2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

LeGO-MM: Learning Navigation for Goal-Oriented Mobile Manipulation via Hierarchical Policy Distillation

Bolei Chen, Liangbai Liu, Shengsheng Yan, Haonan Yang, Ping Zhong†, and Jianxin Wang

***Abstract*— Benefiting from mobility and dexterity, Mobile Manipulation (MM) systems are expected to assist humans with diverse tasks in everyday life. However, since MM tasks (e.g., tidying up a room) require learning multi-stage heterogeneous behaviors (e.g., picking, placing, and opening), existing Reinforcement Learning (RL) agents often face sample inefficiency and progress reversal issues. In addition, such MM agents are limited to learning customized tasks, thus not allowing for the extrapolation to new tasks and real-world scenes. In this work, we propose a Hierarchical Policy Distillation (HPD)-based RL framework to explicitly address these issues, which outperforms existing curriculum learning-based and hierarchical RL-based methods. Specifically, Sub-Skill Distillation (SSD) allows learning both the main MM task and easier sub-skills in a single training loop, facilitating exploration and mitigating process reversal by distilling the relevant sub-skills' experience into the main task. Self-boosting Policy Distillation (SPD) is designed to enhance generalization and address the information asymmetry between MM tasks in a principled way, i.e., distilling the experience of a prior task to a new one. Comparative and ablation studies on different robotic platforms demonstrate that our method significantly outperforms existing methods. Finally, real-world experiments validate the practicality of our method.**

I. INTRODUCTION

Autonomous robots are expected to be a functional part of everyday living in the near future, as shown in Fig. 1. While **R**einforcement **L**earning (RL) has been successful in a variety of settings including robotic navigation [1], [2] and fixed-base manipulation [3], long-term tasks such as **M**obile **M**anipulation (MM) remain a challenge. MM robotic systems incorporate the benefits of mobility and dexterity, due to the enlarged space in which they can move and interact with their environments.

As shown in Fig. 1, MM requires learning heterogeneous behaviors [4], [5] concurrently with sequencing these behaviors to solve the overall task. Therefore, MM agents need to understand both the immediate results of their behaviors and how these results affect overall task completion, which are often intertwined. In other words, these behaviors are interdependent, meaning that the robot can only learn how to navigate to the object placement location in another room if it has first learned how to open the door. In such a multi-stage task, existing RL agents often lead to a reversal of progress[1] [6] because they do not sufficiently explore the state space of the long-horizon task. In addition, existing work [7], [8] generally considers a fixed task setup with a known state space, which is not the case when agents must operate in previously unseen environments. Fixed task settings result in poor generalizability that may be associated with information asymmetry [9], as MM policies trained in a particular task setting do not have full access to the information (state space) of a new MM task in the same environment. Although the previous task may contain a subset of the new task's behaviors, only new task-specific training can access its full state space.

Fig. 1. An illustration of a goal-oriented MM task, which is multi-stage and requires learning heterogeneous behaviors such as navigation, collision avoidance, picking, placing, and opening concurrently.

Several studies [4], [10] use a hierarchical strategy to alleviate the above issues, but these efforts suffer from a two-stage pipeline of first needing to separately train each skill and then deciding how to combine them. Such a scheme is prone to compounding errors due to dynamically and inefficient skill coupling. In addition, some other works [11], [12] try to learn more difficult tasks that require combining multiple behaviors by using simple tasks through curriculum learning [13]. However, these methods strongly rely on manually crafted curriculums, and require many rounds of iterative training to make the agent progressively stronger. Despite promising progress, existing methods expose new weaknesses while presenting no principled solutions to address progress reversal and information asymmetry. In particular, existing methods neglect the mutual promotion of behaviors at different stages and their positive effect on achieving the overall task. In addition, they do not essentially bridge the information gap between MM tasks with different settings (state spaces) to promote generalization and sample-efficient learning.

To address the above issues, this paper proposes a

This work was supported in part by the Key project of Xiangjiang Laboratory (25XJ02003) and in part by the National Natural Science Foundation of China under 62272489, 62332020, and 62350004. This work was carried out in part using computing resources at the High-Performance Computing Center of Central South University.

The authors are with the [1]School of Computer Science and Engineering, Central South University, Changsha 410083, China. Ping Zhong is also affiliated with [2]Xiangjiang Laboratory.

† Corresponding Author (e-mail: ping.zhong@csu.edu.cn).

[1]Progress reversal refers to the situation where a task is divided into multiple stages, the behaviors of subsequent stage may alter the state achieved by the previous stage, leading to the incompleteness of the previous stage.

Hierarchical **P**olicy **D**istillation (HPD)-based RL framework, specifically including **S**ub-**S**kill **D**istillation (SSD) and **S**elf-boosting **P**olicy **D**istillation (SPD). SSD follows an easy-to-hard learning paradigm, concurrently exploring the state spaces of the main MM task and relevant sub-skills in a single training loop, with no task compounding errors and no need to formulate curriculum. Our main insight is that sub-skills are easier to learn compared to the multi-stage and long-horizon MM task. For example, if an episode requires opening a door before arriving in another room to place an object, the relevant task before navigating to the placement location would be "open the door". In this state, the opening skill would be relevant to the main task, while the placing skill would not be. On this basis, we propose distilling experience in sub-skills that explicitly direct task progress into the MM task to facilitate exploration and mitigate process reversal. Unlike SSD, SPD is designed to address the information asymmetry between MM tasks in a principled way, i.e., distilling the experience of a prior task to a new one. Under the residual RL paradigm [7], [14], SPD can generalize a prior policy to a new task without damaging its performance on the previous task. SPD can elegantly close the information gap between partially informed behavior priors from the previous task and MM policy specific to a new task to promote generalization.

In the experimental phase, sufficient comparative and ablation studies on different robotic platforms demonstrate that our method outperforms strong MM baselines using different RL algorithms. Furthermore, real-world robotic experiments demonstrate the sim-to-real generalization and practicality of our method. Overall, our main contributions are as follows:

(1) By distilling the experience of relevant sub-skills to the main MM task in a single training loop, SSD is proposed to facilitate exploration and solve the progress reversal issue.

(2) By distilling the experience of a prior MM policy into a new one without harming its performance, SPD is proposed to address information asymmetry in a principled way.

(3) We evaluate the proposed method on different robotic platforms and further deploy it to a real robotic platform.

II. Related Work

A. Task Decomposition-based Mobile Manipulation

Due to the difficulty of planning in the conjoint space of the mobile base and the robotic arm, early work solve an MM task by decomposing it into sequential navigation and static manipulation tasks. Such decomposition has been popular in methods based on planning [8], [15], reachability analysis [16], [17], and RL [4], [10], [18], [19]. Planning-based methods [8], [15] plan trajectories for robots in joint space to ensure the kinematic feasibility of mobile manipulation tasks. Given the task constraints, inverse reachability maps [16] can be used to seek good placement for the mobile base and thus solve the task smoothly. However, these classical methods are usually unable to instantly respond to diverse unstructured scenes and dynamic changes, and are thus less adaptable. Recently, some RL-based approaches [4], [10], [19] mitigate these issues by using hierarchical policy learning and curriculum learning techniques. The former [4], [10] first trains or is given pre-trained skills and then learns an high-level policy to sequence these skills together to complete long-horizon MM tasks. By formulating an easy-to-hard curriculum, the latter [19] leverages easier MM tasks to learn harder MM tasks that require composing multiple behaviors. In addition, there are methods that use classical methods as behavioral priors to guide sample efficient exploration. For example, BHyRL [7] uses inverse reachability maps as behavioral priors and employs KL regularization to distill them into MM policies to accelerate policy learning. Planning and uncertainty about the observation space can also be used as additional information during training, leading to sample-efficient RL [20], [21].

Despite the promising progress in MM methods based on task decomposition, such a scheme is prone to compounding errors due to dynamically and inefficient skill coupling. In addition, the RL-based methods mentioned above often lead to a reversal of progress because they do not sufficiently explore the state space of the long-horizon task. Unlike existing task decomposition, we propose an SSD technique to learn both the main MM task and the easier sub-skills in a single training loop, facilitating exploration and mitigating the process reversal by distilling the experience of the relevant sub-skills into the main task.

B. Whole-Body Control-based Mobile Manipulation

In contrast to decomposition, whole-body control-based methods simultaneously control all joints of the mobile base and robotic arm during the MM process. Classical optimal control techniques [22], [23] have made promising achievements in this field. These methods explicitly integrate constraints into the objective function and optimize over a roll-out horizon. However, complicated obstacles and dynamic changes usually lead to many constraints that significantly increase the computation so that the real-time nature can not be ensured. Several recent studies have explored whole-body motion control based on RL [24], [25] or diffusion models [26]. Among them, causal strategy gradient-based methods [24] integrate the causal logic between reward and action space into the RL process, significantly improving the learning efficiency.

Most recently, some methods [27], [28] propose an **E**nd-**E**ffector (EE)-centric MM framework, which achieves excellent performance and robustness by solving the mobile base's motion to cooperate with robotic arm's grasping. However, existing work usually considers a fixed task setup with a known state space, neglecting the study of poor generalization due to information asymmetry. In this work, we propose an SPD technique to discuss and solve this problem by distilling the experience of a prior MM policy into a new one.

III. Preliminaries

Problem Definition. The MM problem we consider in this work involves a robot that includes a mobile base and

a robotic arm. The MM tasks described in this paper involve skills such as navigation, collision avoidance, picking, placing, and opening. Such a problem is modeled as a goal-conditional **P**artially **O**bservable **M**arkov **D**ecision **P**rocess (POMDP) defined as a tuple $\mathcal{X}_0 = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{G}, \rho, \gamma)$ with underlying state space $\mathcal{S}$, observation space $\mathcal{O}$, action space $\mathcal{A}$, reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, initial state distribution ρ and discount factor γ. The state transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}_+$ represents the probability density of the next state $s_{t+1} \in \mathcal{S}$ given the current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$. For tasks like object rearrangement [29], the goal space $g \in \mathcal{G}$ is specified using the object's initial pickup pose or the goal pose where the object has to be placed. Our objective is to learn a policy $\pi(a|o, g)$ mapping an observation $o \in \mathcal{O}$ and goal g to an action a that maximizes the sum of discounted rewards $J = \mathbb{E}_{o\sim\mathcal{O}(\cdot|s,a),g\sim\mathcal{G}} \sum_t \gamma^t \mathcal{R}(o, a; g)$.

Common RL-based Solutions. We can generally employ RL-based methods to solve the above problem by training a policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ in a specific long-horizon MM task. In particular, we follow the common practice [7] and model π as a maximum-entropy policy [30], [31] to fully explore the state space of a given task. That is, the policy π maximizes the entropy-augmented cumulative reward and the resulting optimal policy follows a Boltzmann distribution [30] as follows:

$$\pi(a|o, g) = \frac{1}{Z_o} exp(\frac{1}{\alpha} Q^*(o, a; g)), \tag{1}$$

where $Q^*(o, a; g)$ denotes the soft Q-function as defined in [30], which satisfies the soft Bellman equation:

$$\begin{aligned} Q^*(o, a; g) &= \mathcal{R}(o, a; g) \\ &+ \gamma \mathbb{E}_{o'\sim\mathcal{O}(\cdot|s,a)} \left[\alpha \log \int_{\mathcal{A}} \exp\left(\frac{1}{\alpha} Q^*(o', a'; g)\right) da' \right]. \end{aligned} \tag{2}$$

$Z_o = \int_{\mathcal{A}} \exp\left(\frac{1}{\alpha} Q^*(o, a; g)\right) da$ is the normalization constant. α is the coefficient that determines the relative importance of entropy and reward in the maximum-entropy policy. Combining Eq. (1) and Eq. (2), we can derive:

$$Q^*(o, a; g) = \alpha \log \pi(a|o, g) + \alpha \log Z_o, \tag{3}$$

which will be used to derive the subsequent Eq. (6).

IV. Methodology

This section will introduce SSD and SPD separately. The complete HPD is shown in Algorithm 1.

A. Sub-skill Distillation

The core idea of SSD is to learn a robust policy by transferring knowledge of easier sub-skills to the main MM task. We achieve this by learning the main MM task and sub-skills in a single training loop, without using hierarchical strategy learning and explicit curriculum. We denote the main MM task as $\mathcal{X}_0$ with identifier T_0 and define N sub-skills as $\{\mathcal{X}_i\}_{n=1}^N$ with identifier $\{T_i\}_{t=1}^N$. The sub-skills $\{\mathcal{X}_i\}$ share the same state, observation, goal, and action spaces as $\mathcal{X}_0$. However, each $\mathcal{X}_i$ has a individually defined starting state distribution and reward function. We assume that the main

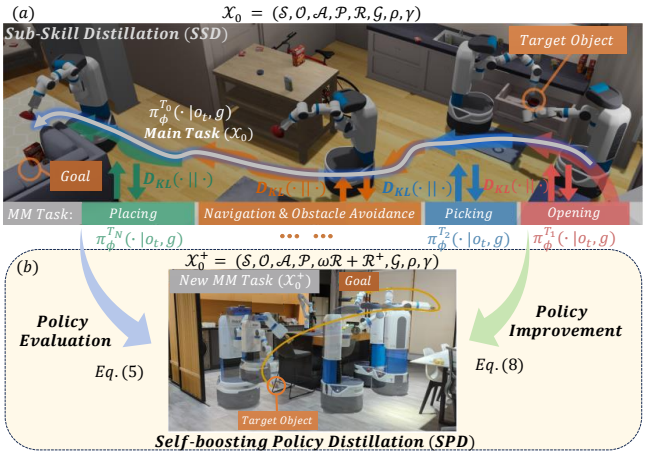

Fig. 2. (a) An illustration of SSD, which allows learning both the main MM task and easier sub-skills in a single training loop, facilitating exploration and mitigating process reversal by distilling the relevant sub-skills' experience into the main task. (b) An illustration of SPD, which is designed to enhance generalization in a principled way by distilling the experience of a prior task to a new one.

task $\mathcal{X}_0$ consists of a subset of sub-skills, and thus sub-skills are easier to learn than the full task $\mathcal{X}_0$. For example, we define sub-skills as interactions required by an agent to complete an entire MM episode, such as picking and opening shown in Fig. 2 (a). Since all sub-skills share the same observation and action space, the policy π_ϕ parameterized by ϕ can act and observe during the learning of the main task $\mathcal{X}_0$ and all skills.

In practice, we create separate environment instances for the main task and N sub-skills, and vectorize them for parallel action execution. SSD optimizes the RL agent by using the average of the main task and sub-skill losses. In addition, SSD distills experience from the sub-skill $\mathcal{X}_i$ to the main task $\mathcal{X}_0$ by forcing the policy to match the action distributions of $\mathcal{X}_0$ and $\mathcal{X}_i$. Therefore, the overall loss function of SSD, including the distillation loss, is as follows:

$$\begin{aligned} J(\pi_\phi) &= \frac{1}{N} \sum_{i=0}^{N} J_{\mathcal{X}_i}(\pi_\phi) \\ &+ \lambda \sum_{i=1}^{N} w_i(s_t) D_{\mathrm{KL}}(\pi_\phi^{T_0}(\cdot|o_t, g) \parallel \pi_\phi^{T_i}(\cdot|o_t, g)). \end{aligned} \tag{4}$$

Here, $J_{\mathcal{X}_i}(\pi_\phi)$ denotes the losses of the policy π_ϕ in $\mathcal{X}_0$ and $\mathcal{X}_i$. The SSD reward is calculated as the KL-divergence of the action distribution under $\mathcal{X}_0$ and $\mathcal{X}_i$ weighted by the relevance of sub-skill $\mathcal{X}_i$ given by $w_i(s_t)$. λ represents the weight of distillation relative to other RL losses. $w_i(s_t)$ can be obtained from the simulator according to the current state s_t during the training phase, which denotes how much the knowledge from $\mathcal{X}_i$ should apply to $\mathcal{X}_0$. Notably, such oracle information is only used during the training phase and is not required during the inference phase. A more detailed algorithmic procedure can be found in Algorithm 1.

At first glance, our SSD has similar insights as hierarchical strategy learning and curriculum learning, except that the main task and sub-skills are learned concurrently. In fact, the SSD has three definite advantages: (1) SSD follows an easy-to-hard learning paradigm and fully explores the sub-skills'

Algorithm 1 The workflow of HPD

Load prior policy π_ϕ, Initialize new policy $\hat{\pi}_\phi$ and Q-function Q_θ
Initialize state-space, observation space and action space $\{\mathcal{S}, \mathcal{O}, \mathcal{A}\}$
Define relevance $w_i : \mathcal{S} \to \{0, 1\}\ \forall\ i \in \{1, 2, \cdots, N\}$
Initialize distillation coefficient, normalization parameters λ
for epoch $\leftarrow 1$ to train-epochs **do**
  // Bold face represents a vector.
  Obtain a batch of B samples $\{\mathbf{r}^i, \mathbf{o}^i, \mathbf{s}^i, \mathbf{a}^i, \mathbf{g}^i\}_{i=0}^N$ by executing $\{\hat{\pi}_\phi^{T_i}\}_{i=0}^N$
  Obtain $\{\hat{\pi}_\phi^{T_i}(\cdot|\mathbf{o}_0, \mathbf{g}_0)\}_{i=1}^N$ by evaluating $\mathbf{o}_0$ with $\{T_i\}_{i=1}^N$
  // Compute losses.
  Compute policy losses: $J_{RL}(\hat{\pi}_\phi) = \frac{1}{N}\sum_{i=0}^N J_{\mathcal{X}_i}(\hat{\pi}_\phi)$ using $\{\mathbf{r}^i, \mathbf{o}^i, \mathbf{s}^i, \mathbf{a}^i, \mathbf{g}^i\}_{i=0}^N$, where $J_{\mathcal{X}_i}(\hat{\pi}_\phi)$ is calculated according to Eq. (8).
  Compute SSD loss: $J_{SSD}(\hat{\pi}_\phi) = \sum_{i=1}^N \sum_{i=0}^B w_i(s_t^0) D_{\mathrm{KL}}(\hat{\pi}_\phi^{T_0}(\cdot|o_t^0, g) \parallel \hat{\pi}_\phi^{T_i}(\cdot|o_t^0, g))$
  Update $\hat{\pi}_\phi$ using the total loss: $J(\hat{\pi}_\phi) = J_{RL}(\hat{\pi}_\phi) + \lambda J_{SSD}(\hat{\pi}_\phi)$
  Calculate the target Q-value using Eq. (6)
  Update Q-function Q_θ by minimizing the temporal-difference error using Eq. (5)
end for

state space while solving the main task, which enhances the agent's exploration. (2) The policy distills experience from the behaviors that are explicitly relevant to the main task, which is sample-efficient. (3) SSD learns to solve a complete MM task from scratch without compounding errors.

B. Self-boosting Policy Distillation

While SSD promotes the learning of a specific MM task, the learned policy may be hard to generalize to other MM tasks in a complex environment due to the information asymmetry[2] [9]. Intuitively, any new MM task can use knowledge from previous tasks (that may contain a subset of sub-skills of the new task) as priors to guide exploration and accelerate learning. Therefore, the core idea of SPD is to generalize a policy trained on a specific MM task to a differentiated or more difficult MM task $\mathcal{X}_0^+$ without harming its performance on the original task $\mathcal{X}_0$, as shown in Fig. 2 (b). We model $\mathcal{X}_0^+$ as a goal-conditional POMDP with an additional reward $\mathcal{R}^+ : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, i.e., $\mathcal{X}_0^+ = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \omega\mathcal{R} + \mathcal{R}^+, \mathcal{G}, \rho, \gamma)$, where ω denotes the reward weight. Since the action space in our problem definition is continuous, the **S**oft **A**ctor-**C**ritic (SAC) algorithm [31] is used to learn a maximum-entropy policy. Therefore, we train the parameterized Q-function $Q_{R^+,\theta}(o, a; g)$, and the policy network $\hat{\pi}_\phi(a|o, g)$ alternately through policy evaluation and improvement.

[2]Information asymmetry exists because the prior policy only accesses the part of the state that is relevant, but the state that has not been accessed is critical for learning new policies.

During policy evaluation, we update the Q-function to evaluate the current policy, and the loss function is essentially to minimize the temporal-difference error by taking gradient steps:

$$J_{Q_R^+}(\theta) = \mathbb{E}_{(o_t, a_t, g, R_t^+, o_{t+1})\sim\mathcal{D}}\Big[\frac{1}{2}\Big(Q_{R^+,\bar{\theta}}^{\text{target}}(o_t, a_t; g) - Q_{R^+,\theta}(o_t, a_t; g)\Big)^2\Big], \tag{5}$$

where $\mathcal{D}$ is the replay buffer and the target Q-value $Q_{R^+,\bar{\theta}}^{\text{target}}(o_t, a_t; g)$ is computed as:

$$Q_{R^+,\bar{\theta}}^{\text{target}}(o_t, a_t; g) = R_t^+ + \gamma\mathbb{E}_{a'\sim\hat{\pi}_\phi}\Big[Q_{R^+,\bar{\theta}}(o_{t+1}, a'; g) + \omega' \log \pi_\phi(a'|o_{t+1}, g) - \hat{\alpha}\log\hat{\pi}_\phi(a'|o_{t+1}, g)\Big]. \tag{6}$$

Eq. (6) is derived from the following Eq. (7) by using a residual Q-function $Q_{R^+,t} := \hat{Q}_t - \omega Q^*$ [32] based on Eq. (2) and Eq. (3). Specifically, the policy evaluation step aims to iteratively compute the soft Q-value of a policy $\hat{\pi}_\phi$ in the SAC algorithm, which relies on repeatedly applying a modified soft Bellman backup operator given by:

$$\hat{Q}_{t+1}(o, a; g) = R^+(o, a; g) + \omega R(o, a; g) + \gamma\mathbb{E}_{o'\sim\mathcal{O}(\cdot|s,a)}\Big[\mathbb{E}_{a'\sim\hat{\pi}_\phi}\Big[\hat{Q}_t(o', a'; g) - \hat{\alpha}\log\hat{\pi}_\phi(a'|o', g)\Big]\Big]. \tag{7}$$

Please see the anonymous link[3] for a step-by-step derivation process of Eq. (6) and Eq. (7).

During policy improvement, we improve the policy $\hat{\pi}_\phi$ leveraging the current estimated Q-value $Q_{R^+,\theta}(o, a; g)$ and the prior policy π_ϕ by minimizing the expected KL-divergence between $\hat{\pi}_\phi$ and the desired maximum-entropy policy [31] given the current $Q_{R^+,\theta}(o, a; g)$, which is equivalent to minimizing the loss function below:

$$J(\hat{\pi}_\phi) = \mathbb{E}_{o_t\sim\mathcal{D}}\Big[\mathbb{E}_{a\sim\hat{\pi}_\phi}\big[\hat{\alpha}\log\hat{\pi}_\phi(a|o_t, g) - Q_{R^+,\theta}(o_t, a; g) - \omega'\log\pi_\phi(a|o_t, g)\big]\Big]. \tag{8}$$

Notably, the new reward function is defined as a weighted sum of the base reward R and the additional reward R^+, allowing the policy fine-tuned by the above RL process satisfies both tasks. In fact, we can follow the technical route in SSD to evolve from π_ϕ to $\hat{\pi}_\phi$ by regularizing the KL-divergence between the fine-tuned and the prior policies. Although the use of KL constraints can shape the learning a new task, it does not address the challenge of effective information transfer from a prior task to a new one. In addition, we can also directly employ the reward function $\omega R + R^+$ to fine-tune π_ϕ to the task $\mathcal{X}_0^+$. However, we experimentally find that this method fails to balance the performance on the previous and new tasks. We believe that addressing the information asymmetry requires a principled way to design learning objectives that jointly optimize the policy's performance on both initial and new tasks. The comparative studies in Tab. II demonstrate the superiority of our SPD.

V. Experiments

A. Experimental Setup

Simulation Platforms and Datasets (Scenes). Following existing work [4], [28], we conduct comparative studies on

[3]https://anonymous.4open.science/r/LeGO-MM-5BB3

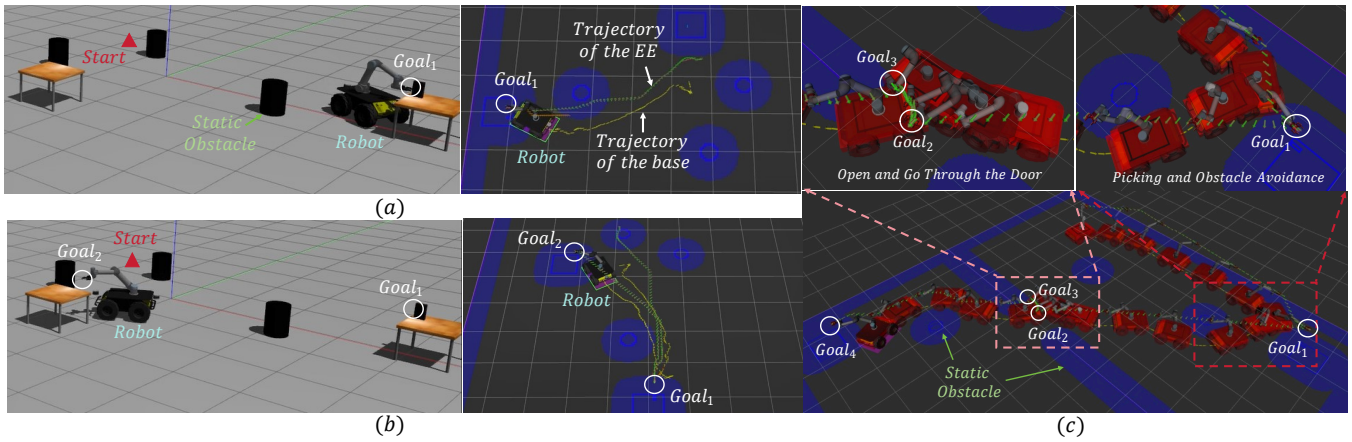

Fig. 3. Illustrations of MM tasks in the *Map-based MM System* setup. The positions of the robot and all objects are randomly initialized at the start of each episode. (a) shows the robot moving from the start position to the picking point (Goal_1) while avoiding static obstacles in Task 1. (b) illustrates the movement from the picking point to the placing point (Goal_2) in Task 1. (c) shows the motion trajectory of the robot to complete Task 2. Task 2 has two more goals than Task 1 for opening the door.

two robotic setups based on mainstream simulation platforms:

(1) Map-based MM System (Gazebo-Husky+UR5 Robot): In this setup, we use the **R**obot **O**perating **S**ystem (ROS)[4] to control an MM robotic system consisting of a Husky[5] mobile base and a 6-DoF UR5[6] robotic arm. Following the EE-centric MM framework [27], given a sequence of 6D poses in $SE(3)$, the robot needs to find appropriate base placements in $SE(2)$ to allow the EE of the robotic arm to reach these 6D goal poses sequentially along dynamically feasible paths. We employ an **I**nverse **K**inematics (IK) solver[7] to solve the robotic arm's motion in joint space. Following this work [28], our objective is to learn the navigation of the mobile base to cooperate with the long-horizon manipulation of the robotic arm. To ensure the robot's kinematic feasibility, the 2-dim action space $\mathcal{A}$ is continuous. The observation space $\mathcal{O}$ consists of the robot joint states, the previous actions a_{t-1}, the EE's pose and velocity, the desired EE's pose, and the robot-centric local occupancy map.

In this setting, we construct diverse training scenes for main MM task and sub-skills learning in the Gazebo simulator[8]. Specifically, the sub-skills we consider include navigation, collision avoidance, picking, placing, and opening, please see the anonymous link for more details. We consider two MM scenarios with different scopes of motion, including single-room (Task 1) and cross-room (Task 2) point-to-point mobile manipulation. As shown in Fig. 3 (a) and (b) for Task 1, in each episode, the positions of the robot, obstacles, desks, picking point (Goal_1), and placing point (Goal_2) are randomly initialized. Unlike Task 1, Task 2 requires the robot to pick up an object and then open a door to another room to place the object in a specified position in the this room, as shown in Fig. 3 (c). Please refer to the anonymous link for more visualizations.

[4]https://wiki.ros.org/
[5]https://wiki.ros.org/Robots/Husky
[6]https://github.com/ros-industrial/universal_robot
[7]https://github.com/TAMS-Group/bio_ik
[8]https://gazebosim.org/home

(2) Vision-based MM System (Habitat-Fetch Robot): We compare our method with the baselines of Habitat 2.0 Object Rearrangement tasks [33] using the experimental setup from [4]. In this setting, the Fetch robot[9] must move an object from a specified start position to a desired goal position using visual observation in an indoor home environment, as shown in Fig. 2 (a). The task is specified by a starting 3D coordinate of the object to be moved and a 3D target coordinate to move the object to. The fetch robot interacts with its environment through a mobile base and a 7-DoF robotic arm. Possible sub-skills involved in each MM episode include navigation, picking, placing, and opening, please see the anonymous link for more details. The episode is successful if the robot places the target object within 15 cm of the goal position within a budget of 1500 steps. The action space $\mathcal{A}$ is a 10-dim continuous space that includes 7-dim joint actions of the robotic arm, 2-dim base actions (linear forwarding and angular velocities), and 1-dim gripper action. The observation space $\mathcal{O}$ consists of head and arm depth images, robot joint states, gripper state in the base frame, goal positions in both base and EE frames, as well as a scalar to indicate whether an object is held.

We use the rearrangement training dataset (11791 episodes) introduced in [29] to train our method and report performance on the easy (100 episodes) and hard (200 episodes) evaluation splits from [4]. In the easy episodes, both the target object and goal are in open receptacles, which means that the robot does not have to interact with the receptacle (e.g., open the receptacle) before picking up and placing it. In the hard episodes, the robot is required to use visual inputs to perceive whether the target object is in a closed receptacle and to open the receptacle before picking it up, as shown in Fig. 2 (a).

Baselines and Evaluation Metrics. For the *Gazebo-Husky+UR5 Robot* setting, we compare our method with the **S**tate-**o**f-**T**he-**A**rt (SoTA) EE-centric MM method N^2M^2 [28], [34], the award-winning hierarchical method BHyRL

[9]http://fetchrobotics.com/

TABLE I

COMPARATIVE STUDIES ARE CONDUCTED IN THE *Map-based MM System* SETUP TO DEMONSTRATE THE SUPERIORITY OF THE SSD TECHNIQUE. THE RESULTS OF THE COMPARED METHODS ARE OBTAINED BASED ON THEIR OPEN-SOURCE CODE. IN EACH TASK, WE EVALUATE 100 EPISODES FOR EACH METHOD. WE APPLY 3 RANDOM SEEDS TO EACH METHOD AND REPORT THE AVERAGE METRICS AND STANDARD DEVIATIONS.

Method	Task 1				Task 2			
	AIKF↓	ABC↓	TCR↑	PSR↑	AIKF↓	ABC↓	TCR↑	PSR↑
End-to-End MM	-	-	$0_{\pm 0}$	$0_{\pm 0}$	-	-	$0_{\pm 0}$	$0_{\pm 0}$
BHyRL	$6.46_{\pm 0.61}$	$0.93_{\pm 0.39}$	$47_{\pm 3}$	$36_{\pm 2}$	$12.06_{\pm 1.21}$	$2.77_{\pm 0.48}$	$21_{\pm 3}$	$19_{\pm 4}$
N^2M^2	$5.96_{\pm 0.53}$	$0.88_{\pm 0.35}$	$56_{\pm 2}$	$48_{\pm 2}$	$11.29_{\pm 1.33}$	$2.16_{\pm 0.55}$	$29_{\pm 4}$	$17_{\pm 2}$
N^2M^2 + SSD	$\mathbf{3.37}_{\pm 0.29}$	$\mathbf{0.26}_{\pm 0.22}$	$\mathbf{83}_{\pm 1}$	$\mathbf{77}_{\pm 3}$	$\mathbf{8.78}_{\pm 1.46}$	$\mathbf{1.27}_{\pm 0.29}$	$\mathbf{57}_{\pm 6}$	$\mathbf{42}_{\pm 7}$

TABLE II

FURTHER COMPARATIVE STUDIES ARE CONDUCTED IN THE *Map-based MM System* SETUP TO DEMONSTRATE THE SUPERIORITY OF OUR SPD TECHNIQUE. FOR ALL METHODS, WE FIRST USE N^2M^2+SSD TO TRAIN AN MM POLICY SPECIFIC TO EITHER TASK 1 OR TASK 2. THEN, WE FURTHER TRAIN THIS MM POLICY ON ANOTHER TASK USING DIFFERENT GENERALIZATION METHODS AND EVALUATE IT ON BOTH TASKS. RL&KL DENOTES RL WITH A KL REGULARIZATION OBJECTIVE.

Method	Task 1				Task 2			
	AIKF↓	ABC↓	TCR↑	PSR↑	AIKF↓	ABC↓	TCR↑	PSR↑
N^2M^2 + SSD (Task-spefic)	$\mathbf{3.37}_{\pm 0.29}$	$\mathbf{0.26}_{\pm 0.22}$	$\mathbf{83}_{\pm 1}$	$\mathbf{77}_{\pm 3}$	$\mathbf{8.78}_{\pm 1.46}$	$\mathbf{1.27}_{\pm 0.29}$	$\mathbf{57}_{\pm 6}$	$\mathbf{42}_{\pm 7}$
N^2M^2 + SSD (Task 1) + RL (Task 2)	$8.94_{\pm 0.76}$	$2.15_{\pm 0.52}$	$44_{\pm 2}$	$31_{\pm 3}$	$8.72_{\pm 1.98}$	$1.36_{\pm 0.53}$	$56_{\pm 3}$	$38_{\pm 5}$
N^2M^2 + SSD (Task 2) + RL (Task 1)	$4.36_{\pm 0.63}$	$0.46_{\pm 0.34}$	$80_{\pm 4}$	$74_{\pm 2}$	$11.80_{\pm 0.98}$	$2.39_{\pm 0.60}$	$36_{\pm 4}$	$29_{\pm 5}$
N^2M^2 + SSD (Task 1) + RL&KL (Task 2)	$4.57_{\pm 0.73}$	$0.82_{\pm 0.36}$	$72_{\pm 1}$	$63_{\pm 3}$	$12.79_{\pm 1.41}$	$2.46_{\pm 0.48}$	$31_{\pm 3}$	$19_{\pm 1}$
N^2M^2 + SSD (Task 2) + RL&KL (Task 1)	$7.25_{\pm 0.81}$	$1.14_{\pm 0.52}$	$50_{\pm 2}$	$39_{\pm 4}$	$9.14_{\pm 1.07}$	$1.98_{\pm 0.62}$	$45_{\pm 3}$	$38_{\pm 1}$
N^2M^2 + SSD (Task 1) + SPD (Task 2)	$3.17_{\pm 0.06}$	$0.07_{\pm 0.09}$	$85_{\pm 1}$	$78_{\pm 3}$	$11.43_{\pm 0.53}$	$1.58_{\pm 0.34}$	$37_{\pm 2}$	$30_{\pm 2}$
N^2M^2 + SSD (Task 2) + SPD (Task 1)	$5.85_{\pm 0.51}$	$0.78_{\pm 0.19}$	$58_{\pm 3}$	$43_{\pm 2}$	$8.85_{\pm 1.33}$	$0.76_{\pm 0.16}$	$58_{\pm 5}$	$45_{\pm 3}$

[7], and an end-to-end method adapted from [28]. As we are committed to sim-to-real transition, we report the **A**verage **IK** solution **F**ailures (AIKF), **A**verage **B**ase **C**ollisions (ABC), **T**ask **C**ompletion **R**ate (TCR), and **P**erfect **S**uccess **R**ate (PSR) metrics based on ROS. A fewer AIKF means a better cooperation between the mobile base and the robotic arm. TCR implies that the MM robot completes the task while allowing for IK solution failures and base collisions. We set the threshold for allowing both kinds of failures to 20. PSR means completing the MM tasks with zero failures. For the *Habitat-Fetch Robot* setting, we compare our HPD-based RL agent with end-to-end [18], [35], hierarchical [4], [10], and curriculum learning-based [18] MM baselines. Following the baseline methods, we report the success rate of MM tasks.

Implementation Details. For all experiments, the parameter λ for SSD is set to 0.05. The parameters α, $\hat{\alpha}$,and ω' used for the SPD are all set to 1.0.

(1) Map-based MM System (Gazebo-Husky+UR5 Robot): We use a 3-layer convolutional network with maximal pooling to encode the local occupancy map, whose size and resolution are 30 $\times$ 30 and 0.1 m, respectively. Other observations, i.e., the state of the robot, are concatenated with the map embeddings and then passed through a multilayer perceptron. We train RL agents for $1e^6$ steps of environment interactions by using a learning rate of $1e^{-4}$ and linearly decay the learning rate to $1e^{-6}$. More implementation details related to the main tasks and sub-skills can be found in the anonymous link.

(2) Vision-based MM System (Habitat-Fetch Robot): Similar to [33], the MM policy is represented as a 2-layer LSTM network with 512 hidden units per layer. The visual observations are encoded using the ResNet50 network. Other observations, i.e., the state of the robot, are concatenated with the visual embeddings and then passed through the LSTM. We train methods for 475M steps of environment interactions by using a learning rate of $3e^{-4}$ and linearly decay the learning rate to 0.

B. Comparative Studies

We first compare our method with several baselines in the *Map-based MM System* setup to demonstrate the superiority of our SSD, as shown in Tab. I. As a hierarchical approach, BHyRL [7] first models the inverse reachability map of the MM robot as a prior policy, and then employs the KL regularizer to boost it to a more powerful policy. N^2M^2 [28] is the SoTA EE-centric RL algorithm capable of handling MM tasks of various complexity, which is recently extended to accommodate different robotic architectures [34]. For a more comprehensive comparison, an End-to-End MM policy is achieved by not relying on the IK solver, but by directly learning to control the entire robot by extending the action space, including the joint velocities of the arm. However, we find this baseline does not work due to the large action space, and N^2M^2 got this conclusion as well. As shown in Tab. I, by applying SSD to N^2M^2 to integrate skill-related experiences into the learning process of the main task, significant performance improvements can be achieved

TABLE III

COMPARATIVE STUDIES ON THE REARRANGEMENT TASK (IN THE *Vision-based MM System* SETUP). THE ALL EPISODES COLUMN IS AN AVERAGE ACROSS BOTH THE EASY AND HARD EPISODES.

Method	Val		
	All Episodes	Easy	Hard
RL Curriculum	$0_{\pm 0}$	$0_{\pm 0}$	$0_{\pm 0}$
M3 [10]	$25_{\pm 1}$	$56_{\pm 1}$	$9_{\pm 2}$
M3 (Oracle) [10]	$27_{\pm 0}$	$57_{\pm 2}$	$12_{\pm 1}$
Galactic [35]	-	$37_{\pm 0}$	-
Skill Transformer [4]	$25_{\pm 1}$	$44_{\pm 1}$	$16_{\pm 1}$
AuxDistill [18]	$49_{\pm 2}$	$74_{\pm 2}$	$36_{\pm 2}$
SSD (Easy) + SPD (Hard)	$52_{\pm 1}$	$76_{\pm 1}$	$40_{\pm 2}$
SSD (Hard) + SPD (Easy)	$53_{\pm 2}$	$78_{\pm 3}$	$37_{\pm 1}$

on both tasks. Please see Fig. 7 in the anonymous link for training details. In particular, the significant reduction in the number of IK solution failures and collisions reflects the ability of the base to cooperate well with the motion of the robotic arm, leading to a significant gain in the PSR metric.

Further comparative studies shown in Tab. II demonstrate the advantages of our SPD. As shown in Tab. II (row2∼row3), employing RL to generalize an MM policy trained on a specific task to a new task without imposing additional constraints leads to severely impairing the performance on the original task. In addition, such a generalization does not result in improved performance on the new task. These results reflect the information asymmetry between tasks, i.e., MM policies trained in a particular task setting do not have full access to the state space of a new MM task in the same environment. Although KL regularization provides constraints for RL, it is only meant to prevent the fine-tuned policy from deviating too far from the prior policy. Therefore, KL regularization objective allows for competitive results on the new task at the cost of a modest performance degradation on the original task (row4∼row5). In contrast, our SPD can achieve promising results on the new task without harming (or even improving) the performance of the original task. Please see Fig. 8 in the anonymous link for training and evaluation details.

Based on publicly available datasets, we further conduct comparative studies between our method and several baselines in the *Vision-based MM System* setup, as shown in Tab. III. In this setting, the MM policy trained using SSD is generalized between Easy and Hard rearrangement episodes through SPD. We find that our method can achieve the SoTA MM performance and contribute to generalization across tasks of varying difficulty. In addition, our HPD can achieve the best MM performance on All Episodes. The performance gains achieved on two robot platforms with different sensing types reflect the robustness of our method.

C. Ablation Studies

The results in Tab. I and Tab. II demonstrate that the attendance of SSD and SPD can effectively improve the MM performance. In addition, we verify the effectiveness of sub-skills in SSD by ablation studies in the *Map-based MM System* setup, as shown in Tab. IV. We find that the absence of collision avoidance and opening sub-skills leads to significant performance degradation on Task 2. In particular, the absence of the collision avoidance sub-skill leads to an increase in the number of collisions (ABC metric). The absence of the opening sub-skill leads to an increase in the number of IK solution failures (AIKF metric), as opening the door requires smooth IK solving. Experience from other sub-skills also contributes to MM performance gains, even if it's not obvious.

TABLE IV

ABLATION STUDIES ARE PERFORMED ON DIFFERENT SUB-SKILLS IN THE *Map-based MM System* SETUP.

Ablation	Task 2			
	AIKF ↓	ABC ↓	TCR ↑	PSR ↑
w/o navigation	**8.69**$_{\pm 1.70}$	1.30$_{\pm 0.49}$	56$_{\pm 6}$	41$_{\pm 6}$
w/o obstacle avoidance	9.57$_{\pm 1.21}$	1.84$_{\pm 0.56}$	52$_{\pm 3}$	37$_{\pm 2}$
w/o picking	8.80$_{\pm 0.66}$	1.39$_{\pm 0.17}$	56$_{\pm 5}$	40$_{\pm 3}$
w/o placing	8.94$_{\pm 0.79}$	1.40$_{\pm 0.32}$	55$_{\pm 4}$	39$_{\pm 4}$
w/o opening	9.92$_{\pm 1.69}$	1.33$_{\pm 0.17}$	54$_{\pm 7}$	38$_{\pm 5}$
all sub-skills	8.78$_{\pm 1.46}$	**1.27**$_{\pm 0.29}$	**57**$_{\pm 6}$	**42**$_{\pm 7}$

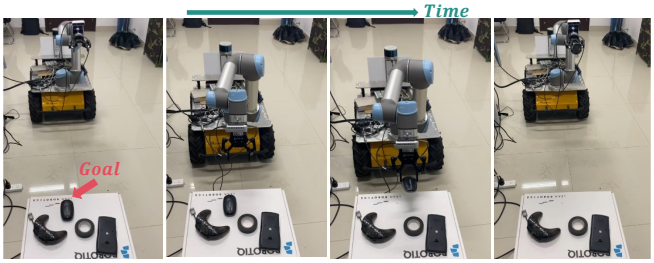

Fig. 4. Real-world experiments in navigating and grasping objects in the *Map-based MM System* setup.

D. Real World Experiments

In the *Map-based MM System* setup, we deploy the policy trained using SSD to a real robotic platform without additional fine-tuning. As shown in Fig. 4, the robot is asked to navigate to the table where the goal object is placed to pick up a mouse and then return to the initial position. In addition, we also require the robot to put the fetched mouse back to its initial position, please see the video in the anonymous link for more details.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose an HPD-based RL framework that explicitly addresses sample inefficiency, progress reversal, and information asymmetry in MM tasks. By distilling the experience of relevant sub-skills into the learning process of the main task, SSD facilitates purposeful exploration and mitigates process reversal. SPD can address the information asymmetry between tasks by generalizing the prior policy to new tasks without harming its performance on the original

task. Sufficient comparative and ablation studies on two robotic platforms with different sensing types demonstrate that our method significantly outperforms the SoTA MM methods. The limitations of our approach are two-fold. First, the goal-oriented MM policy assumes previously known 6-DoF grasping poses, but their estimation is an open research topic in the field of robotics. Second, our MM policy still lacks the ability to accomplish long-horizon tasks.

In future work, we consider addressing these limitations by incorporating large visual language models. In addition, we will evaluate our MM policy in diverse real-world scenarios.

REFERENCES

[1] B. Chen, J. Kang, P. Zhong, Y. Liang, Y. Sheng, and J. Wang, "Embodied contrastive learning with geometric consistency and behavioral awareness for object navigation," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 4776–4785.

[2] B. Chen, J. Kang, P. Zhong, Y. Cui, S. Lu, Y. Liang, and J. Wang, "Think holistically, act down-to-earth: A semantic navigation strategy with continuous environmental representation and multi-step forward planning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 5, pp. 3860–3875, 2023.

[3] H. Nguyen and H. La, "Review of deep reinforcement learning for robot manipulation," in *2019 Third IEEE international conference on robotic computing (IRC)*. IEEE, 2019, pp. 590–595.

[4] X. Huang, D. Batra, A. Rai, and A. Szot, "Skill transformer: A monolithic policy for mobile manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 852–10 862.

[5] Z. Lin, Y. Chen, and Z. Liu, "Autoskill: Hierarchical open-ended skill acquisition for long-horizon manipulation tasks via language-modulated rewards," *IEEE Transactions on Cognitive and Developmental Systems*, 2025.

[6] A. Hundt, B. Killeen, N. Greene, H. Wu, H. Kwon, C. Paxton, and G. D. Hager, ""good robot!": Efficient reinforcement learning for multi-step visual tasks with sim to real transfer," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6724–6731, 2020.

[7] S. Jauhri, J. Peters, and G. Chalvatzaki, "Robot learning of mobile manipulation with reachability behavior priors," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8399–8406, 2022.

[8] S. Jauhri, S. Lueth, and G. Chalvatzaki, "Active-perceptive motion generation for mobile manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 1413–1419.

[9] A. Galashov, S. M. Jayakumar, L. Hasenclever, D. Tirumala, J. Schwarz, G. Desjardins, W. M. Czarnecki, Y. W. Teh, R. Pascanu, and N. Heess, "Information asymmetry in kl-regularized rl," *arXiv preprint arXiv:1905.01240*, 2019.

[10] J. Gu, D. S. Chaplot, H. Su, and J. Malik, "Multi-skill mobile manipulation for object rearrangement," *arXiv preprint arXiv:2209.02778*, 2022.

[11] T. Ni, K. Ehsani, L. Weihs, and J. Salvador, "Towards disturbance-free visual mobile manipulation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5219–5231.

[12] Z. Wang, Y. Jia, L. Shi, H. Wang, H. Zhao, X. Li, J. Zhou, J. Ma, and G. Zhou, "Arm-constrained curriculum learning for loco-manipulation of a wheel-legged robot," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 10 770–10 776.

[13] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," *Journal of Machine Learning Research*, vol. 21, no. 181, pp. 1–50, 2020.

[14] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6023–6029.

[15] M. Arduengo, C. Torras, and L. Sentis, "Robust and adaptive door operation with a mobile robot," *Intelligent Service Robotics*, vol. 14, no. 3, pp. 409–425, 2021.

[16] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1970–1975.

[17] X. Feng, T. Horii, and T. Nagai, "Predictive reachability for embodiment selection in mobile manipulation behaviors," *IEEE Robotics and Automation Letters*, 2025.

[18] A. N. Harish, L. Heck, J. P. Hanna, Z. Kira, and A. Szot, "Reinforcement learning via auxiliary task distillation," in *European Conference on Computer Vision*. Springer, 2024, pp. 214–230.

[19] K. Fang, T. Migimatsu, A. Mandlekar, L. Fei-Fei, and J. Bohg, "Active task randomization: Learning visuomotor skills for sequential manipulation by proposing feasible and novel tasks," *CoRR*, 2022.

[20] A. S. Morgan, D. Nandha, G. Chalvatzaki, C. D'Eramo, A. M. Dollar, and J. Peters, "Model predictive actor-critic: Accelerating robot skill acquisition with deep reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6672–6678.

[21] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[22] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, 2020.

[23] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[24] J. Hu, P. Stone, and R. Martín-Martín, "Causal policy gradient for whole-body mobile manipulation," *arXiv preprint arXiv:2305.04866*, 2023.

[25] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning*. PMLR, 2023, pp. 138–149.

[26] S. Yan, Z. Zhang, M. Han, Z. Wang, Q. Xie, Z. Li, Z. Li, H. Liu, X. Wang, and S.-C. Zhu, "M 2 diffuser: Diffusion-based trajectory optimization for mobile manipulation in 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[27] G. He, X. Guo, L. Tang, Y. Zhang, M. Mousaei, J. Xu, J. Geng, S. Scherer, and G. Shi, "Flying hand: End-effector-centric framework for versatile aerial manipulation teleoperation and policy learning," *arXiv preprint arXiv:2504.10334*, 2025.

[28] D. Honerkamp, T. Welschehold, and A. Valada, "N2m2: Learning navigation for arbitrary mobile manipulation motions in unseen and dynamic environments," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3601–3619, 2023.

[29] A. Szot, K. Yadav, A. Clegg, V.-P. Berges, A. Gokaslan, A. Chang, M. Savva, Z. Kira, and D. Batra, "Habitat rearrangement challenge 2022," https://aihabitat.org/challenge/2022_rearrange, 2022.

[30] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International conference on machine learning*. PMLR, 2017, pp. 1352–1361.

[31] 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.

[32] C. Li, C. Tang, H. Nishimura, J. Mercat, M. Tomizuka, and W. Zhan, "Residual q-learning: Offline and online policy customization without value," *Advances in Neural Information Processing Systems*, vol. 36, pp. 61 857–61 869, 2023.

[33] 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.

[34] R. Schneider, D. Honerkamp, T. Welschehold, and A. Valada, "Task-driven co-design of mobile manipulators," *IEEE Robotics and Automation Letters*, 2025.

[35] V.-P. Berges, A. Szot, D. S. Chaplot, A. Gokaslan, R. Mottaghi, D. Batra, and E. Undersander, "Galactic: Scaling end-to-end reinforcement learning for rearrangement at 100k steps-per-second," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 767–13 777.