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

Harmonic Mobile Manipulation

Ruihan Yang[1*] Yejin Kim[2] Rose Hendrix[2] Aniruddha Kembhavi[2,3] Xiaolong Wang[1] Kiana Ehsani[2]
[1]UC San Diego [2]PRIOR @ Allen Institute for AI [3]University of Washington, Seattle

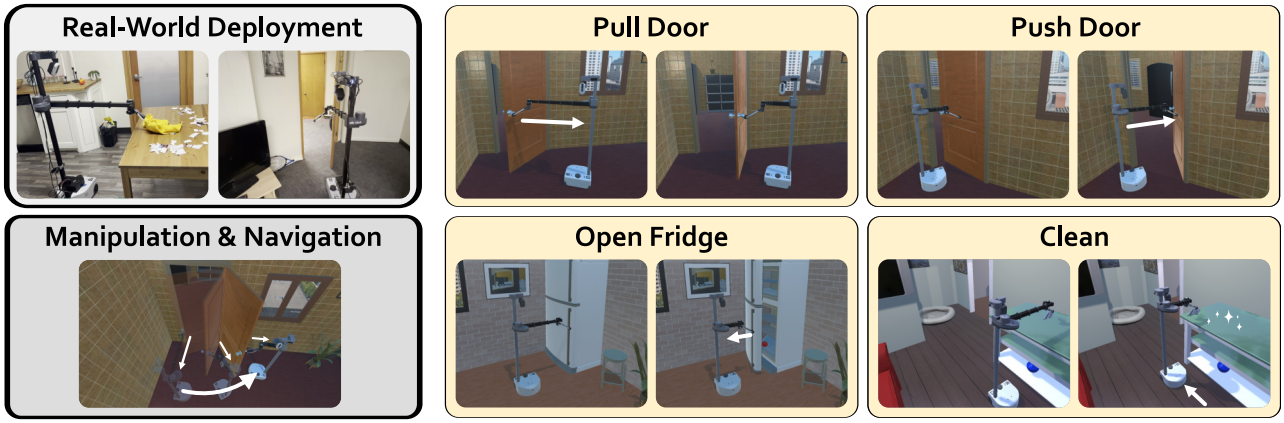

Fig. 1: **Harmonic Mobile Manipulation:** In this work, we address diverse mobile manipulation tasks integral to human's daily life. Trained in a photo-realistic simulation, Our controller effectively accomplishes tasks through harmonious mobile manipulation in a *real-world apartment* featuring a novel layout, *without any fine-tuning or adaptation*.

***Abstract*— Recent advancements in robotics have enabled robots to navigate complex scenes or manipulate diverse objects independently. However, robots are still impotent in many household tasks requiring coordinated behaviors such as opening doors. The factorization of navigation and manipulation, while effective for some tasks, fails in scenarios requiring coordinated actions. To address this challenge, we introduce, HARMONICMM, an end-to-end learning method that optimizes both navigation and manipulation, showing notable improvement over existing techniques in everyday tasks. This approach is validated in simulated and real-world environments and adapts to novel unseen settings without additional tuning. Our contributions include a new benchmark for mobile manipulation and the successful deployment with only RGB visual observation in a real unseen apartment, demonstrating the potential for practical indoor robot deployment in daily life.**

I. INTRODUCTION

The field of robot learning has traditionally treated robot navigation and manipulation as separate domains, leading to significant advancements in each. On one hand, agents have been trained to efficiently explore and navigate environments ([1], [2], [3], [4]), while on the other, substantial progress has been made in performing complex manipulation tasks, such as handling articulated objects with static arms in tabletop settings ([5], [6], [7], [8]).

However, in household settings, many tasks require coordinated body movements while simultaneously manipulating objects with our arms. To address this challenge, recent efforts have focused on enhancing robotic agents with the ability for mobile manipulation. The most prominent approach is invoking navigation and manipulation as separate modules. This separation is achieved either by utilizing a high-level semantic controller, such as a large language model ([9], [10]), or by estimating low-level subgoals using motion planners [11]. In these works, behaviors are split into distinct navigation and manipulation skills.

While this approach has proven effective for long-horizon pick-and-place tasks, it presents certain limitations. Firstly, it simply fails for tasks that demand simultaneous navigation and manipulation, such as opening a door. Without coordinated action, the robot's body may collide with the door, preventing the successful opening of the door. Secondly, the physical constraints of the robot can negatively impact task completion. For example, cleaning a large table with a robot that has limited movement range, like the Stretch robot with a 3-DOF arm, requires concurrent control of the arm and base. Thirdly, the disjoint approach is highly inefficient. Consider the act of opening a door: when we open a door, we instinctively reach for the handle while simultaneously moving towards the door. Separating arm and leg movements can have a negative impact on efficiency and performance.

To address these challenges, we introduce an approach that efficiently coordinates navigation and manipulation for complex mobile manipulation tasks. Leveraging the effectiveness of training models in procedurally generated simulated environments, as shown by recent studies [2], we train our models in the visually diverse environments of ProcTHOR [12]. Recognizing the limitations of ProcTHOR, which lacks features like door opening or table cleaning, we have expanded the simulation to include these functionalities. Our end-to-end model, which relies solely on RGB and arm proprioception, demonstrates an absolute improvement of

[*]Work done during internship at PRIOR @ Allen Institute for AI
More results on `https://rchalyang.github.io/HarmonicMM`

17.6% across four tasks compared to existing baselines and successfully transfers to real-world applications.

Our primary contributions are:

- An end-to-end learning approach that jointly optimizes navigation and manipulation, achieving an absolute improvement of 17.6% in average success rate across tasks compared to previous methods.
- Adding the support for more complex tasks, such as door opening and table cleaning, to ProcTHOR.
- Successful transfer of agents trained in simulation to real-world with only RGB visual observation.
- Introducing a new benchmark for complex mobile manipulation tasks, including opening fridges, cleaning tables, and opening doors by pulling and pushing.

II. Related Work

Embodied AI Benchmark. Over the past few years, a variety of standard benchmarks have emerged to assess progress in the field of embodied AI. These benchmarks primarily focus on high-level planning, scene understanding, and basic interaction tasks like object navigation and house rearrangement ([13], [12], [14], [15], [16], [17], [18], [19], [20]) and generally limit physical world interactions to straightforward pick-and-place actions. Deitke et al.[12] have created diverse houses for training embodied agents to navigate through various everyday scenes. Li et al.[21] tackles various household tasks using predefined motion primitives. In contrast to these benchmarks, our work extends beyond simple pick-and-place tasks. Our robot is designed to perform complex mobile manipulations, requiring tight coordination between navigation and manipulation.

Robotics Manipulation. In robotic manipulation, the focus has traditionally been on either fixed-base tabletop tasks or elementary mobile manipulations in controlled environments ([22], [23], [24], [25], [26], [27], [28], [29], [30]). Mu et al.[24] have developed benchmarks for straightforward mobile manipulation tasks in open spaces. Gu et al.[31] have demonstrated how robots can clean tables using imitation learning in a tabletop setup. While some works have explored basic mobile manipulations, such as door opening in lab settings [28], our efforts extend to more dynamic tasks requiring the robot to actively navigate and interact with its environment, pushing the boundaries of traditional mobile manipulation research.

Mobile Manipulator. In the realm of mobile manipulation, previous studies have designed task-specific mobile manipulators using optimization techniques[32] or employing Task-and-motion-planning (TAMP) methods ([33], [34]). Lew et al.[32] developed a table-cleaning controller using whole-body trajectory optimization. While effective for their intended purposes, these methods tend to be narrowly focused with limited scalability and adaptability to different scenarios. Our pipeline could apply to a wide range of tasks, beyond the constraints of task-specific methodologies and offer a more flexible solution for various mobile manipulation challenges.

Another significant line of work involves learning-based methods ([11], [35]), which offer better generalization in task completion. These methods diverge into two sub-categories: short-horizon whole-body control of robots ([36], [37], [38]), and long-horizon tasks solved at the scene level with predefined motion primitives in an iterative or two-stage manner ([39], [11], [40], [41], [9], [25], [26]). Yokoyama et al.[39], Xia et al.[11] and Ahn et al.[9] exemplify solving long-horizon mobile manipulation tasks using predefined navigation and manipulation skills, either iteratively or in a two-stage approach. Notably, our work proposes a unified learning framework that integrates navigation and manipulation in a seamless manner, addressing the limitations of predefined primitives by focusing on learning adaptable skills for complex, everyday tasks.

III. Daily Mobile Manipulation Task Suite

Humans, on a daily basis, perform long-horizon tasks that necessitate close coordination between navigating the body and manipulating objects within the environment. While we can effortlessly coordinate actions between our hands and feet, teaching robots to do the same is a very challenging problem. To study this problem, first, we introduce a suite of tasks, named the Daily Mobile Manipulation Task Suite, consisting of atomic tasks that one might want a household robot to perform, as in Fig. 1. These tasks require an agent to navigate their world while manipulating objects.

1) Opening Door. In most living spaces, navigating through doors is a key task. This involves identifying and operating the door handle by pulling or pushing while adjusting the body position to avoid collision with the door. The suite includes both *Opening Door (Push)* and *Opening Door (Pull)*.

2) Cleaning Table. A common chore is cleaning surfaces like tables and counters which requires coordinated movement to clean the entire surface area effectively. The suite features *Cleaning Table* to represent such activities.

3) Opening Fridge. Operating household appliances is crucial for a robotic assistant. This task involves locating and opening an appliance like a fridge, including navigation and manipulation. The suite includes *Opening Fridge* as an example of such tasks.

Training Environments. Our training leverages the ProcTHOR[12] environment, preparing agents for our task suite with an emphasis on bridging the simulation-to-reality (Sim2Real) gap through diverse and photorealistic simulations. For *Opening Door* tasks, we use 700 houses featuring two-room configurations with openable doors. *Cleaning Table* is trained with 2000 houses, varying from one to three rooms, each equipped with a table. For fridge-opening scenarios, 1400 houses with realistically furnished kitchens are employed. Environmental factors like textures and lighting are randomized to enhance generalization capabilities, ensuring effective real-world policy application.

IV. Harmonic Mobile Manipulation (HarmonicMM)

Efficiently coordinating navigation and manipulation in mobile robots presents a significant challenge, particularly when using RL in large action spaces. Prior studies often

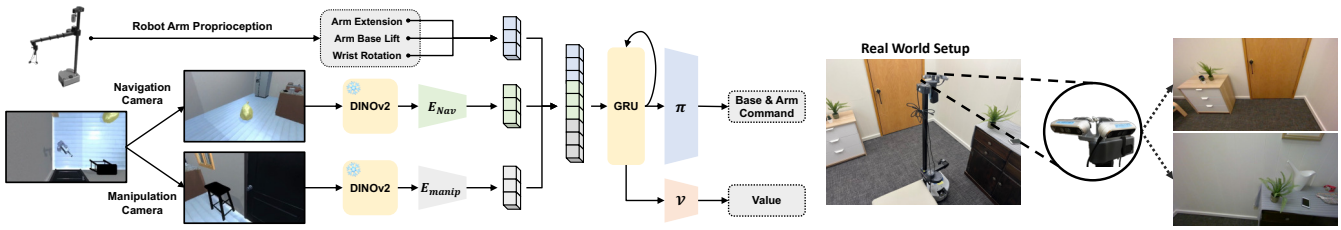

Fig. 2: **HARMONICMM Network Architecture (Left):** Our HARMONICMM controller takes robot proprioception and multi-view visual observations as input and output navigation and manipulation commands at the same time. **Real Visual Observations (Right):** Our robot is shown on the Left and the observations from *Nav Cam* and *Manip Cam* are shown on the Top Right and Bottom Right respectively.

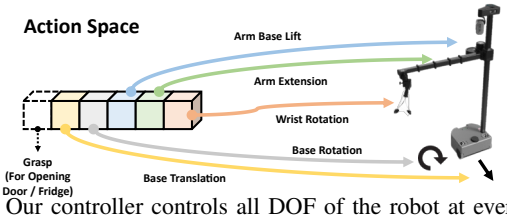

Fig. 3: Our controller controls all DOF of the robot at every step.

split tasks into discrete navigation and manipulation stages, or alternately cycle between these modes, limiting the action space at any given timestep ([9], [26], [17], [11], [36], [42]). This formulation of the action space prevents the robot from performing challenging tasks such as those in Daily Mobile Manipulation Task Suite. Our approach employs the Stretch [43], which navigates and manipulates concurrently, leveraging a 5-dim continuous action space for integrated base and arm movements. As shown in Fig. 3, our action space includes base movements (translation and rotation), arm height and extension adjustments, and wrist rotation, with an additional dimension for grasp actions in specific tasks like door or fridge opening.

A. *Simulated Environment.*

Our simulation utilizes ProcThor [12] for its superior visual diversity, addressing a gap in environments like IsaacGym [44] and Mujoco [45] that lack extensive visual variation critical for real-world skill transfer. To enable the robot to learn the diverse tasks described in Sec III, We augmented the simulation with procedural door/fridge opening logic and photo-realistic table cleaning functionality.

Procedural Door/Fridge Opening: One of our contributions is modifying ProcThor to enable physically realistic articulation manipulation. In the ProcThor[12] simulation environment, doors and fridges are treated as semantic objects, and their open and closed states are not governed by any physical rules. Instead, we estimate their status at each simulation step based on the robot's actions. For pushing actions, the system calculates the direction the robot's end-effector is moving towards. If this direction aligns with the door's normal vector, the door is opened incrementally in the same direction as the end-effector's movement. In the case of pulling actions, the door's movement is determined after the robot command is executed. The door opens in small segments if the end-effector's movement direction aligns with the door's normal vector, provided the robot has already secured the doorknob with a magnetic grasper. Additionally, collision detection algorithms are in place to prevent the door from colliding with the robot's body during these interactions.

Photo-realistic Table Cleaning: We developed a highly realistic table-cleaning simulation in ProcThor[12]. This process involves randomly placing a variety of lifelike dirt objects on a designated table at the start of each episode and attaching a sponge to the robot's end-effector at the start of the episode. During each simulation step, any dirt objects that come into contact with the attached sponge are eliminated. The quantity of dirt objects removed is then tallied for the purpose of reward computation in RL training. This method of simulating photorealistic dirt objects not only allows the robot to visually detect dirt on the simulated table but also to recognize the removal of dirt upon cleaning.

B. *Robot Perception*

We equip the robot with both RGB visual observations as well as the proprioceptive state of the robot, which is available on most robots and easy to obtain.

Multi-camera RGB Observations: In contrast to prior work utilizing a mix of RGB, depth cameras, and LIDAR for mobile manipulation, our approach exclusively uses RGB cameras due to their cost-efficiency, power efficiency, and robustness against environmental noise. RGB cameras have been proven to provide ample scene semantics for navigation and manipulation tasks, supported by advances that have narrowed the simulation-to-reality visual gap ([11], [38], [9], [46], [12], [2], [1]). Our robot is equipped with two RGB cameras: a Navigation Camera (*Nav Cam*) for forward observation and a Manipulation Camera (*Manip Cam*) aimed at the robot's arm, enabling multi-view environmental awareness, as shown in Fig. 2. Despite the photorealism in simulation ([12], [13]), we address the remaining Sim2Real visual discrepancies through augmentation, further detailed in the appendix (Sec VII).

Proprioception: For tasks requiring object manipulation in Daily Mobile Manipulation Task Suite, our controller utilizes a 5-dimensional proprioception vector, detailing the arm base's lift, arm extension, and wrist rotation, to enhance manipulation capabilities.

C. HARMONICMM *Architecture*

As shown in Fig. 2, HARMONICMM starts with a DINOv2 visual backbone that processes input from both RGB cameras. These inputs are further refined by two convolutional visual encoders, E_{Nav} and E_{Manip}, for navigation and manipulation visual observations, respectively, to generate specific visual embeddings. Visual embeddings are concatenated with the robot's proprioception and then fed into a recurrent unit to add

TABLE I: **Task Completion Evaluation (Top):** Our method outperforms baselines with higher Success Rates and higher Progress. **Efficiency Evaluation (Bottom):** Our method outperforms baselines with higher Progress Speed and shorter Episode Length (Eps-Length).

	Cleaning Table		Opening Door (Push)		Opening Door (Pull)		Opening Fridge	
	Success Rate (%)	Progress (%)	Success Rate (%)	Progress (%)	Success Rate (%)	Progress (%)	Success Rate (%)	Progress (%)
Two-Stage	0.0 ± 0.0	4.5 ± 0.7	0.0 ± 0.0	8.5 ± 9.2	0.0 ± 0.0	0.85 ± 0.15	0.0 ± 0.0	0.4 ± 1.1
ReLMoGen[11]	25.1 ± 20.4	69.6 ± 9.6	51.7 ± 31.4	74.5 ± 15.4	40.7 ± 10.7	62.6 ± 8.6	0.0 ± 0.0	6.5 ± 8.8
HARMONICMM (Ours)	$\mathbf{36.7 \pm 22.9}$	$\mathbf{76.0 \pm 11.3}$	$\mathbf{80.0 \pm 19.9}$	$\mathbf{89.7 \pm 5.0}$	$\mathbf{51.2 \pm 12.2}$	$\mathbf{69.8 \pm 2.7}$	$\mathbf{20.0 \pm 3.1}$	$\mathbf{46.2 \pm 5.4}$

	Cleaning Table		Opening Door (Push)		Opening Door (Pull)		Opening Fridge	
	Progress Speed↑	Eps-Length↓	Progress Speed↑	Eps-Length↓	Progress Speed↑	Eps-Length↓	Progress Speed↑	Eps-Length↓
Two-Stage	0.05 ± 0.01	500.0 ± 0.0	0.09 ± 0.09	500.0 ± 0.0	0.00 ± 0.00	500.0 ± 0.0	0.0 ± 0.0	500 ± 0
ReLMoGen[11]	1.01 ± 0.42	443.1 ± 53.2	1.87 ± 1.10	352.8 ± 110.7	1.28 ± 0.49	401.0 ± 44.2	0.065 ± 0.088	500 ± 0
HARMONICMM (Ours)	$\mathbf{1.34 \pm 0.56}$	$\mathbf{410.1 \pm 58.7}$	$\mathbf{3.98 \pm 2.05}$	$\mathbf{218.1 \pm 117.4}$	$\mathbf{1.64 \pm 0.59}$	$\mathbf{370.2 \pm 54.8}$	$\mathbf{0.847 \pm 0.140}$	$\mathbf{446.6 \pm 19.8}$

memory, followed by a linear layer to produce the robot action. More details can be found in Sec VIII. Leveraging a pre-trained visual backbone, as demonstrated by previous research within AI2-THOR ([1], [12], [2], [47]), offers significant advantages in performance and real-world generalization. This choice is informed by the encoder's proven ability to capture detailed semantic information from visual observations and its success in bridging the simulation-to-reality gap, thanks to its self-supervised training on a diverse web-scale dataset.

D. Reward Function

We use the following reward function for all tasks:

$$R = w_{\text{nav}} * R_{\text{nav}} + w_{\text{manip}} * R_{\text{manip}} + w_{\text{efficiency}} * R_{\text{efficiency}}$$

where w_* are coefficients for different reward terms. We provide high-level descriptions of each term below and more details are provided in Sec X.

Navigation Reward R_{nav}: This reward is proportional to the distance the robot moved towards the target object (door/fridge for the *Opening Door/Fridge*, or table for *Cleaning Table*). It provides a reach-target reward when the robot reaches the target and gets cut off afterward.

Manipulation Reward R_{manip}: This reward contains two sub-rewards. The first sub-reward encourages the robot to move its end-effector towards the doorknob, the center of the table, and the fridge handle for *Opening Door*, *Cleaning Table*, and *Opening Fridge* tasks respectively. The second sub-reward is proportional to the progress the robot made towards task completion and offers a significant bonus upon successful task completion. Task progress is measured by the degree of openness of the door or fridge for *Opening Door/Fridge* tasks, and the percentage of dirt removed from the table for *Cleaning Table* tasks.

Efficiency Reward $R_{\text{efficiency}}$: This reward incentivizes the robot to complete the task efficiently while maintaining a safe distance from the object and minimizing excessive movement of its end-effector.

V. EXPERIMENT

We evaluated our method on Daily Mobile Manipulation Task Suite both in simulation and in the real world. We deployed the policy (learned in simulation) in a real apartment with multiple rooms *without any adaptation or fine-tuning*.

A. Training and Evaluation Details

We trained the policies for all methods for each task using DD-PPO([48]) for 10^7 steps using AllenAct[49]. Each experiment was repeated with 3 different seeds. We evaluated the policies on 1200 episodes (300 per task). Our evaluations were conducted in 350 unseen houses with novel layouts.

Robot Initialization Position in Scene: Recognizing the impact of the initial position of the robot on mobile manipulation, we optimized the robot's start points to improve exploration efficiency and task performance. If the robot is too close to the object, it struggles to adjust its pose, while, initializing the robot too far from the object negatively impacts its ability to learn required skills. We place the robot $1-4$ meters away from the table for *Cleaning Table*, and $1-3$ meters away from the door/fridge for *Opening Door/Fridge* tasks.

Metrics: We evaluate the performance of all methods with:

Progress: This metric (as defined in IV-D) indicates how much progress the agent made toward completing the task. Our tasks are complex and long-horizon, making this an insightful metric, especially where the agent does not reach a high completion rate but makes meaningful progress.

Success Rate: Success was defined by the task progress: over 90% for door opening, 70% for fridge opening (since normally fridges are located in a narrower space and humans don't fully open the fridge), and 75% for table cleaning.

Episode Length: Reflects the average steps required for task completion, with episode length capped at 500 steps.

Progress Speed: This metric reflects the efficiency of the agent in making progress toward task completion. It is calculated by: $\frac{\text{Task Progress}}{\text{Episode Length}/\text{Max Episode Length}}$

B. Performance of HARMONICMM in Simulation

Our experiment results demonstrate that HARMONICMM achieves an average success rate of 47% across all tasks (As in Table I). Specifically, our model attains a high success rate of 80% for the task of *Opening Door (Push)* in unseen houses, and achieves average progress rates of 76%, 70%, and 46% for *Cleaning Table*, *Opening Door (Pull)*, and *Opening Fridge*, respectively, indicating significant progress in each task. This achievement is especially impressive given the complex, long-horizon nature of these tasks.

The success rates across tasks vary due to how the robot interacts with its environment. *Opening Door (Push)* has a high success rate as it's less affected by environmental factors: the robot can use the space created by the opened door to continue the task. However, *Opening Door (Pull)* and

Fig. 4: **Real World:** We deployed the learned controller in a real apartment with a novel layout. Each row shows a single trajectory (from left to right) corresponding to Opening Door (Pull), Opening Door (Pull), Opening Door (Push), and Cleaning Table, respectively.

TABLE II: **Proprioception Ablation:** We ablate the necessity of proprioception information on Opening Door tasks. Proprioception information is necessary for tasks requiring precise placement of the end-effector and helpful for other tasks.

	Opening Door (Push)				Opening Door (Pull)			
	Success Rate (%)	Progress (%)	Progress Speed↑	Eps-Length↓	Success Rate (%)	Progress (%)	Progress Speed↑	Eps-Length↓
HarmonicMM w/o proprio	74.6 ± 1.1	83.9 ± 1.4	2.51 ± 0.17	263.4 ± 15.0	5.2 ± 9.0	11.0 ± 18.9	0.16 ± 0.28	489.5 ± 18.1
HarmonicMM	$\mathbf{80.0 \pm 19.9}$	$\mathbf{89.7 \pm 5.0}$	$\mathbf{3.98 \pm 2.05}$	$\mathbf{218.1 \pm 117.4}$	$\mathbf{51.2 \pm 12.2}$	$\mathbf{69.8 \pm 2.7}$	$\mathbf{1.64 \pm 0.59}$	$\mathbf{370.2 \pm 54.8}$

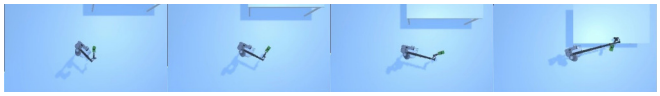

Fig. 5: **Cleaning Table in Simulation**

Opening Fridge are more challenging. These tasks reduce the available space as they progress, especially in narrow areas, leading to lower success rates despite significant progress. *Cleaning Table* involves complex navigation due to obstacles and furniture around the table, requiring intricate movement patterns. This complexity accounts for its lower performance compared to door-related tasks. For *Opening Fridge*, the task's difficulty arises from the smaller size of fridge doors and the need for precise end-effector placement. The often confined placement of fridges in rooms adds to the challenge, affecting the robot's ability to navigate and manipulate effectively.

Comparison with the previous approaches. We hypothesize that jointly learning and executing navigation and manipulation for mobile manipulation tasks achieves better performance than considering them disjointly. To validate this, we compared our method with two common approaches widely used by the community. For a fair comparison, we use the same network architecture stated in Sec IV and the same hyperparameters for baselines.

One common practice, which we note as *Two-Stage* baseline, in embodied AI for mobile manipulation ([17], [35], [50]) is to decompose the tasks into separate stages of manipulation and navigation. In this work, our *Two-Stage* baseline uses a roughly similar high-level structure as OVMM[17] in which the robot navigates to the target object and, after reaching the target, switches to the manipulation. While this approach works for a variety of pick-and-place tasks, it fails for more complex tasks, such as those presented in our benchmark. Table I shows the very low success rate and progress of the *Two-Stage* baseline.

Another common approach is executing manipulation and navigation alternatively. At every step controller invokes separately obtained manipulation or navigation skills. We adapted ReLMoGen [11] to support the embodiment of the Stretch robot. This baseline outputs the target end-effector pose, target base pose in the current robot frame, and choice over navigation or manipulation at every step. Although this approach achieves much higher results than the Two-Stage baseline, there is still a significant 17.6% (37% relative) drop in success rate compared to HarmonicMM.

HarmonicMM is Efficient. Confirming our hypothesis, HarmonicMM not only boosts performance but also enhances efficiency. In table cleaning, our method produces efficient motion of extending arms while approaching the table, as shown in Fig. 5. Table I shows that HarmonicMM makes 32.2% more progress towards completing the task at each step compared to the baselines on *Cleaning Table*, 113.4% on *Opening Door (Push)*, and 27.6% on *Opening Door (Pull)*.

C. HarmonicMM *transfers to the real world.*

Our learned policies were evaluated in a real apartment for *Opening Door (Pull)*, *Opening Door (Push)*, and *Cleaning Table* tasks, showing promising outcomes as in Table III and Fig. 4. The Opening Fridge task is not evaluated since the magnetic seal of the fridge is too strong for our robot to

TABLE III: **Real World Evaluation**

Task	Success Rate	# of trials
Cleaning Table	66.6%	12
Opening Door (Pull)	60.0%	15
Opening Door (Push)	70.0%	10

TABLE IV: **Ablation Study on Opening Door (Pull)**

Opening Door (Pull)	Success Rate	Progress	Reach Door
No Pretrained DINOv2	0%	0%	66.3%
Nav Cam Only	27.3%	47.9%	93%
Manip Cam Only	0%	0.235%	95%
Adapted Skill Trans	0%	0.%	29.7%

pull. To compensate for the simulation's simplified magnetic gripper grasping, a heuristic function for doorknob grasping was implemented in the real world, effectively bridging the simulation-to-reality gap. When our controller initiates a grasp action, the heuristic function is invoked to grasp the doorknob.

Our controller successfully pulled the door fully open in 9 out of 15 attempts in three different rooms. Interestingly, we observed that HARMONICMM exhibited two distinct styles of pulling the door to adapt to the layout differences in each room. When there was enough open space around the door, it continued pulling until the door was fully opened, as shown in the 1st row in Fig. 4. However, when the door was located next to a wall, the agent first pulled the door to start the opening process and then switched to pushing to fully open it (2nd row, Fig. 4). These emergent behaviors showcase the spatial reasoning and scene-understanding abilities of our controller. Similarly, our controller successfully pushed doors fully open 7 times in 4 different rooms out of 10 trials.

For the cleaning table task, we placed the table in four different locations across two rooms, with evenly distributed paper pieces on it for the robot to clean. Our robot navigated to the table and continuously moved its end-effector to clean the entire table, as shown in Fig. 4. Out of 12 trials, our controller successfully cleaned 66.6% of the table's surface area in eight instances.

D. Ablation Study of HARMONICMM

More efficient in longer horizon tasks. we analyzed the success rate of both HARMONICMM and ReLMoGen[11] in the *Cleaning Table* and *Opening Door (Push)* tasks across varying initial distances from the target object of interest (aka. table and door), ranging from $1 \sim 4$m. As in Fig. 6, we observed that the improvement of HARMONICMM over the baseline becomes larger as the initial distance to the target object increases. This trend underscores HARMONICMM's enhanced efficiency, particularly in tasks with longer horizons, where the coordinated integration of navigation and manipulation is pivotal in successful task completion.

Proprioception. Our comparison of HARMONICMM against a proprioception-less variant in door-opening tasks illustrates the proprioception significance (in Table II). We also observe -33.7% / -20% success rate drop, and -8.3% / -26.2% progress drop for *Cleaning Table* and *Opening Fridge* without proprioception. The absence of proprioceptive feedback

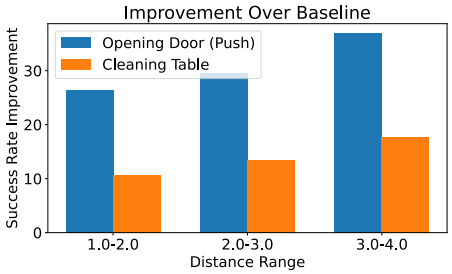

Fig. 6: **Improvement over Range:** The improvement HARMONICMM obtained over ReLMoGen grows as the distance increases.

notably diminishes performance, especially in *Opening Door (Pull)*, where precise end-effector positioning is necessary for pre-grasping the knob. This effect is less pronounced but still present in *Opening Door (Push)*, emphasizing proprioception's overall contribution to task performance and efficiency, even when precise manipulation is not as critical.

Multi-View Visual Observation. We had initial experiments with a single-camera setup showing a significant performance drop compared with multi-camera ones (provided in Table IV). This indicates that complex tasks require simultaneous visual input for both navigation and manipulation, validating our approach of integrated navigation and manipulation.

Pretrained Visual Encoder We substituted the DINOv2 encoder with a trainable CNN. This modification led to a significant drop in success rate and progress (Tab.IV), underscoring the critical role of a pretrained visual encoder

Transformer Variant. We adapted Skill Transformer[51] and evaluated it on the most complicated task *Open Door (Pull)* (As in Table IV). It performs poorly, likely due to the low-sample efficiency of training a large transformer with a ResNet18 encoder from scratch (following the original paper), which is unsuitable for our online RL setting. The original paper trains different low-level skills separately and uses a skill transformer for coordinating skills, which is significantly different from our task setting.

VI. CONCLUSION

In this work, we developed a simulation for complex mobile manipulation tasks, ranging from opening doors/fridges to cleaning tables in daily scenes, using ProcThor[12]. Our HARMONICMM enables robots to solve these tasks using only RGB visual observation and proprioception of the robot through end-to-end learning. Our pipeline significantly outperforms previous baselines in simulation and has successfully transferred to real-world apartments with novel layouts, without any fine-tuning. Code for our benchmark and our method will be released.

Limitation: Despite its success, our HARMONICMM faces limitations related to the kinematic and physical capabilities of the robot, restricting its ability to handle more dynamic real-world tasks, such as lifting heavy objects.

Acknowledgements. We would like to thank the Thor Team: Winson Han, Eli VanderBilt, and Alvaro Herrasti, for their support in setting up the simulations. Special thanks to Winson Han for his expertise and assistance in the visualization of this work.

REFERENCES

[1] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[2] M. Deitke, R. Hendrix, A. Farhadi, K. Ehsani, and A. Kembhavi, "Phone2proc: Bringing robust robots into our chaotic world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9665–9675.

[3] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," 2022.

[4] R. Yang, G. Yang, and X. Wang, "Neural volumetric memory for visual locomotion control," in *Conference on Computer Vision and Pattern Recognition 2023*, 2023. [Online]. Available: https://openreview.net/forum?id=JYyWCcmwDS

[5] R. Mendonca, S. Bahl, and D. Pathak, "Alan: Autonomously exploring robotic agents in the real world," 2023.

[6] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "Mimicplay: Long-horizon imitation learning by watching human play," *arXiv preprint arXiv:2302.12422*, 2023.

[7] 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.

[8] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation," *Conference on Robot Learning (CoRL)*, 2022.

[9] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can and not as i say: Grounding language in robotic affordances," in *arXiv preprint arXiv:2204.01691*, 2022.

[10] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[11] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Leveraging motion generation in reinforcement learning for mobile manipulation," 2021.

[12] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi, and R. Mottaghi, "Procthor: Large-scale embodied ai using procedural generation," in *NeurIPS*, 2022.

[13] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: an interactive 3d environment for visual AI," *arXiv*, 2017.

[14] K. Yadav, J. Krantz, R. Ramrakhya, S. K. Ramakrishnan, J. Yang, A. Wang, J. Turner, A. Gokaslan, V.-P. Berges, R. Mootaghi, *et al.*, "Habitat challenge 2023," https://aihabitat.org/challenge/2023/, 2023.

[15] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva, *et al.*, "Habitat-matterport 3d semantics dataset," *arXiv preprint arXiv:2210.05633*, 2022.

[16] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," *ICCV*, 2019.

[17] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner, *et al.*, "Homerobot: Open-vocabulary mobile manipulation," 2023.

[18] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min, *et al.*, "Habitat 3.0: A co-habitat for humans, avatars and robots," 2023.

[19] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi, "Manipulathor: A framework for visual object manipulation," 2021.

[20] C. Gan, S. Zhou, J. Schwartz, S. Alter, A. Bhandwaldar, D. Gutfreund, D. L. K. Yamins, J. J. DiCarlo, J. McDermott, A. Torralba, and J. B. Tenenbaum, "The threedworld transport challenge: A visually guided task-and-motion planning benchmark for physically realistic embodied ai," 2021.

[21] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, *et al.*, "BEHAVIOR-1k: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=_8DoIe8G3t

[22] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019. [Online]. Available: https://arxiv.org/abs/1910.10897

[23] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," in *International Conference on Learning Representations*, 2023.

[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," 2021.

[25] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," 2023.

[26] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023.

[27] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng, "Large language models as general pattern machines," in *Proceedings of the 7th Conference on Robot Learning (CoRL)*, 2023.

[28] Y. Urakami, A. Hodgkinson, C. Carlin, R. Leu, L. Rigazio, and P. Abbeel, "Doorgym: A scalable door opening environment and baseline agent," 2022.

[29] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," 2022.

[30] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," 2021.

[31] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, P. Sundaresan, P. Xu, H. Su, K. Hausman, C. Finn, Q. Vuong, and T. Xiao, "Rt-trajectory: Robotic task generalization via hindsight trajectory sketches," 2023.

[32] T. Lew, S. Singh, M. Prats, J. Bingham, J. Weisz, B. Holson, X. Zhang, V. Sindhwani, Y. Lu, F. Xia, P. Xu, T. Zhang, J. Tan, and M. Gonzalez, "Robotic table wiping via reinforcement learning and whole-body trajectory optimization," 2022.

[33] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 639–646.

[34] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," 2020.

[35] J. Gu, D. S. Chaplot, H. Su, and J. Malik, "Multi-skill mobile manipulation for object rearrangement," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Z3IClM_bzvP

[36] C. Sun, J. Orbik, C. Devin, B. Yang, A. Gupta, G. Berseth, and S. Levine, "Fully autonomous real-world reinforcement learning with applications to mobile manipulation," 2021.

[37] J. Hu, P. Stone, and R. Martín-Martín, "Causal policy gradient for whole-body mobile manipulation," 2023.

[38] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," *Conference on Robot Learning (CoRL)*, 2022.

[39] N. Yokoyama, A. Clegg, J. Truong, E. Undersander, T.-Y. Yang, S. Arnaud, S. Ha, D. Batra, and A. Rai, "Asc: Adaptive skill coordination for robotic mobile manipulation," 2023.

[40] C. Li, F. Xia, R. Martin-Martin, and S. Savarese, "Hrl4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators," 2019.

[41] 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, jul 2022. [Online]. Available: https://doi.org/10.1109%2Flra.2022.3188109

[42] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," 2023.

[43] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich, "The design of stretch: A compact, lightweight mobile manipulator for indoor human environments," 2022.

[44] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym:

High performance gpu-based physics simulation for robot learning," 2021.
[45] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.
[46] A. Loquercio, A. Kumar, and J. Malik, "Learning visual locomotion with cross-modal supervision," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7295–7302.
[47] R. H. J. S. L. W. K.-H. Z. K. P. S. Y. K. W. H. A. H. R. K. D. S. E. V. A. K. Kiana Ehsani, Tanmay Gupta, "Imitating shortest paths in simulation enables effective navigation and manipulation in the real world," *arXiv*, 2023.
[48] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," 2020.
[49] L. Weihs, J. Salvador, K. Kotar, U. Jain, K.-H. Zeng, R. Mottaghi, and A. Kembhavi, "Allenact: A framework for embodied ai research," *arXiv preprint arXiv:2008.12760*, 2020.
[50] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," 2022.
[51] X. Huang, D. Batra, A. Rai, and A. Szot, "Skill transformer: A monolithic policy for mobile manipulation," 2023.

Appendix

VII. Visual Augmentation

Our applied augmentations include: **ColorJitter**: Adjusts brightness (0.4), contrast (0.4), saturation (0.2), and hue (0.05); **GaussianBlur**: Applies a blur effect with a kernel size of (5,9) and sigma range of (0.1,2); **RandomResizedCrop**: Resizes the input with a scale range of (0.9,1); **RandomPosterize**: Reduces color depth, applied with varying bits (7,6,5,4) and a probability of 0.2 for each setting; **RandomAdjustSharpness**: Enhances or reduces the sharpness with a factor of 2, applied with a probability of 0.5.

VIII. Network Architecture

Our RGB observations initially get processed by DinoV2 model, generating $768 \times 16 \times 16$ features which pooled to a reduced dimension of $768 \times 7 \times 7$. The processed features are passed through separate visual encoders. This results in two distinct sets of $16 \times 7 \times 7$ latent visual features corresponding to each camera view. The key hyperparameters are as follows:

Hyperparameter	Value
RGB Input	384×224
Pretrained Visual Encoder	dinov2_vits14
E_{Nav}	2 Conv(1x1, 16 channels)
E_{Manip}	2 Conv(1x1, 16 channels)
GRU	1 GRU layer with 512 units
Policy(π)	1 FC with 512 units
Value(v)	1 FC with 512 units
Proprioception dim	5
latent feature dim	$7 * 7 * 16 * 2 + 5 = 1573$

IX. RL training Hyperparameters

Hyperparameter	Value
Non-linearity	ReLU
Policy initialization	Standard Gaussian
# of samples per iteration	640
Discount factor	.99
Parallel Environment	20
Batch size	640
Optimization epochs	4
Clip parameter	0.1
Policy network learning rate	5e-5
Value network learning rate	5e-5
Entropy	0.0025

X. Reward Function

In this section, we provide the formulation of our reward and hyperparameters for different tasks in Sec IV-D.

Navigation Reward R_{nav}:

$$R_{\text{nav}} = \delta(w_{\text{nav shaping}} max(d_{\text{closest}} - d_{\text{current}}, 0) + \gamma R_{\text{reach target}})$$

The variable δ is 1 if the robot hasn't reached the target position and 0 otherwise. Similarly, γ is 1 if the robot reaches the target at the current step, and 0 otherwise. d_{closest} is the closest distance robot reached before, d_{current} is the current distance between robot and target object. For all tasks, we use $w_{\text{nav shaping}} = 2, R_{\text{reach target}} = 2, w_{\text{nav}} = 1$.

Manipulation Reward R_{manip}:

$$\begin{aligned} R_{\text{manip}} =& \delta(w_{\text{manip shaping}} R_{\text{manip shaping}}) \\ &+ w_{\text{progress}} \Delta P + \gamma R_{\text{finish task}} \end{aligned}$$

$$\begin{aligned} R_{\text{manip shaping}} =& exp(-5 \times d_{\text{ee current}}) \\ &\times 1000 \times max(d_{\text{ee cloest}} - d_{\text{ee current}}, 0)) \end{aligned}$$

The variable δ is 1 if the end effector has not reached the target region, and 0 otherwise. Similarly, γ is 1 if the robot completes the task at the current step, and 0 otherwise.

d_{closest} is the closest distance the end-effector reached before towards the specified region of the target object, d_{current} is the current distance between the end-effector and the specified region of the target object, P is the task progress. For all tasks, we use $w_{\text{manip}} = 1, w_{\text{manip shaping}} = 0.02, R_{\text{finish task}} = 20$. For opening door/fridge tasks, we use $w_{\text{progress}} = 80$. For the cleaning table task, we use $w_{\text{progress=100}}$. For opening the door (pull) and opening the fridge task, we provide an additional $w_{grasp} = 2$ for grasping the door knob or the edge of the fridge. We consider the grasp successful if the agent has issued the grasp action and the distance between the end-effector and object is smaller than 0.2m.

Efficiency Reward $R_{\text{efficiency}}$:

$$\begin{aligned} R_{\text{efficiency}} =& R_{\text{step penalty}} \\ &+ w_{\text{ee moved}} ||d_{\text{ee moved}}|| + \gamma R_{\text{invalid action}} \end{aligned}$$

where $d_{\text{ee moved}}$ is the distance end-effector moved at the current step, and γ is defined as:

$$\gamma = \begin{cases} 1, & \text{if current action failed} \\ 0, & \text{otherwise} \end{cases}$$

Current action fails if the robot collides with semantic objects in the apartment in simulation, such as a wall or table. For all tasks, we use $R_{\text{step penalty}} = -0.01, w_{\text{ee moved}} = -0.01, R_{\text{invalid action}} = -0.01, w_{\text{efficiency}} = 1$

Hardware Platform: Our simulation and real-world experiment are conducted with Stretch Robot [43] and we use RealSense D455 for our RGB observation.

Future Works: In future work, we aim to enhance the capabilities of HarmonicMM by integrating more complex and dynamic tasks into our task suite. We aim to explore the potential of HarmonicMM in environments with even longer task horizons and more varied challenges.