2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Learning Generalizable Manipulation Policy with Adapter-Based Parameter Fine-Tuning

Kai Lu[1], Kim Tien Ly[2], William Hebberd[2], Kaichen Zhou[1], Ioannis Havoutis[2], Andrew Markham[1]

***Abstract*— This study investigates the use of adapters in reinforcement learning for robotic skill generalization across multiple robots and tasks. Traditional methods are typically reliant on robot-specific retraining and face challenges such as efficiency and adaptability, particularly when scaling to robots with varying kinematics. We propose an alternative approach where a disembodied (virtual) hand manipulator learns a task (i.e., an abstract skill) and then transfers it to various robots with different kinematic constraints without retraining the entire model (i.e., the concrete, physical implementation of the skill). Whilst adapters are commonly used in other domains with strong supervision available, we show how weaker feedback from robotic control can be used to optimize task execution by preserving the abstract skill dynamics whilst adapting to new robotic domains. We demonstrate the effectiveness of our method with experiments conducted in the SAPIEN ManiSkill environment, showing improvements in generalization and task success rates. All code, data, and additional videos are at this GitHub link: https://kl-research.github.io/genrob.**

I. Introduction

Learning generalizable robotic skills is a significant challenge in embodied intelligence, which includes generalization across objects, tasks, and robots. While vision-based object generalization has been extensively studied [1]–[4], generalization across different robots and task trajectories remains relatively underexplored. This capability has a wide impact, enabling robots to efficiently learn new skills or adapt existing skills to similar domains. However, different robots usually have varying kinematic configurations and morphologies, such as body structure and joint limits, leading to different physical constraints and dynamic properties, posing challenges to skill generalization.

Traditional skill learning methods often consider retraining for new tasks on a specific single robot, such as by using reinforcement learning (RL) [5] or imitation learning [6]. However, reinforcement learning often encounters problems with sampling efficiency, and imitation learning struggles to find perfectly corresponding robot samples. Recent works designed skill generalization methods across multiple robots and tasks using large robot learning datasets [7], [8] and foundation models [9], [10], which requires significant computational resources. Another approach is the use of hierarchical or modular network designs [11]–[15], including aligning internal features [11], encoding robotic morphological information [12], and sharing modular policies [13], [14].

[1]: K. Lu, K. Zhou and A. Markham are with the Department of Computer Science, University of Oxford. Email: {kai.lu, rui.zhou,andrew.markham}@cs.ox.ac.uk

[2]: K. T. Ly, W. Hebberd and I. Havoutis are with the Oxford Robotics Institute, University of Oxford, Oxford, UK. Email: {ktien, william.hebberd,ioannis}@robots.ox.ac.uk

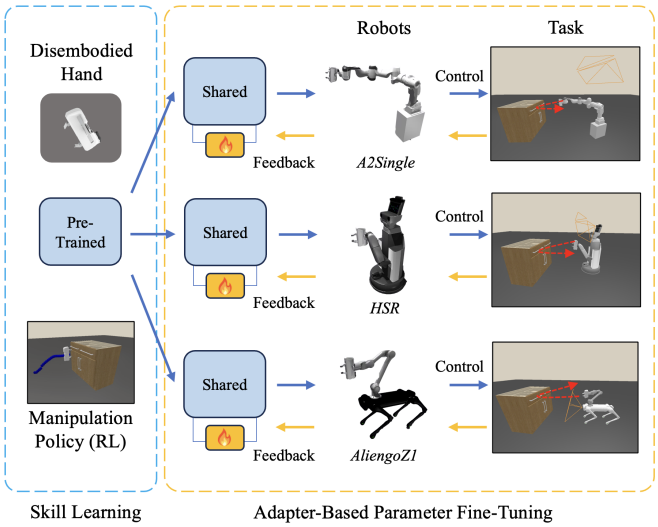


Fig. 1. **Demonstration of the proposed method.** In this work, we study the problem of using adapter-based fine-tuning on pre-trained policy models for generalizable manipulation across different robotic platforms. We teach a disembodied hand to learn tasks like opening a drawer and transfer these skills to a whole-body robot, accounting for the robot's specific constraints.

However, they often focus on simplified robotic morphologies, such as 2D arms [13] or omnidirectional spherical hands [15], or limited tasks like grasping [12]. In this paper, we aim to explore how a shared global skill policy can effectively be applied on multiple high-degree-of-freedom (high-DoF) mobile robot platforms.

Our key innovation is to teach an unconstrained, disembodied hand manipulator to learn a skill, such as opening a drawer or cabinet, and then transfer this skill to a constrained whole-body robot. We consider these constraints as the feasibility of the disembodied hand's trajectory on a specific robot. As the disembodied hand policy is not optimized for the specific robot's constraints and kinematic properties, the trajectories generated by this policy are not optimal or even unfeasible on new whole-body robots. For instance, a robot with more DoFs and longer arms will have greater adaptability, while smaller robots may find it more difficult to follow the poses generated online by RL. To mitigate this issue, we investigate fine-tuning the policy network and introducing robotic control feedback for RL optimization.

Recently, parameter-efficient fine-tuning (PEFT), such as the Adapter technique [16]–[20], has proven effective for pre-trained model fine-tuning and is widely used for generalization in natural language processing [16] and visual generation fields [17]. It achieves specific domain adaptation by inserting additional trainable layers into the existing model. This method allows the model to adapt to new domains with a small number of extra parameters while keeping the original

parameters unchanged, which is advantageous for transfer learning and multi-task learning. Inspired by this, we propose the integration of adapters into the learning of generalizable robotic skills and explore the use of robot feedback in RL to learn these adapters. This approach enables the adaptation to new robots or tasks without the necessity of retraining the entire model. We conjecture that this method can better retain the original model's knowledge of skill dynamics while introducing an understanding of new robots or tasks.

To implement this, we introduce parallel modules into the original network, such as low-rank decomposed (LoRA) [21] adapters, or sequential modules, such as residual adapters [22]. Then, in RL training, we design a feedback reward function from the whole-body robotic control, which requires the robot to solve joint configurations using a Newton-Raphson-based Inverse Kinematics (IK) solver [23] at each step. If an unfeasible solution is returned by the IK solver before the robot completes the task, we will assign a negative reward to RL policy. Hence, the adapter learns to optimize the end-effector (EE) trajectory generated by the original RL model. In addition to generalization across robots, we further explore the application of adapter learning techniques between similar tasks. For example, how skills like pulling a drawer can be adapted and transferred to opening a door. We found that this learning method is effective for transferring robotic skill learning.

In summary, we explore the following three questions:

- How can adapters be used in robotic reinforcement learning to generalize a global skill across different robotic embodiments?
- What impacts do various adapter architectures and fine-tuning strategies have on skill generalization?
- Does adapter technology have broader application scenarios, such as domain adaptation for similar tasks?

We study the generalization of the drawer-opening skill across three mobile manipulation robots in the SAPIEN ManiSkill environment [24], including the ManiSkill A2-Single-Arm Robot (A2Single), the Unitree Aliengo robot with Z1 arm [25] (AliengoZ1), and the Toyota Human Support Robot (HSR) [26]. Our research shows that adapter learning can effectively generalize among robots with different physical constraints. Particularly, LoRA-type adapters improve the success rate on new robots by 11% on A2Single Arm, 15% on AliengoZ1, and 14% on HSR, compared to vanilla full-finetuning. Our task-variant experiments, including door opening and chair pushing tasks, indicate that adapter learning also improves generalization among tasks.

II. Related Work

A. Learning across robotic embodiments

In robotics, the challenge of policy learning across various platforms is a key topic. To address this, many studies focus on developing foundational models for robots, such as RoboCat [9], Gato [10], and RT-X [27]. The training of these models relies on extensive datasets containing diverse robots and demonstrations, including Open X-embodiment [27] and BridgeData [8]. However, these approaches demand substantial computational resources. Different from these works, another path adopts a structured approach to manipulation policy learning, using hierarchical and modular methods to bridge the differences among various robots, such as in [11]. However, these studies often narrow their focus to specific aspects like task simplification, including 2D kinematic setups [13], end-effector morphologies [15], and limited tasks like object grasping [12]. While there are related contributions in robotic navigation and locomotion [28]–[31], these works are only tangentially related to our primary focus on robotic manipulation.

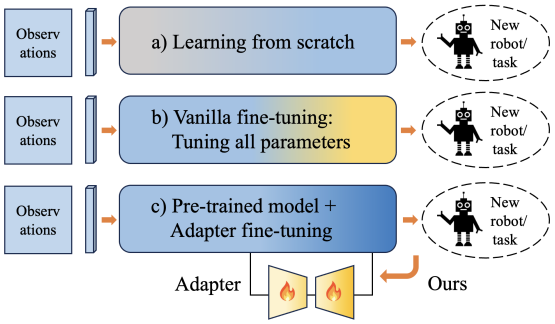


Fig. 2. **Illustration of different approaches.** Commonly seen approaches for new robot/task transfer include learning from scratch and vanilla fine-tuning. In this work, we adopt adapter learning with feedback reward in RL for skill generalization from a pre-trained policy model.

B. Generalizable manipulation policy learning

Learning generalizable manipulation skills is crucial for embodied intelligence. Numerous works in the visual domain have been proposed to address generalization among objects, including domain-invariant 3D feature distillation [2], 3D affordance learning [32], and unified representations of actionable parts [3]. Additionally, various works in visual reinforcement learning and imitation learning have been proposed to solve generalization across objects or tasks. For instance, methods that use decoupled or EE action spaces [33]–[35] or action primitives [36]–[38] have been proposed to improve the efficiency of reinforcement learning and enhance generalizability. On the other hand, modular structures [39], representational alignment [11], and adversarial generative models [4] have proven effective in generalizing across different objects and tasks. With the advent of large models, recent work has involved LLM and VLM for action and semantic matching [40] or open-ended task discovery [41], resulting in policies with improved generalizability.

C. Adapter Learning for Policy Model Fine-Tuning

Parameter-efficient fine-tuning, such as the adapter technique, is extensively utilized for domain adaptation. It was first applied in natural language processing (NLP), such as using low-rank adaptation (LoRA) [21] for fine-tuning GPT-3. Recently, adapters are widely adopted in the vision domain [42]–[44]. For instance, ViT-Adapter [42] has achieved comparable performance to vision-specific transformers. Adapters are also used in robotic imitation learning, such as LoRA-Transformer [45] and TAIL [46], which fine-tune large, task-specific models. Besides, RoboAdapters [47] utilized adapters to adjust upstream visual perception modules for manipulation tasks. Distinct from these applications, our work contemplates the use of adapters in RL for generalization across different robotic platforms (as shown in Fig. 2), and we further explore its adaptability across tasks.

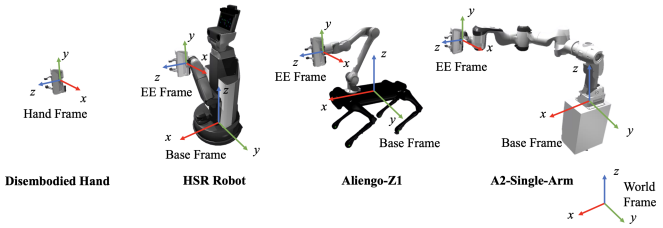


Fig. 3. **Coordinate system of robots and the disembodied hand**. We assess three different mobile manipulators equipped with the same end-effector but featuring varying kinematic configurations.

III. Methodology

Our methodology focuses on transferring a learned skill from a disembodied hand manipulator to different whole-body robotic platforms through the integration of adapters. The adapters serve a dual purpose: fine-tune the model to fit the specific kinematic constraints of a new robot whilst maintaining the integrity of the original learned skill.

A. Problem Statement

Robotic manipulation policy learning can be formulated as a Markov decision process (MDP), which is represented as (S,A,R,T,γ), where S is the set of states, A is the set of actions, $R(s_t,a_t,s_{t+1})$ is the reward function, $T(s_{t+1}|s_t,a_t)$ is the transition function as a probability distribution, and γ is the discount factor for the future rewards. The agent policy $\pi(a|s)$ is the action selecting probability under a given state s. The goal of RL is to maximize the return under the policy $G_\pi = \mathbb{E}_\pi[\sum_t \gamma^t R(s_t,a_t,s_{t+1})]$. In robot learning tasks, we usually need to estimate the task-relevant states from observation O, regarded as $s = f(o)$. This setting is viewed as a partially observable Markov decision process (POMDP) where the policy is $\pi(a|f(o))$.

In this work, we study the problem of generalizing skills across robotic mobile manipulators, where the state space being $s = [s_{obj}, s_{rob}]$ and the action space being $a = [v_{ee}, q_{jaw}]$. We use a shared action space across robots and equip them with the same two-parallel-jaw hand, as shown in Fig. 3.

B. Reinforcement Learning with Disembodied Hand

We first exploit the disembodied hand as an RL agent to learn the abstract skill dynamics, whose DoFs are three virtual prismatic joints and three virtual revolute joints. The action space includes six desired velocities of the virtual joints $v_{ee} \in \mathbb{R}^6$ and two desired positions of the finger joints $q_{f,d} \in \mathbb{R}^2$. For the cabinet environment in ManiSkill [24], we use $s_{obj} = [s_{cab}, s_{link}, s_{hdl}, s_{size}]$, where s_{cab} is the base link pose of the loaded cabinet, s_{link} and s_{hdl} are the current poses and the full poses (i.e., the poses when the drawer is fully opened) of the target drawer link and the handle, and s_{size} is the full length and the opening length of the target drawer. The poses are all represented as world frame coordinates and quaternions. In RL training, $s_{rob} = s_{ee}$ includes the hand joints' positions and velocities. We follow the dense reward function designed in ManiSkill to train the RL model:

$$R_{ms} = \begin{cases} R_{stg} + R_{ee}, & d > d_{ths}, \\ R_{stg} + R_{ee} + R_{link}, & d < d_{ths},\ c < c_{open}, \\ R_{stg} + R_{ee} + R_{link} + R_{stc}, & d < d_{ths},\ c > c_{open}, \end{cases} \tag{1}$$

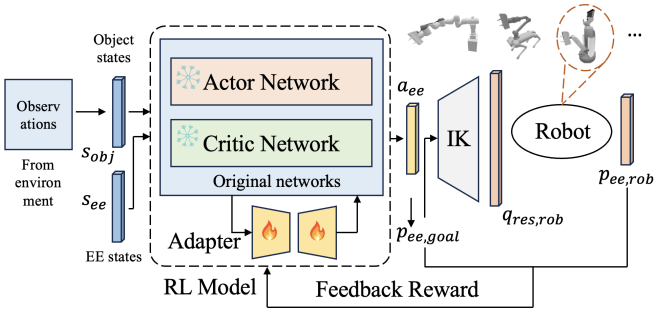


Fig. 4. **Pipeline of our method.** We integrate the adapter module into the RL model and introduce a feedback reward function from the whole-body robotic control. Through this way, the adapter learns to optimize the EE trajectory generated by the original RL model for robot-level generalization.

where R_{stg} increases from the first stage to the final and the stage is defined by the distance between EE and the handle of the target drawer $d \in \mathbb{R}$ and the opening extent of the target drawer $c_{open} \in [0,1]$. The reward R_{ee} encourages the fingers to get closer to the handle, R_{link} encourages the target drawer link to be manipulated to its goal, and R_{stc} expects the drawer to be static after task completion [24]. The model is trained using the soft actor-critic (SAC) [48] algorithm. We view this process as the model pre-training in this work and focus on how to fine-tune the learned policy. The policy is represented as:

$$a_{ee} = \pi_\phi(s_{obj}, s_{ee}). \tag{2}$$

C. Robotic Control Feedback Reward

Given that the RL agent is a disembodied free-floating hand that learns the dynamics of skills without considering the constraints of any specific embodied robot control, trajectories that are unfeasible for the robot may occur during skill transfer. Since different robots have different kinematic properties, it is difficult to design a unified constraint condition on the disembodied hand agent without leading to highly limited or sub-optimal trajectories. Therefore, we choose to introduce adapter techniques to fine-tune the original model, which requires us to design a feedback loop to optimize the pre-trained policy network, as shown in Fig. 4.

We first parallel the environment of the disembodied hand with that of the whole-body robot. The action $a_{ee,t}$ executed in the hand environment yields the next pose x_{t+1}. We aim to synchronize the robot's EE to the pose of the disembodied hand, which is expressed as:

$$x_{t+1} = p_{t+1} = p(q_t + \Delta q), \tag{3}$$

where $p(q)$ denotes the forward kinematics equation, and q represents the robot joint position. This desired joint position can be obtained by computing the Jacobian matrix, where:

$$p_{t+1} = p(q_{t+1}) \approx p(q_t) + J(q_t)\Delta q \tag{4}$$

therefore, the approximated value is:

$$\Delta q \approx J^+(q_t)\Delta p, \tag{5}$$

where J^+ is the pseudoinverse of the Jacobian matrix. We use an IK solver based on the Newton-Raphson method to iterate and find a numerical solution Δq_{res} subject to:

$$p(q_t + \Delta q_{res}) - x_{t+1} < k_{errtol} \tag{6}$$

For robots with high DoFs (considered to be more than 6 here), it is generally not easy to fall into a situation without any IK solutions. However, a global-searching IK solver cannot guarantee a smooth solution between the two EE positions of consecutive time steps. Therefore, we added a restriction to prevent the robot's current configuration from deviating more than k_{maxdev} along each axis. With such a setting, the IK solver iteratively operates through the Jacobian inverse technique. We use the IK-solve-nearby function from the Klampt library [23] to achieve the above setting, represented as:

$$q_{res}, z_{res} = F_{ik}(q, \Delta p, k_{maxdev}, k_{errtol}, k_{maxiter}), \tag{7}$$

where $z_{res} = 1$ when the IK has a feasible solution otherwise $z_{res} = 0$. The solution joint configuration is q_{res}.

For each RL simulation step, we perform the IK calculation for the robot and then control the robot to the solution joint positions before the next RL prediction. Thus, the EE poses that the robot cannot reach through the above IK setting are considered non-compliant with the robot's kinematic constraints, and we use a reward function:

$$R_{ik} = \begin{cases} +1, \ z_{res} = 1, \\ -1, \ z_{res} = 0. \end{cases} \tag{8}$$

This reward varies for different robots, as their kinematic parameters differ, such as joint limits and the structure and number of joints and links. Therefore, the final reward during the fine-tuning process can be expressed as:

$$R = \omega_{ms} R_{ms} + \omega_{ik} R_{ik}. \tag{9}$$

D. Adapter Modules for Parameter Fine-Tuning

When fine-tuning the policy network to a specific robot domain, the vanilla approach involves tuning all of the parameters of the original network. However, this method might overfit to the new domain, reducing the model's capability. Adapter modules offer an alternative by introducing a small number of trainable parameters to adjust the original network. A key benefit of this approach is that transferring skills among different robots only requires attaching the corresponding adapter network. This process typically leaves the original network structure and inference speed unaffected. Furthermore, the parameters of the pre-trained policy network can be shared, streamlining the integration process. Common adapter structures include parallel adapters, such as LoRA [21], and sequential adapters, such as the residual adapter [22], which can improve the model's generalizability.

The principle of LoRA is to parallel two low-dimensional matrices in the network's linear layers, represented as A and B, with B initialized to zero. LoRA hypothesizes that the rank of the parameters that need to be adjusted in the original linear layer matrix is likely not high, e.g., 1/10 of the original matrix, so the adjustment can be achieved through learning A and B. Another commonly seen adapter module is the encoder-decoder with non-linear activation functions using residual structure (ResAdapter) to connect between original layers. The scale parameter of the residual connection points is initialized to zero. Both structures are widely used to adjust networks based on MLP or Transformer architectures.

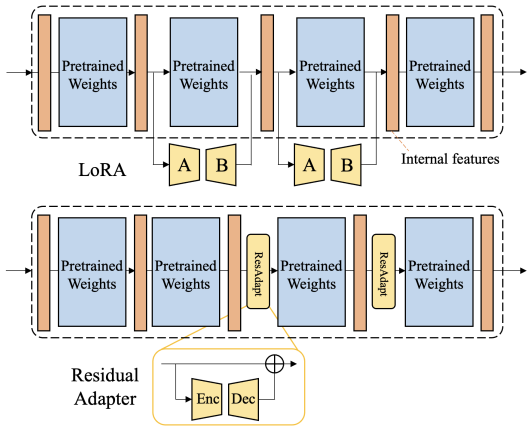


Fig. 5. **Two types of adapters incorporated in this work.** In our implementation, we incorporate the adapters in both the policy (actor) network and the Q-function network (critic) in the RL model.

In this work, we integrate either LoRA or ResAdapter into the linear layers of our MLP, as shown in Fig. 5, and then we use the above reward to finetune the networks. However, we point out that the precise choice of the adapter is not critical to our framework and that our method could relatively easily incorporate other adapters. In this work, we update the parameters of both the actor and critic networks in SAC:

$$\begin{aligned} \pi_{\phi_{nn}, \phi_{adapter}} &: \phi_{adapter} \leftarrow \phi'_{adapter}, \\ Q_{\theta_{nn}, \theta_{adapter}} &: \theta_{adapter} \leftarrow \theta'_{adapter}, \end{aligned} \tag{10}$$

where θ represents the critic (Q-function) parameters, and ϕ represents the actor (policy) parameters.

E. Adjusted Adapter for Task-Specific Constraints

In addition to skill generalization across different robots, we explore more challenging scenarios, namely applying the adapter technique to generalization across tasks. Different tasks typically have distinct trajectories, for instance, as shown in Fig. 6, where pulling a drawer and opening a door are characterized by circular and linear motions respectively, while pushing a chair is usually in the opposite direction to pulling a drawer. Due to significant changes in skill dynamics, we designed a more complex adapter module based on ResAdapter (as shown in Fig. 7) to achieve this adaptation, which can take new inputs and optimize outputs.

To accommodate new tasks, we first align the input states s_{obj} with the corresponding task. For the door opening task, we align the pose of the door handle and door link with the drawer, retaining the original position but leaving the rotation for the Adapter, represented as $\{s_{obj} \setminus s'_{obj}\}$. Additionally, we equate the door's arc length to the drawer's linear length, leaving the door's radius to the Adapter. For the pushing chair task, we map the main body link pose of the chair to the handle and link pose of the drawer but reverse the xy-plane coordinates.

The adjusted residual adapter uses a gate control for the residual connection. The gating signal $d < d_{ths}$ is the distance between the hand and the operational part e.g., the backrest or armrest of the chair. As the new skill dynamics change largely compared to the original skill, in this scenario, we allow the parameters of the original network to be fine-tuned to achieve adaptation to the new states.

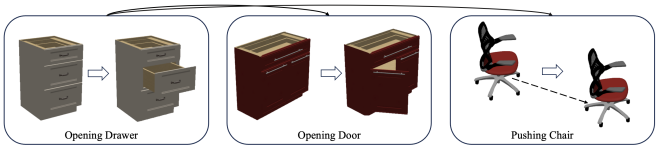


Fig. 6. **Generalization to various tasks.** We demonstrate the adapter technique for generalizing a skill across other manipulation tasks.

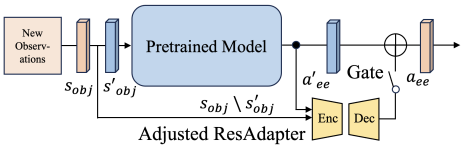


Fig. 7. **Adjusted adapter structure for task-level generalization.** We adjust the structure of ResAdapter to tune the pre-trained model for more challenging scenarios such as new tasks.

IV. Experimental Results

A. Experimental Setups

1) Environments and tasks: We conducted experiments in the SAPIEN ManiSkill simulation environment, choosing the task of opening cabinet drawers as the basic task. The criterion for task success is opening the target joint to $\geq 90\%$ of its extent and that the EE poses are feasible for the whole-body robots. For the drawer-opening task, ManiSkill provides 25 cabinets with different geometries and topologies, of which we randomly select 15 as the training set and 10 as the test set. To evaluate the skill transfer to different robots, the success criteria include the feasibility of the EE poses in the trajectory, which is validated by the Newton-Raphson-based IK solver described in Sec. III-C. We also conducted experiments for task generalization, considering the adaptation of the skill of opening drawers to opening doors and pushing chairs. The success criterion for the door-opening task is to open the specified joint to $\geq \pi/4$ radian. For the pushing chair task, the criterion is that the chair is close to the target position within 0.15 m and remains standing upright. These tasks are also episodic, with a maximum length of 200 steps (each step is 1/20 s).

2) Robotic mobile manipulators: In simulation, we present three mobile manipulation robots, as in Fig. 1:

- **Disembodied Hand**: Modeled as a floating hand with 6-DoF and equipped with two parallel jaws, using the Panda Hand as the collision model.
- **A2Single**: Modeled as an 11-DoF robot with a 4-DoF mobile base allowing for x, y, z translations and yaw rotation. It features a Scirus body and a single 7-DoF Franka Panda arm.
- **HSR**: Modeled as an 8-DoF robot with a 3-DoF mobile base for xy translation and yaw rotation, and it is equipped with a 5-DoF HSR arm.
- **AliengoZ1**: Modeled as a 9-DoF robot with a 3-DoF mobile base for xy translation and yaw rotation. Here, we follow the work of Habitat [49], assuming the base has 3 DoFs of command and ignores the leg movements. The Aliengo robot is equipped with a 6-DoF Z1 arm.

In our real-world experiments, we employ the HSR robot for sim-to-real validation. The outcome and the supplementary videos are available on our project website.

TABLE I
Cross-Robot Generalization

Robot	Method	Success ↑	Length ↓	Reward ↓
Hand	SAC	68%	115.85	-1084.91
A2Single	Direct-Transfer	32%	155.64	-1500.25
	Full-Finetune	25%	168.19	-1563.99
	LoRA Adapter	**36%**$^{\uparrow 11\%}$	**150.61**	**-1452.61**
AliengoZ1	Direct-Transfer	27%	164.98	-1652.33
	Full-Finetune	22%	169.26	-1702.40
	LoRA Adapter	**37%**$^{\uparrow 15\%}$	**150.85**	**-1503.93**
HSR	Direct-Transfer	26%	163.20	-1508.26
	Full-Finetune	23%	170.51	-1619.22
	LoRA Adapter	**37%**$^{\uparrow 14\%}$	**148.29**	**-1410.24**

3) Comparison Methods: We compare different methods of generalization to robots: **a) Direct-Transfer**: directly using the model of the trained disembodied hand agent, **b) Full-FineTune**: tuning all the parameters of the original model, **c) ResAdapter**: using the residual adapter with the original model parameters frozen, and **d) LoRA**: using the LoRA adapter with the original model parameters frozen.

B. Effectiveness of adapter learning for multi-robot transfer

To evaluate the performance of adapter learning for multi-robot skill transfer, we present the average task success rates, episode length in steps, and episode rewards of different approaches in Table I. Comparing Direct-Transfer and LoRA Adapter, we observe that introducing robot inverse kinematics control feedback reward can improve the success rate of tasks and accelerate task completion time. This indicates that RL-generated trajectories are optimized when feedback from the specific robot's kinematic constraints is included, reducing the infeasible EE poses generated by the pre-trained policy. Additionally, comparing Direct-Transfer and Full-Finetune, we find that when the new scenario considerably differs from the original training scenario, directly fine-tuning a pre-trained model may lead to overfitting. The pre-trained model may already be optimized for the specific characteristics of its original robot, and fine-tuning it on a different robot could cause the model to overly adapt to the new scenario, resulting in poor generalization on unseen data.

Furthermore, when comparing the fine-tuned model performance on different robots, we find that the adapter learning method is more effective for HSR and AliengoZ1 than for A2Single. This might indicate that they have more valid samples, i.e., infeasible poses generate corresponding feedback. Since reward in reinforcement learning is a weak supervisory signal, we consider introducing stronger signals such as auxiliary loss in the future to improve tuning efficiency. Our additional experiment on LoRA shown in Fig. 11 demonstrates that applying LoRA to a whole-body robot not only improves performance in the original domain but also facilitates transfer to other robots, showing the effectiveness of the adapter. However, the effect of LoRA on the original robot is higher than on other robots, indicating the necessity of applying adapters for each individual robot.

C. Impact of various adapters for skill generalization

Comparing the three methods of fine-tuning, we find that fine-tuning the entire original model based on constraints

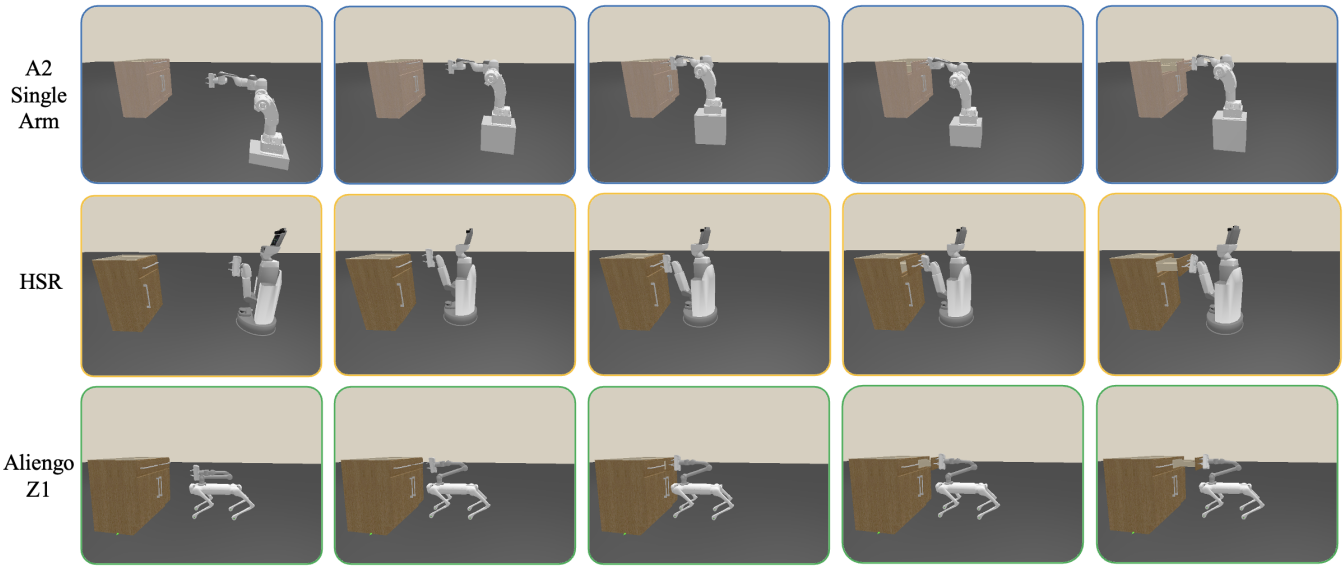


Fig. 8. **Visualization of generalizing a skill across various robotic platforms.**

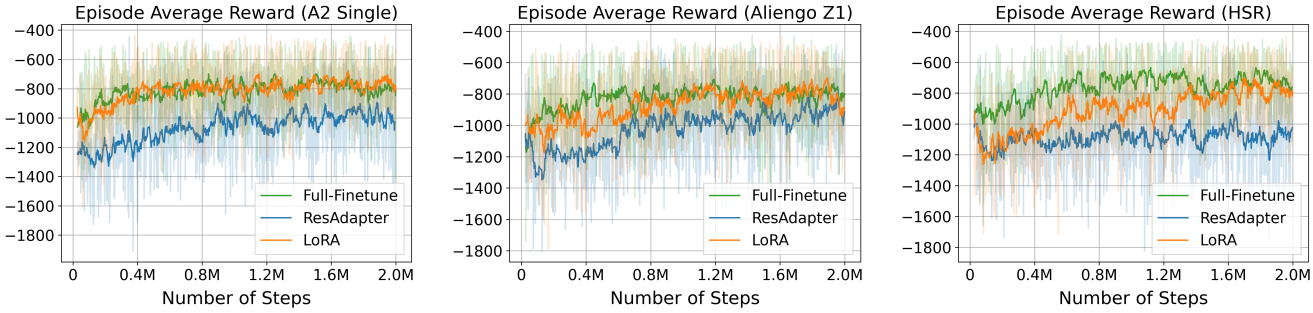


Fig. 9. **Training curve of transferring skill to three different high-DoF robots: A2 Single, Aliengo Z1 and HSR.**

TABLE II
CROSS-ROBOT GENERALIZATION

Robot	Method	Success ↑	Length ↓	Reward ↓
Hand	SAC	68%	115.85	-1084.91
A2Single	ResAdapter	28%	160.18	-1476.89
	LoRA	36%	150.61	-1452.61
AliengoZ1	ResAdapter	27%	162.07	-1558.45
	LoRA	37%	150.85	-1503.93
HSR	ResAdapter	32%	156.35	-1481.21
	LoRA	37%	148.29	-1410.24

presents high success rewards during training, as shown in Fig. 9, but relatively lower performance in the test set. This indicates that although it meets the constraints of the new robot, it leads to forgetting the ability to generalize. ResAdapter presents a better performance in the test set, as shown in Fig. 10, by adjusting the intermediate features of the network and deepening it through concatenation. LoRA overall performs the best, indicating that adjusting the parameters of the MLP linear layer in parallel can allow the network to maintain its original good skill generalization ability while meeting the specific kinematic constraints of the robot. Table II presents the performance of the above two adapter methods. In addition, we visualize the task execution by the three robots in Fig. 8, where their policies are fine-tuned using LoRA modules.

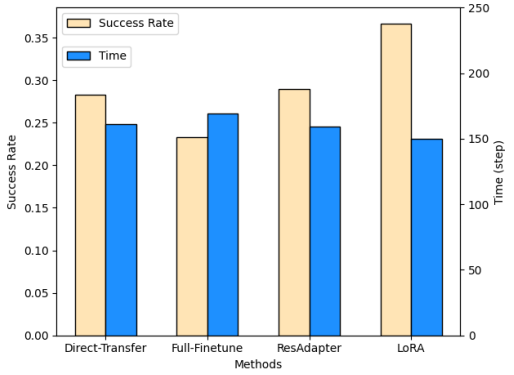
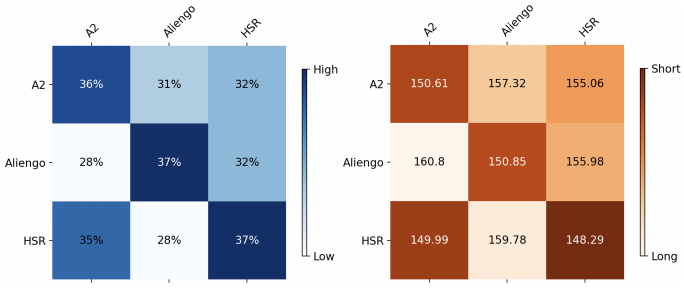


Fig. 10. **Comparison of direct transfer and different tuning methods.** The values are averaged over the three robots.



(a) Task Success Rate (b) Episode length (steps)

Fig. 11. **Cross-robot evaluation.** The adapter is trained on one robot (vertical labels), and tested on the other robot (horizontal labels).

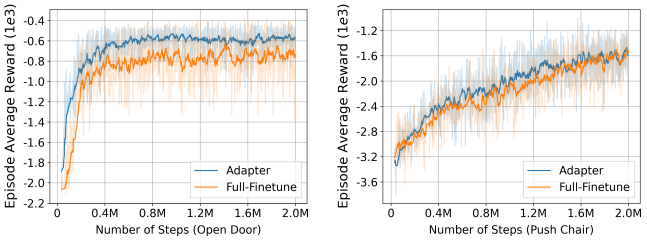


Fig. 12. **Training curves of transferring the opening drawer skill to the opening door task and the pushing chair task.**

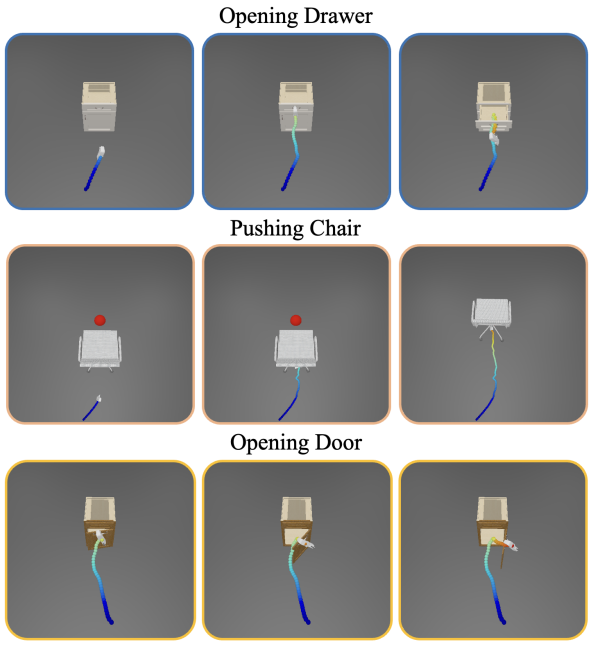


Fig. 13. **Visualization of generalizing a skill across different tasks.**

D. Adapter-Based Tuning for Cross-Task Generalization

We further explore the generalizability of the adapter technique at the task level, including transfer from opening drawers to opening doors, and from opening drawers to pushing chairs. We consider three methods to generalize the original skill: direct alignment (using the input alignment described in the methodology), fine-tuning the original network, and incorporating adapters for fine-tuning. Fig. 12 shows the training process, where using adapters for input processing and output optimization allows the network to converge faster during training and achieve higher rewards in the door-opening task compared to fine-tuning the original network with new inputs.

Table III compares the success rates on the test set, where we find that adapter-based tuning has higher task success rates, suggesting that adding adapter modules can enhance the generalizability of the RL policy across tasks. We also notice that transferring the original skill policy to pushing chairs performs better than transferring to doors opening task. This may be because the skill dynamics of the chair are easier to adapt to regarding the originally learned skill dynamics of the network. Fig. 13 shows typical scenarios for the three different tasks, from which we can see that finetuning the pre-trained model can achieve transformations of the skill dynamics.

TABLE III
CROSS-TASK GENERALIZATION

Robot	Method	Success ↑	Length ↓	Reward ↓
Opening Drawer	SAC	68%	115.85	-1084.91
Opening Door	Direct-Align	0.	200.00	-2026.08
	Full-Finetune	23%	174.76	-1807.69
	Adapter	**31%**$^{\uparrow 8\%}$	**172.09**	**-1735.80**
Push Chair	Direct-Align	3%	196.61	-3794.65
	Full-Finetune	39%	166.17	-2267.71
	Adapter	**54%**$^{\uparrow 15\%}$	**152.77**	**-1954.63**

Fig. 14. **Real-world experiment of generalization across different tasks.**

E. Real-world Experiment

In our real-world experiments, we employ the Toyota HSR robot for sim-to-real validation. Fig. 14 shows the tasks of drawer opening, door opening, and chair pushing. The robot successfully performs the mobile manipulation tasks using the fine-tuned policies from our method.

V. CONCLUSIONS

Our findings indicate that adapter learning is a simple, yet powerful strategy for generalizing robotic skills across different robots and tasks, offering a parameter-efficient alternative to full model retraining. By training first on an entirely virtual disembodied manipulator, the adapter is simply responsible for embodiment-specific adaptation. LoRA-type adapters show improvement in task success rates, supporting their potential for widespread application in robotic learning. Future research may extend these techniques to a broader range of tasks and robots, further enhancing the adaptability and efficiency of robotic systems.

REFERENCES

[1] Y. Zhu, Z. Jiang, P. Stone, and Y. Zhu, "Learning generalizable manipulation policies with object-centric 3d representations," in *7th Annual Conference on Robot Learning*, 2023.

[2] H. Geng, Z. Li, Y. Geng, J. Chen, H. Dong, and H. Wang, "Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[3] Z. Xu, Z. He, and S. Song, "Universal manipulation policy network for articulated objects," *IEEE robotics and automation letters*, vol. 7, no. 2, 2022.

[4] H. Shen, W. Wan, and H. Wang, "Learning category-level generalizable object manipulation policy via generative adversarial self-imitation learning from demonstrations," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, 2022.

[5] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," 2020.

[6] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun, "Survey of imitation learning for robotic manipulation," *International Journal of Intelligent Robotics and Applications*, vol. 3, 2019.

[7] Q. Vuong, S. Levine, H. R. Walke, K. Pertsch, A. Singh, R. Doshi, C. Xu, J. Luo, L. Tan, D. Shah *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," in *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[8] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine, "Bridgedata v2: A dataset for robot learning at scale," 2024.

[9] K. Bousmalis, e. al, and N. Heess, "RoboCat: A Self-Improving Foundation Agent for Robotic Manipulation," 2023.

[10] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, "A Generalist Agent," 2022.

[11] J. Yang, D. Sadigh, and C. Finn, "Polybot: Training One Policy Across Robots While Embracing Variability," 2023.

[12] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, "UniGrasp: Learning a Unified Model to Grasp With Multifingered Robotic Hands," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2286–2293, 2020.

[13] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017.

[14] W. Huang, I. Mordatch, and D. Pathak, "One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control," 2020.

[15] G. Salhotra, I.-C. A. Liu, and G. Sukhatme, "Learning Robot Manipulation from Cross-Morphology Demonstration," 2023.

[16] J. Guo, Z. Zhang, L. Xu, B. Chen, and E. Chen, "Adaptive adapters: An efficient way to incorporate bert into neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, 2021.

[17] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue *et al.*, "Llama-adapter v2: Parameter-efficient visual instruction model," *arXiv preprint arXiv:2304.15010*, 2023.

[18] T. Chen, L. Zhu, C. Deng, R. Cao, Y. Wang, S. Zhang, Z. Li, L. Sun, Y. Zang, and P. Mao, "Sam-adapter: Adapting segment anything in underperformed scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[19] C. Mou, X. Wang, L. Xie, Y. Wu, J. Zhang, Z. Qi, and Y. Shan, "T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 5, 2024.

[20] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, "Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models," *arXiv preprint arXiv:2308.06721*, 2023.

[21] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2021.

[22] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," *Advances in neural information processing systems*, vol. 30, 2017.

[23] K. Hauser, "Klamp't: Kris' Locomotion and Manipulation Planning Toolbox," http://klampt.org, 2022.

[24] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," in *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[25] Unitree, "Unitree Aliengo and Unitree Z1," https://www.unitree.com/.

[26] T. Yamamoto, T. Nishino, H. Kajima, M. Ohta, and K. Ikeda, "Human support robot (hsr)," in *ACM SIGGRAPH 2018 Emerging Technologies*. Association for Computing Machinery, 2018.

[27] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[28] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "GNM: A General Navigation Model to Drive Any Robot," 2023.

[29] K. T. Ly, M. Munks, W. Merkt, and I. Havoutis, "Asymptotically optimized multi-surface coverage path planning for loco-manipulation in inspection and monitoring," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2023.

[30] V. Kurin, M. Igl, T. Rocktäschel, W. Boehmer, and S. Whiteson, "My Body is a Cage: The Role of Morphology in Graph-Based Incompatible Control," 2021.

[31] K. T. Ly, V. Semenov, M. Risiglione, W. Merkt, and I. Havoutis, "R-lgp: A reachability-guided logic-geometric programming framework for optimal task and motion planning on mobile manipulators," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[32] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[33] K. Lu, B. Yang, B. Wang, and A. Markham, "Decoupling skill learning from robotic control for generalizable object manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

[34] K. Lu, J.-X. Zhong, B. Yang, B. Wang, and A. Markham, "Learning to catch reactive objects with a behavior predictor," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[35] 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.

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

[37] K. Pertsch, Y. Lee, and J. Lim, "Accelerating reinforcement learning with learned skill priors," in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, 2021.

[38] K. T. Ly, M. Poozhiyil, H. Pandya, G. Neumann, and A. Kucukyilmaz, "Intent-aware predictive haptic guidance and its application to shared control teleoperation," in *2021 30th IEEE international conference on robot & human interactive communication (RO-MAN)*. IEEE, 2021.

[39] Y. Zhou, S. Sonawani, M. Phielipp, S. Stepputtis, and H. Amor, "Modularity through attention: Efficient training and transfer of language-conditioned policies for robot manipulation," in *Conference on Robot Learning*. PMLR, 2023.

[40] H. Geng, S. Wei, C. Deng, B. Shen, H. Wang, and L. Guibas, "Sage: Bridging semantic and actionable parts for generalizable articulated-object manipulation under language instructions," 2023.

[41] C. Ma, K. Lu, T.-Y. Cheng, N. Trigoni, and A. Markham, "Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors," *arXiv preprint arXiv:2403.13438*, 2024.

[42] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, "Vision transformer adapter for dense predictions," in *The Eleventh International Conference on Learning Representations*, 2022.

[43] Y.-L. Sung, J. Cho, and M. Bansal, "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[44] R. Zhang, R. Fang, W. Zhang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, "Tip-adapter: Training-free clip-adapter for better vision-language modeling," *arXiv preprint arXiv:2111.03930*, 2021.

[45] A. Liang, I. Singh, K. Pertsch, and J. Thomason, "Transformer adapters for robot learning," in *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022.

[46] Z. Liu, J. Zhang, K. Asadi, Y. Liu, D. Zhao, S. Sabach, and R. Fakoor, "Tail: Task-specific adapters for imitation learning with large pretrained models," in *The Twelfth International Conference on Learning Representations*, 2023.

[47] M. Sharma, C. Fantacci, Y. Zhou, S. Koppula, N. Heess, J. Scholz, and Y. Aytar, "Lossless adaptation of pretrained vision models for robotic manipulation," in *The Eleventh International Conference on Learning Representations*, 2023.

[48] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018.

[49] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *Advances in neural information processing systems*, vol. 34, pp. 251–266, 2021.