2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

M-EMBER: Tackling Long-Horizon Mobile Manipulation via Factorized Domain Transfer

Bohan Wu
Department of Computer Science
Stanford University
Stanford, CA, USA
bohanwu@stanford.edu

Roberto Martín-Martín
Department of Computer Science
The University of Texas at Austin
Austin, TX, USA
robertomm@utexas.edu

Li Fei-Fei
Department of Computer Science
Stanford University
Stanford, CA, USA
feifeili@cs.stanford.edu

***Abstract*— In this paper, we propose a novel method to create visuomotor mobile manipulation solutions to long-horizon activities. We propose to leverage the recent advances in robot simulation to train robust visual solutions in simulation that can transfer to the real world. While previous works have shown success applying this procedure to autonomous visual navigation and stationary manipulation, applying it to long-horizon visuomotor mobile manipulation is still an open challenge that demands both perceptual and compositional generalization of multiple skills. In this work, we develop Mobile-EMBER, or M-EMBER, a factorized method that decomposes a long-horizon mobile manipulation activity into a repertoire of primitive visual skills, reinforcement-learns each skill in simulation, and composes these skills to a long-horizon mobile manipulation activity. On a real mobile manipulation robot, we find that M-EMBER completes a long-horizon mobile manipulation activity, `cleaning_kitchen`, achieving over 50% success rate. This requires successfully planning and executing five factorized, learned visual skills, in sequences of up to 48 skills long.**

I. Introduction

Mobile manipulators, robots combining locomotion and interaction capabilities, have the potential to undertake multiple long-horizon activities in human environments. Different from short-horizon stationary manipulation such as pushing or grasping, long-horizon mobile manipulation activities require the correct combination of multiple sensorimotor skills to be accomplished. Moreover, given the large variability in human environments combined with the challenge of moving the base between interactions, true mobile manipulation solutions for the real world have additional demands in generalization and ask for new approaches to learning general solutions.

To acquire generalized visuomotor behaviors for *stationary manipulation* in the real world, the robot learning community has resorted to two main procedures: 1) training in simulation [1–8], or 2) training directly from real-world visual datasets [9–15]. This latter approach has been favored lately, even though the generalization obtained is restricted to that demonstrated in the datasets. In long-horizon *mobile manipulation*, however, the breadth of generalization demanded extends beyond objects. This, combined with the length and compositional variability of each activity (i.e., the same activity may require a different ordering of the same skills to be achieved), renders collecting a sufficiently broad distribution of real-world data rather unfeasible. On the other hand, reaching the necessary generalization for real-world mobile manipulation could be obtained from simulation.

When it comes to domain transfer, multiple solutions have been proposed for visual *stationary manipulation* and *navigation*, but they fall short when applied to mobile manipulation. The most common approach is to try to close the domain gap [16–20]. While successful in stationary manipulation and navigation, these methods may not be sufficient for long-horizon mobile manipulation that demands not only perceptual generalization but also compositionality in the solution. Other methods have also achieved success in domain transfer by choosing input modalities that have lower domain gap [3–7, 21]. While sufficing for navigation and some stationary manipulation, it is unclear if the input modalities chosen in these methods are sufficient for the fine-grained skills involved in many mobile manipulation activities. Finally, a family of adaptive learning [22–28] and system identification [29–33] algorithms have also achieved success in domain transfer in legged locomotion. While these methods close the domain gap in dynamics and action-state transition, it is not yet clear whether these methods can learn visuomotor solutions for a mobile manipulator using only onboard sensors.

To achieve long-horizon mobile manipulation visual solutions in the real-world, we propose Mobile-EMBER, or "M-EMBER"– a factorized method based on the EMBER framework [34]. Concretely, an activity is first factorized into a repertoire of primitive visuomotor skills, and M-EMBER reinforcement-learns each skill in simulation, and transfers and recomposes these skills into a real-world long-horizon mobile manipulation solution for the activity, achieving levels of robustness beyond what EMBER could do (see Sec. V). Thanks to the factorization of skills, M-EMBER copes with initial and task conditions and is able to handle mobile manipulation activities. We demonstrated with extensive evaluations on a real-world mobile manipulator that M-EMBER can complete a complex long-horizon activity (`cleaning_kitchen`) with 53% success by learning five different visuomotor robust skills in simulation, concatenating them autonomously into sequences of up to 48 skills, and generalizing more robustly than existing state-of-the-art solutions.

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

II. Related Work

A. Robot learning via domain transfer

Domain transfer has a rich history in stationary robotic manipulation [1–8], navigation [35–40], and legged locomotion [22–28] [29–33]. This includes the use of photorealism (e.g. photorealistic rendering [1, 2, 35, 41–43] or Generative Adversarial Networks [41, 44–46]) and domain randomization [16–20] to close the domain gap. In comparison, M-EMBER develops factorized domain transfer for long-horizon mobile manipulation that demands both perceptual generalization as well as compositional generalization that can factorize and reuse learned visuomotor skills. Previous works also investigated the use of alternative observations [3–7, 21] to close the domain gap. M-EMBER accepts images as input and performs more fine-grained mobile manipulation than pick-and-place tasks. Finally, adaptive learning [22–28] and system identification [29–33] algorithms have achieved success in closing the dynamics sim-to-real gap. In comparison, M-EMBER attempts to close the visual gap in mobile manipulation.

B. Learning mobile manipulation in simulation or real world

Prior robot learning methods have achieved success in mobile manipulation in simulation [47–49] or the real world [50–59]. Some of these methods collect data in a continuous, online manner [50, 52, 53], while others break data collection into primitives [54, 55] to learn to perform long-horizon mobile manipulation. Inspired by these works, M-EMBER performs long-horizon mobile manipulation, which demands perceptual generalization and compositional generalization that can factorize and reuse learned visuomotor skills.

C. Reinforcement learning for stationary manipulation

Reinforcement learning (RL) has a rich history of being used for robotic control from locomotion [60–62], navigation [35–40], to stationary manipulation [63–65]. Indeed, prior methods in model-free [66–72] and model-based RL [73–84] achieved remarkable success in stationary manipulation. Drawing inspiration from these works, M-EMBER extends EMBER [34], a previous work in this category, to performing long-horizon mobile manipulation.

III. Preliminaries

A. Modeling long-horizon mobile manipulation

In this work, we consider the problem of performing a long-horizon mobile manipulation activity: $\mathcal{M}$. We model the activity's environment as a controlled Markov process represented by the tuple $\mathcal{E} = \langle \mathcal{S}, \rho_0, \mathcal{A}, \mathcal{T}, \gamma, H \rangle$, with an observation space of N cameras of resolution $H \times W$, which are $N = 5$, $H = W = 112$, and the robot's 19 joint angles, and thus $s \in \mathcal{S} = \mathcal{R}\{5 \times 112 \times 112 \times 3 + 19\}$, an initial state distribution ρ_0, an action space $a \in \mathcal{A}$ (see Sec. IV), a dynamics model $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, a discount factor $\gamma \in [0, 1)$, and a finite horizon H. We assume the goal of an activity is defined by a set of symbolic predicates in first-order logic that we obtain from BEHAVIOR [85, 86], a dataset of everyday activities defined in a domain-definition language (BDDL) similar to PDDL [87]. For example, the BEHAVIOR `cleaning_kitchen` activity is defined as:

$\{\forall$ cupboard $\in$ cupboards: $\{\forall$ object $\in$ cupboard: (IN object bucket)$\}$ $\wedge$ ($\neg$ OPENED cupboard) $\wedge$ ($\neg$ DUSTY cupboard)$\}$ $\wedge$ $\{\forall$ drawer $\in$ drawers: $\{\forall$ object $\in$ drawer: (IN object bucket)$\}$ $\wedge$ $\neg$ (OPENED drawer)$\}$

In plain words, this means the goal is to relocate all objects (i.e. pens, markers, screwdrivers, toothbrushes, wiping clothes) inside each cupboard and drawer in the environment into a bucket on the floor and ensure that all cupboards and drawers are closed and that all cupboards are not dusty. Performance of the robot in this activity is binary: "success" if the symbolic goal state is satisfied perfectly within a finite amount of real-clock time, or "failed" otherwise. Let K denote the total number of unique mobile manipulation skills the robot has learned in simulation (in `cleaning_kitchen`, $K = 5$), and $k \in [1, K]$ denote the k^{th} skill in the robot's skill repertoire. Here, each skill is a solution for a different Markov Decision Process (MDP) $\mathcal{M}^k = \langle \mathcal{E}, \mathcal{R}^k \rangle$, where the robot's environment $\mathcal{E}$ is shared across all activities and skills, and $\mathcal{R}^k : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function for the k^{th} skill (e.g. opening, or wiping). This paper will use "primitives" vs. "skills" as well as "factorize" vs. "decompose" interchangeably.

B. Factorization via Example-Driven Model-BasEd RL (EMBER)

M-EMBER is a factorized long-horizon mobile manipulation extension of EMBER [34]. The goal of this work is to provide a solution to long-horizon mobile manipulation activities. To this end, EMBER is not enough: it is unable to cope with the variability and task length involved in mobile manipulation activities. Mobile manipulation requires a large amount of data to cover the activity distribution, and M-EMBER overcomes this challenge by training mobile manipulation skills in simulation and applying them in the real world.

IV. Mobile-EMBER

Mobile-EMBER, or "M-EMBER", is designed to overcome the limitations of EMBER in generalization and to be able to perform long-horizon mobile manipulation. Below, we first describe how a long-horizon mobile manipulation activity is factorized into a repository of visuomotor mobile manipulation skills during training, and recomposed to solve the long-horizon activity at test time. We then discuss how M-EMBER learns each factorized visuomotor skill in simulation. Finally, we describe how M-EMBER enables the factorized skills to be used in the real world.

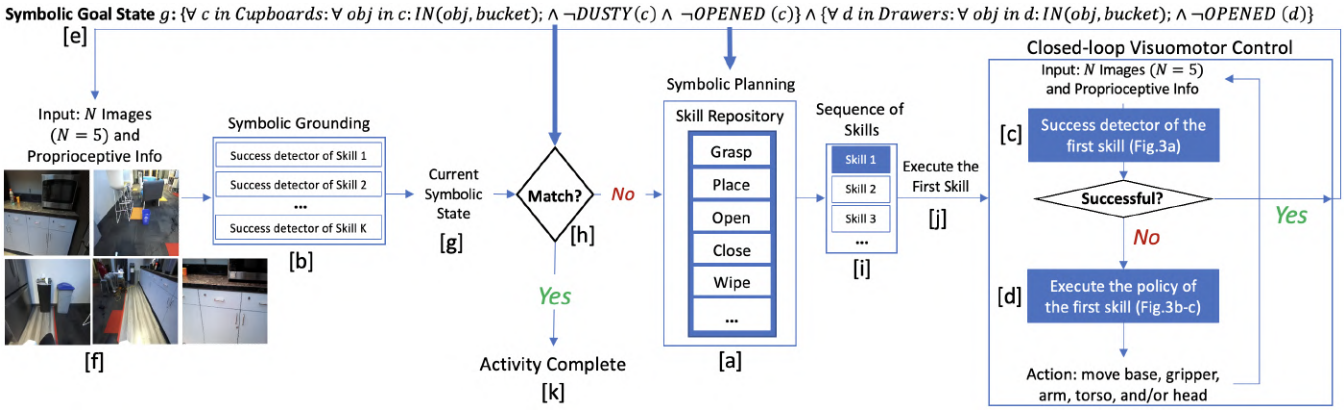


Fig. 1: High-Level Overview of M-EMBER for `cleaning_kitchen` activity. To begin, the human first specifies the activity using a symbolic goal state g (Fig. 1 [e]). For training, the activity is factorized into a repertoire of primitive skills [a] that M-EMBER learns using RL in simulation along with per-skill success detectors [b,c]. In order to detect relevant objects in the environment, M-EMBER also grounds the robot's raw pixel observations into a symbolic representation of the current state [b,g]. At test time, the per-skill success detectors are used to 1) map the input [f] –raw pixel observations (five 112×112 images) and proprioceptive information (joint angles)– into a symbolic representation of the current state [g] to check full-activity completion [h] and perform long-horizon planning [a], and 2) during execution of a skill [d] to detect whether a skill has succeeded [c]. Long-horizon planning is performed by a symbolic planner [a] that computes the sequence of skills to perform [i] and executes the first skill of this sequence [j] based on both the goal [e] and current symbolic state [g]. This procedure repeats until the current symbolic state [g] matches the goal state [e], after which robot execution terminates successfully [k]. Modules [b,c,d] are learned modules elaborated in Sec. IV and Fig. 2.

A. Skill decomposition and recomposition of a long-horizon mobile manipulation activity

To begin, the human first specifies the activity using a symbolic goal state g (Fig. 1 [e]). During training, the long-horizon mobile manipulation activity is factorized into a repository of skills based on the symbolic representation of this goal state (i.e. grasp, place, open, close, and wipe in the case of the `cleaning_kitchen` activity). To verify that the skills are successful, each skill is trained together with a "success detector" that will determine visually when the symbolic component of the activity goal has transitioned to the desired value. In order to detect relevant objects in the environment, M-EMBER also grounds the robot's raw pixel observations into a symbolic representation of the current state (Fig. 1 [b,g]).

To perform the long-horizon activity at test time, M-EMBER first computes the current symbolic state of the environment by passing the raw pixel observations (five 112×112 images) and proprioceptive information (Fig. 1 [f]) to all skills' success detectors (Fig. 1 [b]). Using both the goal (Fig. 1 [e]) and the current symbolic state (Fig. 1 [g]), the symbolic planner computes the sequence of skills to perform (Fig. 1 [i]) and executes the first skill of this sequence (Fig. 1 [j]). This procedure repeats until the current symbolic state (Fig. 1 [g]) matches the goal state perfectly (Fig. 1 [e]), after which robot execution terminates successfully (Fig. 1 [k]).

B. Learning Each Factorized Skill in Simulation

M-EMBER learns to perform each skill by learning three individual components per skill (see Fig. 2): a variational autoencoder (VAE) (f^k_{vae} in Fig. 2), success detectors ($f^k_{\mathcal{R}}$ in Fig. 2), and Q-functions (f^k_Q in Fig. 2). The VAE reduces the dimensionality of the robot's pixel observation into a latent representation that is used as input for the success detectors and the Q-functions; the success detectors allow M-EMBER to learn a binary reward function for each skill; and Q-functions are learned from the binary reward functions and allow M-EMBER to perform close-loop visuomotor control. Accordingly, the VAE optimization objective is:

$$\min_{f^k_{\text{vae}}} \mathbb{E}_{s_t, a_t \sim \mathcal{D}} \Big[-\mathcal{L}_{\text{vae}}(s_t) \Big]$$

where $\mathcal{D}$ is the dataset of simulated trajectories, and $\mathcal{L}_{\text{vae}}$ is the evidence lower bound (ELBO) for the VAE:

$$\max_{p,q} \mathbb{E}_{q(z|s)} \left[\log p\left(s \mid z\right) \right] - D_{\text{KL}} q(z \mid s) p(z)$$

The optimization objective for the success detector (Fig. 2 a) is

$$\max_{f^k_{\mathcal{R}}} \mathbb{E}_{s^+ \sim \mathcal{D}^+, z^+ \sim f^k_{\text{enc}}(\cdot|s^+)} \left[\log \left(f^k_{\mathcal{R}}(z^+) \right) \right] + \mathbb{E}_{s^- \sim \mathcal{D}, z^+ \sim f^k_{\text{enc}}(\cdot|s^+)} \left[\log \left(1 - f^k_{\mathcal{R}}(z^-) \right) \right]$$

Here, $\mathcal{D}^+$ and $\mathcal{D}^-$ are the datasets of images labeled as positive and negative, s^+ and s^- are the images sampled from $\mathcal{D}^+$ and $\mathcal{D}^-$. The Q-function optimization objective is:

$$\min_{f^k_Q} \mathbb{E}_{s_t, a_t, s_{t+1} \sim \mathcal{D}} \left[f^k_Q\left(z_t, a_t\right) - \left(\overline{f^k_{\mathcal{R}}}\left(z_{t+1}\right) + \gamma \overline{f^k_{\mathcal{R}}}\left(z_{t+1}\right) f^k_{Q_{target}} \right) \right]^2$$

where

$$\overline{f^k_{\mathcal{R}}}(z) \equiv \mathbb{1}\{ f^k_{\mathcal{R}}(z) > 0.5 \}$$
$$z_t \sim f^k_{\text{vae}}(\cdot \mid s_t)$$
$$z_{t+1} \sim f^k_{\text{vae}}(\cdot \mid s_{t+1})$$

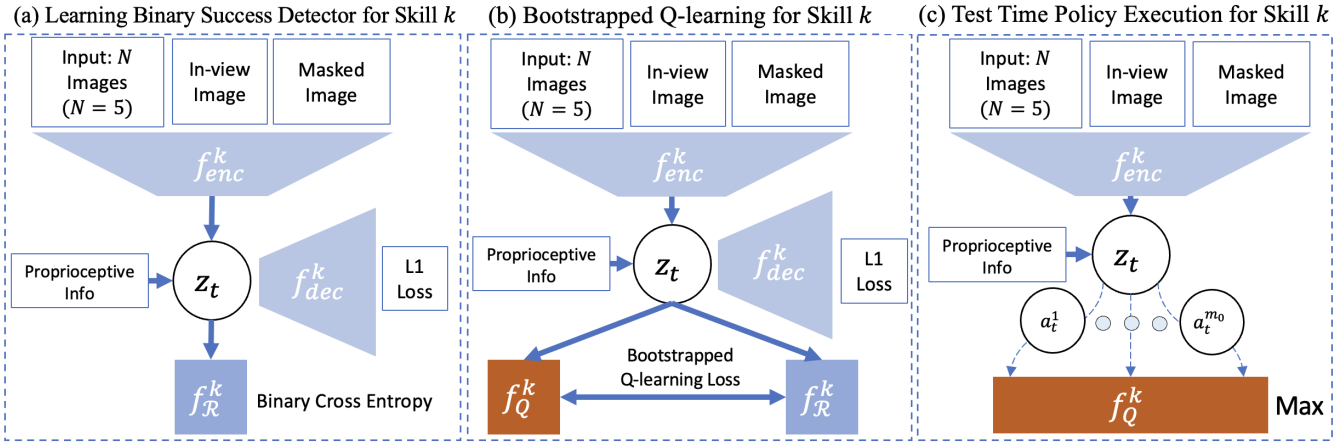


Fig. 2: Low-Level Visualization of M-EMBER's Skill-Learning Process. M-EMBER learns to perform each skill by learning three individual components for each skill: a variational autoencoder (VAE) (f^k_{vae} in Fig. 2), success detectors ($f^k_{\mathcal{R}}$ in Fig. 2), and Q-functions (f^k_Q in Fig. 2). The VAE reduces the dimensionality of the robot's pixel observation that is used as input for the success detectors and the Q-functions; the success detectors allow M-EMBER to learn a binary reward function for each skill; and Q-functions are learned from the binary reward functions and allow M-EMBER to perform close-loop visuomotor control. Kindly see the EMBER paper [34] for details.

The Q-value target is:

$$f^k_{Q_{target}} = \max_{a_{t+1}} f^k_Q(z_{t+1}, a_{t+1})$$

It is computed by maximizing over 200 randomly and uniformly sampled actions $a^{1:200}_{t+1}$. There are three significant differences between M-EMBER and EMBER: 1) while EMBER learns and rolls out a latent dynamics model for MBOLD [84] planning, M-EMBER does not learn or use a latent dynamics model given its poor empirical performance in the real world, which we quantify in Sec. V. Instead, we directly apply cross-entropy method (CEM) on actions sampled from the action space and Q-values queried from the learned Q-functions [67, 72]. 2) In addition to per-camera images, M-EMBER takes as input the image in which the object resides in ("In-view Image" in Fig. 2) as well as the masked image ("Masked Image" in Fig. 2) from the object of interest. These two additional images signal to the learned M-EMBER model which camera the object of interest resides in and which object in this camera image the robot should manipulate. 3) The VAE in M-EMBER is per-skill instead of shared across skills as in EMBER.

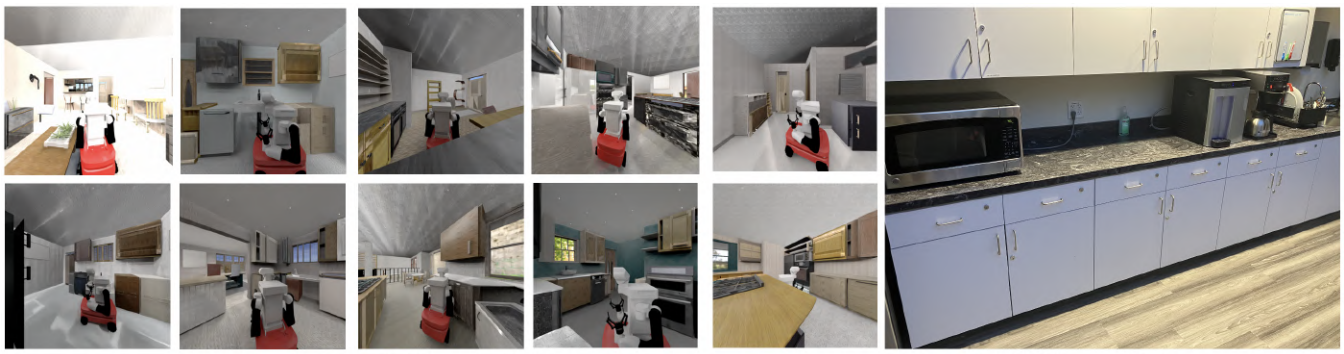
C. Training learned skills in simulation to be used in real-world conditions

To transfer raw pixel inputs from simulation to real-world conditions, we use photorealism and domain randomization in a photorealistic simulator built on top of iGibson [88–90]. To execute actions in the real world, M-EMBER commands small changes in the pose of the robot's two end-effectors (6D) and base (3D), and joint positions of the head, torso and two grippers.

D. Photorealism and domain randomization

We apply photorealistic rendering to the iGibson simulator [88–90] to increase the photorealism of the simulated scenes. We then randomize rendered scenes across 14 dimensions:

- *Object instance:* each object is randomized across instances within the same category
- *Object placement:* when learning each skill, each object is placed in the environment with a randomized 6-DOF pose
- *Object scale:* each object in the environment is randomized across dimensions and scales
- *Object texture:* texture is randomized across diffusion, metallic, roughness, and normal maps
- *Indoor lighting condition:* lighting direction, types, surface area, and intensity are randomized in the indoor environment
- *Outdoor lighting condition:* dome lighting direction and intensity are randomized
- *Initial robot placement:* the robot is initially randomly (2-DOF position and 1-DOF rotation) placed in the kitchen
- *Initial camera viewpoint:* the initial viewpoint of the camera is randomized by the starting joint configuration of the robot, so long as the objects are still visually accessible by at least one of five robot cameras
- *Camera parameters:* each training environment is randomized across intrinsic and extrinsic camera parameters
- *Outdoor environment texture:* outdoor texture is randomized across 500+ HDR environment maps
- *Indoor interior randomization:* all ceilings, walls, and floors are randomized across textures maps
- *Scene randomization:* each training environment is randomized across rooms and floor plans
- *Physics:* each training environment is randomized across 0.5-3.5x frictional and inertial coefficients
- *Robot arm texture:* each training environment is randomized across 20-25 robot arm textures



(a) Training Kitchens (b) Validation Kitchen

Fig. 3: Train and Validation Environments. Fig. **(a)** exhibits a subset of simulated kitchen environments in which the TIAGo robot learns each mobile manipulation primitive skill. The TIAGo robot is equipped with two Robotiq parallel-jaw grippers and five cameras capturing five images at 3Hz. To narrow the domain gap, we use a real world kitchen for validation (Fig. **(b)**), and then evaluate M-EMBER on three kitchens.

V. Experiments

Experiments in this paper aim to answer four main questions: 1) Can M-EMBER's factorized and learned primitive skills generalize? 2) Does latent dynamics prediction in EMBER contribute or degrade M-EMBER's transfer fidelity? 3) Can M-EMBER perform long-horizon activities? 4) How much does photorealism and domain randomization each contribute to transfer fidelity?

A. Experimental setup

To answer these questions, we conduct experiments of both factorized primitive skills and the long-horizon `cleaning_kitchen` BEHAVIOR [85] activity across three kitchens and dozens of object instances (Fig. 4). In Fig. 3, TIAGo, a bi-manual mobile manipulator, has access to five cameras together forming a panoramic view of the robot's surrounding, each capturing 112×112 images at 3Hz, as well as 360° 2D LiDAR scans. All experiments have no April tags or landmarks present and are autonomously executed without human intervention.

B. Comparisons

We compare M-EMBER to two prior methods: EMBER [34] (M-EMBER with MBOLD [84] planning) and BEE [91] (M-EMBER with Visual Foresight, which uses success detectors instead of the Q-functions for model predictive control). We also compare M-EMBER to "M-EMBER with scripted skills", which scripts instead of learning all skills and therefore does not need to perform domain transfer. The training data for all experiments are collected from the same simulator, which contains photorealism and domain randomization techniques outlined in Section IV-C.

C. Factorized skill performance

We compare all methods across five factorized skills: grasp, place, open, close, and wipe. M-EMBER achieves 75-95% success rates across three kitchens and 50+ object instances (Fig. 4). In contrast, EMBER and BEE achieve single-digit success rates for each skill due to the difficulty of using latent dynamics models trained from simulation in the real world.

D. Long-horizon activity performance

For long-horizon experiments, this paper investigates the `cleaning_kitchen` activity, in which the robot is placed in a kitchen, which contains cupboards or drawers. Each cupboard or drawer is closed initially, and there are object instances placed in it. These objects (Fig. 4) range in 5 categories: screwdrivers, toothbrushes, pens, wiping clothes, and markers. There is a bucket (Fig. 4) randomly placed on the floor. There is a piece of wiping cloth (Fig. 4) laying in each cupboard for the robot to wipe the cupboard shelf clean. After wiping, the cloth should also be placed in the bucket. The goal of this activity is to put all objects in each cupboard and drawer into the bucket, wipe all cupboards clean, and close all cupboards and drawers. Wiping is "clean" if 90%+ of the surface area of the cupboard shelf reachable by the robot is wiped.

In Table I, we find that "M-EMBER (Ours)" completes the activity with 53.3% success. In comparison, "M-EMBER with Scripted Skills" achieves single-digit success rates due to the use of scripted, non-learning skills, while EMBER and BEE achieve no success due to poor performance of the latent dynamics models in the real world. Empirically, we observe three major failure cases of M-EMBER in this activity: 1) collision with the kitchen cupboards or drawers; 2) leaving at least one object inside a cupboard or drawer; 3) failed object grasp attempts that let to objects becoming no longer mechanically reachable.

E. Ablation studies

We conduct ablations to quantify the transfer fidelity contribution of photorealism and domain randomization. To quantify the contribution of photorealism, we ablate M-EMBER by turning off photorealism in simulation. In Table I, we find that M-EMBER's factorized skill and long-horizon activity success rates degrade by 35-65% and 53.3% respectively, quantifying the importance of photorealistic

(a) Validation (L), Test (R) Buckets **(b)** Validation (L), Test (R) Object Instances **(c)** Validation (L), Test (R) Clothes

Fig. 4: Validation and Test buckets, object instances, and wiping clothes used in real-robot experiments

TABLE I: Successful trials (out of 30) and percentages of `cleaning_kitchen` activity. K: number of skills composed in each trial.

`cleaning_kitchen`	Number Of Cupboards	Number Of Drawers	M-EMBER (Ours)	Prior Method 1: M-EMBER w/ Scripted Skills	Prior Method 2: EMBER (M-EMBER w/ MBOLD planning) [34]	Prior Method 3: BEE (M-EMBER w/ Visual Foresight) [91]	Ablation 1: M-EMBER w/o Photorealism	Ablation 2: M-EMBER w/ 50% Domain Randomization
Kitchen 1	1	7	5/10	0/10	0/10	0/10	0/10	2/10
Kitchen 2	3	6	5/10	1/10	0/10	0/10	0/10	2/10
Kitchen 3	3	0	6/10	0/10	0/10	0/10	0/10	3/10
Total	7	13	16/30 (53.3%)	1/30 (3.3%)	0/30 (0%)	0/30 (0%)	0/30 (0%)	7/30 (23.3%)

rendering in domain transfer. To ablate domain randomization, we shrink the range of randomization across each dimension of randomization specified in Section IV-C by 50% in simulation. In Table I, we find that M-EMBER's factorized skill and long-horizon activity success rates degrade by 20-45% and 30% respectively, quantifying the importance of rendering randomized environments and objects in domain transfer.

VI. Conclusion

In this work, M-EMBER develops factorized domain transfer for long-horizon mobile manipulation that demands both perceptual and compositional generalization that can factorize and reuse learned skills. Nevertheless, the domain gap is still immense, which is not only reflected in the low success rates achieved in experiments, as well as the limited settings the robot can operate in. Moreover, additional computational investments in simulation will likely result in diminishing returns in performance gains.

VII. Appendix: Additional Implementation Details and Clarifications

1) Using Raw Pixels vs. Depth or RGB-D as Visual Observations: While M-EMBER uses images as input due to legacy considerations, it is a limitation that it currently cannot accept depth input (inclusively or exclusively) as part of its visual observations. Such important extension from raw pixels to depth image is left for future work.

2) Proprioceptive Information as Non-visual Observations: In additional images, M-EMBER takes as input proprioceptive information as non-visual observations. Such proprioceptive information comprises all robot joint angles (e.g. head, torso, arms, and grippers).

3) Use of 360° *2D LiDAR:* The 360° 2D LiDAR readings of the robot provide fail-safe mechanisms for the robot and are not visible to the learning modules in M-EMBER.

4) Simulation Environments: The simulation environments used in this paper are those produced from the iGibson simulator.

5) Using additional photorealistic datasets: While iGibson is used as the primary simulator in this paper, extending training environments to widely used ones such as Matterport, Replica as well as Habitat (including the Rearrangement Challenge) are respectfully left for future work. The authors agree to some extent that submissions to this challenge are probably more realistic baselines for the chosen formulation.

6) The stringent demands of generalization abilities of solutions to the mobile manipulation problem: The authors would like to clarify that the combinatorial nature of the mobile manipulation problem is what makes the demands of the generalization abilities of mobile manipulation solutions particularly stringent.

7) On the use of "unclear" in discussions of related works: The authors acknowledge that instead of conjecturing that it is "unclear" the existing methods will work, they could have explained how they empirically validate these concerns and propose their approach as a solution. However, given the empirical weakness of their own solutions, the authors have opted out of adopting a stronger connotation at this time.

8) Mobile vs. stationary manipulation: The authors acknowledge that the paper might come off as suggesting that mobile manipulation is a totally different problem than stationary manipulation. However, the authors would like to clarify that they do not think mobile manipulation is a fundamentally different problem from stationary manipulation.

VIII. Acknowledgments

The authors would first like to give their special thanks to William Chong, Marion Lepert, and Wesley Guo from the Department of Mechanical Engineering of Stanford University for their outstanding mechanical support for the robot. The authors would also like to thank the entire Stanford Vision and Learning Lab (SVL) for the experimental setup.

REFERENCES

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

[2] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Conference on Robot Learning*, PMLR, 2022, pp. 24–33.

[3] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2442–2447.

[4] J. Mahler *et al.*, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, pp. 1957–1964.

[5] J. Mahler *et al.*, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[6] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *2018 IEEE International Conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 5620–5627.

[7] J. Mahler *et al.*, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, eaau4984, 2019.

[8] Y. Chebotar *et al.*, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 8973–8979.

[9] S. Dasari *et al.*, "Robonet: Large-scale multi-robot learning," in *CoRL*, 2019. arXiv: 1910.11215 [cs.RO].

[10] A. Mandlekar *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conference on Robot Learning*, 2018.

[11] E. Jang *et al.*, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*, PMLR, 2022, pp. 991–1002.

[12] F. Ebert *et al.*, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," *arXiv preprint arXiv:2109.13396*, 2021.

[13] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[14] A. S. Chen, S. Nair, and C. Finn, *Learning generalizable robotic reward functions from "in-the-wild" human videos*, 2021. arXiv: 2103.16817 [cs.RO].

[15] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," *arXiv preprint arXiv:2207.09450*, 2022.

[16] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2017, pp. 23–30.

[17] J. Tobin *et al.*, "Domain randomization and generative models for robotic grasping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3482–3489.

[18] X. Ren *et al.*, "Domain randomization for active pose estimation," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 7228–7234.

[19] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 3803–3810.

[20] Y. Du, O. Watkins, T. Darrell, P. Abbeel, and D. Pathak, "Auto-tuned sim-to-real transfer," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 1290–1296.

[21] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Conference on robot learning*, PMLR, 2017, pp. 291–300.

[22] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, eabk2822, 2022. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.abk2822.

[23] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[24] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing energy consumption leads to the emergence of gaits in legged robots," in *Conference on Robot Learning*, PMLR, 2022, pp. 928–937.

[25] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak, "Coupling vision and proprioception for navigation of legged robots," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 273–17 283.

[26] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, eabk2822, 2022.

[27] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust rough-terrain locomotion with a quadrupedal robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 5761–5768.

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

[29] J. Tan *et al.*, "Sim-to-real: Learning agile locomotion for quadruped robots.," in *Robotics: Science and Systems*, 2018.

[30] W. Yu, V. C. Kumar, G. Turk, and C. K. Liu, "Sim-to-real transfer for biped locomotion," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3503–3510.

[31] A. Allevato, E. S. Short, M. Pryor, and A. Thomaz, "Tunenet: One-shot residual tuning for system identification and sim-to-real robot task transfer," in *Conference on Robot Learning*, PMLR, 2020, pp. 445–455.

[32] A. A. David, S. E. Schaertl, M. Pryor, and A. L. Thomaz, "Iterative residual tuning for system identification and sim-to-real robot learning," *Autonomous Robots*, vol. 44, no. 7, pp. 1167–1182, 2020.

[33] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. Panne, "Learning locomotion skills for cassie: Iterative design and sim-to-real," in *Conference on Robot Learning*, PMLR, 2020, pp. 317–329.

[34] B. Wu, S. Nair, L. Fei-Fei, and C. Finn, "Example-driven model-based reinforcement learning for solving long-horizon visuomotor tasks," in *5th Annual Conference on Robot Learning*, 2021.

[35] P. Anderson *et al.*, "Sim-to-real transfer for vision-and-language navigation," in *Conference on Robot Learning*, PMLR, 2021, pp. 671–681.

[36] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 31–36.

[37] J. Kulhánek, E. Derner, T. De Bruin, and R. Babuška, "Vision-based navigation using deep reinforcement learning," in *2019 European Conference on Mobile Robots (ECMR)*, IEEE, 2019, pp. 1–8.

[38] H. Hu, K. Zhang, A. H. Tan, M. Ruan, C. Agia, and G. Nejat, "A sim-to-real pipeline for deep reinforcement learning for autonomous robot navigation in cluttered rough terrain," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6569–6576, 2021.

[39] W. B. Shen, D. Xu, Y. Zhu, L. J. Guibas, L. Fei-Fei, and S. Savarese, "Situational fusion of visual representation for visual navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2881–2890.

[40] Y. Zhu *et al.*, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 3357–3364.

[41] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "Rl-cyclegan: Reinforcement learning aware simulation-to-real," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 157–11 166.

[42] D. Ho, K. Rao, Z. Xu, E. Jang, M. Khansari, and Y. Bai, "Retinagan: An object-aware approach to sim-to-real transfer," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10 920–10 926.

[43] Y. Narang *et al.*, "Factory: Fast contact for robotic assembly," *arXiv preprint arXiv:2205.03532*, 2022.

[44] S. James *et al.*, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 627–12 637.

[45] D. Ho, K. Rao, Z. Xu, E. Jang, M. Khansari, and Y. Bai, "Retinagan: An object-aware approach to sim-to-real transfer," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 10 920–10 926.

[46] C. Yuan *et al.*, "Sim-to-real transfer of robotic assembly with visual inputs using cyclegan and force control," *arXiv preprint arXiv:2208.14104*, 2022.

[47] F. Xia, C. Li, R. Martın-Martın, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 4583–4590.

[48] C. Li, F. Xia, R. Martin-Martin, and S. Savarese, "Hrl4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators," in *Conference on Robot Learning*, PMLR, 2020, pp. 603–616.

[49] D. Hadjivelichkov, K. Vlachos, and D. Kanoulas, "Improved reinforcement learning coordinated control of a mobile manipulator using joint clamping," *arXiv preprint arXiv:2110.01926*, 2021.

[50] J. Wu *et al.*, "Spatial action maps for mobile manipulation," *arXiv preprint arXiv:2004.09141*, 2020.

[51] C. Wang *et al.*, "Learning mobile manipulation through deep reinforcement learning," *Sensors*, vol. 20, no. 3, p. 939, 2020.

[52] C. Sun *et al.*, "Fully autonomous real-world reinforcement learning with applications to mobile manipulation," in *Conference on Robot Learning*, PMLR, 2022, pp. 308–319.

[53] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," *Advances in neural information processing systems*, vol. 31, 2018.

[54] M. Ahn *et al.*, "Do as i can and not as i say: Grounding language in robotic affordances," in *arXiv preprint arXiv:2204.01691*, 2022.

[55] W. Huang *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[56] X. Zhang, Y. Zhu, Y. Ding, Y. Zhu, P. Stone, and S. Zhang, "Visually grounded task and motion planning for mobile manipulation," *arXiv preprint arXiv:2202.10667*, 2022.

[57] S. Jauhri, J. Peters, and G. Chalvatzaki, "Robot learning of mobile manipulation with reachability behavior priors," *arXiv preprint arXiv:2203.04051*, 2022.

[58] Z. Fu, X. Cheng, and D. Pathak, "Learning a unified policy for whole-body control of manipulation and locomotion," in *6th Annual Conference on Robot Learning*, 2022.

[59] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," *arXiv preprint arXiv:2103.10534*, 2021.

[60] H. Benbrahim and J. A. Franklin, "Biped dynamic walking using reinforcement learning," *Robotics and Autonomous Systems*, vol. 22, no. 3, pp. 283–302, 1997, Robot Learning: The New Wave.

[61] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, 2619–2624 Vol.3, 2004.

[62] R. Tedrake, T. Zhang, and H. Seung, "Stochastic policy gradient reinforcement learning on a simple 3d biped," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2849–2854 vol.3, 2004.

[63] V. Gullapalli, "Skillful control under uncertainty via direct reinforcement learning," *Robotics and Autonomous Systems*, vol. 15, no. 4, pp. 237–246, 1995, Reinforcement Learning and Robotics.

[64] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682–697, 2008, Robotics and Neuroscience.

[65] M. Deisenroth, C. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, Jun. 2011.

[66] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3406–3413, 2016.

[67] D. Kalashnikov *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.

[68] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," 2018.

[69] A. Ghadirzadeh, A. Maki, D. Kragic, and M. Björkman, "Deep predictive policy training using reinforcement learning," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2351–2358, 2017.

[70] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual reinforcement learning with imagined goals," *Advances in Neural Information Processing Systems*, vol. 31, pp. 9191–9200, 2018.

[71] A. Singh, L. Yang, C. Finn, and S. Levine, "End-to-end robotic reinforcement learning without reward engineering," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, Jun. 2019.

[72] D. Kalashnikov *et al.*, "Mt-opt: Continuous multi-task robotic reinforcement learning at scale," *ArXiv*, vol. abs/2104.08212, 2021.

[73] D. Hafner *et al.*, "Learning latent dynamics for planning from pixels," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97, PMLR, Sep. 2019, pp. 2555–2565.

[74] M. Bhardwaj, A. Handa, D. Fox, and B. Boots, "Information theoretic model predictive q-learning," in *Learning for Dynamics and Control*, PMLR, 2020, pp. 840–850.

[75] R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn, "Offline reinforcement learning from images with latent space models," *arXiv preprint arXiv:2012.11547*, 2020.

[76] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel, "Daydreamer: World models for physical robot learning," *arXiv preprint arXiv:2206.14176*, 2022.

[77] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *International Conference on Learning Representations*, 2019.

[78] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in neural information processing systems*, 2016, pp. 64–72.

[79] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 2786–2793.

[80] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[81] F. Ebert, S. Dasari, A. X. Lee, S. Levine, and C. Finn, "Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning," in *Conference on Robot Learning (CoRL)*, 2018.

[82] Y.-C. Lin, M. Bauzá, and P. Isola, "Experience-embedded visual foresight," *ArXiv*, vol. abs/1911.05071, 2019.

[83] H. Suh and R. Tedrake, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," *ArXiv*, vol. abs/2002.09093, 2020.

[84] S. Tian *et al.*, "Model-based visual planning with self-supervised functional distances," *International Conference on Learning Representations (ICLR)*, 2021.

[85] S. Srivastava *et al.*, "Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Conference on Robot Learning*, PMLR, 2022, pp. 477–490.

[86] C. Li *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.

[87] D. McDermott *et al.*, "Pddl–the planning domain definition language–version 1.2," *Yale Center for Computational Vision and Control, Tech. Rep. CVC TR-98-003/DCS TR-1165*, 1998.

[88] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9068–9079.

[89] B. Shen *et al.*, "Igibson 1.0: A simulation environment for interactive tasks in large realistic scenes," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 7520–7527.

[90] C. Li *et al.*, "Igibson 2.0: Object-centric simulation for robot learning of everyday household tasks," *arXiv preprint arXiv:2108.03272*, 2021.

[91] A. S. Chen, H. Nam, S. Nair, and C. Finn, "Batch exploration with examples for scalable robotic reinforcement learning," *arXiv preprint arXiv:2010.11917*, 2020.