:2210.12250*, 2022.

[17] B. Wu, S. Nair, L. Fei-Fei, and C. Finn, "Example-driven model-based reinforcement learning for solving long-horizon visuomotor tasks," *arXiv preprint arXiv:2109.10312*, 2021.

[18] Y. Liu, D. Romeres, D. K. Jha, and D. Nikovski, "Understanding multi-modal perception using behavioral cloning for peg-in-a-hole insertion tasks," *arXiv preprint arXiv:2007.11646*, 2020.

[19] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5628–5635.

[20] B. Wu, F. Xu, Z. He, A. Gupta, and P. K. Allen, "Squirl: Robust and efficient learning from video demonstration of long-horizon robotic manipulation tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9720–9727.

[21] J. Liang, B. Wen, K. Bekris, and A. Boularias, "Learning sensorimotor primitives of sequential manipulation tasks from visual demonstrations," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8591–8597.

[22] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "Learning to generalize across long-horizon tasks from human demonstrations," *arXiv preprint arXiv:2003.06085*, 2020.

[23] D.-A. Huang, S. Nair, D. Xu, Y. Zhu, A. Garg, L. Fei-Fei, S. Savarese, and J. C. Niebles, "Neural task graphs: Generalizing to unseen tasks from a single video demonstration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8565–8574.

[24] S. Nasiriany, H. Liu, and Y. Zhu, "Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7477–7484.

[25] M. Dalal, D. Pathak, and R. R. Salakhutdinov, "Accelerating robotic reinforcement learning via parameterized action primitives," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 847–21 859, 2021.

[26] C. Agia, T. Migimatsu, J. Wu, and J. Bohg, "Stap: Sequencing task-agnostic policies," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7951–7958.

[27] Y. Zhou, M. Do, and T. Asfour, "Learning and force adaptation for interactive actions," in *2016 IEEE-RAS 16th international conference on humanoid robots (humanoids)*. IEEE, 2016, pp. 1129–1134.

[28] B. Nemec, F. J. Abu-Dakka, B. Ridge, A. Ude, J. A. Jørgensen, T. R. Savarimuthu, J. Jouffroy, H. G. Petersen, and N. Krüger, "Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile," in *2013 16th International Conference on Advanced Robotics (ICAR)*. IEEE, 2013, pp. 1–7.

[29] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *2009*

IEEE International Conference on Robotics and Automation. IEEE, 2009, pp. 763–768.

[30] T. Cederborg, M. Li, A. Baranes, and P.-Y. Oudeyer, "Incremental local online gaussian mixture regression for imitation learning of multiple tasks," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 267–274.

[31] Y. Li, G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schaeffer, and E. Burdet, "Force, impedance, and trajectory learning for contact tooling and haptic identification," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1170–1182, 2018.

[32] D. Mitrovic, S. Klanke, and S. Vijayakumar, "Learning impedance control of antagonistic systems based on stochastic optimization principles," *The International Journal of Robotics Research*, vol. 30, no. 5, pp. 556–573, 2011.

[33] E. A. Rückert, G. Neumann, M. Toussaint, and W. Maass, "Learned graphical models for probabilistic planning provide a new class of movement primitives," *Frontiers in computational neuroscience*, vol. 6, p. 97, 2013.

[34] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[35] J. Rey, K. Kronander, F. Farshidian, J. Buchli, and A. Billard, "Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies," *Autonomous Robots*, vol. 42, pp. 45–64, 2018.

[36] M. Kim, S. Niekum, and A. D. Deshpande, "Scape: Learning stiffness control from augmented position control experiences," in *Conference on Robot Learning*. PMLR, 2022, pp. 1512–1521.

[37] M. Bogdanovic, M. Khadiv, and L. Righetti, "Learning variable impedance control for contact sensitive tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6129–6136, 2020.

[38] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, T. Nishi, S. Kikuchi, T. Matsubara, and K. Harada, "Learning force control for contact-rich manipulation tasks with rigid position-controlled robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5709–5716, 2020.

[39] P. Kulkarni, J. Kober, R. Babuška, and C. Della Santina, "Learning assembly tasks in a few minutes by combining impedance control and residual recurrent reinforcement learning," *Advanced Intelligent Systems*, vol. 4, no. 1, p. 2100095, 2022.

[40] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The international journal of robotics research*, vol. 14, no. 1, pp. 19–36, 1995.

[41] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[42] W. Masson, P. Ranchod, and G. Konidaris, "Reinforcement learning with parameterized actions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[43] D. Xu, A. Mandlekar, R. Martín-Martín, Y. Zhu, S. Savarese, and L. Fei-Fei, "Deep affordance foresight: Planning through what can be done in the future," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6206–6213.

[44] P. Mandikal and K. Grauman, "Learning dexterous grasping with object-centric visual affordances," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6169–6176.

[45] N. Vulin, S. Christen, S. Stevšić, and O. Hilliges, "Improved learning of robot manipulation tasks via tactile intrinsic motivation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2194–2201, 2021.

[46] A. Faust, A. Francis, and D. Mehta, "Evolving rewards to automate reinforcement learning," *arXiv preprint arXiv:1905.07628*, 2019.

[47] M. Ulmer, E. Aljalbout, S. Schwarz, and S. Haddadin, "Learning robotic manipulation skills using an adaptive force-impedance action space," *arXiv preprint arXiv:2110.09904*, 2021.

[48] B. Kim, J. Park, S. Park, and S. Kang, "Impedance learning for robotic contact tasks using natural actor-critic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 2, pp. 433–443, 2009.

[49] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

[50] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[51] G. Franzese, A. Mészáros, L. Peternel, and J. Kober, "Ilosa: Interactive learning of stiffness and attractors," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7778–7785.

APPENDIX A
TRAINING & SIMULATION

A. Training Setup

The training codebase used is based on *RLkit*[4], which in turn is based on *rllab*[5]. We document all the hyperparameters used in the training procedure in Table III and IV. An important thing to add is that a target entropy is set for the first 200 epochs primarily to promote exploration for both the primitives and the stiffness parameters.

Fig. 11: Simulation Environments: Lift, Door, Wipe, Cleanup

With regards to the observations used to train the model, the same observation is shared across all environments except Wipe. In those environments, the observations consist of:

- Cartesian Pose
- Object Poses
- Distance from End-Effector to Object(s)
- Gripper State (either 0 or 1)

As for the Wipe environment, the observation becomes:

- Cartesian Pose
- Percentage Wiped
- Stain Centroid and Radius
- Distance from End-Effector to Centroid

B. Simulation Setup

Here, we specify the description of each task setup and specify their success conditions:

- Lift:
  - **Description**: There is a single cube on a tabletop
  - **Success Condition**: The cube is lifted above a height threshold (20 cm)
- Door:
  - **Description**: There is a hinged door with an L-handle
  - **Success Condition**: The handle exceeds a certain position (15 cm) and angle (30°)
- Cleanup:
  - **Description**: There is a jello box, a spam can, and a wooden box on a tabletop
  - **Success Condition**: The jello box is at a threshold distance from the table corner (10 cm) and the spam can is in a wooden box
- Wipe:
  - **Description**: There are stains on a tabletop, which are defined by their table coverage percentage (40%) and stain line width (4 cm)
  - **Success Condition**: There are no stains on the table

[4]https://github.com/rail-berkeley/rlkit
[5]https://github.com/rll/rllab

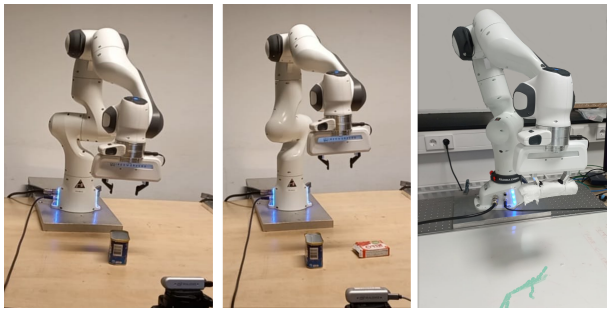

Fig. 12: Real World Experimental Setup: Lift, Cleanup, Wipe

APPENDIX B
COMPOSITIONALITY CALCULATION

Compositionality is an important metric used to evaluate the consistency of primitive selection to solve a given task. This highlights that the policy has learned a repeatable behavior and understands the underlying semantics of the task.

Next, the *Levenshtein distance* is calculated, which represents the minimum number of edits needed to make the two token sequences equal. In other words, longer distances signify less compositional behavior, and vice versa. Consequently, the compositionality of each task is calculated as

$$f_{\text{comp}} = \frac{1}{n(n-1)} \sum_{i \neq j} 1 - \frac{L(K_i, K_j)}{\max(|K_i|, |K_j|)} \tag{10}$$

where n is the combined length of the primitive sequence, $\max(|K\_i|, |K\_j|)$ is the length of the longer primitive sequence, and $L(K\_i, K\_j)$ is the Levenshtein distance between two sequences.

APPENDIX C
REAL WORLD EXPERIMENTS - OBSERVATIONS

As mentioned in Appendix A, the model uses object poses as part of the observation. In order to track the 6D pose of the environment objects in the real world, we use Deep Object Pose[6] with the corresponding YCB objects[7] used in simulation. It is important to note that the wiping task naturally does not involve interactions with solid objects, so we used a simple k-means clustering algorithm to identify the wiping stains based on color.

[6]https://github.com/NVlabs/Deep_Object_Pose
[7]https://ycbbenchmarks.com/object-set/

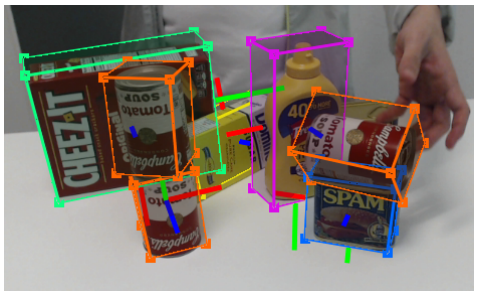

Fig. 13: 6D pose estimation of YCB object set [50]

TABLE III: Network and Optimization Parameters

Hyperparameter	Value
Network Structure (All Networks)	512, 512
Q network and policy activation	ReLU
Q network output activation	None
Policy network output activation	tanh
Optimizer	Adam
Batch Size	1024
Learning rate (all networks)	3×10^{-5}
Target network update rate τ	1×10^{-3}

TABLE IV: Training, Exploration, and Reward Factors

Hyperparameter	Value
Discount Factor	0.99
Replay Buffer Size	1×10^{6}
Reward Scale	5.0
Affordance Score Scale λ	10.0
Number of Training Steps per Epoch	1000
Number of Exploration Actions per Epoch	3000
Horizon Length per Episode	150 actions (except wipe, 300)

In the real-world experiments, the observations for the trained model were obtained using the Franka ROS Interface and the Intel RealSense D435i camera. More specifically, the process entails the following:

- **Cartesian Pose** was extracted directly from the Franka ROS Interface. This information includes the position and orientation of the end-effector in the robot's workspace.
- **Gripper State** was extracted directly from the Franka ROS Interface. The gripper width was used to identify whether it was open or closed.
- **Object Pose** was estimated using Deep Object Pose with the Intel RealSense D435i camera. Using the RGB-D stream, Deep Object Pose analyzes the data and returns 6D object poses at a rate of 15 frames per second.
- **Distance from End-Effector to Object** was calculated directly given that we have both poses

As for the Wipe environment, there are two unique observation elements. First, K-Means clustering was used on the RGB stream, which was followed by color thresholding. This allows us to separate the black stains from the white background. Accordingly, the observations were acquired using the following methods:

- **Percentage Wiped** was calculated using the initial stain as a template. By counting the number of black pixels, we can identify how many have been removed, which corresponds to the percentage wiped.
- **Stain Centroid and Radius** were acquired by pairing the RGB and Depth stream, which allows us to identify the location of the centroid and the radius of the stain.